



(51) International Patent Classification:

H04L 12/803 (2013.01) H04L 29/08 (2006.01)
H04L 12/931 (2013.01) G06F 9/445 (2018.01)
H04L 29/06 (2006.01)

(21) International Application Number:

PCT/SE2017/051279

(22) International Filing Date:

15 December 2017 (15.12.2017)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: TELEFONAKTIEBOLAGET LM
ERICSSON (PUBL) [SE/SE]; 164 83 Stockholm (SE).(72) Inventors: HAMMAM, Tarik; Ringstedtsgatan 8, SE-164
48 KISTA (SE). KÖVÉR, János; Alingsåsvägen 20,

SE-121 48 JOHANNESHOV (SE). FRANZEN, Anders;
Violvägen 7, SE-142 65 TRÅNGSUND (SE).

(74) Agent: SJÖBERG, Mats; Ericsson AB, Patent Unit Kista,
RAN Implementation & Core (PU-KRIC), 164 80 Stock-
holm (SE).

(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, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(54) Title: METHOD AND TRAFFIC PROCESSING UNIT FOR HANDLING TRAFFIC IN A

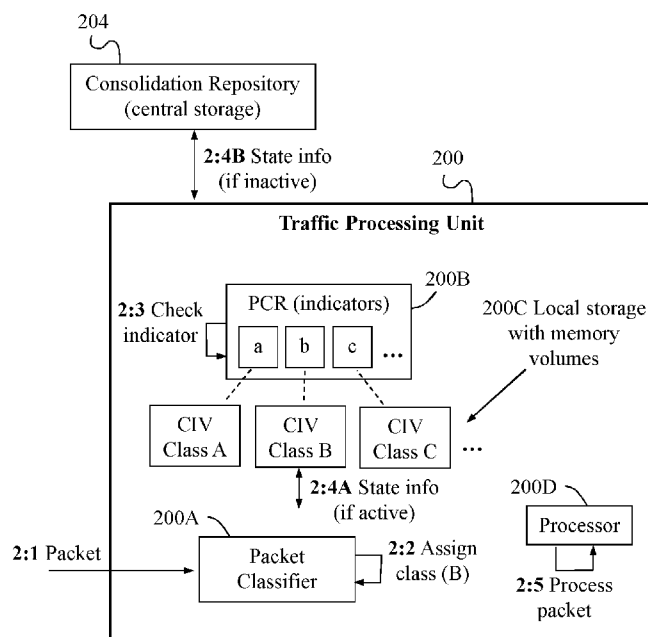


Fig. 2

(57) Abstract: A method and a traffic processing unit (200) for handling traffic in a communication network when the traffic is distributed across a set of traffic processing units. When receiving a packet of a traffic flow distributed to said traffic processing unit, the traffic processing unit (200) assigns a packet class to the received packet, which class can be active or inactive in the traffic processing unit. The traffic processing unit obtains state information of the assigned packet class. If the packet class is detected as active the state information is retrieved from a local storage (200C) in the traffic processing unit, and if the packet class is detected as inactive the state information is fetched from a central storage (204). The traffic processing unit then performs stateful packet processing of the received packet based on the obtained state information.



(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))*

METHOD AND TRAFFIC PROCESSING UNIT FOR HANDLING TRAFFIC IN A COMMUNICATION NETWORK

Technical field

The present disclosure relates generally to a method and a traffic processing unit,
5 for handling traffic in a communication network when the traffic is distributed across a set of traffic processing units.

Background

In communication networks where data is communicated in different traffic flows, it is common to implement a traffic processing function, e.g. firewall processing, by
10 means of resources in a set of more or less uniform processing units so that traffic flows can be distributed across the traffic processing units in some desirable manner. In this description, the processing units of such a set will be denoted “traffic processing units” or traffic processors for short. If the processing must be performed based on some specific state information, referred to as stateful
15 processing, the traffic processing unit that handles a specific traffic flow needs to have access to state information pertaining to that traffic flow. This state information is typically stored locally at the traffic processing unit itself or it can be fetched from some central storage where state information of ongoing traffic flows is maintained.

20 In this context, state information refers to any information, such as parameters, identifiers, descriptors, references, etc., that is somehow related to the ongoing traffic flow. For example, in the case of a processing unit having the function of a stateful firewall, the state information usually reflects some part of a state changing history of a communication protocol state machine which the firewall function
25 tracks in order to detect illicit protocol manipulation.

The above distribution of processing resources across a set of multiple traffic processing units, which may be done by a distribution device or the like connected to each traffic processing unit, can be beneficial to employ for several reasons. For example, the traffic can be load balanced so that the processing workload is
30 spread out evenly across the traffic processing units so as to achieve high

capacity and efficiency and to avoid delaying bottlenecks as well as under-used units. Another reason for employing such distribution of traffic flow may be to achieve a flexible or “scalable” capacity of the processing function so that it can be adapted to current needs and requirements, e.g. by adding more traffic processing units to the set to increase the total capacity, or removing traffic processing units from the set if the total capacity is larger than needed. It may further be desirable to adapt the processing capacity in a more or less dynamic manner, e.g. by hiring processing equipment as virtual machines in a cloud-like environment, which may be useful in situations where the load on the processing function varies greatly.

However, it is a problem that when the processing is stateful and requires some specific state information, and a traffic processing unit currently used for an ongoing traffic flow is inactivated for whatever reason, the state information must be migrated from the old traffic processing unit to a new one so that the new traffic processing unit can continue the processing based on the same state information.

There are several solutions to accomplish such a migration but they introduce considerable complexity which could be difficult and/or costly to realize.

Summary

It is an object of embodiments described herein to address at least some of the problems and issues outlined above. It is possible to achieve this object and others by using a method and a node as defined in the attached independent claims.

According to one aspect, a method is performed by a traffic processing unit for handling traffic in a communication network when the traffic is distributed across a set of traffic processing units. In this method, the traffic processing unit receives a packet of a traffic flow distributed to said traffic processing unit, and assigns a packet class to the received packet, out of a set of predefined classes, based on information derived from the received packet. The assigned packet class can be detected as being active or inactive in the traffic processing unit based on an activity indicator of the packet class maintained in the traffic processing unit.

The traffic processing unit further obtains state information pertaining to the assigned packet class. This is done such that if the packet class is detected as active the state information is retrieved from a local storage in the traffic processing unit, and if the packet class is detected as inactive the state information is fetched from a central storage where state information of different packet classes is maintained. The traffic processing unit then performs stateful packet processing of the received packet based on the obtained state information.

