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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 September 2016 (15.09.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/141996 A1

(51) International Patent Classification:
H04L 12/721 (2013.01) *H04L 12/24* (2006.01)
H04L 12/715 (2013.01) *G06F 21/75* (2013.01)

(21) International Application Number:
PCT/EP2015/055227

(22) International Filing Date:
12 March 2015 (12.03.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **NEC EUROPE LTD.** [DE/DE]; Kurfürsten-Anlage 36, 69115 Heidelberg (DE).

(72) Inventors: **BIFULCO, Roberto**; Bergheimerstraße 38, 69115 Heidelberg (DE). **KARAME, Ghassan**; Ringstraße 3, Heidelberg Heidelberg (DE). **KLAEDTKE, Felix**; Brunnengasse 8, 69117 Heidelberg (DE). **CUI, Heng**; - (DE).

(74) Agent: **ULLRICH & NAUMANN**; Schneidmühlstraße 21, 69115 Heidelberg (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))*

(54) Title: METHOD FOR FORWARDING DATA IN A NETWORK, FORWARDING ELEMENT FOR FORWARDING DATA AND A NETWORK

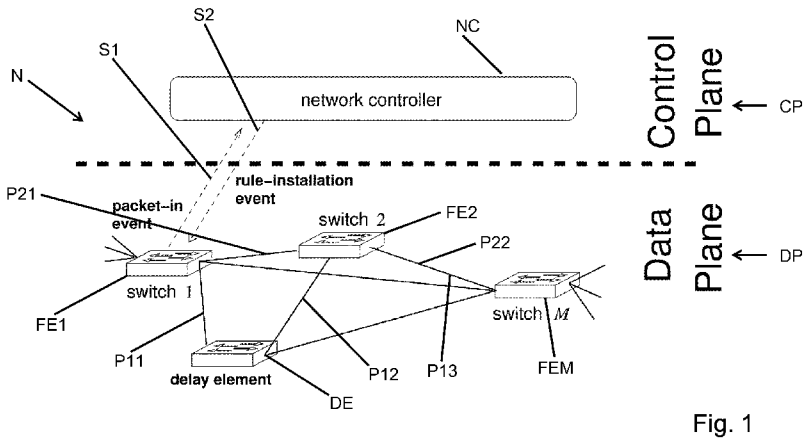

Fig. 1

(57) Abstract: The present invention relates to a method for forwarding data, preferably data in form of flows, in a network N, preferably a software-defined network, wherein the network comprises one or more forwarding elements FE1, FE2, FEM for forwarding data, wherein in case the data matches to a present forwarding rule on said forwarding element said data is forwarded with a time delay - delay forwarding - according to a time delay policy and generated by a delay entity DE §such that a first number of packets - first packets - of said data is forwarded by said forwarding element with a first forwarding time delay and wherein a second number of packets - second packets - of the same data is forwarded with a second forwarding time delay, wherein the first forwarding time delay and the second forwarding time delay are having a certain time difference from each other. A controller-forwarding element interaction S1, S2 for packet forwarding is thus obfuscated by introducing delays in the forwarding of the first packets of a network flow without incurring considerable modifications to existing network elements. The delay entity may be emulated by a software component.

WO 2016/141996 A1

METHOD FOR FORWARDING DATA IN A NETWORK, FORWARDING ELEMENT FOR FORWARDING DATA AND A NETWORK

The present invention relates to a method for forwarding data, preferably data in form of flows, in a network, preferably a software-defined network, wherein the network comprises one or more forwarding elements for forwarding data.

The present invention further relates to a forwarding element for forwarding data, preferably data in form of flows, in a network, and connectable to a delay entity.

The present invention even further relates to a network, preferably a software-defined network, for forwarding data, preferably data in form of flows, in a network, wherein the network comprises one or more forwarding elements for forwarding data.

Although applicable to networks in general, the present invention will be described with regard to networks in form of software-defined networks.

Although applicable to data in general, the present invention will be described with regard to data in form of data flows.

Although applicable to any kind of forwarding element in, the present invention will be described with regard to forwarding elements in form of switches, in particular OpenFlow Switches.

Software Defined Networks 'SDN' enable network administrators to manage network services by abstracting lower-level functionalities of the physical network. This is provided by decoupling the system making decisions about where traffic is sent from the underlying system, e.g. switches, that forwards traffic to the selected destination. Software defined networks require procedures for providing a communication between the so-called control plane and the so-called data plane. Such a procedure providing communication between both planes is e.g. the so-called OpenFlow, which is disclosed in the non-patent literature of the Open

Networking Foundation (ONF), OpenFlow Switch Specification, Version 1.3.2, Wire Protocol 0x04, April 25, 2013.

