

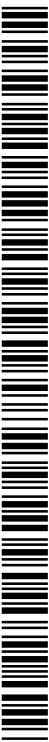


- (51) **International Patent Classification:**
H04L 12/715 (2013.01) *H04L 12/741* (2013.01)
- (21) **International Application Number:**
PCT/EP2016/050549
- (22) **International Filing Date:**
13 January 2016 (13.01.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant: HUAWEI TECHNOLOGIES CO., LTD.**
[CN/CN]; Huawei Administration Building Bantian Long-gang District, Shenzhen, Guangdong 518129 (CN).
- (72) **Inventors; and**
- (71) **Applicants (for US only): WEI, Qing** [CN/DE]; c/o Huawei Technologies Duesseldorf GmbH Riesstr. 25, 80992 (DE). **PEREZ, David** [ES/DE]; c/o Huawei Technologies Duesseldorf GmbH Riesstr. 25, 80992 Munich (DE).
- (74) **Agent: KREUZ, Georg;** Huawei Technologies Duesseldorf GmbH, Riesstr. 8, 80992 Munich (DE).
- (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, 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.

- (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))



WO 2017/121471 A1

(54) **Title:** DYNAMIC FORWARDING RULES IN SDN

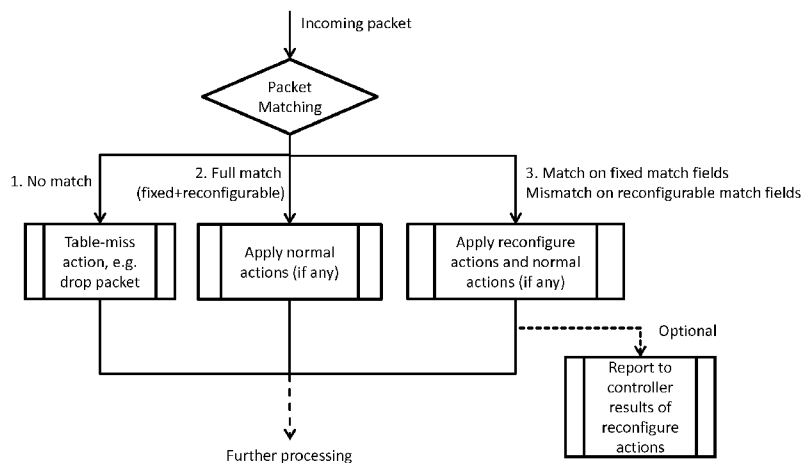


Fig. 9

(57) **Abstract:** A forwarding unit (210, 220) for a software defined networking, SDN, system (10) is provided. The forwarding unit comprises a storage module (212) for storing at least one forwarding rule (300), wherein each of the at least one forwarding rule comprises a match field (305) comprising at least one reconfigurable element (320) and an instruction set (330) comprising at least one reconfiguration action. The forwarding unit (210, 220) is configured to apply a forwarding rule of the at least one forwarding rule to a received data packet by comparing the received data packet with the match field of the forwarding rule (300). The forwarding unit is further configured to carry out the at least one reconfiguration action of the forwarding rule if the received data packet does not match one or more of the at least one reconfigurable element of the forwarding rule.

DESCRIPTION**Dynamic Forwarding Rules in SDN**5 TECHNICAL FIELD

10 The present invention relates to the technical field of data transmission networks, in particular to components used in software defined networks, SDN. Specifically, the present invention relates to a forwarding unit (may also be referred to as switch or switching unit) for a SDN system, a control unit (may also be referred to as controller) for a SDN system, an SDN system, and a method for applying forwarding rules to data packets.

BACKGROUND

15

Software defined networking, SDN, is an approach that basically relates to decoupling the management and controlling tasks from data packet forwarding tasks. The management and controlling tasks are usually referred to as control plane whereas the forwarding tasks are referred to as data plane. This decoupling can simplify the structure of a network and can standardize interfaces between individual components and between the control and data plane. Usually, the data plane is configured such that it requires control commands from the control plane in order to meet the forwarding tasks. Simply speaking, the 'intelligence' of an SDN system is provided in the control plane whereas the data plane simply carries out commands and instructions received from the control plane. One mechanism which relates to and defines the communication between the control plane and the data plane is OpenFlow. It should be understood that any reference to OpenFlow in the following is of exemplary nature and may generally relate to any mechanisms and interfaces which define the communication between the control plane and the data plane in SDN. Reference to OpenFlow is made representative for any of these mechanisms and interfaces.

35 A typical SDN network is composed of simple switches (or forwarding elements) in the forwarding plane and an intelligent SDN controller that configures how those switches behave by installing flow (or forwarding) rules on the switches. For the sake of clarity it should be noted that the terms forwarding rules and flow rules are

used herein as synonyms. In the simplest view, the forwarding rules can be thought of as match-action pairs. The context information (such as the incoming switch port), header and/or other parts of an incoming data flow, frame, packet, datagram or segment (in the following called packet for simplicity) may be matched to the contents of the flow table of the switch, and, in case of a match, the switch may trigger actions, such as forwarding to a certain port, dropping the packet altogether, redirecting the packet to the controller and so on. In case an incoming packet does not match any of the forwarding rules defined on the switch then the switch can send what is called a PACKET_IN consisting of the incoming packet as well as some switch context information (such as incoming port) to the controller. The controller may analyze this PACKET_IN (or an application on top of the controller), which consequently typically will result in either i) installation of new forwarding rules on some switches, e.g. to handle future packets somehow corresponding to the initial one and/or ii) sending out of some packets, e.g. relaying the originally received one. The controller is a logical entity that gathers and keeps an up-to-date per-flow network state. However, such centralization introduces obvious scalability issues as a result of congestion at the control plane and overload of the controller.

Having a centralized controller in the SDN architecture may have many advantages. However, the performance of such controller may decrease in a big scale network or in a highly dynamic environment. More specifically, the limitation of current SDN architecture appears in the following aspects:

Scalability: The performance of the controller is closely related to the number of flows to be managed in parallel;

Latency: The response time to dynamic changes depends on the distance between the controller and SDN switches as well as the number of dynamic events to be processed at the controller simultaneously;

Resiliency: Since the controller is the brain of the SDN network, resilience of the controller and control channel greatly affects the correct operation of the complete SDN network

There have been described several attempts to solve the problems stated above. US 2015/163151 A1 describes a hierarchical control architecture at the expense of increasing the synchronization signaling between the distributed controllers. Other approaches focus on agent based solutions, where a local agent is installed in the

SDN switch and takes over some of the tasks from the applications sitting on the central controller. SoftOffload: "A programmable approach toward collaborative mobile traffic offloading", MobiSys '14, describes, for instance, delegating some of the offloading functionalities to local agents to reduce the redundant signaling and the processing overhead at the controller. However, agent based approaches require additional application layer processing capability at the SDN switch, which might not always be possible.

Open vSwitch (see <http://openswitch.org/support/dist-docs/ovs-ofctl.8.txt>) provides an action "learn" that allows modifying a flow table based on the content of the flow currently being processed. In this case, the switch must have the processing capability to perform a set of logical operations (e.g. store/load values in an internal switch registry). The processing of "learn" action is software based. Whenever the switch gets the "learn" command from the controller, it will always perform expensive "learn" action regardless whether "learn" is needed or not. Open vSwitch manual suggests not use "learn" mechanism in the main flow table (table=0) of the switch because of performance reasons. The "learn" action can only operate on forwarding rules that were produced by the same action.

SUMMARY

There may be a need of reducing the load of the control plane in an SDN system.

This object is achieved by the features of the independent claim. Further implementation forms are apparent from the dependent claims, the description and the figures.

According to an aspect of the invention, a forwarding unit for a software defined networking (SDN) system is provided. The forwarding unit comprises a storage module for storing at least one forwarding rule, wherein each of the at least one forwarding rule comprises a match field comprising at least one reconfigurable element and an instruction set comprising at least one reconfiguration action. The forwarding unit is configured to apply a forwarding rule of the at least one forwarding rule to a received data packet by comparing the received data packet with the match field of the forwarding rule. The forwarding unit is further configured to carry out the at least one reconfiguration action of the forwarding rule if the received data packet

does not match one or more of the at least one reconfigurable element of the forwarding rule.

5 The forwarding unit can be described as a switch, i.e., an element of the data plane of the SDN. In the following, the terms forwarding unit and switch shall be interpreted as synonyms, unless stated otherwise. Such a switch may have at least one or more than one forwarding rule or generally any number of forwarding rules. The forwarding rule(s) is (are) stored in a storage module which can be generally referred to as a memory of the forwarding unit and which in particular can comprise
10 a persistent memory.

The match field is a part of the forwarding rule, wherein the match field comprises at least one condition (this condition can be referred to as “elements” of the match field in the following) which must be met in order to carry out the actions defined by the
15 forwarding rule. In other words, there is a conditional relation between the match field and the actions of a forwarding rule.

The reconfigurable element is part of the match field and triggers the reconfiguration action if the reconfigurable element is not met by a data packet. The match field may
20 comprise n (with n equal to or greater than 1) reconfigurable elements where the reconfiguration action is carried out if at least one of the reconfigurable elements is not met. Not meeting a reconfigurable element may also be referred to as a mismatch.

25 A forwarding rule is applied to a received data packet, i.e., the content of fields of the data packet is compared with the match field of a forwarding rule which can also be described as examining the data packet in view of the conditions defined in the match field of a forwarding rule. The match field may comprise one or more elements, i.e., conditions. In case of a rule match, all conditions of the match field
30 are met by definition. A partial match (which necessarily is a partial mismatch), will be referred to as a mismatch in the following.

It is understood that when discussing a single condition, matching a condition is equivalent to mismatching an opposite condition.

35

A reconfiguration action defined in the instruction set of a forwarding rule does not contribute to forwarding data packet but relates to an internal configuration action of the forwarding unit. The reconfiguration action can be applied to that forwarding rule which contains the reconfiguration action and/or to another forwarding rule stored in the storage module. In other words, the reconfiguration is used to carry out any configuration or reconfiguration of the forwarding unit and the forwarding rules contained in the forwarding unit.

