

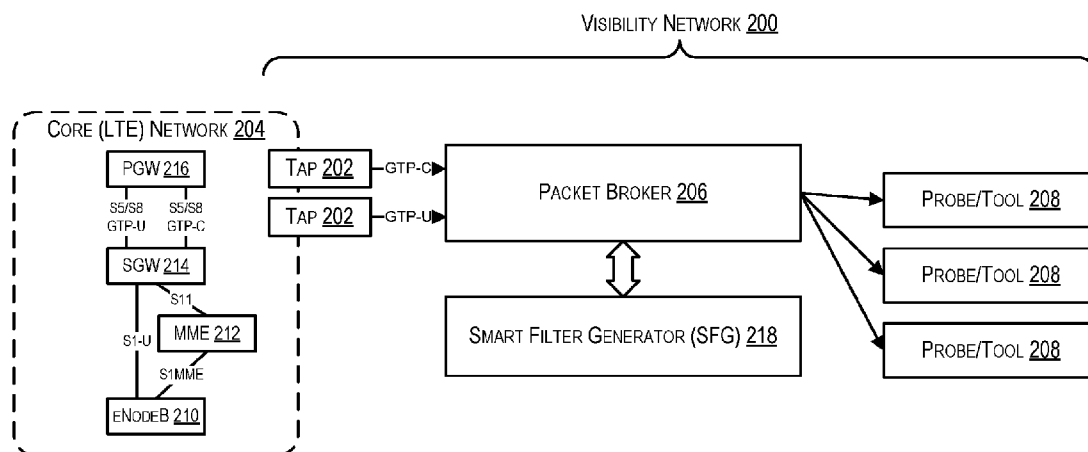
(51) International Patent Classification:
H04N 19/635 (2014.01)(21) International Application Number:
PCT/US2017/055191(22) International Filing Date:
04 October 2017 (04.10.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
201641035761 19 October 2016 (19.10.2016) IN
15/467,766 23 March 2017 (23.03.2017) US(71) Applicant: **EXTREME NETWORKS, INC.** [US/US];
6480 Via del Oro, San Jose, California 95119 (US).(72) Inventors: **NAGARAJ, Manjunath Meda**; 130 Holger
Way, San Jose, California 95134 (US). **ISHTIAQ, Imran**
Ahmed; 130 Holger Way, San Jose, California 95134 (US).**VEDAM, Jude Pragash**; 130 Holger Way, San Jose, Cal-
ifornia 95134 (US).(74) Agent: **FEUSTEL, Richard, M.** et al.; Haley Guiliano
LLP, 75 Broad Street, Suite 1000, New York, NY 10004
(US).(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,
KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.(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,

(54) Title: SMART FILTER GENERATOR

**FIG. 2**

(57) **Abstract:** Techniques for implementing a smart filter generator in a visibility network are provided. In one set of embodiments, the smart filter generator can maintain at least one mapping between (1) a first-order parameter found in network traffic replicated from a core network monitored by the visibility network, and (2) a second-order parameter related to the first-order parameter, where the second-order parameter is not found in the network traffic replicated from the core network. The smart filter generator can further receive, from a user, a user-defined packet filter definition comprising a filtering criterion that makes use of the second-order parameter. The smart filter generator can then translate, based on the at least one mapping, the filtering criterion into a version that makes use of the first-order parameter, and can generate a new packet filter comprising the translated version of the filtering criterion.



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))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

SMART FILTER GENERATOR

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] The present application claims priority to India Provisional Application No. 201641035761, filed October 19, 2016, entitled “SMART EXPERT FILTER GENERATOR.” The present application also claims priority to U.S. Nonprovisional Application No. 15/467,766, filed March 23, 2017, entitled “SMART FILTER GENERATOR.” The entire contents of these applications are incorporated herein by reference in their entireties for all purposes.

BACKGROUND

[0002] In the field of computer networking, a visibility network (also known as a “visibility fabric”) is a type of network that facilitates the monitoring and analysis of traffic flowing through another, “core” network (e.g., a production network). The reasons for deploying a visibility network are varied and can include network management and optimization, business intelligence/reporting, compliance validation, service assurance, security monitoring, and so on.

[0003] FIG. 1 depicts an example visibility network 100 according to an embodiment. As shown, visibility network 100 includes a number of taps 102 that are deployed within a core network 104. Taps 102 are configured to replicate data and control traffic that is exchanged between network elements in core network 104 and forward the replicated traffic to a packet broker 106 (note that, in addition to or in lieu of taps 102, one or more routers or switches in core network 104 can be tasked to replicate and forward data/control traffic to packet broker 106 using their respective SPAN or mirror functions). Packet broker 106 can perform various packet processing functions on the replicated traffic, such as removing protocol headers, filtering/classifying packets based on user-defined filters/rules, and so on. Packet broker 106 can then forward the processed traffic to one or more analytic probes/tools 108, which can carry out various calculations and analyses on the traffic in accordance with the business goals/purposes of visibility network 100.

[0004] With respect to traffic filtering, existing packet brokers can accept and apply user-defined filters that are based on parameters explicitly present in the traffic replicated from a core network (referred to herein as “first-order” parameters). For example, assume that core

network 104 of FIG. 1 is a mobile network and that the traffic replicated from core network 104 is GTP-C/GTP-U traffic. In this scenario, existing implementations of packet broker 106 can accept/apply user-defined filters based on first-order parameters that explicitly appear in GTP traffic such as IMSI, IMEI, APN, QCI, RAT, ULI, etc.

5 **[0005]** However, existing packet brokers generally cannot accept or apply user-defined filters based on parameters that may be associated with, but are not explicitly present in, the traffic replicated from the core network (referred to herein as “second-order” parameters). For instance, returning to the GTP example above, existing implementations of packet broker 106 cannot accept/apply user-defined filters based on second-order parameters that do not
10 appear in GTP traffic such as, e.g., characteristics of the end-user device connected to a particular GTP session (CPU type, RAM amount, screen size, device type, etc.), geographic location of the end-user device, and others.

[0006] If an operator of a visibility network wishes to analyze replicated traffic based on second-order parameters, it is possible to work around this limitation by configuring the
15 network’s packet broker to forward all replicated traffic to the analytic probes/tools. The analytic probes/tools can then store the traffic and perform a post-hoc analysis of the stored data to identify the packets of interest. However, in cases where the volume of traffic generated by the core network is high, this approach will generally require a significant amount of compute and storage resources on the analytic probes/tools in order to store and
20 analyze all of the replicated traffic, which undesirably increases the cost and complexity of the visibility network.

SUMMARY

[0007] Techniques for implementing a smart filter generator in a visibility network are
25 provided. In one set of embodiments, the smart filter generator can maintain at least one mapping between (1) a first-order parameter found in network traffic replicated from a core network monitored by the visibility network, and (2) a second-order parameter related to the first-order parameter, where the second-order parameter is not found in the network traffic replicated from the core network. The smart filter generator can further receive, from a user,
30 a user-defined packet filter definition comprising a filtering criterion that makes use of the second-order parameter. The smart filter generator can then translate, based on the at least one mapping, the filtering criterion into a version that makes use of the first-order parameter,

and can generate a new packet filter comprising the translated version of the filtering criterion.

[0008] The following detailed description and accompanying drawings provide a better understanding of the nature and advantages of particular embodiments.

5

BRIEF DESCRIPTION OF DRAWINGS

[0009] FIG. 1 depicts an example visibility network.