According to another aspect, a traffic processing unit is arranged to handle traffic in a communication network when the traffic is distributed across a set of traffic processing units. The traffic processing unit is configured to receive a packet of a traffic flow distributed to said traffic processing unit, and to assign a packet class to the received packet, out of a set of predefined classes, based on information derivable from the received packet, wherein the packet class is detected as being active or inactive in the traffic processing unit based on an activity indicator of the packet class maintained in the traffic processing unit.

The traffic processing unit is also configured to obtain state information pertaining to said assigned packet class, such that if the packet class is detected as active the state information is retrieved from a local storage in the traffic processing unit, and if the packet class is detected as inactive the state information is fetched from a central storage where state information of different packet classes is maintained. The traffic processing unit is further configured to perform stateful packet processing of the received packet based on the obtained state information.

The above method and traffic processing unit may be configured and implemented according to different optional embodiments to accomplish further features and benefits, to be described below.

Brief description of drawings

The solution will now be described in more detail by means of exemplary embodiments and with reference to the accompanying drawings, in which:

Fig. 1 is a communication scenario illustrating how traffic is distributed across a set of traffic processing units in a communication network, where the solution could be used.

Fig. 2 is a block diagram illustrating an example of how the solution may be employed in a traffic processing unit, according to some example embodiments.

Fig. 3 is a flow chart illustrating a procedure in a traffic processing unit, according to further example embodiments.

Fig. 4 is a flow chart illustrating an example of how a traffic processing unit may operate in more detail, according to further example embodiments.

Fig. 5 is a block diagram illustrating how a traffic processing unit may be structured, according to further example embodiments.

Detailed description

Briefly described, a solution is provided that enables a data processing unit to obtain and use state information in an easy manner for processing an incoming traffic flow, regardless of whether the state information already exists in the data processing unit or must be acquired otherwise. **Fig. 1** illustrates a communication scenario where the solution could be employed for handling traffic in a communication network when the traffic, i.e. packets, is distributed across multiple traffic processing units 100. It is shown that a distribution device 102 distributes a packet to one of the traffic processing units 100A which then obtains state information for the packet and performs stateful processing of the packet based on the obtained state information.

After the processing is completed, the traffic processing unit 100A may send the packet to a collection device 104 or the like which is responsible for routing the packet towards its destination. The solution and embodiments to be described herein are directed to how valid state information can be obtained for a packet in an efficient manner without causing significant delays, by either retrieving the state information from a local memory in the traffic processing unit 100A or by fetching it from a central storage.

This can be accomplished by assigning a packet class to the received packet and checking an activity indicator of the packet class maintained in the traffic processing unit. If the activity indicator indicates the packet class as active, this means that valid state information is available from a local memory in the traffic processing unit 100A. On the other hand, if the activity indicator indicates the packet class as inactive, this means that no valid state information is available from the local memory and it must be fetched from a central storage where state information of different packet classes is maintained.

The solution and various embodiments thereof will be described herein in terms of functionality in a traffic processing unit, such as the above-mentioned traffic processing unit 100A, which operates to handle traffic in a communication network when the traffic is distributed across a set of traffic processing units.

An example of how such a traffic processing unit may be configured with various functionalities will first be described with reference to the block diagram illustrated in **Fig. 2**, and with further reference to Fig. 1. The traffic processing unit 200 is shown to comprise the following functional entities which may be implemented in any suitable manner, e.g. by a combination of hardware and software. However, the solution is not limited to this example of functional entities.

The traffic processing unit 200 comprises a function for packet classification denoted **packet classifier 200A**. An **action 2:1** illustrates that a packet is received at the traffic processing unit 200, which may have been distributed thereto by a distribution device 102. The packet classifier 200A then assigns a packet class to the received packet based on information derived from the received packet, in an **action 2:2**. For example, the information in the received packet may comprise at least one of a source address and a destination address, which thus indicate from where the packet has been sent and to where it should be further sent. Such information can thus be used for determining the packet class of the received packet. In further examples, the packet class could be determined by applying a mathematical function or logical hash operation on any information in the packet, for example, a logical/Boolean operator "Exclusive OR

(XOR)” may be applied to information fields in the packet header, such as Source address XOR Destination address XOR Source port XOR Destination port XOR Protocol.

The traffic processing unit 200 in this example also comprises a **packet class**

- 5 **record 200B** holding activity indicators of different packet classes for traffic flows that have been handled previously by the traffic processing unit. In the packet class record 200B a number of activity indicators a, b, c ... may be stored and each activity indicator indicates that a particular associated packet class A, B, C... is either active or inactive in the traffic processing unit 200.

- 10 The packet class record 200B may hold activity indicators of all possible packet classes, such that those packet classes that have valid associated state information already available in the traffic processing unit 200 are indicated as active, while those packet classes that do not have any valid state information available in the traffic processing unit 200 are indicated as inactive. Any state
- 15 information may be maintained as valid in the traffic processing unit 200 for a prescribed duration until a suitable timeout occurs after which the state information can be deleted and the corresponding activity indicator is changed from active to inactive. The latter may be done even if the expired state information remains in the traffic processing unit 200.

- 20 The traffic processing unit 200 further comprises a **local storage 200C** where state information is being stored in memory volumes for different packet classes that have been assigned to packets when previously processed by the traffic processing unit 200. These memory volumes are denoted Class Information Volumes, CIVs. In this example, activity indicator “a” indicates that class A is
- 25 active and has associated state information stored in a CIV allocated for class A in the local storage 200C. Likewise, activity indicators “b” and “c” indicate that classes B and C are also active by having associated state information stored in corresponding CIVs in the local storage 200C.

- The traffic processing unit 200 is further operable to obtain state information
- 30 pertaining to the assigned packet class, e.g. by checking the activity indicator of

the packet class assigned to the received packet and obtaining the state information according to the activity indicator. As indicated above, the activity indicator can be checked in the PCR 200B as illustrated by an **action 2:3**. If the activity indicator indicates the assigned packet class as active, the state information is available in the traffic processing unit and can thus be retrieved from the local storage 200C, as illustrated by an **action 2:4A**. If the activity indicator indicates the assigned packet class as inactive, the state information is fetched from a central storage 204 where state information of different packet classes is maintained, as illustrated by an alternative **action 2:4B**. The central storage 204 may also be denoted a "Consolidation Repository", as in Fig. 2.

Having obtained the state information in either of actions 2:4A and 2:4B, the traffic processing unit 200 is able to perform stateful packet processing of the received packet based on the obtained state information as illustrated by an **action 2:5**.

