Information Distribution in Autonomic Networking
draft-liu-anima-grasp-distribution-10

Abstract

   This document discusses the requirement of capability of information
   distribution among autonomic nodes in autonomic networks. In general,
   information distribution can be categorized into two different modes:
   1) one autonomic node instantly sends information to other nodes in
   the domain; 2) one autonomic node publishes some information and
   asynchronously some other interested nodes request the published
   information. In the former case, information data will be generated
   and consumed instantly. In the latter case, information data live
   longer in the network.

   These capabilities are basic and fundamental to an autonomous network
   system (i.e. ANI [I-D.ietf-anima-reference-model]). This document
   clarifies possible use cases of information distribution in ANI and
   requirements to ANI so that rich information distribution can be
   natively supported. Possible options to realize the information
   distribution function are also briefly discussed.

Status of this Memo

material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

Table of Contents

1 Introduction

   In an autonomic network, autonomic functions (AFs) running on
   autonomic nodes would exchange information constantly, both for
   controlling/management signaling and data exchange. This document
   discusses the information distribution capability of such exchanges
   between AFs.

   According to the number of participants, information distribution can
   happen with the following scenarios:

      1) Point-to-point (P2P) Communication: information are exchanged
      between two communicating parties from one node to another node.

      2) One-to-Many Communication: information exchanges involve an
      information source and multiple receivers.

   The approaches of distributing information could be categorized into
   two basic models:

      1) An instant communication: a sender connects and sends the
      information data (e.g. control/management signaling,
      synchronization data and so on) to the receiver(s) immediately.

      2) An asynchronous communication: a sender saves the information
      in the network, may or may not publish the information to the
      other who is interested in the published information, to which a
      node asks to retrieve.

   The ANI should have provided a generic way to support these various
   scenarios, rather than assisted by other transport or routing
   protocols (HTTP, BGP/IGP as bearing protocols etc.). In fact, GRASP
   already provides part of the capabilities.

   In this document, we first analyze requirements of information
   distribution in autonomic networks (Section 3), and then introduce
   its relationship to the other modules in ANI (Section 4). After that,
   the node behaviors and extensions to the existing GRASP are
   introduced in Section 5 and Section 6, respectively.

[2](#). **Terminology**

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

[3](#). **Requirements of Advanced Information Distribution**

   If the information exchanged is just short and simple, this can be
   done instantly. In practice, however, this is not always the case. In
   the following cases, a mixture of instant and asynchronous
   communication models is more appropriate.

   1) Long Communication Intervals. The time interval of the
   communication is not necessarily always short and instant.
   Advanced AFs may rather involve heavy jobs/tasks (e.g. database
   lookup, authentication etc.) when gearing the network, so the
   instant mode may introduce unnecessary pending time and become
   less efficient. If simply using an instant mode, the AF has to
   wait until the tasks finish and return. A better way is that an AF
   instantly sends the request but switches to an synchronous mode,
   once the jobs are finished, AFs will get notified.

   2) Common Interest Distribution. As mentioned, some information
   are common interests among AFs. For example, the network intent is
   distributed to network nodes enrolled, which is a typical one-to-
   many scenario. We can also finish the intent distribution by an
   instant flooding (e.g. via GRASP) to every network nodes across
   the network domain. Because of network dynamic, however, not every
   node can be just ready at the moment when the network intent is
   flooded. Actually, nodes may join in the network sequentially. In
   this situation, an asynchronous communication model could be a
   better choice where every (newly joining) node can subscribe the
   intent information and will get notified if it is ready (or
   updated).

   3) Distributed Coordination. With computing and storage resources
   on autonomic nodes, alive AFs not only consumes but also generates
   data information. For example, AFs coordinating with each other as
   distributed schedulers, responding to service requests and
   distributing tasks. It is critical for those AFs to make correct
   decisions based on local information, which might be asymmetric as
   well. AFs may also need synthetic/aggregated data information
   (e.g. statistic info, like average values of several AFs, etc.) to
   make decisions. In these situations, AFs will need an efficient
   way to form a global view of the network (e.g. about resource
   consumption, bandwidth and statistics). Obviously, purely relying
   on instant communication model is inefficient, while a scalable,

common, yet distributed data layer, on which AFs can store and
share information in an asynchronous way, should be a better
choice.