[0010] FIG. 2 depicts a visibility network comprising a smart filter generator according to an embodiment.

10 **[0011]** FIG. 3 depicts a workflow for that can be executed by the smart filter generator of FIG. 2 according to an embodiment.

[0012] FIG. 4 depicts an example architecture for the smart filter generator of FIG. 2 according to an embodiment.

[0013] FIG. 5 depicts an example network switch/router according to an embodiment.

15 **[0014]** FIG. 6 depicts an example computer system according to an embodiment.

DETAILED DESCRIPTION

[0015] In the following description, for purposes of explanation, numerous examples and details are set forth in order to provide an understanding of various embodiments. It will be
20 evident, however, to one skilled in the art that certain embodiments can be practiced without some of these details, or can be practiced with modifications or equivalents thereof.

1. Overview

[0016] Embodiments of the present disclosure describe a smart filter generator that can communicate with, or be integrated within, a packet broker of a visibility network to facilitate
25 the filtering of traffic replicated from a core network based on second-order parameters (i.e., parameters that are not explicitly present in the replicated traffic). According to one set of embodiments, the smart filter generator can maintain a knowledge base comprising one or more sets of mappings between (1) second-order parameters that a user may be interested in using as a basis for filtering traffic from the core network, and (2) first-order parameters

associated with the second-order parameters. Merely by way of example, if the core network is a mobile network, the knowledge base may comprise a first set of mappings between various user equipment (UE) device characteristics (second-order parameters) and IMEI TAC (first-order parameter); a second set of mappings between various geographic identifiers or classifiers (second-order parameters) and ENodeB ID/IP address (first-order parameters); a third set of mappings between various UE software/browser/OS identifiers (second-order parameters) and a user agent string (first-order parameter); and so on. Each of these different sets of mappings can be stored as separate databases in the knowledge base.

[0017] The smart filter generator can further receive, from a user, a packet filter definition that includes a filtering criterion comprising one or more of the second-order parameters included in the knowledge base and one or more corresponding values. For example, the user may provide a packet filter definition that performs a “drop” action on all traffic meeting the filtering criterion (device type=“iPhone”).

[0018] Upon receiving the user-defined packet filter definition, the smart filter generator can consult the knowledge base and translate, based on the mappings in the knowledge base, the second-order parameters and values included in the filtering criterion into corresponding first-order parameters and values. For example, returning to the example above, the smart filter generator can access the device database of the knowledge base and retrieve a list of all IMEI TACs mapped to the device type “iPhone.”

[0019] The smart filter generator can then generate a new packet filter definition that includes, as its filtering criterion, the one or more first-order parameters and values determined via the translation. Finally, this newly generated packet filter definition can be communicated to the packet broker, which can apply the packet filter (in the form of, e.g., an access control list, or ACL) to traffic that is replicated/received from the core network.

[0020] With the general approach described above, the smart filter generator can enable the packet broker to effectively accept and apply user-defined packet filters that are based on second-order parameters not typically found in that traffic. This allows the visibility network to identify/analyze traffic based on such parameters, without needing to perform resource-intensive post-hoc analysis or querying on the analytic probes/tools. As a result, the complexity and cost of the visibility network (in particular with respect to the compute/storage needs of the analytic probes/tools) can be kept low.

[0021] The foregoing and other aspects of the present disclosure are described in further detail below.

2. Visibility Network

[0022] FIG. 2 depicts a visibility network 200 in accordance with an embodiment of the present disclosure. As shown, visibility network 200 includes a number of taps 202 that are deployed in a core network 204 and are configured to replicate traffic exchanged in network 204 to a packet broker 206. In FIG. 2, core network 204 is a mobile LTE network that comprises network elements specific to this type of network, such as an eNodeB 210, a mobility management entity (MME) 212, a serving gateway (SGW) 214, and a packet data network gateway (PGW) 216 which connects to an external packet data network such as the Internet. Further, in this particular example, taps 202 are configured to replicate and forward GTP-C and GTP-U traffic that is exchanged on certain interfaces of core network 204. However, it should be appreciated that core network 204 can be any other type of computer network known in the art, such as a mobile 3G network, a landline local area network (LAN) or wide area network (WAN), etc.

[0023] Upon receiving the replicated traffic via taps 202, packet broker 206 can perform various types of packet processing functions on the traffic (e.g., filtering, classifying, correlating, etc.) as configured by a user/administrator and can forward the processed traffic to one or more analytic probes/tools 208 for analysis. In one embodiment, packet broker 206 can be implemented solely in hardware, such as in the form of a network switch or router that relies on ASIC or FPGA-based packet processors to execute its assigned packet processing functions based on rules that are programmed into hardware memory tables (e.g., CAM tables) resident on the packet processors and/or line cards of the device. In another embodiment, packet broker 206 can be implemented solely in software that runs on, e.g., one or more general purpose physical or virtual computer systems. In yet another embodiment, packet broker 206 can be implemented using a combination of hardware and software, such as a combination of a hardware-based basic packet broker and a software-based “session director” cluster as described in co-owned U.S. Patent Application No. 15/205,889, entitled “Software-based Packet Broker,” the entire contents of which are incorporated herein by reference in its entirety for all purposes.

[0024] As noted in the Background section, while existing packet brokers can accept and apply user-defined packet filters that filter replicated traffic based on first-order parameters

(i.e., parameters that are present in the replicated traffic), existing packet brokers generally cannot filter replicated traffic based on second-order parameters (i.e., parameters which do not appear in the replicated traffic). It is possible to identify traffic that matches one or more second-order parameters by querying the analytic probes/tools of the visibility network or
5 implementing additional/special probes in the core network to select the traffic of interest; however, these solutions generally increase the cost and complexity of the visibility network.

[0025] To address the foregoing and other similar limitations, visibility network 200 of FIG. 2 is enhanced to include a novel smart filter generator (SFG) 218. SFG 218 can be implemented in software, hardware, or a combination thereof. Further, SFG 218 can be
10 implemented as an entity that is separate from packet broker 206 (as shown in FIG. 2), or as an integral component of packet broker 206. As described in further detail below, SFG 218 can enable packet broker 206 to extend its traffic filtering capabilities to filter replicated traffic from core network 204 based on second-order parameters that are not readily available in the replicated traffic. Examples of such second-order parameters in the context of mobile
15 LTE network 204 include end-user equipment capabilities, client browser type, roaming subscriber info, and geographic attributes (e.g., ZIP code, postal address, GPS coordinates, etc.). In this way, SFG 218 can provide more flexible and useful filtering functions to the user/operators of visibility network 200, without increasing the cost and/or complexity of the network.

[0026] It should be appreciated that FIG. 2 is illustrative and not intended to limit embodiments of the present disclosure. For example, the various entities shown in FIG. 2 may be arranged according to different configurations and/or include subcomponents or functions that are not specifically described. One of ordinary skill in the art will recognize other variations, modifications, and alternatives.

3. High-Level SFG Workflow and Architecture

[0027] FIG. 3 depicts a high-level workflow 300 that can be executed by SFG 218 of FIG. 2 to facilitate the filtering of replicated traffic on packet broker 206 based on second-order parameters according to an embodiment.