Thereby, it is an advantage that the traffic processing unit 200 can easily and promptly obtain state information either locally or from a central storage by means of packet classes and associated activity indicators which have been predefined in the traffic processing unit 200. A packet class can easily be assigned to incoming packets based on information in the packet. It is also an advantage that the traffic processing unit 200 can immediately determine whether the state information is available locally or must be fetched from a central storage, based on the associated activity indicator. Thereby, it is not necessary that the distribution device 102 must be aware of how and where any required state information can be obtained since that is determined by the traffic processing unit 200 itself. As a result, the distribution device 102 can be implemented with very simple functionality and to low costs.

An example will now be described with reference to the flow chart in **Fig. 3**, of how the solution may be employed in terms of actions performed by a traffic processing unit such as the above-described traffic processing unit 200. Fig. 3 is described below with further reference to Fig. 2 although without limitation to such a configuration of the traffic processing unit 200. Some optional example embodiments that could be used in this procedure will also be described below.

The actions shown in Fig. 3 are thus performed by a traffic processing unit 200 for handling traffic in a communication network when the traffic is distributed across a set of traffic processing units. It is assumed that the traffic processing unit 200 is capable of receiving traffic flows, e.g. when distributed by traffic distributor such as the above-described distribution device 102.

A first **action 300** illustrates that the traffic processing unit 200 receives a packet of a traffic flow distributed to said traffic processing unit, which corresponds to action 2:1 above. In a next **action 302**, the traffic processing unit 200 assigns a packet class to the received packet, out of a set of predefined classes, based on information derived from the received packet, which corresponds to action 2:2 above. In another **action 304**, the packet class may be detected or noted or in any way observed, as being either active or inactive in the traffic processing unit based on an activity indicator of the packet class maintained in the traffic processing unit, which corresponds to action 2:3 above.

In a further **action 306**, the traffic processing unit 200 obtains state information pertaining to said assigned packet class, which can be made as follows. If the packet class is detected or noted as active, this indicates that the state information is maintained locally in the traffic processing unit 200 which therefore can retrieve the state information from a local storage 200C in the traffic processing unit, which corresponds to action 2:4A above. On the other hand, if the packet class is detected or noted as inactive, this indicates that the state information is not maintained locally in the traffic processing unit 200 which therefore needs to fetch the state information from a central storage 204 where state information of different packet classes is maintained. The latter alternative corresponds to action 2:4B above.

A final **action 308** illustrates that the traffic processing unit 200 performs stateful packet processing of the received packet based on the obtained state information, that is either from the local storage 200C or from the central storage 204.

Some optional example embodiments that could be used in this procedure will now be described. In one example embodiment, if the packet class is detected as

inactive, meaning that any state information pertaining to the packet class was fetched from the central storage in action 306, the traffic processing unit 200 further stores the fetched state information in a memory volume of said local storage associated with the packet class. Thereby, the traffic processing unit 200
5 can build up its own database of state information for different packet classes by repeating the fetching action for further traffic flows, and the possibility of finding state information locally will gradually increase which in turn reduces the load on the network and on the central storage. In particular, fetching all the state
10 information pertaining to the inactive packet class is advantageous since the local storage is thereby updated with valid state information for the new packets that arrive to the processing unit and which are “assigned” the same packet class.

In case the packet class was detected as inactive and state information was fetched and stored as above, another example embodiment may be that the traffic processing unit 200 changes the activity indicator of said packet class from
15 inactive to active. This will indicate that the newly fetched state information is now available locally as stored in the memory volume associated with the packet class. For example, if the locally stored state information is not retrieved and used by the traffic processing unit 200 for a certain prescribed time-out period, the state
20 information may be considered to have expired and can therefore be deleted from the local storage, to save memory space therein.

In another example embodiment, the activity indicator may in the traffic processing unit 200 be maintained in a packet class record 200B with activity indicators of different packet classes for traffic flows that have been handled by the traffic processing unit. This embodiment has also been described above with reference
25 to the PCR 200B in Fig. 2.

In another example embodiment, the local storage 200C in the traffic processing unit 200 may comprise multiple memory volumes containing state information of different packet classes. In another example embodiment, each local memory volume may contain state information of one or more packet classes. It is also
30 possible that state information of one packet class is distributed across more than one local memory volume. How the state information of different packet classes is

stored in one or more local memory volumes is dependent on how the latter are dimensioned and arranged and on the size of the state information.

It was mentioned above that a packet class is assigned to the received packet based on information in the received packet. In another example embodiment, such information derived from the received packet may comprise at least one of a source address and a destination address, which thus can be used for determining the packet class of the received packet. It is also possible that each packet of a flow comprises a flow descriptor or identifier that could be used for assigning the packet class.

10 In another example embodiment, if the state information for said traffic flow is changed, the traffic processing unit 200 may update the local memory volume and the central storage accordingly. In another example embodiment, the traffic processing unit 200 may receive the traffic when distributed by a stateless distribution device 102 such as a layer-3 Ethernet switch. In yet another example embodiment, the traffic processing unit could be implemented as a Virtual Machine, VM, in a cloud environment.

It was mentioned above that the activity indicator for a packet class is changed from inactive to active as state information is fetched from the central storage, and also that it may be changed from active to inactive if the locally stored state information has expired and/or is deemed to be no longer valid. Some examples of how a change of the activity indicator for a packet class may be triggered will be outlined below, with further reference to Fig. 2.

Change from inactive to active

25 In one example, the change from inactive to active may occur due to a concluded fetching of state information from the central storage, i.e. the consolidation repository 204.

In another example, the change from inactive to active may occur as an immediate result of the arrival at the traffic processing unit 200 of the first packet of a packet class. The packet processing for this packet class is then held in abeyance until

the fetching of data from the consolidation repository 204 is concluded for said packet class.

In yet another example, the change from inactive to active may occur in response to a signal from the consolidation repository 204, or in response to a signal from a
5 separate monitoring system function, not shown.

Change from active to inactive

In one example, the change from active to inactive may occur in response to expiry of a timer (timeout) indicating that no packets for the packet class in question has arrived before said timer expiry.

10 In another example, the change from active to inactive may occur in response to deletion of invalid state information from the memory volume in the local storage 200C.

In yet another example, the change from active to inactive may occur in response to a signal from the consolidation repository 204, or in response to a signal from
15 the above-mentioned separate monitoring system function.

An example of how a traffic processing unit may operate in more detail will now be described referring to the flow chart in **Fig. 4**, again with further reference to Fig. 2 but without limitation to the structure depicted therein. The situation in this example is basically the same as in Fig. 3, although certain features and variations
20 are illustrated in more detail.

A first **action 400** illustrates that the traffic processing unit 200 receives a packet of a flow and assigns a packet class to the packet, basically in the manner described above for actions 300-302. In a next **action 402**, the traffic processing unit 200 checks the activity indicator of the assigned packet class, e.g. as
25 maintained in a PCR 200B as of Fig. 2.