For ANI, in order to support various communication scenarios, an
information distribution module is required, and both instant and
asynchronous communication models should be supported.

4. Information Distribution in ANI

This section describes how the information distribution module fits
into the ANI including what extensions of GRASP are required [I-
D.ietf-anima-grasp].

```
                        +-------------------+
                        |       ASAs        +
                        +-------------------+
                                 ^
                                 |
                                 v
 +--------------------------Info-Dist. APIs----------------------+
 | +---------------+   +-------------------+   +---------------+ |
 | |  Event Queue  |-|-| Selective Flooding|-|-| Info. Storage | |
 | +---------------+   +-------------------+   +---------------+ |
 +--------------------------------------------------------------+
                             ^
                             |
                             v
            +-------------GRASP APIs----------------+
            | +---------------+   +---------------+ |
            | |  GRASP Base   |-|-|   Extension | | |
            | +---------------+   +---------------+ |
            +---------------------------------------+
```
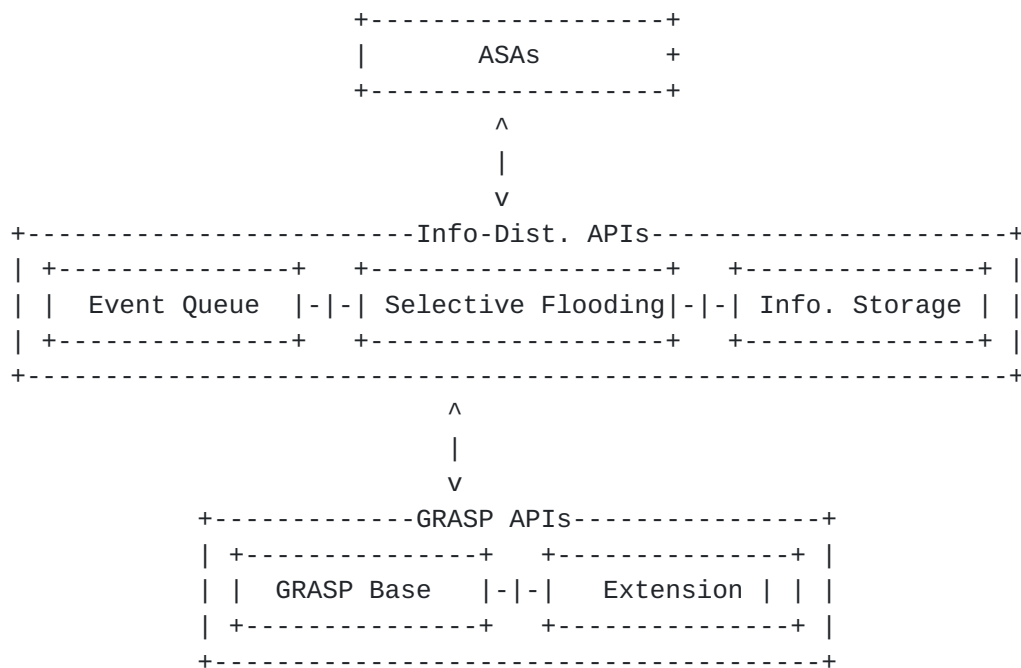
Figure 1. Information Distribution Module and GRASP Extension.

As the Fig 1 shows, the information distribution module includes
three sub-modules, all of which provides APIs for ASAs. Specific
behaviors of these modules is described in Section 5. In order to
support the modules, the GRASP is also extended, which is described
in Section 6.

5. Node Behaviors

ANI is a distributed system, so the information distribution module
must be implemented in a distributed way as well. This means that
every node participate to contribute. In this section, we discuss how

each autonomic node should behave in order to realize the information
distribution module. Node interactions and information data exchange
between network nodes are necessary in order to support the instant
and asynchronous information distribution, which will be introduced
in the follow sections, respectively.

5.1 Instant Information Distribution

In this case, sender(s) and receiver(s) are specified. Information
will be directly sent from the sender(s) to the receiver(s). This
requires that every node is equipped by some signaling/transport
protocols so that they can coordinate with each other and correctly
deliver the information.

5.1.1 Instant P2P and Flooding Communications

Current GRASP already provides the capability to support instant P2P
and flooding. It is natural to use the GRASP Synchronization message
directly for P2P distribution. Furthermore, it is also natural to use
the GRASP Flood Synchronization message for 1-to-all distribution.