The forwarding unit described herein enables carrying out reconfiguration actions, which are in particular related to a forwarding rule. Therefore, the forwarding unit must not generate and send a PACKET_IN message to the controller in case of a rule mismatch, for example if the forwarding unit does not contain any forwarding rule which can be applied to a data packet. As a result, communication between the forwarding unit and a controller can be reduced and the data traffic and the risk of network congestion on the control plane as well as the load of the controller may be reduced. In other words, simple forwarding rule modification tasks are outsourced from the controller to the forwarding unit which can be carried out with low computational power such that the forwarding unit still meets the requirements of SDN.

The condition for initiating the reconfiguration action being carried out is a mismatch between the data packet and the reconfigurable element of the match field. The reconfiguration action does not necessarily amend the content of the reconfigurable element but may do so. The reconfiguration action is carried out if the data packet does not match at least one reconfigurable element of the match field.

According to an embodiment of the invention, the match field of the forwarding rule comprises at least one static element, wherein the forwarding unit is configured to carry out the at least one reconfiguration action of the forwarding rule if the received data packet additionally matches the at least one static element of the forwarding rule.

The static elements define a minimum set of criteria which must be met by a data packet in order to match the forwarding rule. If the reconfigurable elements are additionally met, the forwarding rule as a whole (all conditions of the match field) is met and the reconfiguration action of the instruction set is not carried out. In this

case, a forwarding action of the forwarding unit will be carried out typically, if the forwarding rule contains such a forwarding action. On the contrary, in case of a mismatch of the reconfigurable element (if the reconfigurable element is not met), the reconfiguration action is carried out. Thus, a forwarding rule can be defined such
5 that the reconfiguration actions can react on an amended content of specific fields of data packets.

A forwarding rule can be defined such that the reconfiguration action is carried out under very specific conditions, i.e. when the static elements are met and when at
10 least one of the reconfigurable elements is not met. The static elements in the match field are required to identify a flow entry (a specific flow rule or forwarding rule) in the flow table. If the match field does not comprise any static elements, the incoming packets will be treated as a „table miss“- „no matching“ and trigger the reconfiguration action. A forwarding rule can comprise only reconfiguration action
15 without any forwarding action

According to a further embodiment of the invention, the match field of the forwarding rule comprises more than one reconfigurable elements and the forwarding unit is configured to carry out the at least one reconfiguration action of the forwarding rule if
20 the received data packet does not match all of the at least one reconfigurable elements of the forwarding rule.

According to a further embodiment of the invention, the match field of the forwarding
25 rule comprises a forwarding action, wherein the forwarding unit is configured to carry out the forwarding action of the forwarding rule if a received data packet matches the match field of the forwarding rule.

In case a data packet matches the match field (both static and reconfigurable
30 elements), the data packet is forwarded according to the forwarding action and the reconfiguration action will not be carried out.

The forwarding unit is configured to determine whether or not there is a match of a data packet to a forwarding rule. In case of a match, the forwarding action is carried
35 out while in case of mismatching at least one reconfigurable element of the match

field, the reconfiguration action is carried out while the forwarding action is not carried out.

5 According to a further embodiment of the invention, the forwarding unit is configured to modify at least one reconfigurable element of the at least one forwarding rule (in one embodiment: all forwarding rules stored in the storage unit) when carrying out the at least one reconfiguration action of the forwarding rule (and/or of other forwarding rules) if a received data packet does not match one or more reconfigurable elements of the forwarding rule.

10

In case of a rule mismatch, the forwarding unit can by itself modify a forwarding rule and no PACKET_IN to the controller is required. The reconfiguration action can be formulated such that in case of a mismatch a reconfigurable element of a match field of the mismatched rule can be adapted and/or reconfigurable elements of one or 15 more forwarding rules can be adapted, where these forwarding rules may be identified in the reconfiguration action. The reconfiguration action can be defined when configuring the forwarding rule, i.e. the reconfiguration action is part of the forwarding rule when being transmitted from the control unit to the forwarding unit.

20 According to a further embodiment of the invention, the forwarding unit is configured to delete at least one of the at least one forwarding rule if a received data packet does not match one or more reconfigurable elements of the forwarding rule.

25 The storage space in the storage module is limited and the number of active forwarding rules is limited due to that space limitation, too. Deleting a rule may release storage space in the storage module and other forwarding rules can be activated or stored therein alternatively.

30 According to a further embodiment of the invention, the forwarding unit is configured to modify at least one or a forwarding rule if a received data packet does not match one or more reconfigurable elements of the forwarding rule.

35 Modifying a forwarding rule may include duplicating an existing rule (optionally with subsequent modifications of at least one field of the forwarding rule) or modifying an existing forwarding rule (this may be the forwarding rule containing the reconfiguration action and/or another forwarding rule stored in the storage module).

The copied/duplicated forwarding rule can be used to modify one of these forwarding rules while the other one is kept as it is. This may be useful under certain conditions where the data packets of a data flow alternate in time, thus resulting in a first type and second type of data packet. Duplication and modification of one of the forwarding rules enables that there is a forwarding rule for both types of data packets while at the configuration time of the switch only one forwarding rule is transmitted to it. The specific forwarding rule for the second type of data packets is dynamically created by the switch based on the reconfiguration instructions and by using simple operations like replacing the content of existing fields.

10

According to a further embodiment of the invention, the forwarding unit is configured to generate at least one forwarding rule if a received data packet does not match one or more reconfigurable elements of the forwarding rule.

15 The reconfiguration action may contain all relevant information for generating a forwarding rule. By generating a forwarding rule from the scratch, the forwarding table of the forwarding unit can be adapted to a specific data transmission scenario (type of data packets transmitted to and handled by the switch) or to a varying data transmission scenario without requiring additional commands from the controller.

20

According to a further embodiment of the invention, the reconfiguration action of the forwarding rule contains an association instruction and the forwarding unit is configured to associate the forwarding rule to at least a second forwarding rule.

25 In other words, a first forwarding rule can refer to at least one second forwarding rule, which can be any forwarding rule in the same forwarding unit.

The forwarding rule association can be applied to existing forwarding rules or to newly generated forwarding rules. If a new forwarding rule is generated based on the reconfiguration information of a first forwarding rule, the new forwarding rule is associated to the first forwarding rule. This may simplify the forwarding rule management.

30 Forwarding rule association may be a general feature of the forwarding unit as described herein. Forwarding rule association may particularly be used to point to

35

the object of reconfiguration action (e.g., the forwarding rule containing said reconfiguration action or any other forwarding rule).

5 In the following, a forwarding rule may be referred to as dynamic forwarding rule (DFR). This terminology indicates that the forwarding rule, or DFR, can be modified dynamically. Forwarding rule association may be used to show the parent/child relationship between the modifying DFR and modified/generated DFR. This relationship can be used for the forwarding rule management afterwards, e.g., when the parent DFR changes, the child DFR may be changed as well.

10

For example, the reconfiguration action of a first forwarding rule may act on a second forwarding rule (which is the associated forwarding rule, while there may be more than one associated forwarding rules to the first forwarding rule). A reconfiguration action may contain a reference to a second forwarding rule. The association of a forwarding rule to a second one is decided by the controller unit or the forwarding unit in case it generates a forwarding rule based on an existing forwarding rule.

15

A first forwarding rule's reconfiguration action may act on a second forwarding rule (which is the associated forwarding rule, the second forwarding rule can be more than one). A reconfiguration action may contain a reference to a second forwarding rule. The association of a forwarding rule to another one is decided by the controller unit or the forwarding unit in case it generates a forwarding rule based on an existing forwarding rule.

25

According to a further embodiment of the invention, the forwarding unit is configured to generate a forwarding rule having a match field and an action field, wherein the match field has at least one static element and at least one reconfigurable element and wherein the action field has at least one forwarding action and at least one reconfiguration action.

30

Thus, the reconfiguration action can generate dynamic forwarding rules and the generated forwarding rules themselves are reconfigurable.

35 According to a further embodiment of the invention, the forwarding unit is configured to adapt an output port of a forwarding rule when carrying out the at least one

reconfiguration action of the forwarding rule, wherein the output port is adapted based on an input port of a received data packet.

5 For example, in case of data communication between two hosts A and B and if one of these hosts is a mobile host (say for example host A) and its connection port to a forwarding unit (this is the input port or incoming port of the forwarding unit at which the data packet from host A is received) changes, the reconfiguration action can adapt the forwarding rule for the reverse data flow to use the incoming port of data packets of the mobile host A (the data packets sent from mobile host A to host B) as
10 the outgoing port for the response data packets (the data packets from host B sent to the mobile host A). The reconfiguration action can be formulated such that the forwarding unit uses information of the received data packets to reconfigure existing forwarding rules. In the example above, the incoming port of a data packet is used to adapt an existing forwarding rule.

15

According to a further embodiment of the invention, the forwarding unit is configured to adapt the output port of a forwarding rule based on a source address and the input port of a received data packet.

20 Thus, a flow path can be established without asking the controller of the SDN. The static fields contain at least the destination address where the reconfigurable fields contain the source address and the incoming port and where the reconfiguration action is defined. The reconfiguration action contains the new output port while the source address and/or the input port of the data packet are reconfigurable match
25 fields. Thus, depending on the source address and the input port, the forwarding unit can act as a bridge without asking the controller.

30 According to a further embodiment of the invention, the forwarding unit is configured to report result of the at least one reconfiguration action to a control unit of the SDN system.

By reporting the reconfiguration result to a controller (control unit), the forwarding unit updates the controller so that the controller is aware of the current status of the SDN. At least the content of the forwarding rules of the forwarding units of a SDN
35 are used for defining the status of the SDN.

The processing capability of a switch determines how a DFR will be implemented. For instance, if a switch is DFR-enabled, the controller can install a DFR as is in such switch. If the switch is not DFR-enabled, the controller needs first to decompose a DFR into one or several flow mod operations. Such decomposition
5 also depends on other processing capabilities of the target switch, such as pipeline processing, or version of OpenFlow, for example. This step can decouple the definition of DFR at the controller from the actual implementation of DFR at the switch and provide a general solution for heterogeneous SDN networks with different types of SDN switches.