If it is found in a next determining **action 404** that the activity indicator indicates the assigned packet class is active, the traffic processing unit 200 can retrieve the state information from a memory unit in the local storage 200C in an **action 406**,

which has also been described above for action 2:4A. On the other hand, if the activity indicator indicates the assigned packet class is inactive, the traffic processing unit 200 must fetch the state information from the central storage 204 in another **action 408**, which has also been described above for action 2:4B. A further **action 410** illustrates that the traffic processing unit 200 performs stateful packet processing of the received packet based on the obtained state information, which has also been described above for action 2:5.

In the procedure of Fig. 4, the traffic processing unit 200 further checks in an **action 412** whether the state information has changed or not, e.g. modified depending on the processing made in action 410. If there is no reason to change the state information, the traffic processing unit 200 can basically move back to action 400 and repeat the procedure for a next received packet. Otherwise, the traffic processing unit 200 will update the state information in both the central storage and the local storage in an **action 414**, and then move back to action 400 for handling the next received packet. The procedure of Fig. 4 can thus be performed for any received packet in one or more traffic flows.

The block diagram in **Fig. 5** illustrates a detailed but non-limiting example of how a traffic processing unit 500 may be structured to bring about the above-described solution and embodiments thereof. The traffic processing unit 500 may be configured to operate according to any of the examples and embodiments for employing the solution as described herein, where appropriate and as follows. The traffic processing unit 500 is shown to comprise a processor P and a memory M, said memory comprising instructions executable by said processor whereby the traffic processing unit 500 is operable as described herein. The traffic processing unit 500 also comprises a communication circuit C with suitable equipment for sending and receiving packets in the manner described herein.

The communication circuit C may be configured for communication with a distribution device, such as the distribution device 102 shown in Fig. 1, including receiving packets of a traffic flow as in action 300, and for communication with a collection device, such as the collection device 104 shown in Fig. 1, using suitable protocols and interfaces. Such communication may be performed over any

suitable interfaces and nodes depending on the implementation, although this is not necessary to describe here as such in any detail. The solution and embodiments herein are thus not limited to using any specific types of networks, technology or protocols for communication.

- 5 The traffic processing unit 500 comprises means configured or arranged to basically perform the actions in Fig. 3 and at least some of the actions in Fig. 4, and more or less as described above for the traffic processing unit 200 in various examples. In Fig. 5, the traffic processing unit 500 is arranged or configured to handle traffic in a communication network when the traffic is distributed across a
10 set of traffic processing units including the traffic processing unit 500.

The traffic processing unit 500 is configured to receive a packet of a traffic flow distributed to said traffic processing unit. This receiving operation may be performed by a **receiving module 500A** in the traffic processing unit 500, e.g. in the manner described above for action 300.

- 15 The traffic processing unit 500 is further configured to assign a packet class to the received packet, out of a set of predefined classes, based on information derivable from the received packet, wherein the packet class is detected or noted as being active or inactive in the traffic processing unit based on an activity indicator of the packet class maintained in the traffic processing unit. The above assigning
20 operation may be performed by an **assigning module 500B** in the traffic processing unit 500, e.g. as described above for action 302. The assigning module 500B could alternatively be named a logic module or analyzing module. Further, a **detecting module 500C** may be employed in the traffic processing unit 500 to perform the above active or inactive detection of the assigned packet class. In
25 case no such detecting module 500C is employed, the detecting or noting the packet class as being active or inactive may basically be performed by the assigning module 500B.

- The traffic processing unit 500 is also configured to obtain state information pertaining to said assigned packet class. If the packet class is detected as active,
30 the traffic processing unit 500 is configured to retrieve the state information from a

local storage in the traffic processing unit, e.g. from the memory M, e.g. as described above for action 406. If the packet class is detected as inactive, the traffic processing unit 500 is configured to fetch the state information from a central storage where state information of different packet classes is maintained, e.g. as described above for action 408. This obtaining operation may be performed by an **obtaining module 500D** in the traffic processing unit 500, basically as described above for action 306. The obtaining module 500D could alternatively be named a state information module or acquiring module.

The traffic processing unit 500 is further configured to perform stateful packet processing of the received packet based on the obtained state information. This operation may be performed by a **processing module 500E** in the traffic processing unit 500, e.g. as described above for action 308. The processing module 500E could alternatively be named to indicate what type of processing it performs, such as a firewall module in case the traffic processing unit 500 is operable as a firewall.

It should be noted that Fig. 5 illustrates various functional units or modules in the traffic processing unit 500, and the skilled person is able to implement these functional modules or in practice using suitable software and hardware. Thus, the solution is generally not limited to the shown structures of the traffic processing unit 500, and the functional units or modules 500A-E therein may be configured to operate according to any of the features and embodiments described in this disclosure, where appropriate.

The functional modules or units 500A-E described above can be implemented in the traffic processing unit 500 by means of suitable hardware and program modules of a computer program comprising code means which, when run by the processor P causes the traffic processing unit 500 to perform at least some of the above-described actions and procedures.

In Fig. 5, the processor P may comprise a single Central Processing Unit (CPU), or could comprise two or more processing units such as CPUs. For example, the processor P may include a general purpose microprocessor, an instruction set

processor and/or related chip sets and/or a special purpose microprocessor such as an Application Specific Integrated Circuit (ASIC). The processor P may also comprise a storage for caching purposes.

Each computer program may be carried by a computer program product in the traffic processing unit 500 in the form of a memory having a computer readable medium and being connected to the processor P. The computer program product or memory in the traffic processing unit 500 may thus comprise a computer readable medium on which the computer program is stored e.g. in the form of computer program modules or the like. For example, the memory may be a flash memory, a Random-Access Memory (RAM), a Read-Only Memory (ROM), an Electrically Erasable Programmable ROM (EEPROM) or Hard Drive storage (HDD), and the program modules could in alternative embodiments be distributed on different computer program products in the form of memories within the traffic processing unit 500.

The solution described herein may thus be implemented in the traffic processing unit 500 by a computer program comprising instructions which, when executed on at least one processor, cause the at least one processor to carry out the actions according to any of the above embodiments and examples, where appropriate. The solution may also be implemented in a carrier containing the above computer program, wherein the carrier is one of an electronic signal, an optical signal, a radio signal, or a computer readable storage product or computer program product.

Some further examples of how the above-described embodiments and features could be employed in practice will now be outlined. The actual processing functionality described herein will be generally referred to as a Central Processing Unit, CPU, which is a common term in the field of computers and data processing.