[0028] Starting with block 302, SFG 218 can receive, via one or more data provisioning
30 interfaces (e.g., CSV using SCP or FTP, CLI, REST API using JSON or XML, SNMP, etc.), mappings between (1) second-order parameters that a user/operator of packet broker 206 may wish to use as a basis for filtering traffic from core network 204, and (2) first-order

parameters that explicitly appear in that traffic. For example, as mentioned previously, in the case where the core network is a mobile LTE network as shown in FIG. 2, the second-order parameters may include UE device capabilities, geographic location information, user agent information, etc., while the first-order parameters may include IMSI, IMEI, APN, QCI, RAT, ULI, MCC, MNC, etc. The mappings may be entered manually by a user or in an automated manner via a provisioning application or script.

[0029] At block 304, SFG 218 can store the received mappings in a local knowledge base. As discussed in further detail below, this knowledge base can comprise a number of separate databases, where each database is configured to maintain mappings for a particular related set of second-order and first-order parameters (e.g., one database for device capability-related parameters, another database for location-related parameters, etc.).

[0030] Once the knowledge base has been populated with parameter mappings pertaining to at least one second-order parameter P1 and at least one corresponding first-order parameter P2, SFG 218 can receive, via a user configuration interface (e.g., CLI, REST API, etc.), a definition of a packet filter from a user, where the user-defined packet filter definition includes a filtering criterion based on second-order parameter P2 (block 306). For example, if P2 is a UE “deviceType” parameter, the packet filter definition received at block 306 may include the filtering criterion (deviceType=“iPhone”).

[0031] Then, at blocks 308-312, SFG 218 can parse the user-defined packet filter definition, identify the use of second-order parameter P2 in the filter’s filtering criterion, and translate, based on the mappings in the knowledge base, the filtering criterion into a version that makes use of corresponding first-order parameter P1 (rather than second-order parameter P2). For instance, returning to the deviceType example above, if the knowledge base includes mappings between the deviceType “iPhone” and two IMEI TACs “ABCDEFGH” and “12345678,” SFG 218 can translate the filtering criterion from (deviceType=“iPhone”) to (IMEI TAC=[“ABCDEFGH”, “12345678”]).

[0032] Once the filtering criterion has been translated, SFG 218 can generate a new packet filter definition that makes use of the translated criterion (block 314). Finally, at block 316, SFG 218 can communicate the newly generated packet filter definition to packet broker 206, which in turn can configure itself to apply the packet filter (in the form of, e.g., an ACL) and thereby use it to filter replicated traffic received from core network 204.

[0033] It should be appreciated that workflow 300 of FIG. 3 is illustrative and various modifications are possible. For example, although shown sequentially, in certain embodiments the execution of blocks 302-304 (which pertain to the receipt and storage of parameter mappings in the knowledge base) may overlap with blocks 306-316 (which pertain to packet filter generation). This may occur if, e.g., SFG 218 receives additional/updated mapping information from users or from an automated provisioning component (e.g., a central support portal) on a periodic basis. One of ordinary skill in the art will recognize other variations, modifications, and alternatives.

[0034] FIG. 4 is a block diagram of one possible architecture (400) for SFG 218 according to an embodiment. As shown, architecture 400 includes a provisioning sub-system 402 that exposes various provisioning interfaces (e.g., CSV, CLI, REST API, and SNMP) usable for populating a knowledge base 404 with parameter mappings. As mentioned previously, this provisioning can be carried out manually by a user or automatically via, e.g., a remote update agent/server that is configured to update the contents of knowledge base 404 on a periodic basis.

[0035] Knowledge base 404 comprises a number of databases 406, 408, 410, and 412 which are used to store the parameter mapping data provisioned via provisioning sub-system 402. Each of these databases may store parameter mappings pertaining to a particular type of filter that a user may wish to define; for example, in FIG. 4, knowledge base 404 includes device, location, user agent, and home network databases. However, it should be appreciated that these are merely exemplary and other types of databases are also possible.

[0036] In one set of embodiments, the interface between provisioning sub-system 402 and knowledge base 404 can be, e.g., an ODBC interface if a MySQL-like database system is used. In other embodiments, the interface between provisioning sub-system 402 and knowledge base 404 can make use of standard inter-process communication (IPC) if a memory-based data structure is used to host the databases of knowledge base 404.

[0037] The following are example database schemas for the device, location, user agent, and home network databases shown in FIG. 4 respectively:

OS Type	Manufacturer	Memory size	Screen Size	Device Name	Device Type	IMEI TAC
Android	Google	4GB	5 Inch	Pixel	Smart Phone	35174605

iOS	Apple	2GB	4.7 inch	iPhone 6	Smart Phone	35925406
Android	Ford	4GB	NA	Car Brand	Connected Car	11238753

Table 1 : Device Database Schema

EnodeB ID	EnodeB IP	MCC	MNC	GPS Co-ordinate	ZIP/Postal Code	City	Region
45461	72.1.6.76	310	560	37.7749° N, 122.4194° W	94104	SFO	Downtown SFO
12344	56.5.6.44	310	580	40.7831° N, 73.9712° W	10029	New York	Manhattan
34344	44.55.6.7	404	45	12.9716° N, 77.5946° E	560061	Bangalore	South Bangalore

Table 2 : Location Database Schema

5

Browser Name	Operating System	Browser Version	User Agent in Regex format
Mozilla/5.0	Windows	5.0	.*?[/\Mozilla][\d.]+
Safari	Mac OS X	600.7.12	.*?[/\ AppleWebKit][\d.]+
Chrome	Android	31.0.1650.59	.*?[/\Chrome][\d.]+

Table 3 : User Agent Database Schema

MCC	MNC	Country	Operator
310	560	USA	AT&T Mobility
310	580	USA	T-Mobile
404	45	India	Airtel

Table 4 : Home Network Database Schema

10

[0038] In addition to provisioning sub-system 402 and knowledge base 404, SFG architecture 400 further includes a user interface sub-system 414 and a filter generation sub-system 416. As shown, user interface sub-system 414 exposes a CLI and/or REST API interface which enables one or more users to provide/enter packet filter definitions. Upon receiving a user-defined packet definition, user interface subsystem 414 can pass the definition to filter generation sub-system 416. In response, filter generation sub-system 416 can parse the user-defined packet filter definition, translate the second-order

15

parameters/values included in the filtering criteria of the user-defined packet definition into corresponding first-order parameters/values based on the parameter mappings in knowledge base 404, and generate a new packet filter definition with the translated criteria.

[0039] Finally, filter generation sub-system 416 can communicate the newly generated packet filter definition to packet broker 206 via an appropriate interface. In cases where SFG 218 is implemented as a software process running within packet broker 206, SFG 218 can communicate the packet filter definition (i.e., program it on the packet broker) using a local CLI interface. Alternatively, in cases where SFG 218 is implemented as a separate/remote entity, SFG 218 can communicate the packet filter definition to packet broker 206 via a remote CLI interface or a REST API interface. One of ordinary skill in the art will recognize other variations, modifications, and alternatives.

4. Example Filters

[0040] The remaining sections of this disclosure provide examples of four types of user-defined, second-order filters that may be supported by SFG 218 in the context where core network 204 is a mobile LTE network: (1) subscriber property-based filters, (2) location/address-based filters, (3) end user equipment (UE) device-based filters, and (4) user plane-based filters.