In more detail, when packets of a data flow in the SDN are forwarded this involves only the data plane of a switch: Incoming network packets are matched with flow rules installed at the switches and upon matching are forwarded according to rules' specific actions. However, for some packets, the forwarding happens only after some interaction between the data plane of the switch and control plane. More precisely, a switch can be configured to generate a notification event to the control plane upon reception of packets belonging to specified flows. The control plane answers to said notification by generating a forwarding decision, which is installed in the switch's data plane in the form of a flow rule. When the flow rule is installed, the packet is finally forwarded by the switch.

In a software-defined network, the control plane is usually implemented as a remote entity - the so-called controller - and its invocation significantly delays packet forwarding. These delays are intrinsic in software-defined networks because

a) the controller runs on commodity computers and processes the packets in software whereas switches use highly specialized hardware, and
b) installing flow rules in the switches' flow tables is a costly operation.

By monitoring the network traffic, an attacker can observe these delays and guess - with a considerable probability - for which data flows the controller is invoked. In this way, the attacker may learn how the network is configured and how it operates. The leakage of this information may expose the network to a number of attacks, compromising its security. In more detail in a conventional software-defined network an attacker can measure several features like dispersion, round-trip times, etc. of packet forwarding by observing the network traffic. The attacker can thereby identify - with a very high probability - if the switches have matching flow rules installed for a network packet or if the switches interact with the controller for forwarding the packet. Currently, only prohibitive defenses against making such observations exist, e.g. sending all network packets to the controller or deleting and reinstalling flow rules at the switches. However, these conventional

solutions add a significant workload to the controller and the installation of flow rules is a costly operation in terms of network resources, e.g. additional network traffic.

It is therefore an objective of the present invention to provide a method for forwarding data, a forwarding element and a network for forwarding data, which provide a higher level of security and are easy to implement while consuming only minor additional network resources.

The aforementioned objectives are accomplished by a method for forwarding data, preferably data in form of flows, in a network, preferably a software-defined network, wherein the network comprises one or more forwarding elements for forwarding data.

Said method is characterized in that in case the data matches to a present forwarding rule on said forwarding element said data is forwarded with a time delay – delay forwarding – according to a time delay policy and generated by a delay entity such that a first number of packets – first packets - of said data is forwarded by said forwarding element with a first forwarding time delay and wherein a second number of packets – second packets - of the same data is forwarded with a second forwarding time delay, wherein the first forwarding time delay and the second forwarding time delay are having a certain time difference from each other.

The aforementioned objectives are also accomplished by a forwarding element for forwarding data, preferably data in form of flows, in a network, and connectable to a delay entity, preferably for performing with a method according to one of the claims.

Said forwarding element is characterized in that said forwarding element being adapted to send received data in case said data matches to a present forwarding rule on said forwarding element to said delay entity according to a time delay policy such that a first number of packets – first packets - of said data is forwarded with a first forwarding time delay and wherein a second number of packets –

second packets - of the same data is forwarded with a second forwarding time delay, wherein the first forwarding time delay and the second forwarding time delay are having a certain time difference from each other.

The aforementioned objectives are also accomplished by a network, preferably a software-defined network, for forwarding data, preferably data in form of flows, in a network, wherein the network comprises one or more forwarding elements for forwarding data, preferably for performing with a method according to one of the claims.

Said network is characterized by a delay entity adapted to forward received data with a time delay – delay forwarding – and said forwarding element being adapted to send data in case said data matches to a present forwarding rule on said forwarding element to said delay entity according to a time delay policy such that a first number of packets – first packets - of said data is forwarded with a first forwarding time delay and wherein a second number of packets – second packets - of the same data is forwarded with a second forwarding time delay, wherein the first forwarding time delay and the second forwarding time delay are having a certain time difference from each other.

In particular a rule or forwarding rule comprises matching information and one or more actions to be performed when data or more specifically packets match to the matching information. One of the actions to be performed is then preferably specified as delay forwarding, such that upon matching delay forwarding is applied. The actions can be specified using a group table. A group table may comprise one or more action buckets. Each bucket may comprise one or more actions that are then applied to a packet.

According to the invention it has been recognized that in particular a controller-forwarding element interaction for packet forwarding is obfuscated by introducing delays in the forwarding of the first packets of a network flow without incurring considerable modifications to existing network elements, preferably in a software defined network.

According to the invention it has been further recognized that the controller-forwarding element interaction is mimicked at the start of a network flow in the presence of a matching flow rule.

According to the invention it has been even further recognized that only minor additional network resources are consumed and the controller's packet-forwarding logic is not additionally complicated.

According to the invention it has been even further recognized that security is significantly enhanced since delay forwarding of packets in networks, preferably software-defined networks, in which an attacker can measure several features like dispersion, round trip times, etc. of packet forwarding by observing the network traffic masks typical time patterns or mimics typical time patterns leading to a confusion of a potential attacker about the network configuration.

Further features, advantages and preferred embodiments are described in the following subclaims.