In the example of a multi-CPU scalable stateful firewall used for high capacity (line-rate) traffic interception, which firewall could be deployed in a so-called "bump-in-the-wire" configuration, the embodiments herein enable scaling-in and scaling-out of stateful firewall processing capacity. This can be done by spreading

the traffic to be intercepted over a set of available CPUs by using a simple stateless flow-based packet distribution device which could be achieved by an ordinary layer-3 Ethernet switch or similar, which is based on a low cost component, e.g. an application-specific standard product, ASSP.

- 5 The embodiments herein can advantageously be realized in a range of products where there is a need for high throughput and dynamical scalability of processing capacity in conjunction with complex processing of state information acquired on-the-fly through deep packet inspection of the traffic. This is often the case for products used in network boundary gateway points, such as, firewalls (FW),
10 security gateways (SEG) and session border gateways (SBG), and in arrangements where scalability is required the embodiments herein enable the use of simple and cost effective stateless distribution devices.

Multiple CPU hardware instances and corresponding parallel firewall processing units can be used to achieve firewall processing capacity at high speed. The CPU
15 processing capacity could be increased by distributing the firewall arriving traffic flow to a set of in-parallel processing firewall units allocated to a corresponding number of CPU hardware instances, using a distribution method called flow based Equal Cost Multi Paths (ECMP). In this distribution method, the arriving packet traffic flow is divided into subsets of traffic flow based on a classification of IP
20 datagram packets according to information in the IP packet header, and then packet transport is conducted in such a way that the subsets of traffic flow are spread out over the set of available "next-hop devices", that is, firewall processing units.

After packets arrive on an individual firewall processing unit which is allocated to a
25 CPU in the described multiprocessor configuration, by the embodiments herein, the datagram packets of each subset of ECMP distributed traffic flow are classified anew into a set of enumerated packet classes, wherein packets of each enumerated packet class are subjected to firewall processing software instructions with a dedicated memory volume at disposal for storage of traffic state information
30 pertaining to packets of the enumerated packet class. This state information may be related to a collection of individual firewall sessions. However, if the

enumerated packet class is detected inactive, the memory volume dedicated for the enumerated packet class will be updated with state information for the enumerated packet class fetched from a consolidation repository, or from a location indicated by the consolidation repository. Then the traffic packets can be
5 subjected to firewall processing software instructions using the fetched state information.

One purpose of the embodiments herein may be to use a simple stateless flow based ECMP distribution device as ingress traffic distributor, and yet facilitate a reconfiguration event of a CPU with an allocated stateful firewall processing unit,
10 being added or removed while the system is in service while still handling ongoing traffic.

For arriving packet traffic, it is common in a stateful firewall to observe protocol state transitions with respect to acquired state information and protocol specific governing rules, and thereafter separate the processing of illicit traffic packets from
15 the processing of legitimate traffic packets. Further, the traffic state information is often dynamic and may be acquired through traffic interception and packet inspection. The traffic state information may also be of semi-static nature, for example, data indicating that a certain traffic-flow is to be rate limited or copied. It should be noted that when packets arrive to firewall processing units the simple
20 stateless ingress ECMP distribution device is completely unaware of any firewall inspected traffic state information data on the firewall processing units, and hence the traffic distribution to firewall processing units is decoupled from the dependencies of data location, that is, where specific firewall session data is located.

25 The classification of packets into enumerated packet classes on the firewall processing units as described herein can be done in a more fine-granular way than the division into subsets of traffic flow done by the bump-in-the-wire ingress ECMP distributor, that is, a subset of flows distributed to an instance of a firewall processing unit, can be classified into one or more enumerated packet classes.

One potential area to which the embodiments herein apply is the function of in-service re-configuration of the firewall in response to adding or removing CPUs. In this case, consistency with respect to the sets of in parallel distributed traffic flows and the sets of in-firewall memorized state information needs to be preserved, e.g. in the event of a re-configuration taking place as a planned operation and maintenance activity adding an extra CPU unit, or when a re-configuration occurs spontaneously, triggered by a detected processor failure.

The embodiments herein may achieve an advantage in that the ingress ECMP device is decoupled from the tasks of relocating firewall state information across CPUs in a reconfiguration scenario such that the traffic distribution device, which maintains no knowledge of how firewall state information is organized or stored on firewall processing units, may on its own accord, for instance in response to detecting the loss of CPU or firewall processing unit (or discovery thereof), change the ECMP distribution pattern of traffic flow. In case one or more subsets of traffic flow spontaneously arrives at another available firewall processing unit than before the change of distribution pattern, the firewall processing units will automatically be adjusted to the new distribution pattern with the rearranged set of available firewall processing units. With the decoupling of the above described tasks, which can be achieved by the embodiments herein, a simple traffic splitting device without any built-in advanced state information aware logic and purpose-specific coordination protocol can be used for the ingress stage of traffic distribution.

In the case of a bump-in-the-wire scalable firewall, the embodiments herein enable an effective bulk relocation of memorized state information, handled per enumerated packet class, done by using a simple stateless ECMP distribution device. It is also possible to perform the task of splitting the traffic at line rate by a distributor-device realized by a merchant silicon ASSP, whereas complex stateful firewall processing is done by general purpose processors.

While the solution has been described with reference to specific exemplifying embodiments, the description is generally only intended to illustrate the inventive concept and should not be taken as limiting the scope of the solution. For example, the terms "traffic processing unit", "traffic distribution device", "state

information", "packet class", "local storage", "memory volume", "central storage" and "activity indicator" have been used throughout this disclosure, although any other corresponding entities, functions, and/or parameters could also be used having the features and characteristics described here. The solution is defined by the appended claims.

5

CLAIMS

1. A method performed by a traffic processing unit (200) for handling traffic in a communication network when the traffic is distributed across a set of traffic processing units, the method comprising:
- 5 - receiving (2:1, 300) a packet of a traffic flow distributed to said traffic processing unit,
- assigning (2:2, 302) a packet class to the received packet, out of a set of predefined classes, based on information derived from the received packet, wherein the packet class is detected (2:3, 304) as being active or inactive in the
- 10 traffic processing unit based on an activity indicator of the packet class maintained in the traffic processing unit,
- obtaining (2:4A/B, 306) state information pertaining to said assigned packet class, such that if the packet class is detected as active the state information is retrieved (2:4A) from a local storage (200C) in the traffic processing unit, and if the
- 15 packet class is detected as inactive the state information is fetched (2:4B) from a central storage (204) where state information of different packet classes is maintained, and
- performing (2:5, 308) stateful packet processing of the received packet based on the obtained state information.
- 20 2. A method according to claim 1, wherein if the packet class is detected as inactive the fetched state information is stored in a memory volume of said local storage associated with the packet class.
3. A method according to claim 2, wherein the activity indicator of said packet class is changed from inactive to active.
- 25 4. A method according to any of claims 1-3, wherein the activity indicator is maintained in a packet class record (200B) with activity indicators of different packet classes for traffic flows that have been handled by the traffic processing unit.