4.1 Subscriber Property-based Filters

4.1.1 Roaming Subscriber Filters

[0041] This type of filter can enable a user to drop/redirect/replicate/sample the traffic generated by roaming subscribers. The listing below shows an example set of CLI commands that may be entered by the user for providing a definition of this type of filter to SFG 218.

```
sd (config) > filter_type=SEF
```

```
sd (config-SEF_Filter) > set SEF_Type=Roaming
```

```
sd (config-SEF_Filter) > add rule Roaming=<ALL |Country Name |Operator  
Name> Action=< Port-Group | Drop | Sample | Replicate>
```

[0042] Upon receiving this filter definition, SFG 218 can use the “SEF” type field to query the home network database. The result of this query is the network identifier (MNC and MCC) of the network on which the packet broker is deployed. This could be one pair of MCC-MNC or a list. SFG 218 can then generate a new filter using wild cards, as IMSI has MCC and MNC as constituent fields.

[0043] In certain embodiments, this particular type of filter can be enhanced to filter based on name of the country of origin of subscribers. For example, subscribers roaming from Japan or USA can be filtered. This can be achieved by modifying the query to filter the given country name.

- 5 [0044] This type of filter can also be further enhanced to filter by the specific operator and/or country of origin of subscribers (e.g., Vodafone subscribers from the UK).

4.1.2 Tethering Subscriber Filtering

- [0045] This type of filter can enable a user to drop/redirect/replicate/sample the traffic generated by subscribers who are tethering from their mobile devices. The listing below shows an example set of CLI commands that may be entered by the user for providing a definition of this type of filter to SFG 218.

```
sd (config) > filter_type=SEF
sd (config-SEF_Filter) > set SEF_Type=Tethering
sd (config-SEF_Filter) > add rule Tethering=<Device Name> Action=< Port-
15 Group | Drop | Sample | Replicate>
```

- [0046] Upon receiving this filter definition, SFG 218 can use the “SEF” type field to query the device database. Two values can be retrieved from the device database in response to this query: (1) an IMEI TAC code belonging to the device in input, and (2) Operating System. From (2), the user agent database can be queried to extract the possible user agents the operating system may support. From these two lists, two packet filters can be generated and applied in succession (i.e., chained) on packet broker 206.

4.2 Location-based Filters

- [0047] This type of filter can enable a user to drop/redirect/replicate the traffic generated by subscribers present in a particular location. The listing below shows an example set of CLI commands that may be entered by the user for providing a definition of this type of filter to SFG 218.

```
sd (config) > filter_type=SEF
sd (config-SEF_Filter) > set SEF_Type=Location
sd (config-SEF_Filter) > add rule Location=<Zip Code | GPS Lat Long1 GPS
30 Lat Long2 > Action=< Port-Group | Drop | Sample | Replicate>
```

[0048] Upon receiving this filter definition, SFG 218 can use the “SEF” type field to query the location database. The result of this query is a list of ENodeB IDs. SFG 218 can then generate a packet filter based on eNodeB ID and communicate the filter to packet broker 206.

[0049] In certain embodiments, this filter can be enhanced to filter based on postal address/ZIP code and/or the name of a particular city or region such as “South San Francisco.”

4.3 Device-based Filters

[0050] This type of filter can enable a user to drop/redirect/replicate the traffic generated from end user equipment with specific capabilities. The listing below shows an example set of CLI commands that may be entered by the user for providing a definition of this type of filter to SFG 218.

```
sd (config) > filter_type=SEF
sd (config-SEF_Filter) > set SEF_Type=Device
sd (config-SEF_Filter) > add rule Device=< Device Type | Device
Manufacturer | Device Model | Screen Size | Memory Size | Operating System
> Action=< Port-Group | Drop | Sample | Replicate >
```

[0051] Upon receiving this filter definition, SFG 218 can use the “SEF” type field to query the device database. The result of this query is a list of IMEI TAC codes. SFG 218 can then generate a packet filter based on the retrieved list of IMEI TACs and can communicate this filter to packet broker 206.

[0052] In certain embodiments, this filter can be enhanced to filter based on any parameter present in the device database schema or any combination of those parameters. For example, Device = Smart Phone AND Screen Size larger than 5” AND Memory Size larger than 4GB.

4.4 User Plane-based Filters

[0053] This type of filter enables a user to drop/redirect/replicate the traffic generated from a specific browser. The listing below shows an example set of CLI commands that may be entered by the user for providing a definition of this type of filter to SFG 218.

```
sd (config) > filter_type=SEF
sd (config-SEF_Filter) > set SEF_Type=Useragent
```

```
sd (config-SEF_Filter) > add rule Device=< Browser Type | Browser
Type=<value> Version=<Value> > Action=< Port-Group | Drop | Sample |
Replicate >
```

[0054] Upon receiving this filter definition, SFG 218 can use the “SEF” type field to query the user agent database. The result of this query is a list of user agents in regex format. SFG 218 can then generate a packet filter based on the retrieved list and can communicate the filter to packet broker 206.

[0055] In certain embodiments, this filter can be enhanced to filter based on any parameter present in the user agent database schema or any combination of those parameters. For example, Browser = Mozilla AND version = 43.

4.5 S-Flow Generation

[0056] In certain embodiments, this filter action (which enables a user to generate S-Flow records for the traffic generated from a filter) can be added any of the filters described above. The listing below shows an example set of CLI commands that may be entered by the user for enabling an S-Flow filter action with respect to a device-type filter.

```
sd (config) > filter_type=SEF

sd (config-SEF_Filter) > set SEF_Type=Device | Tethering | Location |
Useragent | Roaming

sd (config-SEF_Filter) > set sflow = <enable/disable>

sd (config-SEF_Filter) > add rule Device=< Device Type | Device Model |
Screen Size | Memory Size > Action=< Port-Group >
```

[0057] When this action clause is enabled, SFG 218 can enable S-Flow records at the corresponding port of packet broker 206. Hence, S-Flow records belonging to the match in any filter can be generated.

5. Example Network Device

[0058] FIG. 5 depicts an example network device (e.g., switch/router) 500 according to an embodiment. Network switch/router 500 can be used to implement (either wholly in in part) packet broker 206 described throughout this disclosure.

[0059] As shown, network switch/router 500 includes a management module 502, a switch fabric module 504, and a number of line cards 506(1)-506(N). Management module 502 includes one or more management CPUs 508 for managing/controlling the operation of the

device. Each management CPU 508 can be a general purpose processor, such as a PowerPC, Intel, AMD, or ARM-based processor, that operates under the control of software stored in an associated memory (not shown).

5 [0060] Switch fabric module 504 and line cards 506(1)-506(N) collectively represent the data, or forwarding, plane of network switch/router 500. Switch fabric module 504 is configured to interconnect the various other modules of network switch/router 500. Each line card 506(1)-506(N) can include one or more ingress/egress ports 510(1)-510(N) that are used by network switch/router 500 to send and receive packets. Each line card 506(1)-506(N) can also include a packet processor 512(1)-512(N). Packet processor 512(1)-512(N) is a
10 hardware processing component (e.g., an FPGA or ASIC) that can make wire speed decisions on how to handle incoming or outgoing traffic.

[0061] It should be appreciated that network switch/router 500 is illustrative and not intended to limit embodiments of the present disclosure. Many other configurations having more or fewer components than switch/router 500 are possible.

15 6. Example Computer System

[0062] FIG. 6 depicts an example computer system 600 according to an embodiment. Computer system 600 can be used to implement (either wholly or in part) packet broker 206 described throughout this disclosure.