According to a preferred embodiment the first packets are beginning packets of said data and the second packets are the subsequent packets of said data preferably wherein the first number and/second number of packets is predetermined or fixed. This enables to confuse an attacker is a simple way.

According to a further preferred embodiment the first forwarding time delay is equal to a controlling time within a certain deviation interval, wherein the controlling time representing the time for generating and initially providing said present forwarding rule to said forwarding element. This enables in an easy way to mimic the installation of forwarding rules for said data even if no forwarding rule has to be installed. This further confuses a potential attacker when measuring time features of packet forwarding.

According to a further preferred embodiment first packets are forwarded via a different output port of said forwarding element than said second packets, wherein the delay entity is connected to the output port for the first packets. This enables

an easy transmission to the delay entity for delaying the packets so that the forwarding element does not have to mimic the delay itself which packets are to be delay forwarded or not.

According to a further preferred embodiment when the time between subsequent packets exceeds a certain time threshold, then first packet exceeding said time threshold is considered as the first packet of a further first packets of a new data flow. This enhances further the security since additional delays for a data flow are introduced such that an adversary or attacker is further confused: When a data flow first arrives at a switch then the rule is installed causing a delay in forwarding the first packets. Further when packets of the same initial data flow are exceeding a certain time threshold between subsequent packets then a further delay is introduced to the packets mimicking an installation of a new forwarding rule and therefore causing an adversary to assume that the data flow has ended and that a new flow has arrived at the forwarding element.

According to a further preferred embodiment on only a subset of data arriving at said forwarding element, delay forwarding is applied. This enables to further enhance a confusion of an adversary.

According to a further preferred embodiment priority information assigned to said data is checked and said subset is excluded from delay forwarding when said priority information exceeds a predefined priority threshold. This enhances the flexibility since for example depending on specific implementation decisions some high-priority flows can be forwarded always without delay ensuring for example high quality of service QoS or quality of experience QoE.

According to a further preferred embodiment said subset of data is selected probabilistically. This further enhances the security such that an adversary cannot extract or guess any network modifications by measuring features of packet forwarding in the network.

According to a further preferred embodiment said delay entity is a) integrated into the forwarding element or b) located within the network outside the forwarding

element, wherein in case of b) said forwarding element marks the data, preferably each packet, to be delayed with a delay indication for later evaluation by the delay entity.

While a) enables a particular easy implementation within the forwarding element itself avoiding installation of a delay entity besides the forwarding element, b) enables an easy and flexible use of the delay element by further other forwarding elements avoiding costs for delay element for each forwarding element.

According to a further preferred embodiment the delay entity is emulated by a software-component on a corresponding network entity of the network, preferably wherein said network entity is provided in form of said forwarding element and/or a controller for controlling said forwarding element. This avoids the installation of further hardware and corresponding integration into the network. When the delay entity is emulated on the forwarding element or on a controller of a Software Defined Network, already present hardware can be used for providing the delay entity and thus delay forwarding.

According to a further preferred embodiment different packets belonging to the same data are forwarded by said forwarding element and/or said delay entity via different paths in said network. This further enables to defend against an adversary who is trying to measure the dispersion between back-to-back packets in order to estimate the capacity of network paths or to acquire statistics about delay paths in the network, for example by using standard deviation or the like.

According to a further preferred embodiment said forwarding via different paths is performed for load balancing with the network. This enables an optimization with regard to network resources and also further confusing an adversary.

According to a further preferred embodiment on at least part, preferably all packets of the data having the same or a similar header, delay forwarding is applied, wherein one or more parameters for headers are defined being evaluated when checking for a similarity. This even further enhances the security: An adversary may still learn whether the controller is installed in fine-grained rules or coarse-

grained rules. This can be done for example if the adversary slightly changes the probe packet headers in order to see if additional delays are triggered. By delaying all packets whose headers are equal or marginally different the information leakage about the flow rule installation logic is therefore limited.

There are several ways how to design and further develop the teaching of the present invention in an advantageous way. To this end it is to be referred to the patent claims subordinate to the independent patent claims therein on the one hand and to the following explanation of preferred embodiments of the invention by way of example, illustrated by the figure on the other hand. In connection with the explanation of the preferred embodiments of the invention by the aid of the figure, generally preferred embodiments and further developments of the teaching will be explained.

In the drawings

Fig. 1 shows a system according to a first embodiment of the present invention; and

Fig. 2 shows a schematic view of delay forwarding of packets according to a second embodiment of the present invention.

Fig. 1 shows a system according to a first embodiment of the present invention.