5. A method according to any of claims 1-4, wherein said local storage comprises multiple memory volumes (200C) containing state information of different packet classes.
6. A method according to claim 5, wherein each local memory volume
5 contains state information of one or more packet classes.
7. A method according to any of claims 1-6, wherein said information derived from the received packet comprises at least one of a source address and a destination address, used for determining the packet class of the received packet.
- 10 8. A method according to any of claims 1-7, wherein if the state information for said traffic flow is changed, the local memory volume and the central storage are updated (414) accordingly.
9. A method according to any of claims 1-8, wherein the traffic is received when distributed by a stateless distribution device (102) such as a layer-3 Ethernet
15 switch.
10. A method according to any of claims 1-9, wherein the traffic processing unit is implemented as a Virtual Machine, VM, in a cloud environment.
11. A traffic processing unit (500) arranged to handle traffic in a communication network when the traffic is distributed across a set of traffic
20 processing units, wherein the traffic processing unit (500) is configured to:
- receive (500A) a packet of a traffic flow distributed to said traffic processing unit,
 - assign (500B) a packet class to the received packet, out of a set of predefined classes, based on information derivable from the received packet, wherein the packet class is detected as being active or inactive in the traffic processing unit
25 based on an activity indicator of the packet class maintained in the traffic processing unit,

- obtain (500D) state information pertaining to said assigned packet class, such that if the packet class is detected as active the state information is retrieved from a local storage (M) in the traffic processing unit, and if the packet class is detected as inactive the state information is fetched from a central storage where state
5 information of different packet classes is maintained, and
- perform (500E) stateful packet processing of the received packet based on the obtained state information.

12. A traffic processing unit (500) according to claim 11, wherein if the packet class is detected as inactive the traffic processing unit (500) is configured to store
10 the fetched state information in a memory volume of said local storage associated with the packet class.

13. A traffic processing unit (500) according to claim 12, wherein the traffic processing unit (500) is configured to change the activity indicator of said packet class from inactive to active.

15 14. A traffic processing unit (500) according to any of claims 11-13, wherein the traffic processing unit (500) is configured to maintain the activity indicator in a packet class record with activity indicators of different packet classes for traffic flows that have been handled by the traffic processing unit.

15. A traffic processing unit (500) according to any of claims 11-14, wherein
20 said local storage comprises multiple memory volumes adapted to contain state information of different packet classes.

16. A traffic processing unit (500) according to claim 15, wherein each local memory volume is adapted to contain state information of one or more packet classes.

25 17. A traffic processing unit (500) according to any of claims 11-16, wherein the traffic processing unit (500) is configured to derive from the received packet said information comprising at least one of a source address and a destination address, used for determining the packet class of the received packet.

18. A traffic processing unit (500) according to any of claims 11-17, wherein if the state information for said traffic flow is changed, the local memory volume and the central storage are updated accordingly.

5 19. A traffic processing unit (500) according to any of claims 11-18, wherein the traffic is received when distributed by a stateless distribution device (102) such as a layer-3 Ethernet switch.

20. A traffic processing unit (500) according to any of claims 11-19, wherein the traffic processing unit is implemented as a Virtual Machine, VM, in a cloud environment.

10 21. A computer program comprising instructions which, when executed on at least one processor, cause the at least one processor to carry out the method according to any one of claims 1-10.

15 22. A carrier containing the computer program of claim 21, wherein the carrier is one of an electronic signal, optical signal, radio signal, or computer readable storage medium.