However, as mentioned in Section 3, in some scenarios one node needs
to actively send some information to another. GRASP Synchronization
just lacks such capability. An un-solicited synchronization mechanism
is needed. A relevant GRASP extension is defined in Section 6.

5.1.2 Instant Selective Flooding Communication

When doing selective flooding, the distributed information needs to
contain the criteria for nodes to judge which interfaces should be
sent the distributed information and which are not. Specifically, the
criteria contain:

   o Matching condition: a set of matching rules.

   o Matching object: the object that the match condition would be
      applied to. For example, the matching object could be node itself
      or its neighbors.

   o Action: what behavior the node needs to do when the matching
      object matches or failed the matching condition. For example, the
      action could be forwarding or discarding the distributed message.

The criteria information must be include in the message that carries
the distributed information from the sender. The receiving node
decides the action according to the criteria carried in the message.
Still considering the criteria attached with the distributed
information, the node behaviors can be:

o When the Matching Object is "Neighbors", then the node matches
the relevant information of its neighbors to the Matching
Condition. If the node finds one neighbor matches the Matching
Condition, then it forwards the distributed message to the
neighbor. If not, the node discards forwarding the message to the
neighbor.

o When the Matching Object is the node itself, then the node
matches the relevant information of its own to the Matching
Condition. If the node finds itself matches the Matching
Condition, then it forwards the distributed message to its
neighbors; if not, the node discards forwarding the message to the
neighbors.

An example of selective flooding is briefly described in the Appendix
A.

5.2 Asynchronous Information Distribution

Asynchronous information distribution happens in a different way
where sender(s) and receiver(s) are normally not immediately
specified. Both senders and receivers may come up in an asynchronous
way. First of all, this requires that the information can be stored;
secondly, it requires an information publication and subscription
(Pub/Sub) mechanism (corresponding protocol specification of Pub/Sub
is defined in Section 6).

As we sketched in the previous section, in general, each node
requires two modules: 1) Information Storage (IS) module and 2) Event
Queue (EQ) module in the information distribution module. We
introduce details of the two modules in the following sections.

5.2.1 Information Storage

IS module handles how to save and retrieve information for ASAs
across the network. The IS module uses a syntax to index information,
generating the hash index value (e.g. a key) of the information and
mapping the hash index to a certain node in ANI. Note that, this
mechanism can use existing solutions. Specifically, storing
information in an ANIMA network will be realized in the following
steps.

1) ASA-to-IS Negotiation. An ASA calls the API provided by
information distribution module (directly supported by IS sub-
module) to request to store the information somewhere in the
network. Such a request will be checked by the IS module who will
be responsible for the request whether such a request is feasible
according to criteria such as permitted information size.

2) Destination Node Mapping. The information block will be handled by the IS module in order to calculate/map to a destination node in the network. Since ANIMA network is a peer-to-peer network, a typical way is to use dynamic hash table (DHT) to map information to a unique index identifier. For example, if the size of the information is reasonable, the information block itself can be hashed, otherwise, some meta-data of the information block can be used to generate the mapping.

3) Destination Node Negotiation Request. Negotiation request of storing the information will be sent from the IS module to the IS module on the destination node. The negotiation request contains parameters about the information block from the source IS module. According to the parameters as well as the local available resource, the destination node will feedback the source IS module.

4) Destination Node Negotiation Response. Negotiation response from the destination node is sent back to the source IS module. If the source IS module gets confirmation that the information can be stored, source IS module will prepare to transfer the information block; otherwise, a new destination node must be discovered (i.e. going to step 7).

5) Information Block Transfer. Before sending the information block to the destination node that accepts the request, the IS module of the source node will check if the information block can be afforded by one GRASP message. If so, the information block will be directly sent by calling a GRASP API. Otherwise, bulk data transmission with GRASP will be triggered, where multi-time GRASP message sending will be used so that one information block will be transferred by smaller pieces [I-D.ietf-anima-reference-model].

6) Information Writing. Once the information block (or a smaller block) is received, the IS module of the destination node will store the data block in the local storage, which is accessible.