In Fig. 1 a software-defined network N is shown with a control plane CP and a data plane DP. In the control plane a network controller NC is provided which is responsible for controlling switches FE1, FE2, FEM in the data plane DP. The switches FE1, FE2, FEM are connected with each other for forwarding packets of data. Further in the data plane DP a delay element DE is provided directly connected to forwarding element FE1 and FE2. Upon a packet-in event, i.e. upon arrival of a first packet of a data flow, the first switch FE1 communicates in a first step S1 with the network controller NC when no matching rule for said arriving data flow is present. The network controller NC then initiates rule-installation in a second step S2 on the switch FE1 such that the arrived packets of said data flow

can be forwarded according to the installed and matched forwarding rule. To achieve delay of packets, a packet forwarding logic may be installed on the network switches FE1, FE2 ... which selects different output ports for a flow depending on its status.

Preferably and in more detail a flow can be associated with one of two possible statuses: New, Old. All the flows that are in a "New" status are forwarded to said delay element DE, which mimics the delay introduced preferably by the handling of the packets at the controller CN.

The switch then preferably applies this forwarding logic, per flow:

```
if (status = NEW):
  forward with additional delay;
else:
  (execute normal forwarding logic);
```

The checking "status=NEW" of a flow can be performed in different ways.

One possible way is to check a flow timer, which tracks the time passed since the last packet belonging to the flow was received by the switch. If this time is above a given threshold, a newly received flow's packet can be considered as the start of a new flow, thus the flow status is set NEW.

Instead, a flow status is set to OLD when a given number of packets has been received by the flow while staying in status NEW.

Applying the above definitions of NEW and OLD status, the previous forwarding logic would then look as follows:

```
Variable: counter;
if (last_packet_time > time_threshold or (counter > 0 and counter <= num_packets)
): //status NEW
  forward with additional delay;
```

```
  counter = counter + 1;
else: //status OLD
  counter = 0;
  (execute normal forwarding logic);
```

Depending on the specific implementation decisions, either all or a subset of the flows can be handled with this logic. For instance, some high-priority flows may have packets always forwarded without delays. Likewise, it is possible to define just a subset of new flows to be delayed, for instance, using some probabilistic selection of the subset.

The delay element DE, which purpose is to introduce a delay which is preferably comparable with the one introduced by the controller interaction, can be implemented in different ways: For example a software switch can be used as a delay element DE for delaying the packets. The delay element DE can also be deployed in different ways:

- Integrated in the switch FE1, FE2, FEM itself. This can be achieved explicitly sending the packets to be delayed to e.g., the switch's slow path, or by queueing them in a slow output queue designed for the purpose;
- Connected to one of the switch's ports. The delay element DE is connected to a dedicated port of the switch FE1, FE2, FEM, where the packets are delivered when they need to be delayed;
- The delay element DE is deployed somewhere in the network N. The switch FE1, FE2, FEM tags the packets to which the delay has to be applied so that they are delivered eventually to the delay element DE.

To implement delay forwarding in switches FE1, FE2, …, a rule may be defined for which one of the actions to be performed when the rule is matched with "delay" routing these packets to the delay element DE prior to a further forwarding to the next forwarding element FE2, FEM or the like.

In more detail when the switch FE1, FE2, FEM is an OpenFlow switch said switch FE1, FE2, ... may define a pipeline of flow tables to look up the actions that have to be applied to the packets of a given network flow.

The OpenFlow switch (e.g. the OpenFlow specification >1.3) defines a pipeline of flow tables to lookup the actions that have to be applied to the packets of a given network flow F. The actions can be specified using a group table. A group table may comprise one or more action buckets. Each bucket may comprise one or more actions that are then applied to a packet.

A group table is associated with a bucket selection logic, when the group table type is “select”. In this embodiment it can be implemented as bucket selection logic.

Assuming a group table like the following:

Group number	Actions
1	Delay; Forward to port X
2	Forward to port X

The bucket selection logic of the group table would be:

```
Variable: counter;
if (last_packet_time > time_threshold or (counter > 0 and counter <= num_packets)
): //status NEW
  select bucket 1
  counter = counter + 1;
else: //status OLD
  counter = 0;
  select bucket 2
```

For delaying the first few packets of a connection after selectively forwarding the first packets to a delay element DE a feedback loop between the controller C and

the delay element DE may be established to ensure that packets are delayed according to a realistic delay distribution as dictated by the controller C.

Alternatively the delay element DE can be emulated by controller C itself.

Further delay element DE can be deployed in the following way:

1. The delay element DE is emulated by the controller NC itself. Here, the action of Group 1 could be for example to forward the first few packets of a given connection to the controller NC.
2. The delay element DE is a dedicated software switch FE1, FE2, ... that is part of the network N. This requires one to reserve one port of the OpenFlow switch FE1, FE2, ... in order to set up the selective forwarding to the software switch FE1, FE2, ... Alternatively, one can tag packets and install a (non-OpenFlow) switch to selectively forward them either to another hardware OpenFlow switch FE1, FE2, ... or to a software-based switch FE1, FE2, ...
3. The delay element DE can be emulated by the software component of the OpenFlow switches FE1, FE2, ...