1/4

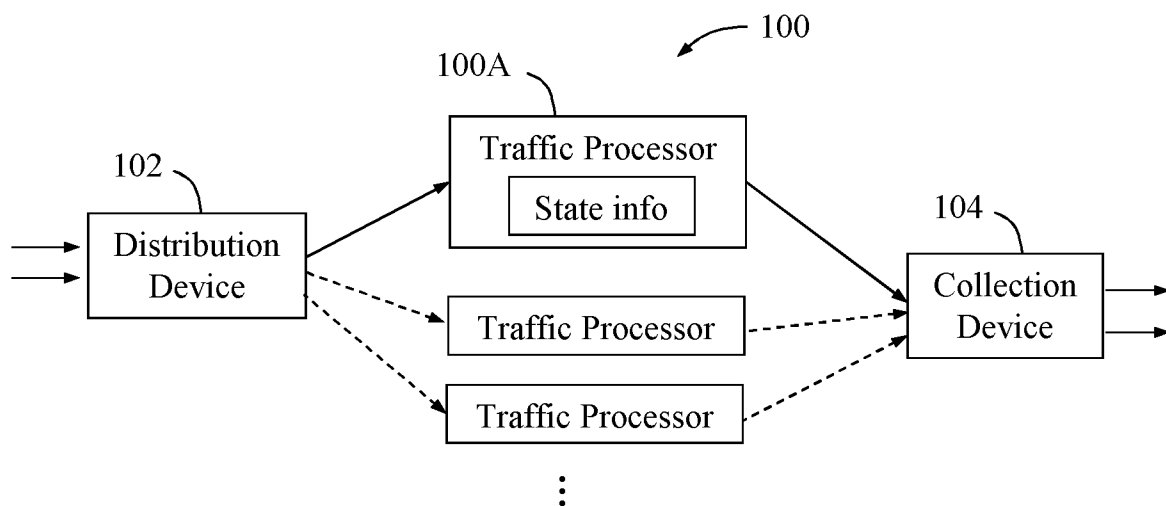


Fig. 1

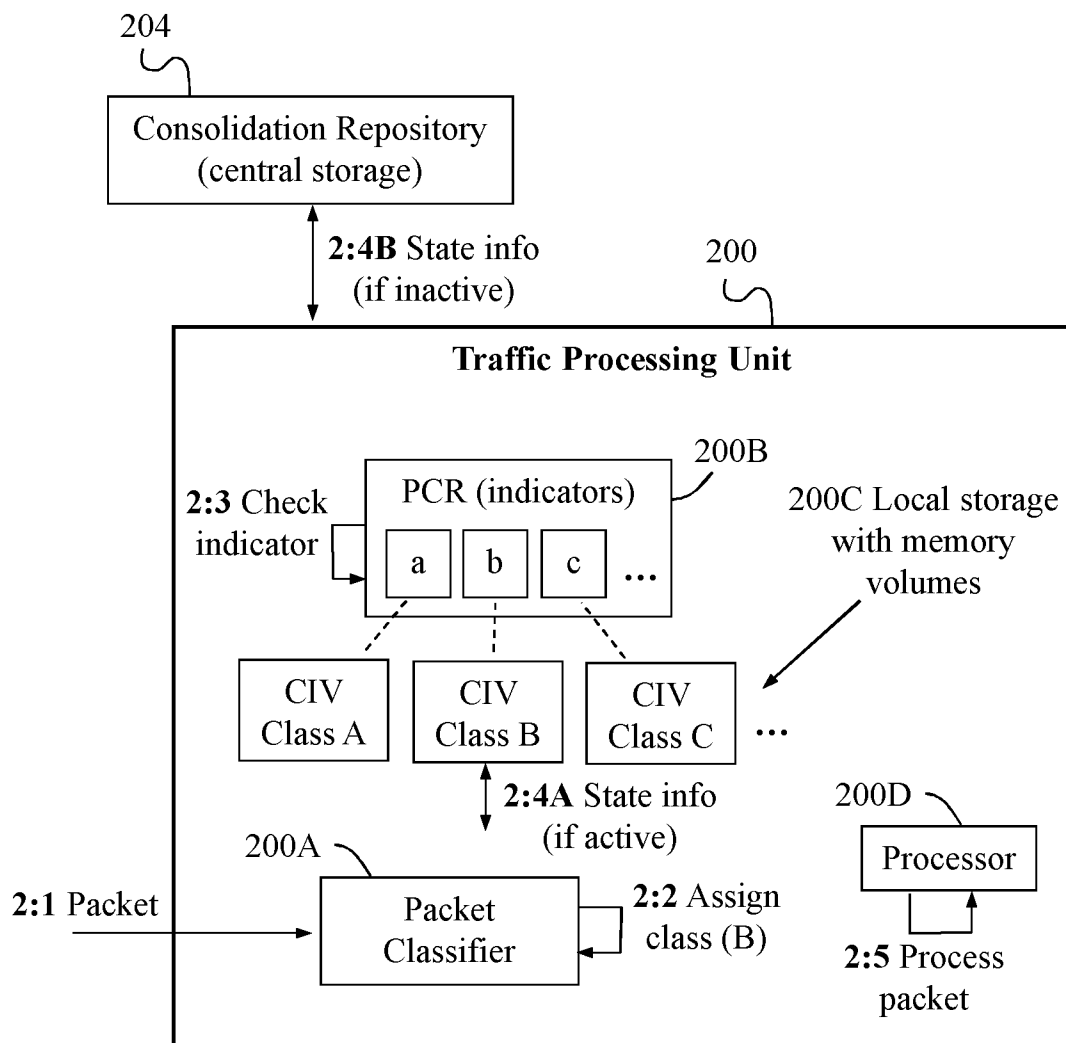
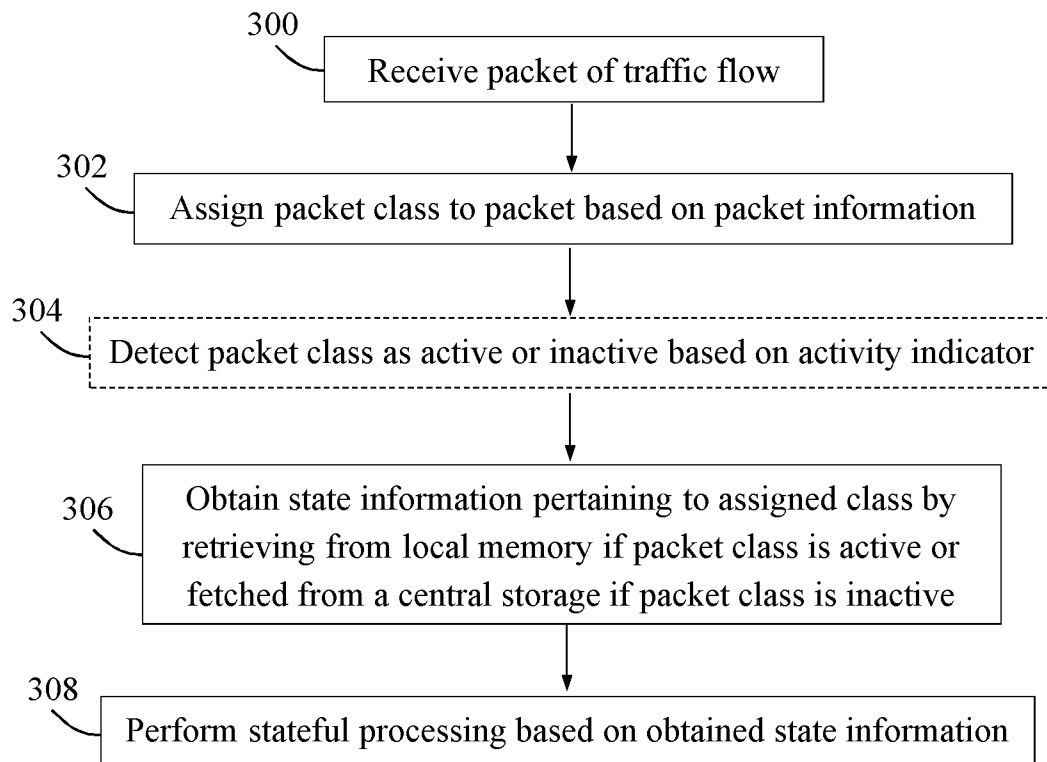