7) (Optional) New Destination Node Discovery. If the previously selected destination node is not available to store the information block, the source IS module will have to identify a new destination node to start a new negotiation. In this case, the discovery can be done by using discovery GRASP API to identify a new candidate, or more complex mechanisms can be introduced.

Similarly, Getting information from an ANIMA network will be realized in the following steps.

1) ASA-to-IS Request. An ASA accesses the IS module via the APIs exposed by the information distribution module. The key/index of

the interested information will be sent to the IS module. An
assumption here is that the key/index should be ready to an ASA
before an ASA can ask for the information. This relates to the
publishing/subscribing of the information, which are handled by
other modules (e.g. Event Queue with Pub/Sub supported by GRASP).

2) Destination Node Mapping. IS module maps the key/index of the
requested information to a destination node, and prepares to start
to request the information. The mapping here follows the same
mechanism when the information is stored.

3) Retrieval Negotiation Request. The source IS module sends a
request to the destination node identified in the previous step
and asks if such an information object is available.

4) Retrieval Negotiation Response. The destination node checks the
key/index of the requested information, and replies to the source
IS module. If the information is found and the information block
can be afforded within one GRASP message, the information will be
sent together with the response to the source IS module.

5) (Optional) New Destination Request. If the information is not
found after the source IS module gets the response from the
original destination node, the source IS module will have to
discover where the location storing the requested information is.

IS module can reuse distributed databases and key value stores like
NoSQL, Cassandra, DHT technologies. storage and retrieval of
information are all event-driven responsible by the EQ module.

5.2.2 Event Queue

The main job of Event Queue (EQ) module is to help ASAs to show
interests to particular information and notify the occurrences of
that in asynchronous communication scenarios. In ANI, information
generated on network nodes is labeled as an event identified with an
event ID, which is semantically related to the topic of the
information. Key features of EQ module are summarized as follows.

1) Event Group: EQ module provides isolated queues for different
event groups. If two groups of AFs could have completely different
purposes or interests, EQ module allows to create multiple queues
where only AFs interested in the same topic will be aware of the
corresponding event queue.

2) Event Prioritization: Events do not have to be delivered in the
same priority. This corresponds to how much important the event
implies. Some of them are more urgent than regular ones.

Prioritization allows AFs to differentiate events (i.e. information) they publish/subscribe.

3) Event Matching: an information consumer has to be identified from the queue in order to deliver the information from the provider. Event matching keeps looking for the subscriptions in the queue to see if there is an exact published event there. Whenever a match is found, it will notify the upper layer to inform the corresponding ASAs who are the information provider and subscriber(s) respectively.

The procedure of how EQ module on every network node works is introduced as follows.

1) Event ID Generation: If information of an ASA is ready, an event ID is generated according to the content of the information. This is also related to how the information is stored/saved by the IS module introduced before. Meanwhile, the type of the event is also specified where it can be of control purpose or user plane data.

2) Priority Specification: According to the type of the event, the ASA may specify its priority to say how this event is wanted to be processed. By considering both aspects, the priority of the event will be determined and ready for enqueuing.

3) Event Enqueue: Given the event ID, event group and its priority, a queue is identified locally if all criteria can be satisfied. If there is such a queue, the event will be simply added into the queue, otherwise a new queue will be created to accommodate such an event.

4) Event Propagation: The published event will be propagated to the other network nodes in the ANIMA domain. A propagation algorithm can be employed to here in order to optimize the propagation efficiency of the updated event queue states.

5) Event Match and Notification: While propagating updated event states, EQ module in parallel keeps matching published events and its interested consumers. Once a match is found, the provider and subscriber(s) will be notified for final information retrieval.

Event contains the address where the information is stored, after a subscriber is notified, it directly retrieves the information from the given location.

5.2.3 **Interface between IS and EQ Modules**

EQ and IS modules are correlated. When an AF publishes information, not only an publishing event is translated and sent to EQ module, but also the information is indexed and stored simultaneously. Similarly, when an AF subscribes information, not only subscribing event is triggered and sent to EQ module, but also the information will be retrieved by IS module at the same time.

5.3 Summary

In summary, the general requirements for the information distribution module on each autonomic node are two sub-modules handling instant communications and asynchronous communications, respectively. For instant communications, node requirements are simple, in which signaling protocols have to be supported. With minimum efforts, reusing the existing GRASP is possible. For asynchronous communications, information distribution module requires event queue and information storage mechanism to be supported.