20 [0063] As shown in FIG. 6, computer system 600 can include one or more general purpose processors (e.g., CPUs) 602 that communicate with a number of peripheral devices via a bus subsystem 604. These peripheral devices can include a storage subsystem 606 (comprising a memory subsystem 608 and a file storage subsystem 610), user interface input devices 612, user interface output devices 614, and a network interface subsystem 616.

25 [0064] Bus subsystem 604 can provide a mechanism for letting the various components and subsystems of computer system 600 communicate with each other as intended. Although bus subsystem 604 is shown schematically as a single bus, alternative embodiments of the bus subsystem can utilize multiple buses.

30 [0065] Network interface subsystem 616 can serve as an interface for communicating data between computer system 600 and other computing devices or networks. Embodiments of network interface subsystem 616 can include wired (e.g., coaxial, twisted pair, or fiber optic Ethernet) and/or wireless (e.g., Wi-Fi, cellular, Bluetooth, etc.) interfaces.

[0066] User interface input devices 612 can include a keyboard, pointing devices (e.g., mouse, trackball, touchpad, etc.), a scanner, a barcode scanner, a touch-screen incorporated into a display, audio input devices (e.g., voice recognition systems, microphones, etc.), and other types of input devices. In general, use of the term “input device” is intended to include all possible types of devices and mechanisms for inputting information into computer system 600.

[0067] User interface output devices 614 can include a display subsystem, a printer, a fax machine, or non-visual displays such as audio output devices, etc. The display subsystem can be a cathode ray tube (CRT), a flat-panel device such as a liquid crystal display (LCD), or a projection device. In general, use of the term “output device” is intended to include all possible types of devices and mechanisms for outputting information from computer system 600.

[0068] Storage subsystem 606 can include a memory subsystem 608 and a file/disk storage subsystem 610. Subsystems 608 and 610 represent non-transitory computer-readable storage media that can store program code and/or data that provide the functionality of various embodiments described herein.

[0069] Memory subsystem 608 can include a number of memories including a main random access memory (RAM) 618 for storage of instructions and data during program execution and a read-only memory (ROM) 620 in which fixed instructions are stored. File storage subsystem 610 can provide persistent (i.e., nonvolatile) storage for program and data files and can include a magnetic or solid-state hard disk drive, an optical drive along with associated removable media (e.g., CD-ROM, DVD, Blu-Ray, etc.), a removable flash memory-based drive or card, and/or other types of storage media known in the art.

[0070] It should be appreciated that computer system 600 is illustrative and not intended to limit embodiments of the present disclosure. Many other configurations having more or fewer components than computer system 600 are possible.

[0071] The above description illustrates various embodiments of the present invention along with examples of how aspects of the present invention may be implemented. The above examples and embodiments should not be deemed to be the only embodiments, and are presented to illustrate the flexibility and advantages of the present invention as defined by the following claims. For example, although certain embodiments have been described with respect to particular process flows and steps, it should be apparent to those skilled in the art

that the scope of the present invention is not strictly limited to the described flows and steps. Steps described as sequential may be executed in parallel, order of steps may be varied, and steps may be modified, combined, added, or omitted. As another example, although certain embodiments have been described using a particular combination of hardware and software,
5 it should be recognized that other combinations of hardware and software are possible, and that specific operations described as being implemented in software can also be implemented in hardware and vice versa.

[0072] The specification and drawings are, accordingly, to be regarded in an illustrative rather than restrictive sense. Other arrangements, embodiments, implementations and
10 equivalents will be evident to those skilled in the art and may be employed without departing from the spirit and scope of the invention as set forth in the following claims.

WHAT IS CLAIMED IS:

1 1. A method comprising:
2 maintaining, by a smart filter generator (SFG) in a visibility network, at least one
3 mapping between:
4 a first-order parameter found in network traffic replicated from a core
5 network monitored by the visibility network; and
6 a second-order parameter related to the first-order parameter, wherein the
7 second-order parameter is not found in the network traffic replicated from the core network;
8 receiving, by the SFG from a user, a user-defined packet filter definition
9 comprising a filtering criterion that makes use of the second-order parameter;
10 translating, by the SFG based on the at least one mapping, the filtering criterion
11 into a version that makes use of the first-order parameter; and
12 generating, by the SFG, a packet filter comprising the translated version of the
13 filtering criterion.

1 2. The method of claim 1 further comprising:
2 communicating the generated packet filter to a packet broker of the visibility
3 network.

1 3. The method of claim 2 wherein the packet broker is configured to apply
2 the generated packet filter to the network traffic replicated from the core network.

1 4. The method of claim 1 wherein the at least one mapping is received from a
2 user via a command line interface (CLI) or a comma separated values (CSV) file interface
3 exposed by the SFG.

1 5. The method of claim 1 wherein the at least one mapping is received from
2 an automated provisioning application or agent via a Representational State Transfer (REST)
3 application programming interface (API) exposed by the SFG.

1 6. The method of claim 1 wherein the at least one mapping comprises
2 multiple sets of mappings between first-order parameters and second-order parameters, each set

3 of mappings corresponding to a related group of parameters and being stored in a separate
4 database of the SFG.

1 7. The method of claim 6 wherein the core network is a mobile network, and
2 wherein the multiple sets of mappings include a set of mappings corresponding to user
3 equipment (UE) device related parameters.

1 8. The method of claim 6 wherein the core network is a mobile network, and
2 wherein the multiple sets of mappings include a set of mappings corresponding to location or
3 address related parameters.

1 9. The method of claim 6 wherein the core network is a mobile network, and
2 wherein the multiple sets of mappings include a set of mappings corresponding to user agent
3 related parameters.

1 10. The method of claim 6 wherein the core network is a mobile network, and
2 wherein the multiple sets of mappings include a set of mappings corresponding to home network
3 related parameters.

1 11. The method of claim 1 wherein the SFG is integrated into a packet broker
2 of the visibility network.

1 12. The method of claim 1 wherein the SFG is implemented as a physical or
2 virtual server that is separate from a packet broker of the visibility network.

1 13. A non-transitory computer readable storage medium having stored thereon
2 program code executable by a smart filter generator (SFG) in a visibility network, the program
3 code causing the SFG to:

4 maintain at least one mapping between:

5 a first-order parameter found in network traffic replicated from a core
6 network monitored by the visibility network; and

7 a second-order parameter related to the first-order parameter, wherein the
8 second-order parameter is not found in the network traffic replicated from the core network;

9 receive, from a user, a user-defined packet filter definition comprising a filtering
10 criterion that makes use of the second-order parameter;

11 translate, based on the at least one mapping, the filtering criterion into a version
12 that makes use of the first-order parameter; and

13 generate a packet filter comprising the translated version of the filtering criterion.

1 14. The non-transitory computer readable storage medium of claim 13
2 wherein the program code further causes the SFG to:

3 communicate the generated packet filter to a packet broker of the visibility
4 network.

1 15. The non-transitory computer readable storage medium of claim 14
2 wherein the packet broker is configured to apply the generated packet filter to the network traffic
3 replicated from the core network.

1 16. The non-transitory computer readable storage medium of claim 13
2 wherein the at least one mapping comprises multiple sets of mappings between first-order
3 parameters and second-order parameters, each set of mappings corresponding to a related group
4 of parameters and being stored in a separate database of the SFG.