10

According to a further aspect of the invention, a control unit for a software defined networking (SDN) system is provided. The control unit is configured to configure at least one forwarding unit of the SDN system. The control unit comprises a configuration module for generating configuration commands for a forwarding unit,
15 wherein the configuration command comprises at least one forwarding rule having a match field with at least one reconfigurable element and an instruction set comprising at least one reconfiguration action to be applied to a forwarding rule by the forwarding unit in case a data packet received by the forwarding unit does not match one or more of the at least one reconfigurable element.

20

Generally, in an SDN the control unit configures the forwarding unit, as described above. In respect of DFR, the control unit configures the forwarding unit such that the control unit provides at least some ability of reconfiguration to the forwarding unit and the control unit does not need to be involved in any reconfiguration action. The
25 configuration commands used by the control unit for configuring the forwarding unit must therefore contain at least the match field of at least one forwarding rule and the instruction set with the reconfiguration action. In respect of the structure of the forwarding rule, reference is made to the details provided above relating to the forwarding unit.

30

During runtime (when the SDN and its components are used to forward data packets), the forwarding unit carries out the reconfiguration action in case of a mismatch while, prior to carrying out data packet forwarding, the control unit provides specific configuration commands to the forwarding unit. The control
35 commands may comprise the reconfigurable forwarding rules. In other words, the configuration commands of the control unit are functionally interrelated with the

forwarding rules of the forwarding unit. Thus, reference can be made to the details provided above relating to the function of the forwarding unit, while these details are not repeated herein.

- 5 The configuration commands may be provided to the forwarding unit via a configuration interface.

According to a further aspect of the invention, a software defined network, SDN, system is provided. The SDN system comprises at least one control unit as
10 described above and at least one forwarding unit as described in any one of the embodiments above, wherein the at least one control unit is configured to configure at least one forwarding rule of the at least one forwarding unit.

According to a further aspect of the invention, a method for applying forwarding
15 rules of a software defined networking, SDN, system is provided. The method comprises the steps of:

- receiving at least one forwarding rule which comprises a match field with at least one reconfigurable element and an instruction set with at least one reconfiguration action;
- 20 applying a forwarding rule of the at least one forwarding rule to a received data packet by comparing the received data packets with the match field of the forwarding rule;
- carrying out the at least one reconfiguration action of the forwarding rule if the received data packet does not match one or more of the at least one reconfigurable
25 element of the forwarding rule.

For details relating to the method according to this aspect, reference is made to the details provided above when describing the function of the forwarding unit. Summed up, the forwarding unit receives forwarding rules provided by the control unit and
30 applies the forwarding rules to data packets. In case the conditions for carrying out a reconfiguration action are met (at least one reconfiguration action is not met), said reconfiguration action is carried out.

In other words, the SDN system and its components, i.e., the forwarding unit and the
35 control unit, may be described as follows: The SDN system supports Dynamic Forwarding Rules (DFR), wherein the controller defines the DFRs and selects the

candidate SDN switches to install the DFRs, where the DFR includes in one embodiment: two different match fields, where one can be a fixed match field and the other one can be a reconfigurable match field, reconfiguration actions and forwarding actions, wherein the SDN switches are enhanced to execute the DFRs. A
5 DFR can have different implementation at the SDN switches, e.g., two flow entries with different priority and flow entries in linked flow tables or one flow entry with both fixed and reconfigurable match field. The DFR can be converted into a set of configuration messages according to the capability of target SDN switches by the controller; or DFR can be converted into a set of flow entries by target SDN switches
10 according to their capability. The fixed match field may be used also as the trigger for the reconfiguration actions. The reconfiguration actions include modifying the DFR itself and/or associated forwarding rules at the same SDN switch and/or generate associated forwarding rules at the same SDN switch. The SDN switch may be configured to detect the trigger of DFR (two steps/priority based matching/parallel
15 matching), perform reconfiguration actions, gather the required information, identify the associated forwarding rules, identify the reconfigurable match fields, modify the reconfigurable match fields of itself or associated forwarding rules, generate associated forwarding rules, notify the controller on the modification of the DFRs and generated associated forwarding rules. When the DFR is removed or expired,
20 the associated forwarding rules/DFRs may be removed as well.

The approach described herein does not require complex logical operations, as it relies on standard matching operations. So it can be easily accelerated by hardware or implemented in the kernel space instead of user space, and thus
25 providing increased performance. In addition, the approach described herein is highly flexible in the sense that it allows a first forwarding rule to perform modifications on associated forwarding rules that were not installed by the first forwarding rule.

30 The goal is to increase the programmability of SDN switches and increase the efficiency of SDN network in large scale, dynamic environments (e.g., mobile network). A forwarding rule based mechanism is described which gives certain flexibility to the SDN switches to adapt their forwarding behaviour according to local dynamic information. This approach is based on processing capabilities of SDN
35 switches, it does not require a local agent. It is also independent of the control architecture and can avoid the need for complex control architecture in some case.

DFRs are computed and installed by the controller. However, different from static forwarding rule, a DFR includes a triggering condition and additional reconfigure actions to the normal (for example forwarding) actions. This enables DFR to modify itself and/or generate/modify associated forwarding rules, providing a fast response to local changes.

Below, different procedures to install a DFR at the SDN switches are described. Furthermore, the procedure to process the DFR at the SDN switches using their existing processing capabilities (i.e., matching, forwarding rule configuration) and three possible implementations thereof are described.

The usage of DFR is exemplified using two use cases: UE mobility management and local bridging. Basically, DFR can be applied to all use cases where it is beneficial to have forwarding rules adaptation locally at the SDN switch without asking the controller.

Compared to static forwarding rules, DFR provides a mechanism to reconfigure the switch flow table locally and automatically, which reduces the configuration time for the SDN switches as well as the computation load at the controller. Comparing to the "learn" capability of virtual SDN switches, DFR does not depend on the intelligent processing capability (i.e., programming) of a virtual SDN switch. It uses a fixed format of forwarding rules and the actual implementation can be easily adapted to the capability of different target SDN switch. This enables the general usage of DFR in a big scale network with different types of SDN switches and improves the performance of the data plane processing.

Similar to other approaches to overcome the limitations of SDN networks, DFR provides short reconfiguration time and fast response to changes at the switch level. It increases the scalability of SDN networks, as it delegates processing from controller to SDN switches and reduces the signaling from the controller. The difference is that DFR reuses the existing matching and configuration capability of the SDN switch. It has low complexity compared to agent based approach or hierarchical/distributed controller approach. No additional trigger signal is needed due to the smart combination of fixed and reconfigurable match field. Meanwhile, the flow table size is reduced by local forwarding rule adaptation/generation. DFR thus

provides a general mechanism which makes the SDN network more flexible and dynamic. It increases the dimension of programming space in SDN networks.

BRIEF DESCRIPTION OF THE FIGURES

5

Embodiments of the invention will be described with respect to the following figures, in which:

10 Fig. 1 schematically shows an SDN system according to an exemplary embodiment of the invention;

Fig. 2 schematically shows a forwarding unit according to an exemplary embodiment of the invention;

15 Fig. 3 schematically shows a control unit and a forwarding unit according to an exemplary embodiment of the invention;

Fig. 4 schematically shows the underlying infrastructure of an SDN system based on a global network view;

20

Fig. 5 shows a schematic functional overview of a forwarding unit according to the OpenFlow specification;

Fig. 6 schematically shows a flow table processing pipeline of a control unit;

25

Fig. 7 schematically shows the structure of a forwarding rule in a forwarding unit according to an exemplary embodiment of the invention;

30 Fig. 8 schematically shows the steps carried out by a control unit according to an exemplary embodiment of the invention;

Fig. 9 schematically shows the steps carried out by a forwarding unit according to an exemplary embodiment of the invention;

35 Fig. 10 schematically shows a scenario for applying forwarding rule reconfiguration according to an embodiment of the invention;

Fig. 11 schematically shows a scenario for applying forwarding rule reconfiguration according to an embodiment of the invention.

5 DETAILED DESCRIPTION OF EMBODIMENTS

Fig. 1 provides an overview of the structure of an SDN system 10. The control plane 100 is separated from the data plane 200. The control plane 100 is formed by at least one control unit 110 which controls the configuration and the functioning of the data plane, in particular the configuration of forwarding rules on the forwarding unit 210, 220 of the data plane.

The SDN system 10 and its components are configured to carry out the functions as described above with reference to the forwarding unit, the control unit, the SDN system and the method for applying forwarding rules to data packets.

Fig. 2 shows a forwarding unit 210 comprising a first port 211A, a second port 211B, and a storage module 212.

The first port may be an incoming port (receiving data packets) and the second port may be an outgoing port (transmitting outgoing data). The forwarding unit may comprise more than two ports and any one of these ports can be configured to receive and transmit data packets. A forwarding rule may be configured to determine the outgoing port for any received data packet thus defining the route of the data packet through the data plane.

Fig. 3 illustrates the relation between a control unit 100 which comprises a configuration module 112 and a forwarding unit 210. The control unit is configured to send configuration commands to the forwarding unit 210, wherein the configuration commands define the forwarding rules at the forwarding unit including the match field and the instruction set as described above and hereinafter.

Figs. 4, 5, and 6 provide a schematic overview of SDN.

In Software Defined Networking (SDN), control and data plane are decoupled, as can be seen in Fig. 4. A logically centralized controller defines the packet forwarding

and processing behaviour of SDN switches in the data plane by installing forwarding rules in their flow tables.

SDN is a dynamic and manageable architecture. It decouples the network control
5 and forwarding planes from each other. The network can be now programmed from
a logically centralized controller. Such controller can host many control applications
that decide the forwarding behaviour of the underlying infrastructure based on a
global network view as shown in Fig. 4. The desired behaviour is installed in the
network forwarding elements (e.g. SDN switches) by means of forwarding rules by
10 the controller. One of the most widely used protocols to perform this kind of
configuration in SDN switches is OpenFlow (see Fig. 5). Usually, forwarding rules
are static, i.e., a forwarding rule does not change its content.