6. Protocol Specification (GRASP extension)

There are multiple ways to integrate the information distribution module. The principle we follow is to minimize modifications made to the current ANI.

We consider to use GRASP as an interface to access the information distribution module. The main reason is that the current version of GRASP is already an information distribution module for the cases of P2P and flooding. In the following discussions, we introduce how to complete the missing part.

6.1 Un-solicited Synchronization Message (A new GRASP Message)

In fragmentary CDDL, a Un-solicited Synchronization message follows the pattern:

    unsolicited_synch-message = [M_UNSOLDSYNCH, session-id, objective]

A node MAY actively send a unicast Un-solicited Synchronization message with the Synchronization data, to another node. This MAY be sent to port GRASP_LISTEN_PORT at the destination address, which might be obtained by GRASP Discovery or other possible ways. The synchronization data are in the form of GRASP Option(s) for specific synchronization objective(s).

6.2 Selective Flooding Option

In fragmentary CDDL, the selective flood follows the pattern:

```
   selective-flood-option = [O_SELECTIVE_FLOOD, +O_MATCH-CONDITION,
                             match-object, action]
   O_MATCH-CONDITION = [O_MATCH-CONDITION, Obj1, match-rule, Obj2]
      Obj1 = text
      match-rule = GREATER / LESS / WITHIN / CONTAIN
      Obj2 = text
   match-object = NEIGHBOR / SELF
   action = FORWARD / DROP
```

The selective flood option encapsulates a match-condition option
which represents the conditions regarding to continue or discontinue
flood the current message. For the match-condition option, the Obj1
and Obj2 are to objects that need to be compared. For example, the
Obj1 could be the role of the device and Obj2 could be "RSG". The
match rules between the two objects could be greater, less than,
within, or contain. The match-object represents of which Obj1 belongs
to, it could be the device itself or the neighbor(s) intended to be
flooded. The action means, when the match rule applies, the current
device just continues flood or discontinues.

6.3 Subscription Objective Option

In fragmentary CDDL, a Subscription Objective Option follows the
pattern:

```
   subscription-objection-option = [SUBSCRIPTION, 2, 2, subobj]
   objective-name = SUBSCRIPTION
   objective-flags = 2
   loop-count = 2
   subobj = text
```

This option MAY be included in GRASP M_Synchronization, when
included, it means this message is for a subscription to a specific
object.

6.4 Un_Subscription Objective Option

In fragmentary CDDL, a Un_Subscribe Objective Option follows the
pattern:

```
   Unsubscribe-objection-option = [UNSUBSCRIB, 2, 2, unsubobj]
   objective-name = SUBSCRIPTION
   objective-flags = 2
   loop-count = 2
   unsubobj = text
```

This option MAY be included in GRASP M_Synchronization, when
included, it means this message is for a un-subscription to a

specific object.

6.5 Publishing Objective Option

In fragmentary CDDL, a Publish Objective Option follows the pattern:

```
publish-objection-option = [PUBLISH, 2, 2, pubobj] objective-name
= PUBLISH
objective-flags = 2
loop-count = 2
pubobj = text
```

This option MAY be included in GRASP M_Synchronization, when included, it means this message is for a publish of a specific object data.

Note that extended GRASP messages with new arguments inside here will trigger interactions/actions of the underlying information distribution module introduced in Section 5.

7. Security Considerations

The distribution source authentication could be done at multiple layers:

   o Outer layer authentication: the GRASP communication is within
      ACP (Autonomic Control Plane,
      [I-D.ietf-anima-autonomic-control-plane]). This is the default
      GRASP behavior.

   o Inner layer authentication: the GRASP communication might not
      be within a protected channel, then there should be embedded
      protection in distribution information itself. Public key
      infrastructure might be involved in this case.

8. IANA Considerations

    TBD.

9. References

9.1 Normative References

    [I-D.ietf-anima-grasp]
               Bormann, D., Carpenter, B., and B. Liu, "A Generic
               Autonomic Signaling Protocol (GRASP)", draft-ietf-
               animagrasp-15 (Standard Track), October 2017.