1 17. A computer system operating as smart filter generator (SFG) in a visibility
2 network, the computer system comprising:

3 a processor; and

4 a non-transitory computer readable medium having stored thereon program code
5 that, when executed by the processor, causes the processor to:

6 maintain at least one mapping between:

7 a first-order parameter found in network traffic replicated from a
8 core network monitored by the visibility network; and

9 a second-order parameter related to the first-order parameter,

10 wherein the second-order parameter is not found in the network traffic replicated from the core
11 network;

12 receive, from a user, a user-defined packet filter definition comprising a
13 filtering criterion that makes use of the second-order parameter;

14 translate, based on the at least one mapping, the filtering criterion into a
15 version that makes use of the first-order parameter; and
16 generate a packet filter comprising the translated version of the filtering
17 criterion.

1 18. The computer system of claim 17 wherein the program code further causes
2 the processor to:
3 communicate the generated packet filter to a packet broker of the visibility
4 network.

1 19. The computer system of claim 18 wherein the packet broker is configured
2 to apply the generated packet filter to the network traffic replicated from the core network.

1 20. The computer system of claim 17 wherein the at least one mapping
2 comprises multiple sets of mappings between first-order parameters and second-order
3 parameters, each set of mappings corresponding to a related group of parameters and being
4 stored in a separate database of the computer system.

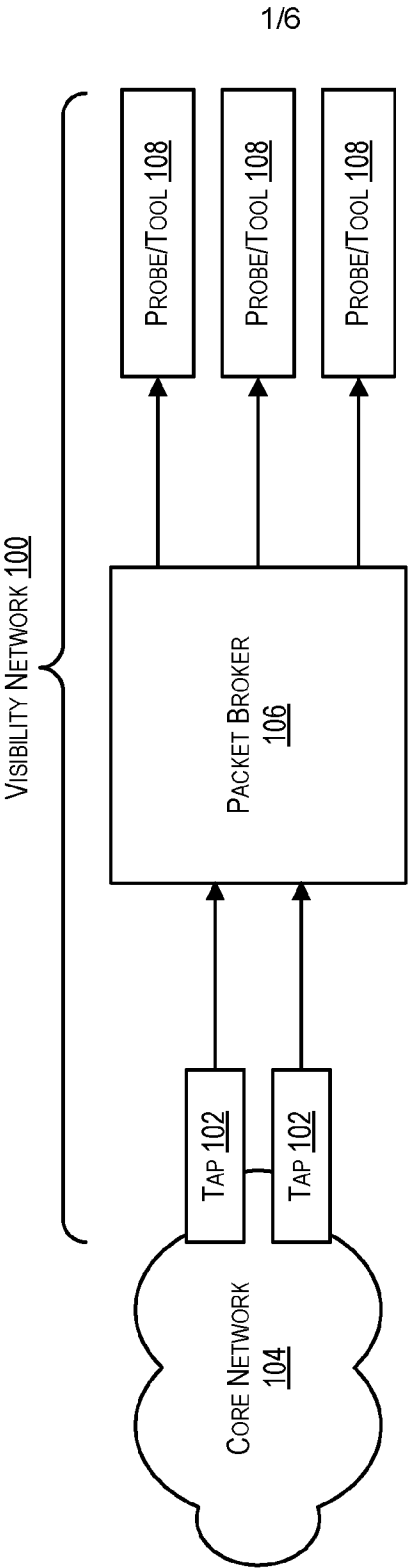


FIG. 1
(Prior art)

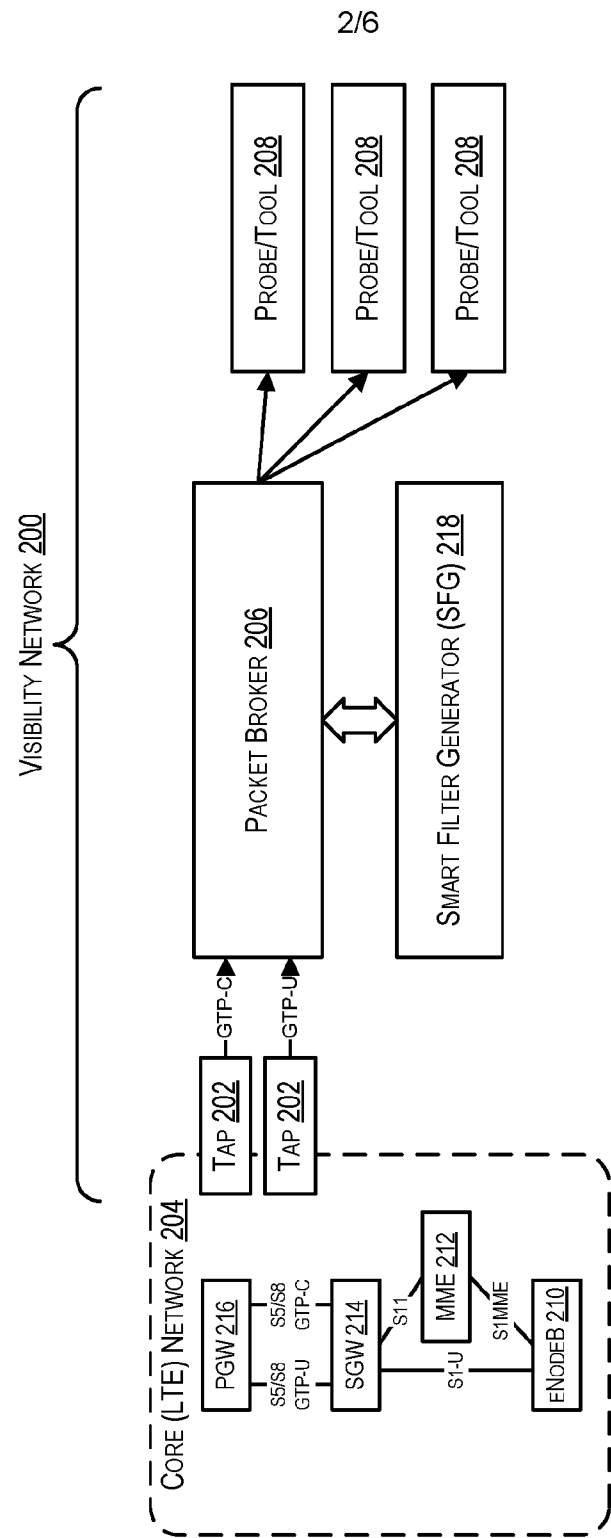
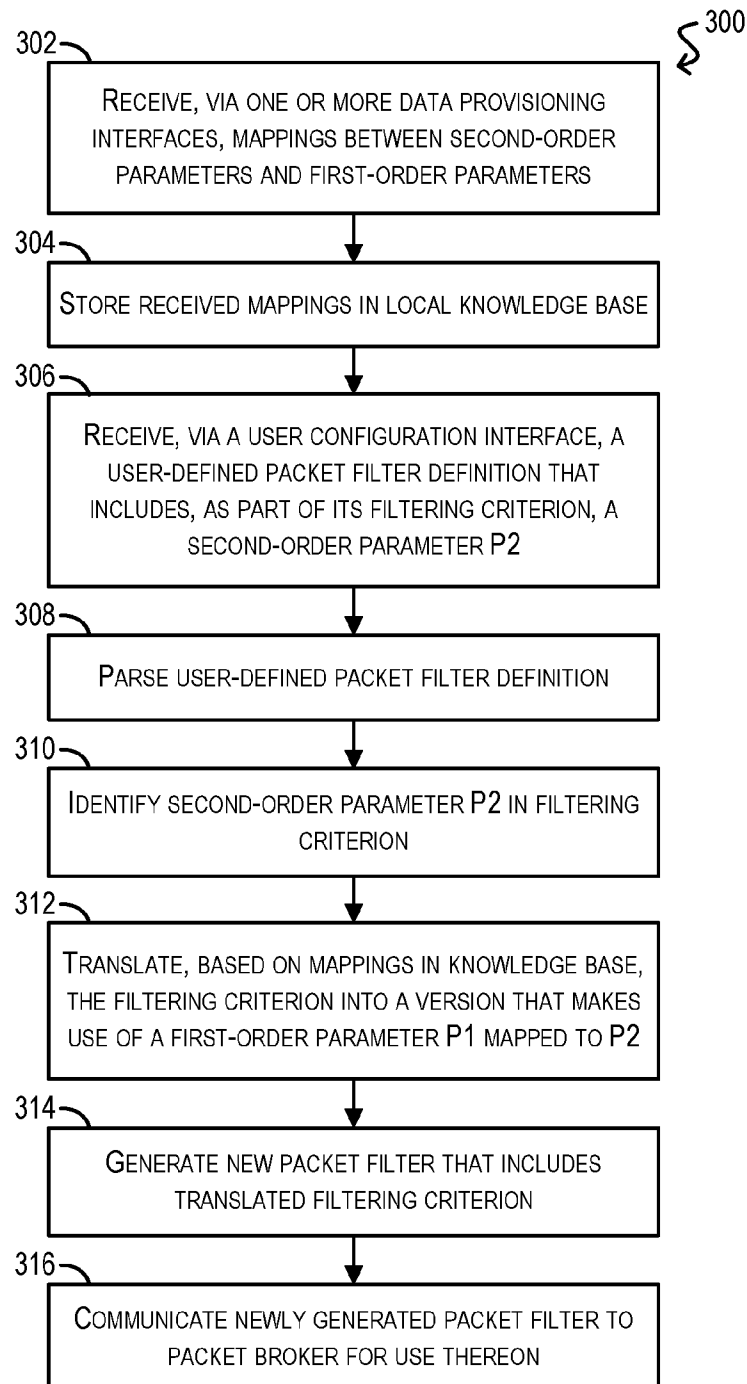