The organization of the flow tables in the SDN switch is shown in Fig. 6. Except for
15 the group table, the SDN switch maintains a chain of flow tables (pipeline
processing). Forwarding rules are specified in the flow table as table entries that
include match fields and actions (instruction set). The set of possible actions include
sending the incoming packet to the controller, packet header modification, sending
the packet to another flow table, output the packet to a given switch port, etc. An
20 incoming packet will be processed by the table pipeline according to the match fields
and actions.

After the forwarding rules are specified by the controller, they are statically installed
at the SDN switches. If a forwarding rule needs to be modified (e.g. change match
25 fields or actions), the controller needs to interact again with the given switch in order
to reconfigure it.

In case the matching condition of a packet changes (e.g., packet coming from a
different port in the switch due to User Equipment (UE) mobility), if there are no
30 forwarding rules installed for this new condition, the switch will send the incoming
packet to the controller wrapped into an SDN control message (PACKET_IN
message). The controller will analyse the incoming packet and update the existing
forwarding rule accordingly. This round trip of control signalling to the controller
consumes time and introduces signalling overhead. In case of a network with many
35 moving UEs (e.g., one eNB (ENodeB, Evolved Node B) of a mobile communication
network has 3-6 cells and each cell can support ca. 200 active UEs, and one mobile

network could have 100,000+ eNBs), the control signalling to the controller due to simultaneous UE mobility events could generate congestion of the control channel and huge processing load at the controller. The response latency to the local changes includes the processing time at the controller, queuing time at the control
5 channel, round-trip time of the control signalling and the configuration time of individual SDN switch. The response latency may be lower bounded by several milliseconds. Such latency may not be acceptable for certain time critical applications (e.g., Tactile internet may require less than 1ms end to end delay) and limits the space of the SDN architecture.

10

Fig. 7 schematically shows a forwarding rule 300 comprising a match field 305 and an instruction set 330. The match field 305 comprises a static element 310 and a reconfigurable element 320, wherein the reconfigurable element 320 can be reconfigured by a reconfiguration action stored in the instruction set 330 of the same
15 or another forwarding rule.

Using such reconfigurable forwarding rules (also referred to as DFR) may have several advantages, as described in the following.

20 Similar to static forwarding rules, DFR consists of two main elements, match fields 305 and instruction set 330. Basically, DFR can be installed in any flow table where static forwarding rule is installed. DFR can also be installed or partially installed in different flow table as in the case of static forwarding rules.

25 The DFR 300 of Fig. 7 includes two types of match fields: fixed match fields (static element 310) and reconfigurable match fields (reconfigurable element 320). The match field 305 is being applied to an incoming data packet in order to determine if the matching criteria are met. The reconfigurable match fields 320 follow the same format as the fixed match fields. The initial values of the reconfigurable match fields
30 are specified by the controller. Both fixed and reconfigurable match fields match one or several packet header fields, e.g. source IP address, the metadata value added by the SDN switch, e.g., the packet ingress port, and other pipeline fields. The SDN switch triggers some actions (describe below) when a change in the current value of the reconfigurable match field(s) is detected, e.g., a change in the input port (set as reconfigurable match field) of a packet matching the fixed match fields triggers an
35 action that modifies another forwarding rule in the flow table.

The instruction set 330 includes normal (forwarding) actions, e.g., forward the packet to a port, modify the packet by decrementing the TTL field, or change its state, etc. In addition, it includes reconfigure actions, which act on the flow table of the SDN switch, e.g., modification of the reconfigurable match fields, or
5 modification/generation of an associated forwarding rule. If a packet matches fixed and reconfigurable match fields, but no change in the reconfigurable match field(s) is detected by the SDN switch, normal actions (if specified) are executed.

10 A DFR 300 can have an associated forwarding rule e.g., the downlink forwarding rule belonging to a mobile node can be associated with its uplink forwarding rule. A new generated forwarding rule can be associated with the DFR which generates it. Such associated forwarding rule can be changed or installed as specified in the reconfigure action of the DFR. The associated forwarding rule can be either a
15 normal forwarding rule or a DFR. In static forwarding rules, there is no explicit rule association concept but rule identification method. E.g., the controller uses cookie field to identify a certain/group of forwarding rules for certain action. While for DFR case, forwarding rule association may provide a hierarchical forwarding rule relationship chart, e.g., DFR which generates a forwarding rule is the parent and a
20 forwarding rule generated by that DFR is the child. An uplink DFR can be the parent of a downlink DFR. Such a chart facilitates the local control of the forwarding rules at the SDN switches.

Fig. 8 shows the procedure at the controller side to prepare using the approach of
25 DFR.

After receipt of a DFR request (e.g., from a service), the controller specifies the DFR (the fixed and reconfigurable match fields, normal actions if any, and reconfigure actions) and associated forwarding rules (if any). The associated forwarding rules
30 can be DFRs as well. Afterwards, the controller selects the candidate SDN switches to install the DFR (and associated forwarding rules if any). The capability of the candidate SDN switches, (e.g., whether it can do parallel matching of several match fields), affects the actual implementation of DFR at the SDN switch. Therefore, there are three options here:

35

Option 1: the controller converts the DFR and associated forwarding rules (if any) into a set of flow table configuration messages according to the capability of the candidate SDN switches, e.g. a DFR is converted into several flow entries that are to be installed in one or several flow tables at the candidate SDN switch. The
5 converted flow entries provide the same behavior of such DFR

Option 2: the candidate SDN switch converts the DFR into the correspondent flow entries in its flow tables according to its own capability. In this case, the SDN switch interprets the DFR sent by the controller and decides how it is decomposed into
10 forwarding rules and in which flow tables they are installed.

Option 3: there is no need to convert the DFR into correspondent configuration messages (e.g., assume all SDN switch will support DFR in the future).

15 In case of option 1, the controller has an additional conversion step before distributing the configuration messages to the candidate SDN switches as shown in Fig. 8. In case of option 2 and 3, this step is not needed.

The distribution procedure of the flow table configuration message may use the
20 existing data path configuration procedures.

Fig. 9 schematically shows the procedure at the forwarding unit to execute the DFR. Before execution, the DFR must be installed.

25 When a forwarding unit (SDN switch) receives the DFR related configuration messages from the controller (option 1 mentioned above), it installs the related flow entries in its flow tables. When an SDN switch receives the DFR directly from the controller (options 2 and 3 mentioned above), it may convert the DFR into the correspondent flow entries before installs them in its flow tables (option 2), or install
30 DFR directly into its flow tables (option 3).

When the SDN switch detects the match of both fixed match fields and reconfigurable match fields (branch 2 in Fig. 9), it executes the normal forwarding actions in the instruction set. When the SDN switch detects the match of the fixed
35 match fields and mismatch of the reconfigurable match fields (branch 3 in Fig. 9), it executes the reconfigure actions (optionally in addition to normal forwarding actions)

in the instruction set. The normal action can be optional and the SDN switch will do nothing in branch 2 and perform only reconfigure actions related tasks in branch 3. When the SDN detects the mismatch of both match fields (branch 1 in Fig. 9), it executes the “flow table miss” action, which normally is (but not limited to) either
5 “drop packet” or “send to controller”.

To perform the reconfigure actions, the SDN switch needs to gather the required information which will be used to modify the reconfigurable match fields of the DFR/or the associated forwarding rules, and/or generate associated forwarding
10 rules. In case of existence of associated forwarding rules, the associated forwarding rules need to be identified. In case of generation of an associate forwarding rule, the new generated forwarding rule should be associated with the DFR which generates it. After SDN switch performs the reconfigure actions, the local modification may be reported to the controller.

15

Fig. 10 and 11 describe different use cases with DFR. While Fig. 10 schematically shows the aspect of mobility management, Fig. 11 illustrates local bridging.

DFR can be used in mobility management to automatically reconfigure the flow path
20 of a UE after a mobility event. In this use case, DFR reduces the load at the controller when multiple mobility events happen simultaneously and also the flow path reconfiguration latency, as the SDN switch does not need to wait for instructions from the controller in order to redirect the UE flow path.

25 Taking as a reference the scenario shown in Fig. 10, a procedure using DFR to handle mobility events in the data plane is explained below. The numbering of the steps described below is used in the figures as reference, while the number of each step is circled in the drawing.

30 Step 0: The controller computes a DFR to be installed in s1 to handle traffic between m1 (IP 'A') and an external host (IP 'B') in certain Packet Data Network (PDN). Such DFR is installed in s1 during the configuration process of the attach procedure of m1 and is described as follows:

35 Fixed match fields: [src: IP A, dst: IP B];
Reconfigurable match fields: [in_port];

Action: [output => PDN];

Associated flow entry: Downlink flow IP B → IP A, Match fields: [src: IP B, dst: IP A]

Action: [output => s2];

5 Trigger for reconfigure action: incoming packet with [src: IP A, dst: IP B] and different in_port;

Reconfigure action: Update current flow entry with new value of in_port [in_port => R'], and change the Action of the associated flow entry IP B → IP A to [output => R'], where R' is the new in_port.

10 Step 1: Incoming packet in s1 from m1 [IP src: 'A', IP dst: 'B', in_port: 's2']. The packet is forwarded to PDN as specified in the flow table of s1 [output => PDN].

Step 2: m1 moves to a new location and reattaches to a neighbour access point.

15 Step 3: Incoming packet in s1 from m1 [IP src: 'A', IP dst: 'B', in_port: 's3']. The installed DFR in s1 detects that a changed on the input port for the flow IP 'A' → IP 'B' occurred. DFR updates the current flow entry to match the tuple [IP src: 'A', IP dst: 'B', in_port: 's3'] and also the associated downlink flow entry IP 'B' → IP 'A' with the action [output => 's3'];

20

Step 4: [optional] The update in the flow table of s1 is reported to the controller by s1.