9.2 Informative References

    [RFC7575] Behringer, M., "Autonomic Networking: Definitions and
               Design Goals", RFC 7575, June 2015

    [I-D.ietf-anima-autonomic-control-plane]
               Eckert, T., Behringer, M., and S. Bjarnason, "An Autonomic
               Control Plane (ACP)", draft-behringer-anima-autonomic-
               control-plane-13, December 2017.

    [I-D.ietf-anima-stable-connectivity-10]
               Eckert, T., Behringer, M., "Using Autonomic Control Plane
               for Stable Connectivity of Network OAM", draft-ietf-anima-
               stable-connectivity-10, February 2018.

    [I-D.ietf-anima-reference-model]
               Behringer, M., Carpenter, B., Eckert, T., Ciavaglia, L.,
               Pierre P., Liu, B., Nobre, J., and J. Strassner, "A
               Reference Model for Autonomic Networking", draft-ietf-
               anima-reference-model-05, October 2017.

    [I-D.du-anima-an-intent]
               Du, Z., Jiang, S., Nobre, J., Ciavaglia, L., and M.
               Behringer, "ANIMA Intent Policy and Format", draft-
               duanima-an-intent-05 (work in progress), February 2017.

    [I-D.ietf-anima-grasp-api]
               Carpenter, B., Liu, B., Wang, W., and X. Gong, "Generic
               Autonomic Signaling Protocol Application Program Interface
               (GRASP API)", draft-ietf-anima-grasp-api-00 (work in
               progress), December 2017.

    [I-D.carpenter-anima-grasp-bulk-02]
               Carpenter, B., Jiang, S., Liu, B., "Transferring Bulk Data
               over the GeneRic Autonomic Signaling Protocol (GRASP)",
               draft-carpenter-anima-grasp-bulk-02 (work in progress),

June 30, 2018

[3GPP.29.500]
            3GPP, "Technical Realization of Service Based
            Architecture", 3GPP TS 29.500 15.1.0, 09 2018

[3GPP.23.501]
            3GPP, "System Architecture for the 5G System", 3GPP TS
            23.501 15.2.0, 6 2018,
            <http://www.3gpp.org/ftp/Specs/html-info/23501.htm>.

[3GPP.23.502]
            3GPP, "Procedures for the 5G System", 3GPP TS 23.502
            15.2.0, 6 2018, <http://www.3gpp.org/ftp/Specs/html-
            info/23502.htm>.

[5GAA.use.cases]
            White Paper "Toward fully connected vehicles: Edge
            computing for advanced automotive communications", 5GAA
            <http://5gaa.org/news/toward-fully-connected-vehicles-
            edge-computing-for-advanced-automotive-communications/>

Appendix A.

   GRASP includes flooding criteria together with the delivered
   information so that every node will process and act according to the
   criteria specified in the message. An example of extending GRASP with
   selective criteria can be:

      o Matching condition: "Device role=IPRAN_RSG"

      o Matching objective: "Neighbors"

      o Action: "Forward"

   This example means: only distributing the information to the
   neighbors who are IPRAN_RSG.

Authors' Addresses

   Bing Liu
   Huawei Technologies
   Q27, Huawei Campus

     No.156 Beiqing Road
     Hai-Dian District, Beijing 100095
     P.R. China

     Email: leo.liubing@huawei.com

     Sheng Jiang
     Huawei Technologies
     Q27, Huawei Campus
     No.156 Beiqing Road
     Hai-Dian District, Beijing 100095
     P.R. China

     Email: jiangsheng@huawei.com

     Xun Xiao
     Munich Research Center
     Huawei technologies
     Riesstr. 25, 80992, Muenchen, Germany

     Emails: xun.xiao@huawei.com

     Artur Hecker
     Munich Research Center
     Huawei technologies
     Riesstr. 25, 80992, Muenchen, Germany

     Emails: artur.hecker@huawei.com

     Zoran Despotovic
     Munich Research Center
     Huawei technologies
     Riesstr. 25, 80992, Muenchen, Germany

     Emails: zoran.despotovic@huawei.com

Appendix A   Real-world Use Cases of Information Distribution

The requirement analysis in Section 3 shows that generally
information distribution should be better of as an infrastructure
layer module, which provides to upper layer utilizations. In this
section, we review some use cases from the real-world where an
information distribution module with powerful functions do plays a
critical role there.