Fig. 2

2/4

**Fig. 3**

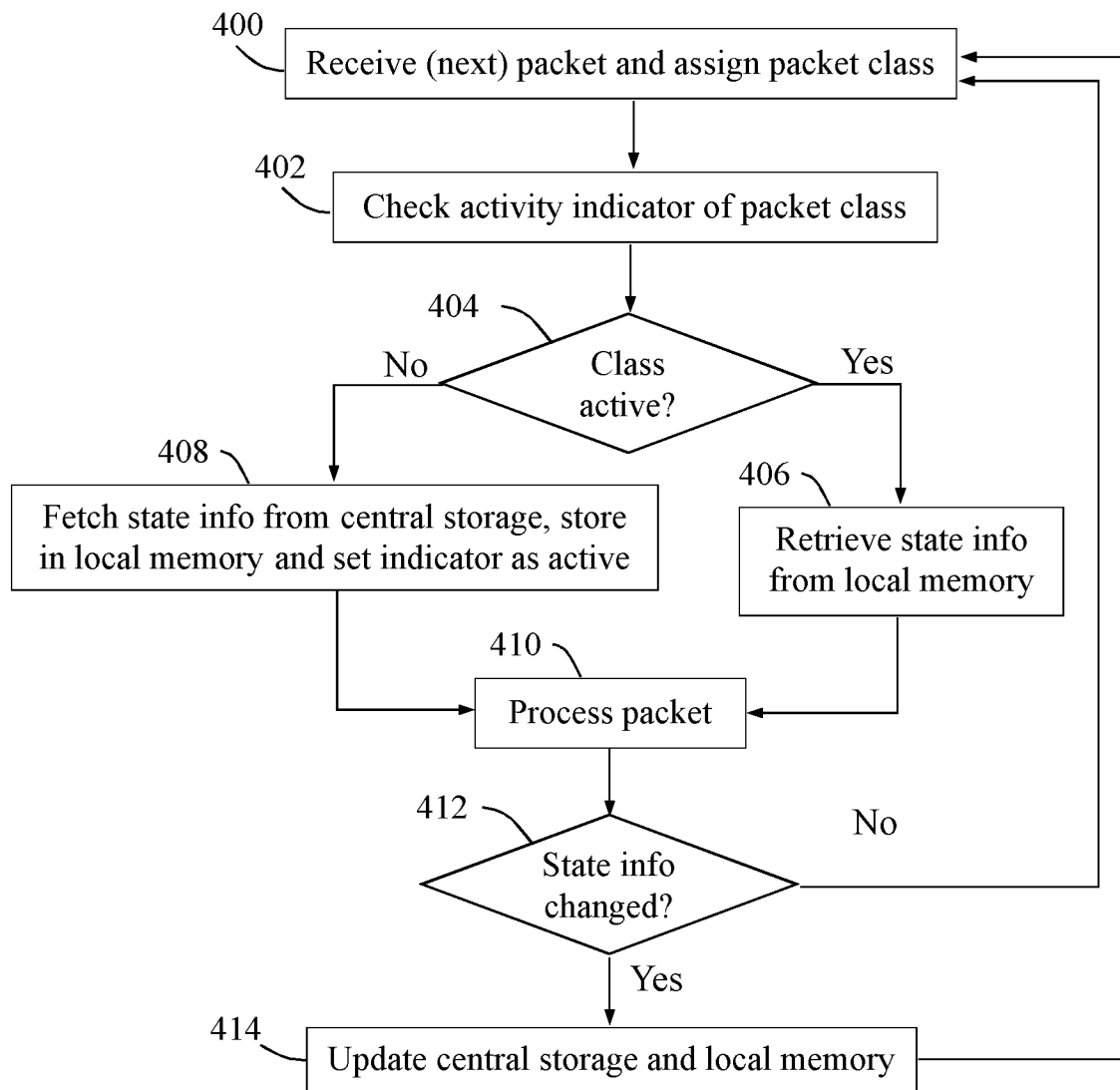


Fig. 4

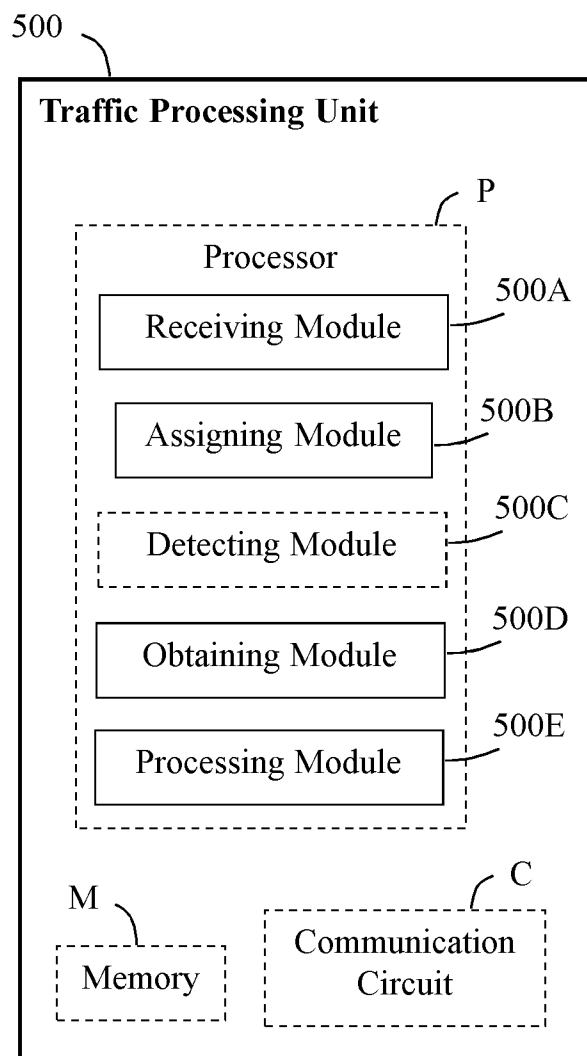


Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE2017/051279

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F, H04L

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

SE, DK, FI, NO classes as above

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

EPO-Internal, PAJ, WPI data, COMPENDEX, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 20130308438 A1 (CALLAN DAVID F ET AL), 21 November 2013 (2013-11-21); abstract; paragraphs [0077]-[0090] --	1-22
A	US 20140226466 A1 (PETTERSSON STEN ET AL), 14 August 2014 (2014-08-14); abstract; paragraphs [0039]-[0049] --	1-22
A	US 20140347993 A1 (HAMMAM TARIK ET AL), 27 November 2014 (2014-11-27); abstract; paragraphs [0029]-[0035] -- -----	1-22



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

"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

28-09-2018

Date of mailing of the international search report

28-09-2018

Name and mailing address of the ISA/SE

Patent- och registreringsverket

Box 5055

S-102 42 STOCKHOLM

Facsimile No. + 46 8 666 02 86

Authorized officer

Anders Edlund

Telephone No. + 46 8 782 28 00

Continuation of: second sheet

International Patent Classification (IPC)

H04L 12/803 (2013.01)

H04L 12/931 (2013.01)

H04L 29/06 (2006.01)

H04L 29/08 (2006.01)

G06F 9/445 (2018.01)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/SE2017/051279

US	20130308438 A1	21/11/2013	US	9288141 B2	15/03/2016
			US	9197545 B2	24/11/2015
			US	20130308459 A1	21/11/2013
			US	20130308439 A1	21/11/2013
			WO	2013173758 A2	21/11/2013
US	20140226466 A1	14/08/2014	US	9923997 B2	20/03/2018
US	20140347993 A1	27/11/2014	EP	2803178 A1	19/11/2014
			US	9935880 B2	03/04/2018
			WO	2013104956 A1	18/07/2013