Another use case for DFR is enabling local bridging at the SDN switch, as shown in

25 Fig. 11. Bridging here refers to the connection of two traffic flows at the SDN switch.

Taking as a reference the scenario shown in Fig. 11 when m1 communicates with m2, S1 "bridges" the traffic between m1 and m2. DFR can be used to automatically set up a flow path between m2 and m1 without asking the controller. The detailed

30 procedure is the following:

Step 0: The controller computes a DFR to be installed in s1 for a flow between IP 'B' and IP 'A' with the following characteristics:

35 Fixed match fields: [dst: IP 'A'];

Reconfigurable match fields: [src: IP 'B', in_port: 'PDN'];

Action: [output => s2];

Trigger for reconfigure action: incoming packet with [dst: IP 'A'] and different values for [src, in_port];

Reconfigure action: Install a new flow entry with the new values of [src, in_port] and
5 action [output => s2];

Step 1: m2 sends a packet to m1, but no flow is set up in s1.

Step 2: Incoming packet in s1 from m2: [IP src: 'C', IP dst: 'A', in_port: 's3']. The
10 installed DFR in s1, detects a change in [IP src, in_port]. The DFR reconfigure
action is executed and a new flow entry is installed with the match fields [IP src: 'C',
IP dst: 'A', in_port: 's3'] and action [output => s2].

Step 3: [optional] The installation of the new flow entry is reported to the controller
15 by s1.

In the following, examples of implementing DFR in SDN shall be briefly described.
Depending on the actual capability of target SDN switch, DFR can be implemented
in different flavours. The controller is responsible to translate DFR into a set of
20 configuration message recognized by the target SDN switch.

Priority based implementation

One option for DFR implementation in the SDN switch is to leverage on forwarding
25 rule priorities. Priority based implementation installs DFRs as multiple flow entries
with different priorities in the SDN switch flow table:

The high priority one matches all the specified match fields, including both fixed and
reconfigurable match fields;

30 With a lower priority than the previous entry, the second one matches only the fixed
match fields. It includes a configure action that installs or modifies the specified
associated flow entries with the values of the reconfigurable match fields that were
not matched by the high priority flow entry.

35 Considering the use case shown in Fig. 10, m1, with IP 'A' has established
connection to a server in an external network (IP 'B') via s2. In step 2, m1 moves to

a neighbouring access point, and such connection needs to be redirected to s3 in step 3. In this case, a DFR can be installed in s1 to automatically perform the required redirection when m1 moves as described below.

5 The DFR that needs to be installed in s1 is defined as:

DFR ID: #A-B		
IP src: IP A IP dst: IP B	in_port: [R]	Normal action: output => PDN Reconf action: Update flow #A-B (new in_port = R') Update flow #B-A (new fwd action: output => R')

where the input port [R] is the reconfigurable match field. Let R be the old value of input port and R' be the current value of the input port of the matched packet. As reconfiguration actions, the DFR will update its reconfigurable match field from R to R', and the associated flow entry with ID #B-A with R' as output action. Such DFR can be implemented as two flow entries (e.g., #A-B-1/2) with different priorities as follows:

Flow ID	Prio	Match fields	Instruction set	
			Normal Action	Reconfigure Action
#A-B-1	1	IP src: IP A, IP dst: IP B, in_port: s2	output => PDN	
#A-B-2	2	IP src: IP A, IP dst: IP B, in_port: ANY	output => PDN	Modify flow #A-B-1, new match: in_port: [= #A-B-2 in_port] Modify flow #B-A, new action: output => [= #A-B-2 in_port]
#B-A	3	IP src: IP B, IP dst: IP A	output => s2	

Flow entry #A-B-2 will only be matched when the input port changes with regards to #A-B-1's. As action, the input port in #A-B-1 is updated to the new value registered by #A-B-2 and, at the same time, the associated flow entry #B-A is updated with the new value as output port.

20 Pipeline based implementation

DFR implementation can also leverage on pipeline processing. In this case, the fixed match fields are matched by a flow entry installed in a first flow table, as action, the packet is forwarded to a second table for further processing. In this second table, only the reconfigurable match fields are matched following the approach already described herein. Pipeline based implementation separates the fixed match field and reconfigurable match field into different tables. Therefore, it can be used to

improve the performance of SDN switch by leverage the flow entry (re)configuration frequency and latency in different tables.

If this implementation option is applied to the use case shown in Fig. 10, DFR could be implemented in an SDN switch as a pipeline of flow tables as shown below:

Table 1			
Flow ID	Prio	Match fields	Instruction set
			Normal Action Reconfigure Action
#A-B	1	IP src: IP A, IP dst: IP B	go to Table 2
#B-A	2	IP src: IP B, IP dst: IP A	output => s2

Table 2			
Flow ID	Prio	Match fields	Instruction set
			Normal Action Reconfigure Action
#A-B-1	1	in_port: s2	output => PDN
#A-B-2	2	in_port: ANY	output => PDN Modify flow #A-B-1, new match: in_port: [= #A-B-2 in_port] Modify flow #B-A in Table 1, new action: output => [= #A-B-2 in_port]

Parallel matching implementation

10

DFR can also be implemented as a single flow entry in the flow table of a SDN switch. In this approach, the SDN switch needs to explicitly differentiate between fixed and reconfigurable match fields. The SDN switch keeps track of the values of reconfigurable match fields and checks the matching results of both the fixed and reconfigurable match fields. If the fixed fields match while the reconfigurable fields do not, the SDN switch executes the reconfigure action associated to that flow entry. If the use case shown in Fig. 10 is considered, the DFR defined in the section relating to priority based implementation can be implemented as follows:

15

Flow ID	Prio	Match fields		Instruction set	
		Fixed	Reconfigurable	Normal Action	Reconfigure Action
#A-B	1	IP src: IP A, IP dst: IP B	in_port: s2	output => PDN	Modify flow #A-B, new match: in_port: [= #A-B in_port] Modify flow #B-A, new action: output => [= #A-B in_port]
#B-A	2	IP src: IP B, IP dst: IP A		output => s2	

20

where the reconfigure action in #A-B is executed when the value of the input port for the same entry changes. As in the previous implementation proposal, the reconfigure action updates the value of the input port in flow entry #A-B and sets such value as output port in flow entry #B-A.

5

10

CLAIMS

1. A forwarding unit (210, 220) for a software defined networking, SDN, system (10), comprising:
- 5 a storage module (212) for storing at least one forwarding rule (300), wherein each of the at least one forwarding rule comprises a match field (305) comprising at least one reconfigurable element (320) and an instruction set (330) comprising at least one reconfiguration action;
- wherein the forwarding unit (210, 220) is configured to apply a forwarding
- 10 rule of the at least one forwarding rule to a received data packet by comparing the received data packet with the match field of the forwarding rule (300);
- wherein the forwarding unit is further configured to carry out the at least one reconfiguration action of the forwarding rule if the received data packet does not match one or more of the at least one reconfigurable element of the forwarding rule.
- 15
2. The forwarding unit (210, 220) of claim 1,
- wherein the match field (305) of the forwarding rule comprises at least one static element (310);
- wherein the forwarding unit is configured to carry out the at least one
- 20 reconfiguration action of the forwarding rule if the received data packet additionally matches the at least one static element (310) of the forwarding rule.
3. The forwarding unit (210, 220) of claim 1 or 2,
- wherein the match field (305) of the forwarding rule comprises more than one
- 25 reconfigurable elements (320) and wherein the forwarding unit is configured to carry out the at least one reconfiguration action of the forwarding rule if the received data packet does not match all of the at least one reconfigurable elements of the forwarding rule.
- 30
4. The forwarding unit (210, 220) of any one of the preceding claims,
- wherein the match field (305) of the forwarding rule comprises a forwarding
- action;
- wherein the forwarding unit (210, 220) is configured to carry out the
- forwarding action of the forwarding rule if a received data packet matches the match
- 35 field of the forwarding rule.

5. The forwarding unit (210, 220) of any one of the preceding claims,
wherein the forwarding unit is configured to modify at least one
reconfigurable element (320) of the at least one forwarding rule when carrying out
the at least one reconfiguration action of the forwarding rule if a received data
5 packet does not match one or more reconfigurable elements of the forwarding rule.
6. The forwarding unit (210, 220) of any one of claims 1 to 4,
wherein the forwarding unit is configured to delete at least one of the at least
one forwarding rule if a received data packet does not match one or more
10 reconfigurable elements of the forwarding rule.
7. The forwarding unit (210, 220) of any one of claims 1 to 5,
wherein the forwarding unit is configured to modify a forwarding rule if a
received data packet does not match one or more reconfigurable elements of the
15 forwarding rule.
8. The forwarding unit (210, 220) of any one of the preceding claims,
wherein the forwarding unit is configured to generate at least one forwarding
rule if a received data packet does not match one or more reconfigurable elements
20 of the forwarding rule.
9. The forwarding unit (210, 220) of any one of the preceding claims,
wherein the reconfiguration action of the forwarding rule contains an
association instruction and wherein the forwarding unit is configured to associate the
25 forwarding rule to at least a second forwarding rule.
10. The forwarding unit (210, 220) of any one of the preceding claims,
wherein the forwarding unit is configured to adapt an output port of a
forwarding rule when carrying out the at least one reconfiguration action of the
30 forwarding rule;
wherein the output port is adapted based on an input port of a received data
packet.
11. The forwarding unit (210, 220) of claim 10,

wherein the forwarding unit is configured to adapt the output port of a forwarding rule based on a source address and a incoming port of a received data packet.

5 12. The forwarding unit (210, 220) of any one of the preceding claims, wherein the forwarding unit is configured to report result of the at least one reconfiguration action to a control unit (110) of the SDN system.

10 13. A control unit (110) for a software defined networking, SDN, system (10), the control unit being configured to configure at least one forwarding unit (210, 220) of the SDN system (10), the control unit comprising:

a configuration module (112) for generating configuration commands for a forwarding unit;

15 wherein the configuration command comprises at least one forwarding rule (300) having a match field (305) with at least one reconfigurable element (320) and an instruction set (330) comprising at least one reconfiguration action to be applied to a forwarding rule by the forwarding unit (210, 220) in case a data packet received by the forwarding unit does not match one or more of the at least one reconfigurable element.