FIG. 2

3/6

**FIG. 3**

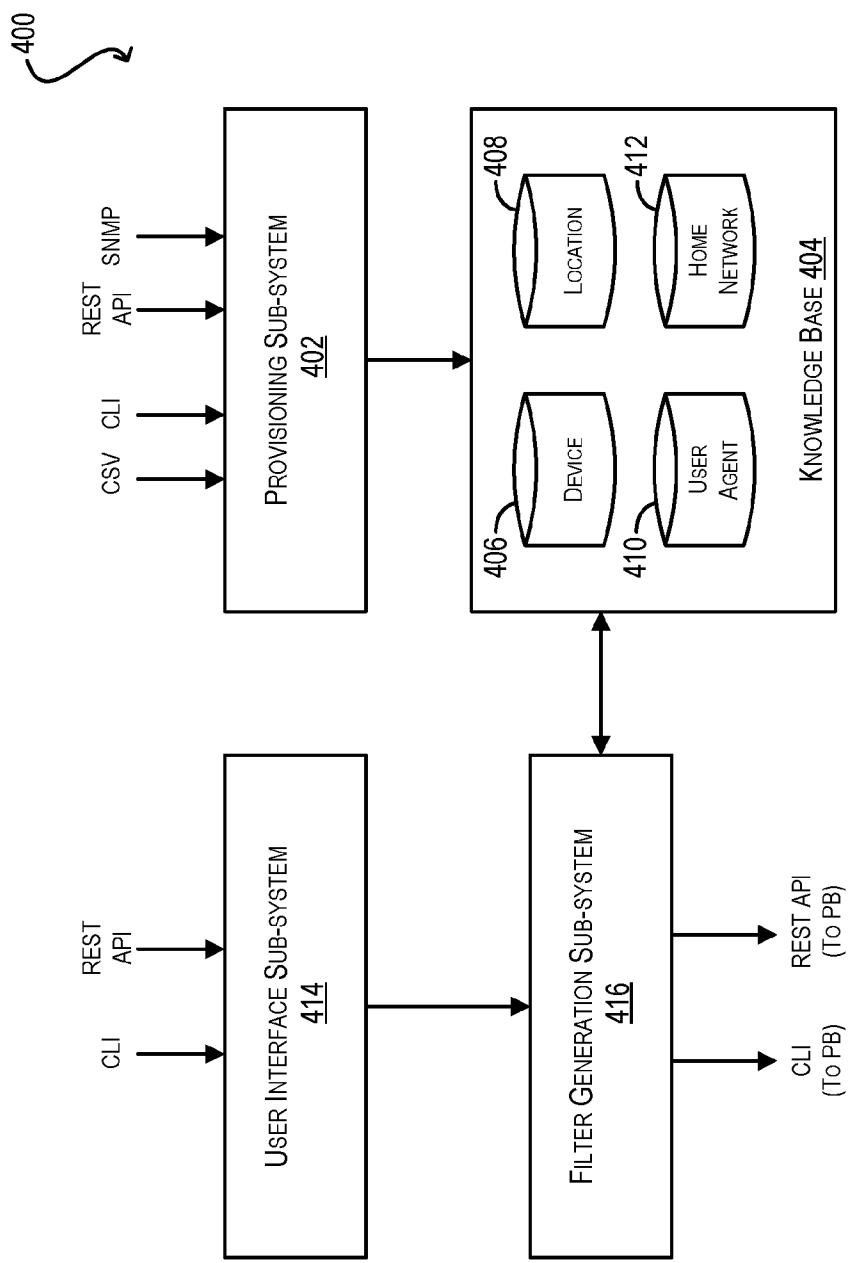
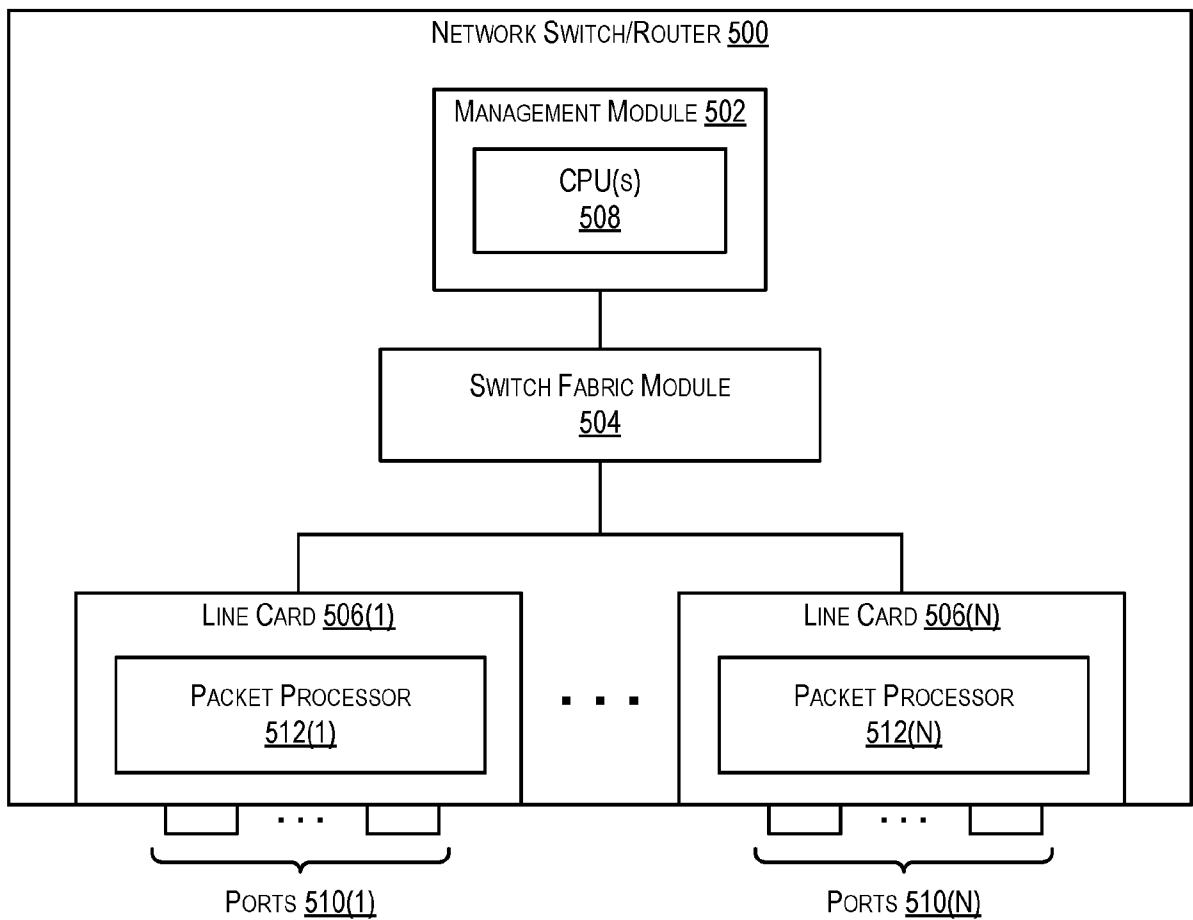
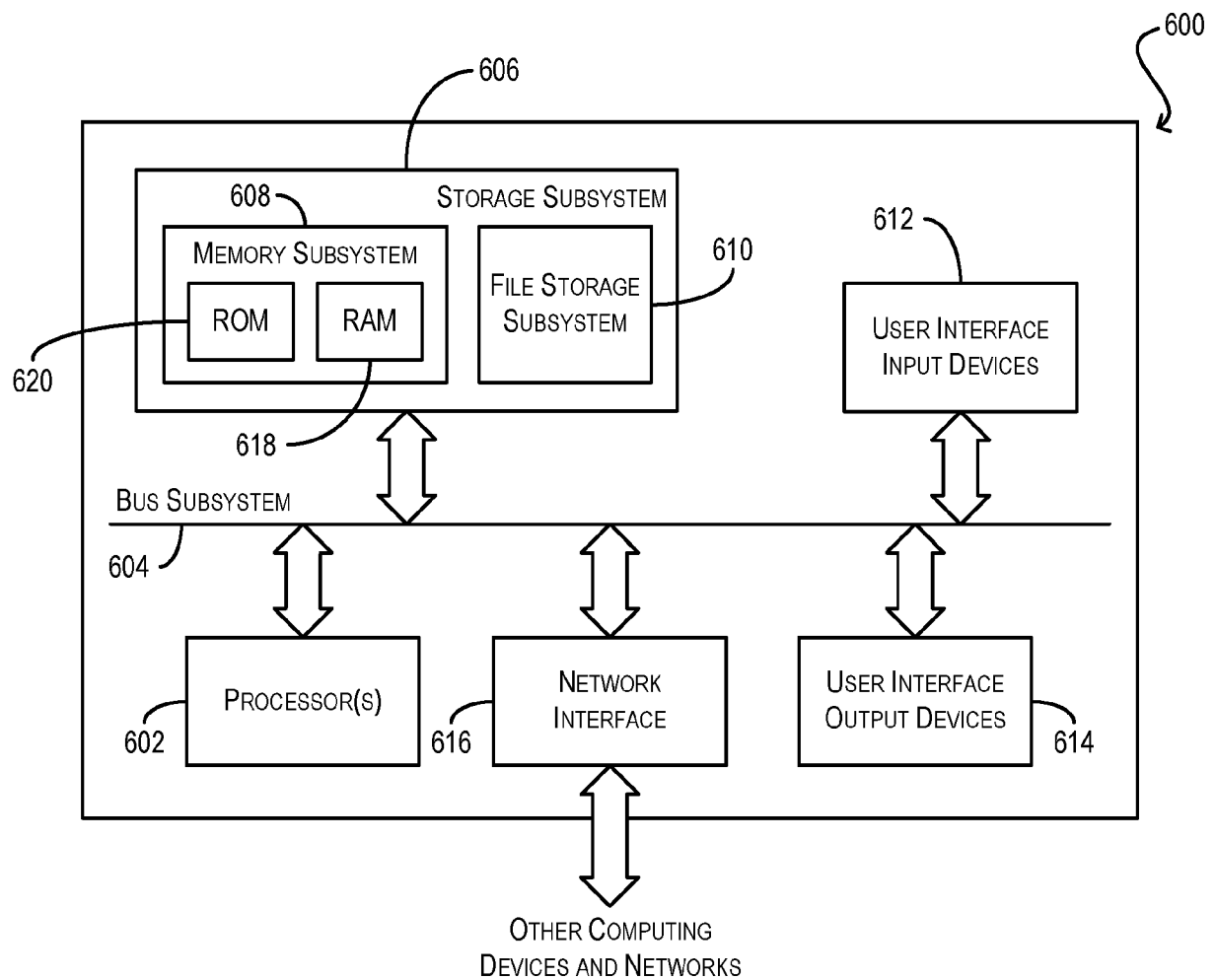


FIG. 4

5/6

**FIG. 5**

6/6

**FIG. 6**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2017/055191

A. CLASSIFICATION OF SUBJECT MATTER <i>H04N 19/635 (2014.01)</i>		
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) H04N 19/00-19/645, G06F 3/00-3/01, 12/00-12/06		
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) PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2015/0319070 A1 (YOUVAL NACHUM) 05.11.2015, abstract, paragraphs [0030], [0039], [0051], [0052], [0064], [0065], [0068], [0078]	1-20
Y	WO 2010/017308 A1 (MOVIK NETWORKS) 11.02.2010, pages 2, 5, 10, 11, 12, 14, 20, 24	1-20
Y	US 2016/0087861 A1 (CHIA-CHEE KUAN et al) 24.03.2016, paragraphs [0002], [0005], [0006], [0028], [0059], [0060]	2, 3, 11, 12, 14, 15, 18, 19
Y	WO 2016/164633 A1 (AMAZON TECHNOLOGIES, INC.) 13.10.2016, paragraphs [0012]-[0014].	4, 5
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
*	Special categories of cited documents:	"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
"A"	document defining the general state of the art which is not considered to be of particular relevance	
"E"	earlier document but published on or after the international filing date	
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	
"O"	document referring to an oral disclosure, use, exhibition or other means	
"P"	document published prior to the international filing date but later than the priority date claimed	
Date of the actual completion of the international search 07 February 2018 (07.02.2018)		Date of mailing of the international search report 14 February 2018 (14.02.2018)
Name and mailing address of the ISA/RU: Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37		Authorized officer V. Aleksandrov Telephone No. (499) 240-25-91