

ANIMA WG
Internet-Draft
Intended status: Informational
Expires: August 18, 2017

Z. Du
S. Jiang
Huawei Technologies Co., Ltd
J. Nobre
Federal University of Rio Grande do Sul
L. Ciavaglia
Alcatel Lucent
M. Behringer
Cisco Systems
February 14, 2017

ANIMA Intent Policy and Format
draft-du-anima-an-intent-05

Abstract

One of the goals of autonomic networking is to simplify the management of networks by human operators. Intent Based Networking (IBN) is a possible approach to realize this goal. With IBN, the operator indicates to the network what to do (i.e. her intent) and not how to do it. In the field of Policy Based Management (PBM), the concept of intent is called a declarative policy. This document proposes a refinement of the intent concept initially defined in [RFC7575] for autonomic networks by providing a more complete definition, a life-cycle, some use cases and a tentative format of the ANIMA Intent Policy.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 2 |
| 2. Requirements Language and Terminology | 3 |
| 3. Concept of ANIMA Intent Policy | 4 |
| 4. Intent Life Cycle | 4 |
| 5. Use Cases for ANIMA Intent Policy | 6 |
| 6. Distribution of ANIMA Intent Policy | 7 |
| 7. Management of ANIMA Intent Policy | 7 |
| 8. Interpretation of ANIMA Intent Policy | 7 |
| 9. Uniform Format of the ANIMA Intent Policy | 8 |
| 10. Security Considerations | 9 |
| 11. IANA Considerations | 9 |
| 12. Acknowledgements | 9 |
| 13. Change log [RFC Editor: Please remove] | 9 |
| 14. References | 9 |
| Authors' Addresses | 10 |

1. Introduction

One of the goals of autonomic networking is to simplify the management of networks by human operators. Intent Based Networking (IBN) is a possible approach to realize this goal. With IBN, the operator indicates to the network what to do (i.e. her intent) and not how to do it. In the field of Policy Based Management (PBM), the concept of intent is called a declarative policy. This document proposes a refinement of the intent concept initially defined in [RFC7575] for autonomic networks by providing a more complete definition, a life-cycle, some use cases and a tentative format of the ANIMA Intent Policy.

An Autonomic Network must be able to operate with minimum intervention from human operators. However, it still needs to

receive some form of guidance (e.g. ANIMA Intent Policies) in order to fulfill the operator requirements.

In PBM, the Policy Continuum defines the levels at which the policies are defined (policy creation point), consumed (policy execution point) and translated (policy interpretation point). Using PBM, the operator can manage the network as a whole, and does not need to configure each individual devices in the network. The transformation of the high-level/abstract policies to the low-level device configurations is realized automatically by a set of functions usually regrouped inside a Policy Engine.

The use of policies and in particular of declarative policies assumes that the entities in the Autonomic Network receiving the ANIMA Intent Policy are capable of processing (refining and/or executing) the policy with no ambiguity. For that, the format of the ANIMA Intent Policy and the hierarchy of policy levels must be specified.

This document proposes a base format of the ANIMA Intent Policy. Application-specific extensions of the base format should be defined on a per need basis in dedicated documents.

2. Requirements Language and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

Autonomic Function: A feature or function which requires no configuration, and can derive all required information either through self-knowledge, discovery or through Intent.

Autonomic Node: A node which employs exclusively Autonomic Functions.

Legacy Node: A non-autonomic node, i.e., a node which employs some non-autonomic functions.

Autonomic Network: A network containing exclusively Autonomic Nodes. It may contain one or several Autonomic Domains.

Autonomic Domain: A collection of autonomic nodes that instantiate the same Intent.

Autonomic Service Agent: An agent implemented on an Autonomic Node which implements an Autonomic Function.

Intent: An abstract, high-level policy used to operate the network.

ANIMA Intent Policy: A declarative type of policy used in Autonomic Networks.

Configlet: Intent is interpreted on the Autonomic Node, and the results will be interpreted and stored in a local format on the Autonomic Node. This stored version is known as a "configlet".

NOC: A network operations center is the location where network monitoring and control is exercised.

3. Concept of ANIMA Intent Policy

In the scope of autonomic networking, the definition of intent can be found in [I-D.ietf-anima-reference-model], in which intent is described as "an abstract, declarative, high-level policy used to operate an autonomic domain, such as an enterprise network."

An Autonomic Network will comprise multiple ANIMA Intent Policies. Different ANIMA Intent Policies will be "interpreted" by different entities in autonomic networks, and the "level" of understanding of the intent will impact how the intent will be presented to this entity. So there should be "intermediate" mechanisms/functions that cater for the intent translation continuum across the heterogeneity (in policy capabilities) of the network entities. Also, ANIMA Intent Policies will possibly overlap and this overlapping should be managed (e.g., avoid conflicts, resolve applicable policies in context).