20

14. A software defined network, SDN, system (10), comprising at least one control unit (110) of claim 13; at least one forwarding unit (210, 220) of any one of claims 1 to 12; wherein the at least one control unit (110) is configured to configure at least
25 one forwarding rule of the at least one forwarding unit (210, 220).

15. A method for applying forwarding rules (300) of a software defined networking, SDN, system (10), the method comprising:

30 receiving at least one forwarding rule which comprises a match field (305) with at least one reconfigurable element (320) and an instruction set (330) with at least one reconfiguration action;

applying a forwarding rule of the at least one forwarding rule to a received data packet by comparing the received data packets with the match field of the forwarding rule;

carrying out the at least one reconfiguration action of the forwarding rule if the received data packet does not match one or more of the at least one reconfigurable element of the forwarding rule.

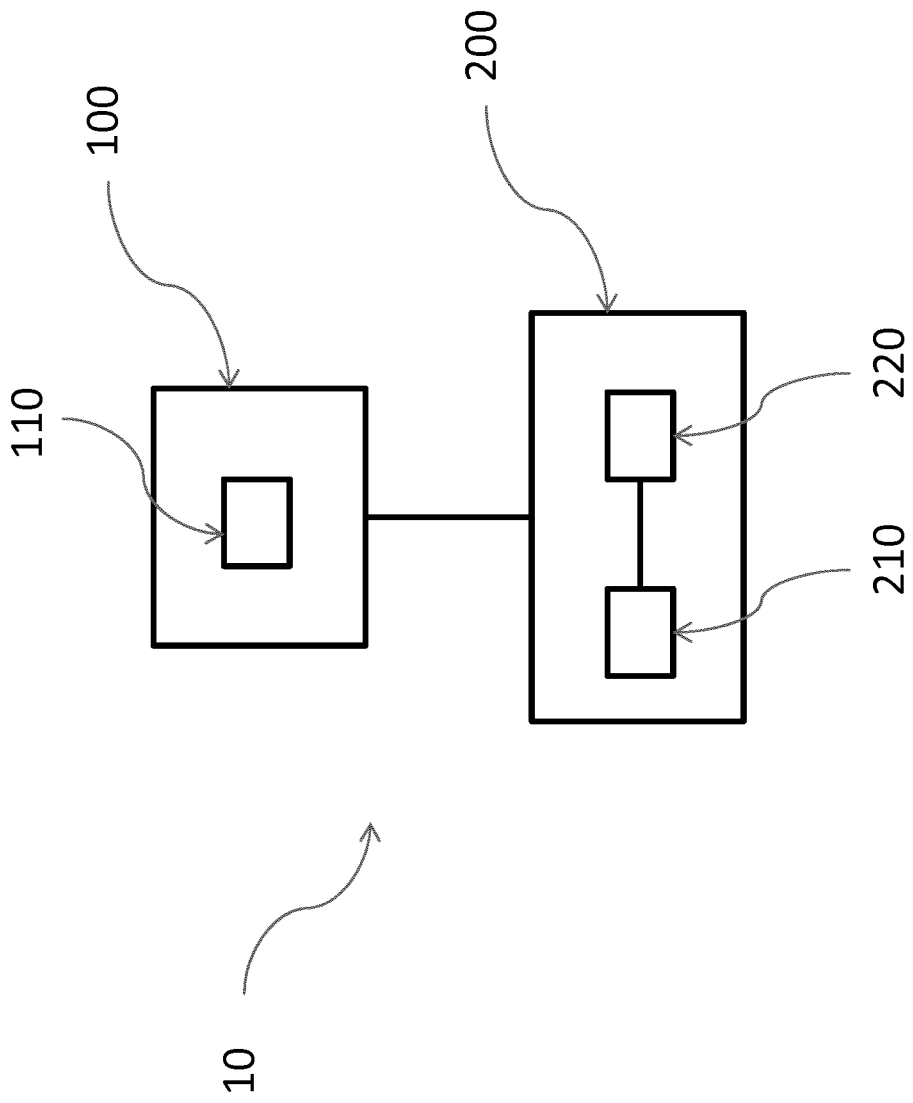


Fig. 1

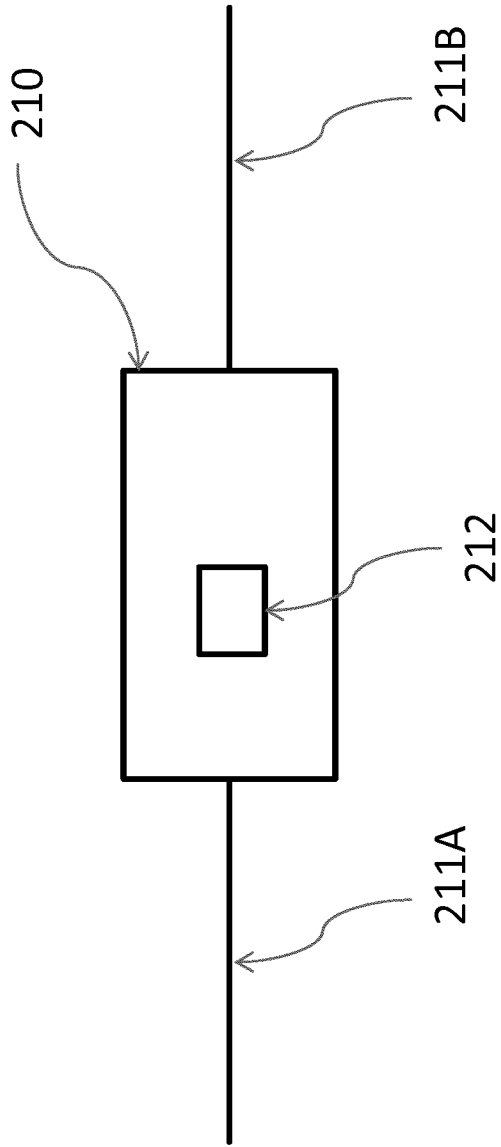


Fig. 2

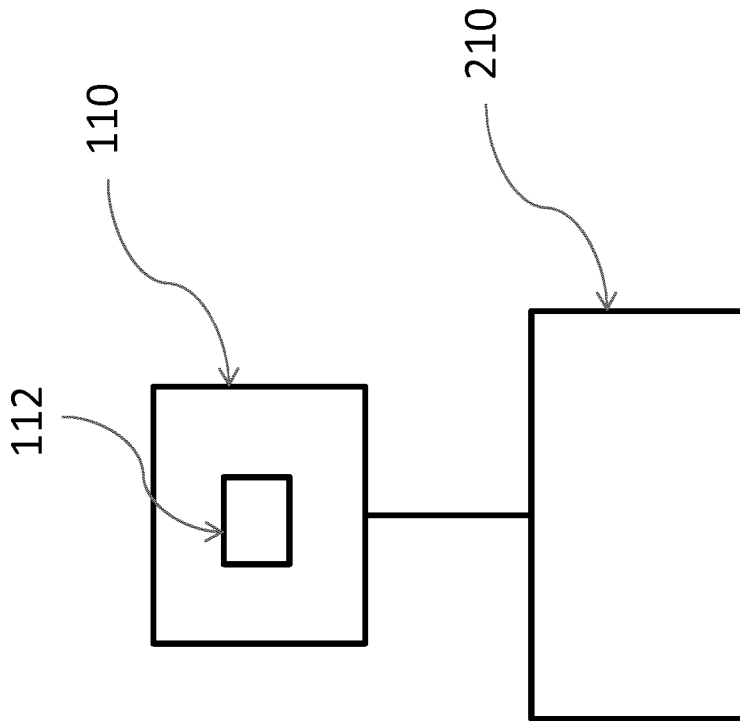


Fig. 3

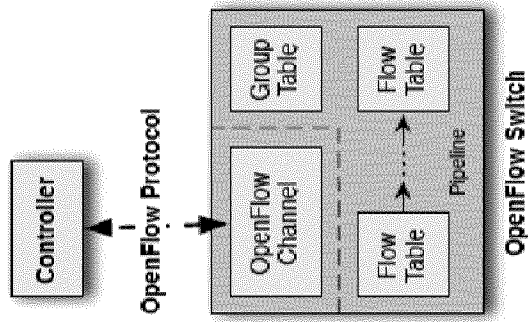


Fig. 5

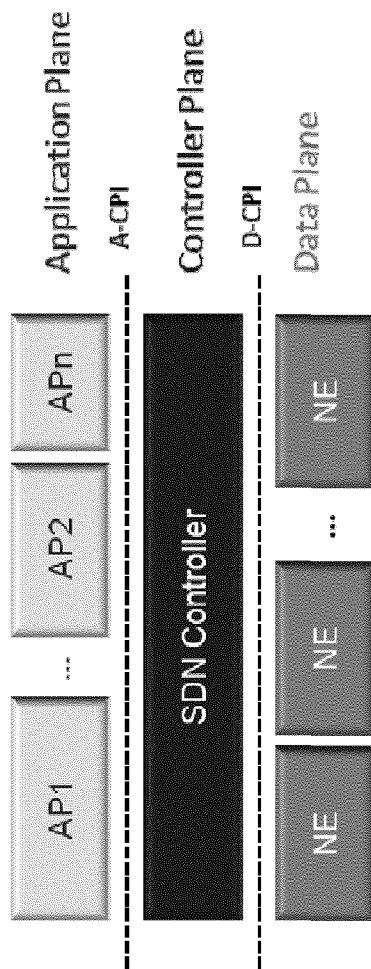
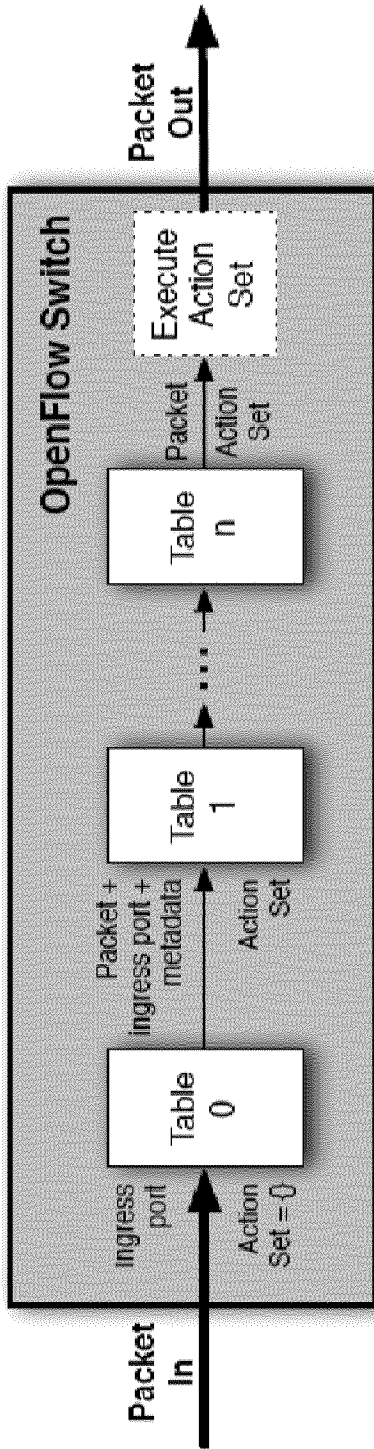
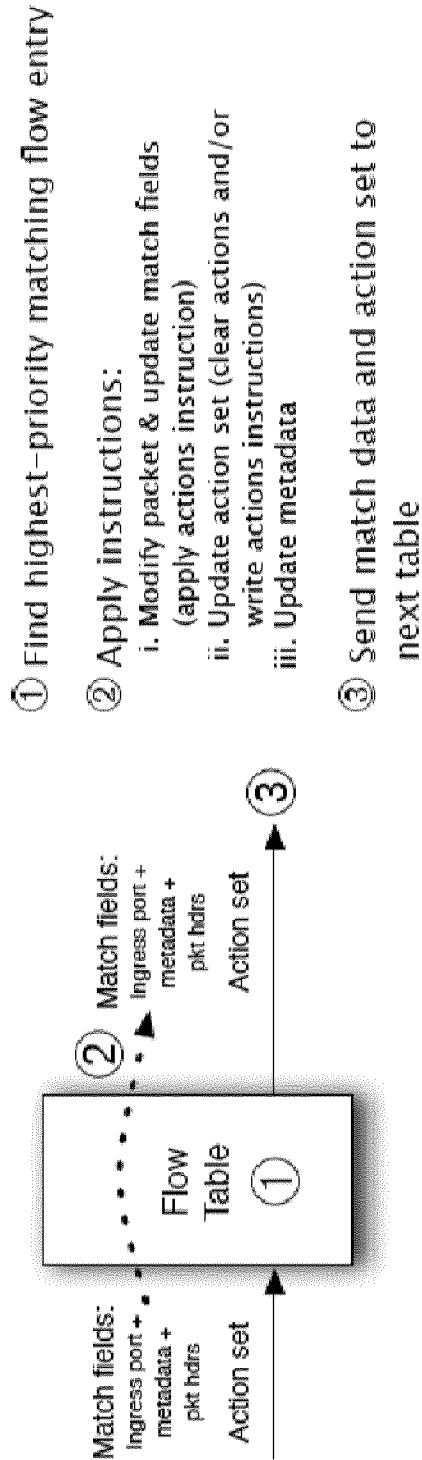