The aforementioned group tables can not only be used in order to delay the first packets of a given connection but may also be used in order to selectively forward different packets of the same flow F across different redundant paths P11, P12, P13 or P21, P22 in the network N. This has the advantage that load balancing across different paths P11, P12, P13 or P21, P22 can be provided and further to defend against an adversary who is trying to measure the dispersion between back-to-back packets in order to estimate the capacity of the network paths P11, P12, P13 or P21, P22 respectively or to acquire statistics about the delay paths in the network N, for example by looking at the standard deviation.

However, an adversary may still learn that the controller NC is installing fine-grained rules or coarse-grained rules. An adversary can for example slightly change probe packet headers in order to see if additional delays are triggered. To prevent this information leakage a logic within group tables can be installed in

order to preferably probabilistically, delay paths or all packets who's header are marginally different. As such the information leakage about the flow rule installation logic is limited.

Fig. 2 shows a schematic view of delay forwarding of packets according to a second embodiment of the present invention.

In Fig. 2 the time line t of subsequent packets NP, NP11, NP21, NP12 of a flow F of data D is shown. All the packets NP, NP11, NP21, NP12 shown in Fig. 2 are network packets NP with headers matching a flow rule. The network packets NP11 are the packets for which the switch FE1 sends packet-in events to the controller NC, whereas for the packets NP21 the switch FE1 does not send further packet in event for the controller NC. So the packets for which the switch FE1, FE2, FEM sends packet-in events to the controller NC are first packets NP11 and the subsequent packets are the second packets NP21. If the maximum time difference TD between subsequent packets NP21E and NP12A is greater than a predetermined time threshold, then the packets NP12A and the corresponding subsequent packets are to be recognized as "new" flow F2 compared to the prior "old" flow F1, whereby performing again delay forwarding with new first packets and new second packets of the now further flow F2.

An attacker can understand if a flow rule has been installed by observing the distribution of the forwarding delays for few packets at the start of a network flow. Assuming that the forwarding delay of each of the packets of a flow NP1, NP2, …, NPi is around a mean value FT1, and that the forwarding delay for the each of the subsequent packets of the same flow NPi+1, …, NPj is around a mean value FT2, if FT2<<FT1 the attacker can guess that a flow rule, for such flow, has been installed. In fact, in a reactive setup the first few packets of the flow are handled at the controller NC, while the subsequent ones are forwarded directly by a flow rule in the switch FE1, FE2, FEM. Instead, when a flow rule is already in place for a given flow, then FT1 will be very similar to FT2.

To summarize the present invention, it enables leveraging internal times, internal counters and group tables of flow switches in order to selectively forward certain packets of each flow.

Further the present invention enables to selectively forward the first few packets per flow to a delay element in the network in order to emulate a controller-switch interaction.

Even further the present invention enables to selectively forward back-to-back packets of the same flow across multiple paths in order to prevent an external adversary to estimate the bandwidth capacity of the network.

The present invention in particular provides a method comprising the following steps: When a new packet arrives at a forward element FE:

1. The forwarding element FE checks the packet header of the received packet to identify which flow it belongs to. If the flow rule is installed, the forwarding element FE looks up the corresponding group table to decide on the next action.
2. Depending on the time during which a packet from that specific flow was seen, and depending on the packet counter number, the switch chooses one pre-determined bucket from the group table.
3. The forwarding element FE forwards the packet according to the policy determined by the chosen bucket.
4. Depending on the policy, the forwarding element FE might forward the packet to a delay element.

If a forwarding element FE1, FE2, ... receives a network packet for which it has to contact the controller NC, for example if no matching flow rule in the switches flow table is installed, it sends as usual a notification event to the controller NC.

The present invention has inter alia the following advantages: The present invention obfuscates in particular a controller-switch interaction for packet

forwarding without incurring significant modifications in existing software-defined network elements.

The present invention further only consumes minor additional network resources and does not complicate a controller's packet forwarding logic.
The present invention enables a mimicking at the data plane of a controller-switch interaction at the start of a network flow in the presence of a matching flow rule.

Many modifications and other embodiments of the invention set forth herein will come to mind to the one skilled in the art to which the invention pertains having the benefit of the teachings presented in the foregoing description and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

Claims

1. A method for forwarding data (D), preferably data (D) in form of flows (F1, F2), in a network (N), preferably a software-defined network, wherein the network (N) comprises one or more forwarding elements (FE1, FE2, FEM) for forwarding data (D),
characterized in that
in case the data (D) matches to a present forwarding rule on said forwarding element (FE1, FE2, FEM) said data (D) is forwarded with a time delay – delay forwarding – according to a time delay policy and generated by a delay entity (DE) such that
a first number of packets – first packets (NP1) – of said data (D) is forwarded by said forwarding element (FE1, FE2, FEM) with a first forwarding time delay and wherein
a second number of packets – second packets (NP21) – of the same data (D) is forwarded with a second forwarding time delay, wherein
the first forwarding time delay and the second forwarding time delay are having a certain time difference from each other.