A.1 Service-Based Architecture (SBA) in 3GPP 5G

In addition to Internet, the telecommunication network (i.e. carrier
mobile wireless networks) is another world-wide networking system.
The architecture of the upcoming 5G mobile networks from 3GPP has
already been defined to follow a service-based architecture (SBA)
where any network function (NF) can be dynamically associated with
any other NF(s) when needed to compose a network service. Note that
one NF can simultaneously associate with multiple other NFs, instead
of being physically wired as in the previous generations of mobile
networks. NFs communicate with each other over service-based
interface (SBI), which is also standardized by 3GPP [3GPP.23.501].

In order to realize an SBA network system, detailed requirements are
further defined to specify how NFs should interact with each other
with information exchange over the SBI. We now list three
requirements that are related to information distribution here.

   1) NF Pub/Sub: Any NF should be able to expose its service status
   to the network and any NF should be able to subscribe the service
   status of an NF and get notified if the status is available. An
   concrete example is that a session management function (SMF) can
   subscribe the REGISTER notification from an access management
   function (AMF) if there is a new user entity trying to access the
   mobile network [3GPP.23.502].

   2) Network Exposure Function (NEF): A particular network function
   that is required to manage the event exposure and distributions.
   In specific, SBA requires such a functionality to register network
   events from the other NFs (e.g. AMF, SMF and so on), classify the
   events and properly handle event distributions accordingly in
   terms of different criteria (e.g. priorities) [3GPP.23.502].

   3) Network Repository Function (NRF): A particular network
   function where all service status information is stored for the
   whole network. An SBA network system requires all NFs to be
   stateless so as to improve the resilience as well as agility of
   providing network services. Therefore, the information of the
   available NFs and the service status generated by those NFs will
   be globally stored in NRF as a repository of the system. This
   clearly implies storage capability that keeps the information in
   the network and provides those information when needed. A concrete
   example is that whenever a new NF comes up, it first of all
   registers itself at NRF with its profile. When a network service
   requires a certain NF, it first inquires NRF to retrieve the
   availability information and decides whether or not there is an
   available NF or a new NF must be instantiated [3GPP.23.502].

(Note: 3GPP CT might finally adopt HTTP2.0/JSON to be the protocol
communicating between NFs, but autonomic networks can also load
HTTP2.0 with in ACP.)

A.2 Vehicle-to-Everything

Carrier networks On-boarding services of vertical industries are also
one of some blooming topics that are heavily discussed. Connected car
is clearly one of the important scenarios interested in automotive
manufacturers, carriers and vendors. 5G Automotive Alliance - an
industry collaboration organization defines many promising use cases
where services from car industry should be supported by the 5G mobile
network. Here we list two examples as follows [5GAA.use.cases].

   1) Software/Firmware Update: Car manufacturers expect that the
   software/firmware of their car products can be remotely
   updated/upgraded via 5G network in future, instead of onsite
   visiting their 4S stores/dealers offline as nowadays. This
   requires the network to provide a mechanism for vehicles to
   receive the latest software updates during a certain period of
   time. In order to run such a service for a car manufacturer, the
   network shall not be just like a network pipe anymore. Instead,
   information data have to be stored in the network, and delivered
   in a publishing/subscribing fashion. For example, the latest
   release of a software will be first distributed and stored at the
   access edges of the mobile network, after that, the updates can be
   pushed by the car manufacturer or pulled by the car owner as
   needed.

   2) Real-time HD Maps: Autonomous driving clearly requires much
   finer details of road maps. Finer details not only include the
   details of just static road and streets, but also real-time
   information on the road as well as the driving area for both local
   urgent situations and intelligent driving scheduling. This asks
   for situational awareness at critical road segments in cases of
   changing road conditions. Clearly, a huge amount of traffic data
   that are real-time collected will have to be stored and shared
   across the network. This clearly requires the storage capability,
   data synchronization and event notifications in urgent cases from
   the network, which are still missing at the infrastructure layer.

A.3 Summary

Through the general analysis and the concrete examples from the real-
world, we realize that the ways information are exchanged in the
coming new scenarios are not just short and instant anymore. More
advanced as well as diverse information distribution capabilities are
required and should be generically supported from the infrastructure

layer. Upper layer applications (e.g. ASAs in ANIMA) access and
utilize such a unified mechanism for their own services.