Fig. 4



(a) Packets are matched against multiple tables in the pipeline



(b) Per-table packet processing

Fig. 6

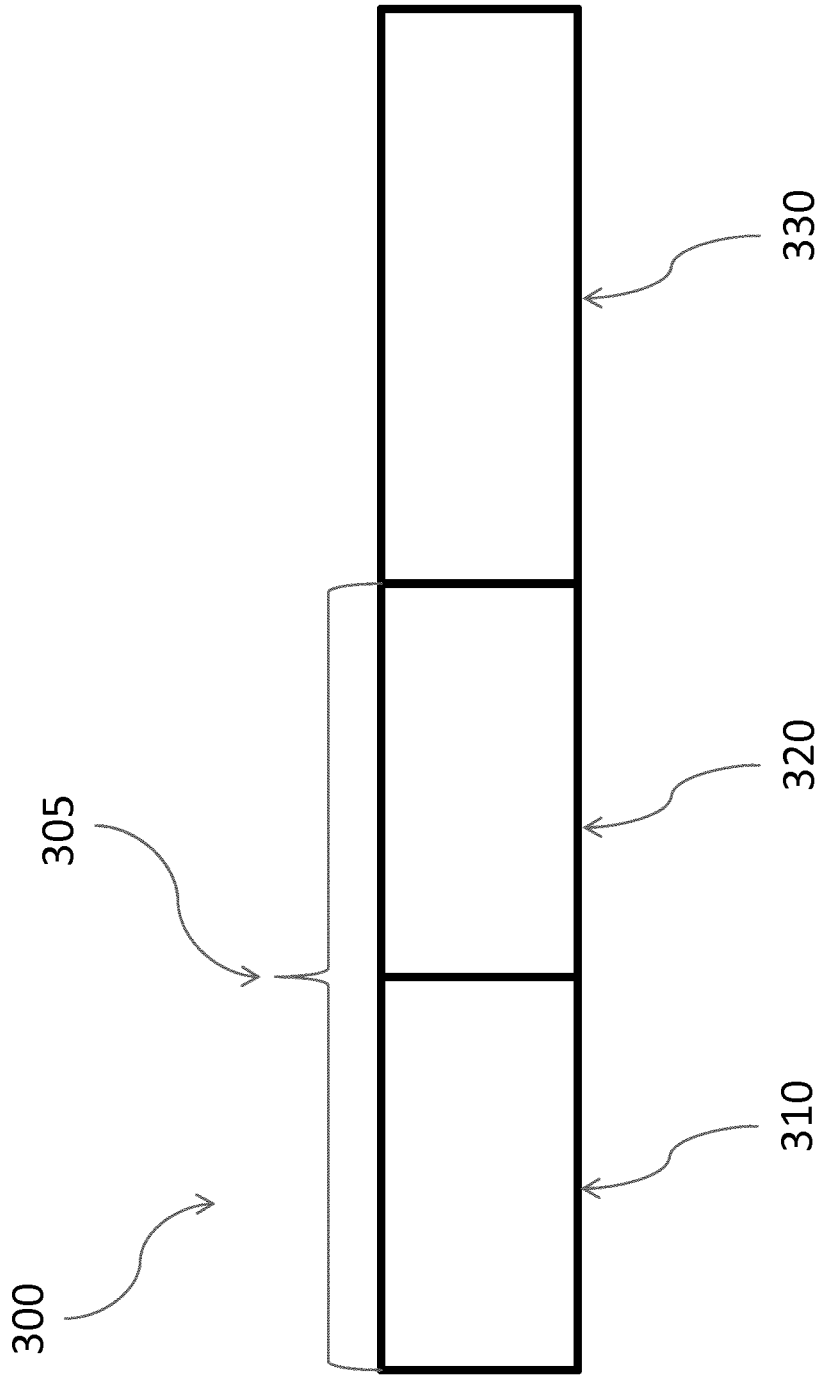


Fig. 7

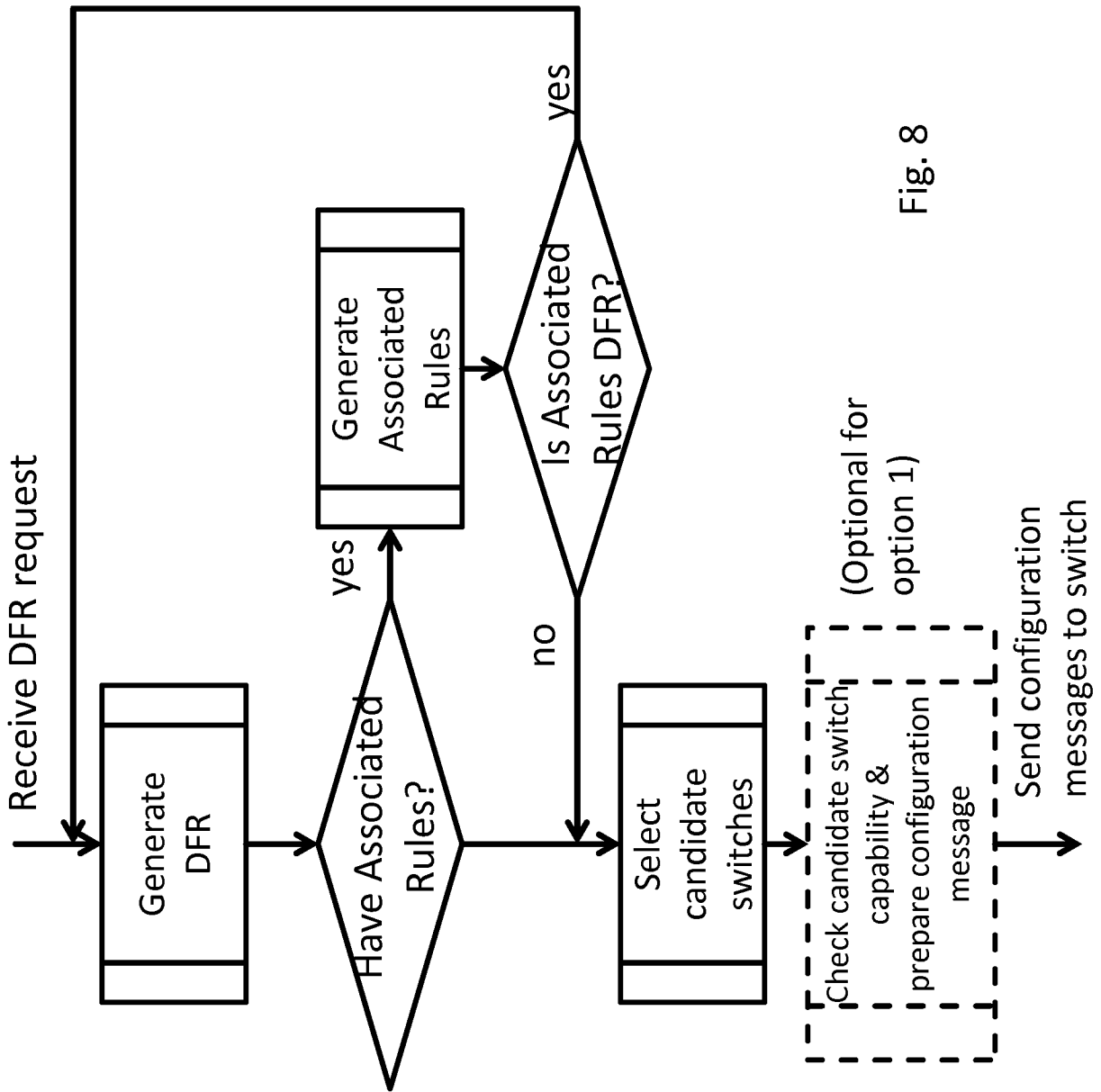


Fig. 8

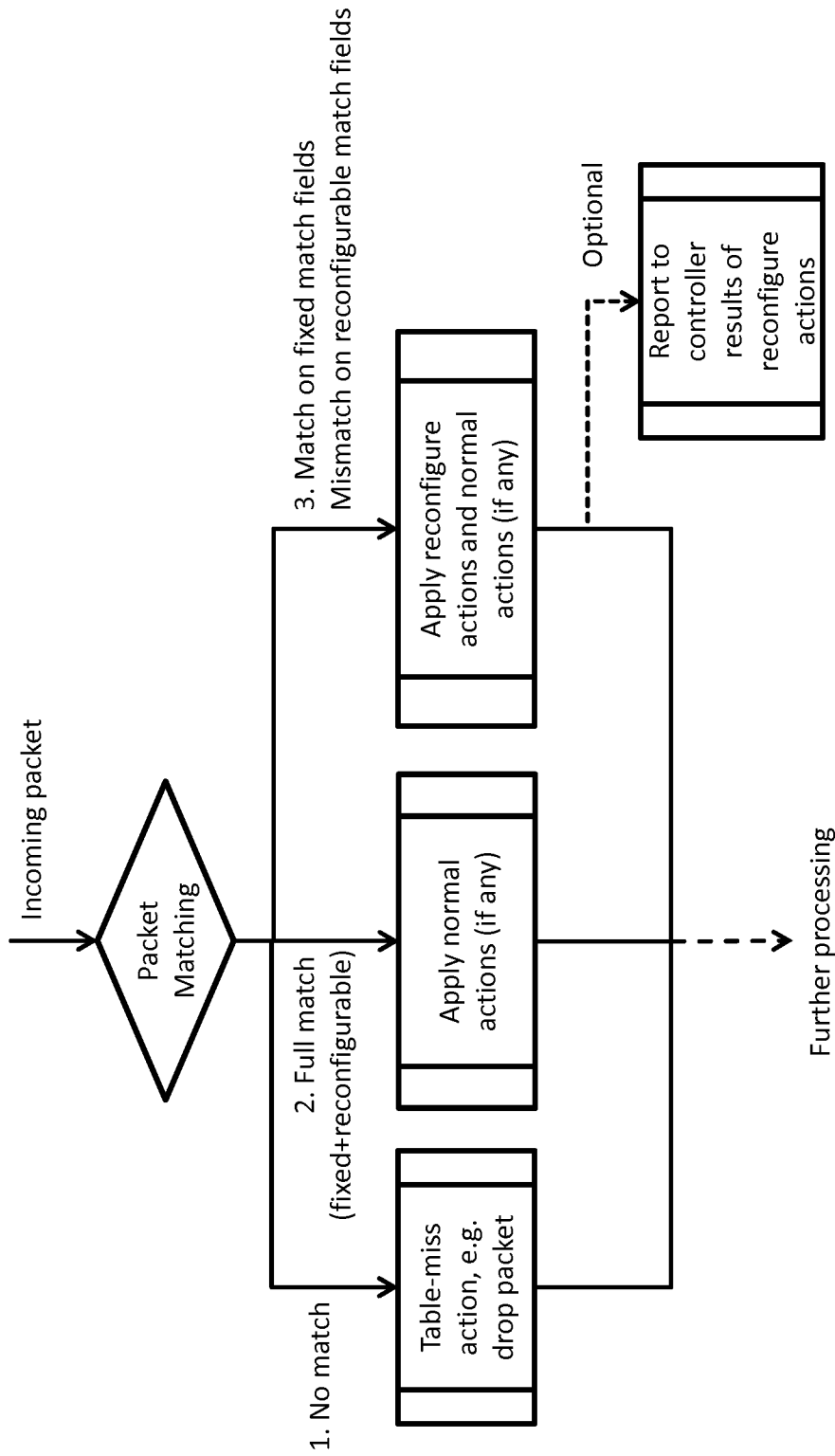


Fig. 9

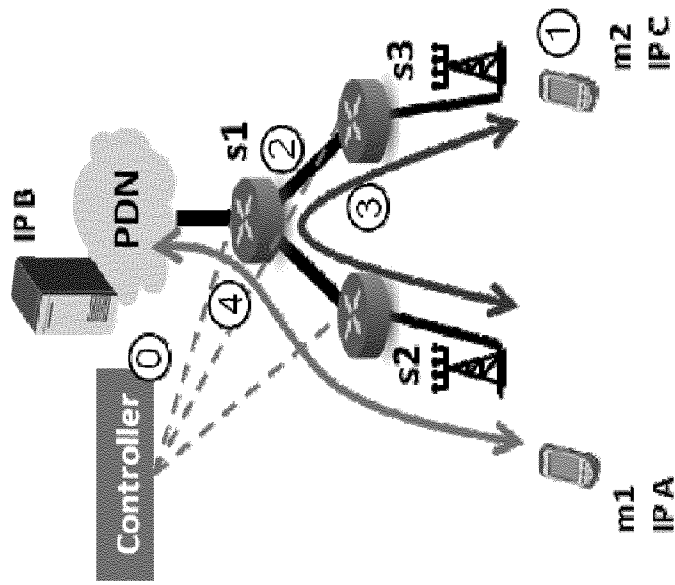


Fig. 11

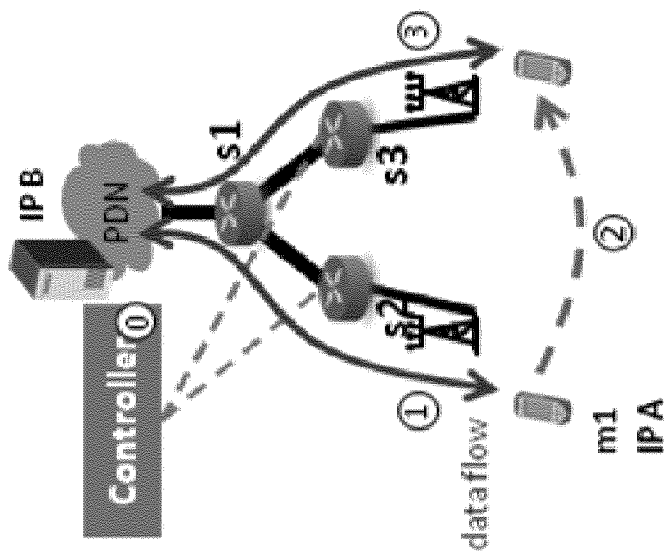


Fig. 10

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/050549

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/715 H04L12/741
ADD.
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

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 practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2013/163426 A1 (BELIVEAU LUDOVIC [US] ET AL) 27 June 2013 (2013-06-27) abstract figures 1,2,5,6 page 1, paragraph 9 - page 2, paragraph 10 page 4, paragraph 39 - paragraph 42 page 5, paragraph 45 - page 6, paragraph 54 page 6, paragraph 56 - paragraph 57 page 7, paragraph 61 - paragraph 62 page 10, paragraph 77 - paragraph 79 page 10, paragraph 90 - page 11, paragraph 96 ----- -/--	1-15

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 12 August 2016	Date of mailing of the international search report 19/08/2016
---	--

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/050549