2. The method according to claim 1, characterized in that the first packets (NP11) are beginning packets of said data (D) and the second packets (NP21) are the subsequent packets of said data (D), preferably wherein the first number and/second number of packets is predetermined or fixed.

3. The method according to one of the claims 1-2, characterized in that the first forwarding time delay is equal to a controlling time within a certain deviation interval, wherein the controlling time representing the time for generating and initially providing said present forwarding rule to said forwarding element (FE1, FE2, FEM).

4. The method according to one of claims 1-3, characterized in that first packets (NP11) are forwarded via a different output port of said forwarding

element (FE1, FE2, FEM) than said second packets (NP21), wherein the delay entity (DE) is connected to the output port for the first packets (NP11).

5. The method according to one of the claims 1-4, characterized in that when the time (TD) between subsequent packets (NP21E, NP12A) exceeds a certain time threshold, then the first packet (NP12A) exceeding said time threshold is considered as the first packet of a further first packets of a new data flow (F2).

6. The method according to one of the claims 1-5, characterized in that on only a subset of data (D) arriving at said forwarding element delay forwarding is applied.

7. The method according to claim 6, characterized in that priority information assigned to said data (D) is checked and said subset is excluded from delay forwarding when said priority information exceeds a predefined priority threshold.

8. The method according to claim 6, characterized in that said subset of data (D) is selected probabilistically.

9. The method according to one of the claims 1-8, characterized in that said delay entity (DE) is
a) integrated into the forwarding element (FE1, FE2, FEM) or
b) located within the network (N) outside the forwarding element (FE1, FE2, FEM),
wherein in case of b) said forwarding element (FE1, FE2, FEM) marks the data (D), preferably each packet (NP11, NP21, NP12), to be delayed with a delay indication for later evaluation by the delay entity (DE).

10. The method according to claim 9, characterized in that when the delay entity (DE) is emulated by a software-component on a corresponding network entity of the network (N), preferably wherein said network entity is provided in form of said forwarding element (FE1, FE2, FEM) and/or a controller, for controlling said forwarding element (FE1, FE2, FEM).

11. The method according to one of the claims 1-10, characterized in that different packets (NP11, NP21) belonging to the same data (D) are forwarded by said forwarding element (FE1, FE2, FEM) and/or said delay entity (DE) via different paths in said network.

12. The method according to claim 11, characterized in that said forwarding via different paths (P11, P12, P13; P21, P22) is performed for load balancing within the network (N).

13. The method according to one of the claims 1-12, characterized in that on at least part, preferably all, packets (NP11, NP21) of the data (D) having the same or a similar header, delay forwarding is applied, wherein one or more parameters for headers are defined being evaluated when checking for similarity.

14. A forwarding element (FE1, FE2, FEM) for forwarding data (D), preferably data (D) in form of flows (F, F1, F2), in a network, and connectable to a delay entity (DE), preferably for performing with a method according to one of the claims 1-13,
characterized in that
said forwarding element (FE1, FE2, FEM) being adapted to send received data (D) in case said data (D) matches to a present forwarding rule on said forwarding element (FE1, FE2, FEM) to said delay entity (DE) according to a time delay policy such that
a first number of packets – first packets (NP11) – of said data (D) is forwarded with a first forwarding time delay and wherein
a second number of packets – second packets (NP21) – of the same data (D) is forwarded with a second forwarding time delay, wherein
the first forwarding time delay and the second forwarding time delay are having a certain time difference from each other.

15. A network (N), preferably a software-defined network (N), for forwarding data (D), preferably data in form of flows (F, F1, F2), in a network (N), wherein the network (N) comprises one or more forwarding elements (FE1, FE2, FEM) for

forwarding data according to one or more rules, preferably for performing with a method according to one of the claims 1-13,
characterized by
a delay entity (DE) adapted to forward received data (D) with a time delay – delay forwarding – and
said forwarding element (FE1, FE2, FEM) being adapted to send data (D) in case said data (D) matches to a present forwarding rule on said forwarding element (FE1, FE2, FEM) to said delay entity (DE) according to a time delay policy such that
a first number of packets – first packets (NP11) – of said data (D) is forwarded with a first forwarding time delay and wherein
a second number of packets – second packets (NP21) – of the same data (D) is forwarded with a second forwarding time delay, wherein
the first forwarding time delay and the second forwarding time delay are having a certain time difference from each other.

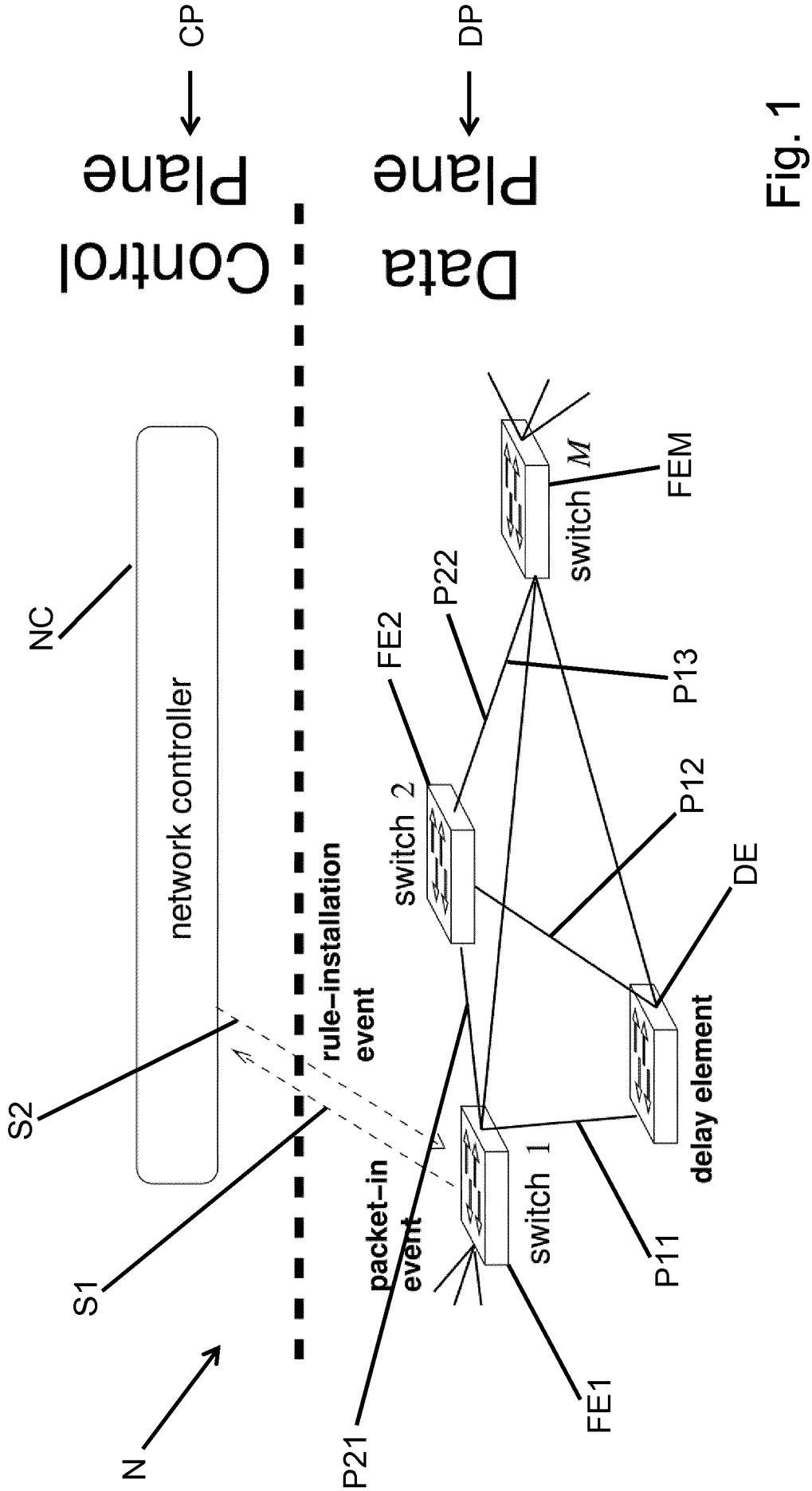
CP
DP
Control Plane
Data Plane
NC
network controller
S2
S1
N
rule–installation event
packet–in event
P21
switch 1
FE1
P11
delay element
DE
P12
switch 2
FE2
P22
P13
switch M
FEM