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PAT BOSSHART ET AL: "Forwarding metamorphosis", SIGCOMM, ACM, 2 PENN PLAZA, SUITE 701 NEW YORK NY 10121-0701 USA, 27 August 2013 (2013-08-27), pages 99-110, XP058030628, DOI: 10.1145/2486001.2486011 ISBN: 978-1-4503-2056-6	1,13-15
A	abstract figures 1,3 page 100, column 2, line 14 - line 52 page 101, column 1, line 50 - column 2, line 50 page 103, column 1, line 1 - line 10 page 103, column 1, line 29 - line 51 page 104, column 1, line 12 - line 15 page 105, column 2, line 10 - line 21 -----	2-12
X	CN 104 009 921 A (OPENNET SCIENCE & TECHNOLOGY BEIJING CO LTD) 27 August 2014 (2014-08-27)	1,13-15
A	the whole document -----	2-12
A	US 2015/078386 A1 (JACKSON ETHAN J [US] ET AL) 19 March 2015 (2015-03-19) the whole document -----	1-15
A	US 2014/098669 A1 (GARG VIPIN [US] ET AL) 10 April 2014 (2014-04-10) the whole document -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2016/050549

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013163426	A1	27-06-2013	
		CN 103999430 A	20-08-2014
		EP 2795873 A1	29-10-2014
		US 2013163426 A1	27-06-2013
		US 2014204948 A1	24-07-2014
		WO 2013093858 A1	27-06-2013

CN 104009921	A	27-08-2014	NONE

US 2015078386	A1	19-03-2015	NONE

US 2014098669	A1	10-04-2014	
		CN 104704779 A	10-06-2015
		EP 2904745 A2	12-08-2015
		US 2014098669 A1	10-04-2014
		WO 2014057375 A2	17-04-2014