Fig. 1

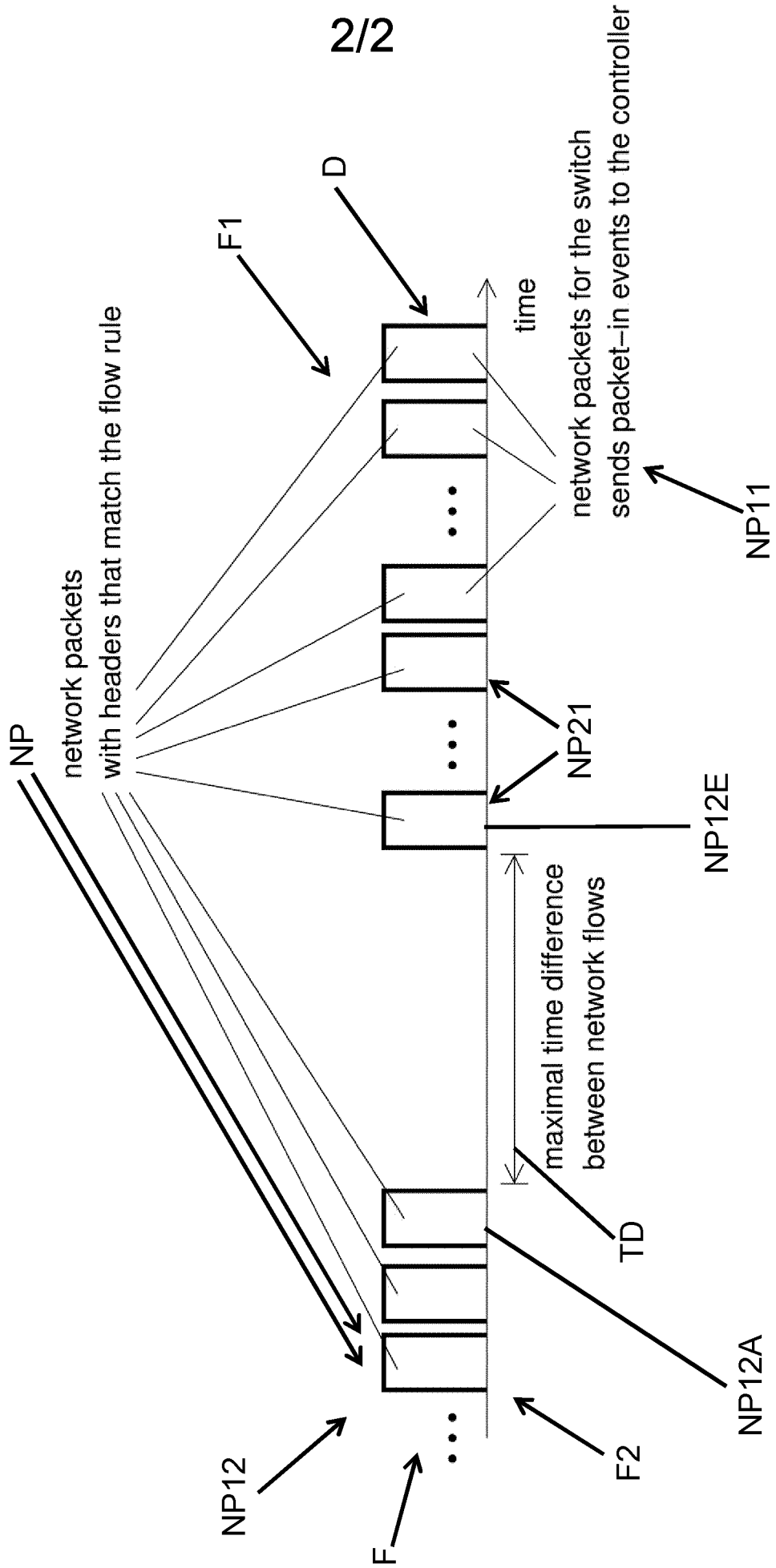
NP
network packets
with headers that match the flow rule
F1
D
time
network packets for the switch
sends packet–in events to the controller
NP11
NP21
NP12E
maximal time difference
between network flows
TD
NP12A
NP12
F
F2

Fig. 2

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/055227

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/721 H04L12/715 H04L12/24 G06F21/75
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 G06F

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

Electronic data base consulted during the international search (name of data base and, where 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 2008/119180 A1 (SMALL DAVID [US] ET AL) 22 May 2008 (2008-05-22) paragraphs [0012], [0017] - [0024]; figures 2,3	1-15
A	US 2006/101513 A1 (GAMMEL BERNDT [DE] ET AL) 11 May 2006 (2006-05-11) paragraphs [0010] - [0014]	1-15

-/--

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

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
5 November 2015	13/11/2015

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/055227

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WASSERMAN S HARTMAN PAINLESS SECURITY D ZHANG HUAWEI M: "Security Analysis of the Open Networking Foundation (ONF) OpenFlow Switch Specification; draft-mrw-sdnsec-openflow-analysis-01.txt" , SECURITY ANALYSIS OF THE OPEN NETWORKING FOUNDATION (ONF) OPENFLOW SWITCH SPECIFICATION; DRAFT-MRW-SDNSEC-OPENFLOW-ANALYSIS-01.TXT, INTERNET ENGINEERING TASK FORCE, IETF; STANDARDWORKINGDRAFT, INTERNET SOCIETY (ISOC) 4, RUE DES FALAISES CH- 1205 GENE, 17 April 2013 (2013-04-17), pages 1-7, XP015091451, [retrieved on 2013-04-17] the whole document -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2015/055227

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008119180 A1	22-05-2008	US 2008119180 A1	22-05-2008
		WO 2008063718 A1	29-05-2008
US 2006101513 A1	11-05-2006	DE 10310781 A1	30-09-2004
		EP 1602017 A2	07-12-2005
		US 2006101513 A1	11-05-2006
		WO 2004081971 A2	23-09-2004