4. Intent Life Cycle

This section describes a top-down flow about how an ANIMA Intent Policy is derived through an autonomic network.

1. Business goals: The network owner wants the network to follow some business goals. These goals are initially not formalised in a particular way. A Domain Specific Language (DSL) is used to format these goals in a form subsequent components can interpret and process.
2. ANIMA Intent Policy (or Intent): Is the formalisation of business goals so that computer can deal with them. It is encoded as a file (or several files), and this file must be "given to the network".

3. Ingestion: The Intent file(s) get instantiated on an autonomic node. On a particular node, an intent file is "ingested". After that, it needs to be distributed.
4. Intent Distribution: Intent is flooded to all nodes in a network. Every node has a copy of the original "Intent" file(s), without modification. Each node re-distributes the original Intent files, without modification. Therefore, Intent is optional and transitive in nature. The Intent files must now be interpreted by each node. Editor's note: need to better defined meaning of "optional" and "transitive".
5. Intent splitting (on each node): Intent is split into sections, one for the ANI itself, others for specific Autonomic Functions. ASAs are notified if there is new Intent for them. Some intent sections may not apply to a particular node. Now each component of a node (ANI, all ASAs) know their respective Intent.
6. Intent Interpretation (on each node, by each function): The ANI as well as all ASAs on a node interpret their respective Intent section(s). It gets translated into a "target configuration", taking into account local state. For this translation, it may be necessary for ASAs to communicate with ASAs on other nodes, to pass on resources (IP addresses), to negotiate, etc. All such communications may be triggered by Intent, but the communications themselves are not Intent. (NB: This interpretation could also be done centrally, and the resulting configurations distributed; This is of course an option, but out the scope of ANIMA.) After interpreting Intent locally on each node, each node has target configlet to apply. Editor's note: define new terms such as "configlet"
7. Conflict Resolution with non-autonomic management (on each node): The target configlet resulting from Intent has the lowest priority; meanwhile, any other management method (CLI, NETCONF, etc.) overrides Intent.
8. Conflict Resolution between autonomic components (on each node): Each autonomic function needs to register with a "conflict resolution function" which parameters it modifies; in case of conflict, the conflict resolution function takes a decision and feeds that back to the autonomic functions. This may modify the target configlet.
9. Applying the target configlet.
10. Feedback loops to NOC: The NOC needs to know about certain conditions, such as conflicts with non-autonomic management.

Not all conflicts can be resolved automatically, so they may require NOC actions. Undesirable states (deviations from expected default behaviour) may have to be communicated too. To some extent, Intent itself can specify which conditions should trigger feedback loops to the NOC. Feedback loops may happen at other phases as well (ex: 8).

5. Use Cases for ANIMA Intent Policy

In this section, some use cases are introduced to clarify the concept of ANIMA Intent Policy. It should be noted that intent is defined per Autonomic Function, and can also be a general one related to multiple AFs.

The first example is about "arranging VM guest distribution". The autonomic network is supposed to be able to monitor the CPU/power utilization on each host machine, and control the status of each host machine (e.g. turn on/off). The operator may have an intent "there should be enough hosts to keep CPU utilization less than 70%", and also another one "there are few enough hosts powered so that electricity isn't wasted".

These two intents can both influence the ASA responsible for controlling how many hosts are needed. The final decision is made according to multiple factors, including network environment and intents entered by the operators.

In this case, the first intent should have a higher priority than the later one. The two intents should be analyzed and coordinated to ensure the ASA act rightly.

Another example is about coordination of "load balancing" intent and "energy saving" intent. Autonomic Network of Operator A is composed of Autonomic Function Agents such as load balancing (LB_AFA) and energy saving (ES_AFA). Operator A wants to limit the proportion of links loaded over a certain threshold and thus defines an Intent to activate load balancing if the load is superior to 0.6 on more than 30% of the links.

Meanwhile, operator A wants different load balancing policies per (technology, administrative, topology) domain. Let's consider a metropolitan network domain and a core network domain, or different LB policy for border routers than interior routers. For the metropolitan network domain, Operator A defines an Intent to minimize the link load variance. For the core network domain, Operator A applies the previously defined intent (activate load balancing if the load is superior to 0.6 on more than 30% of the links).

The intents will be distributed to the right network domain, and take effect after being interpreted and coordinated, and it is easy to change them without the need to configure every device manually.

6. Distribution of ANIMA Intent Policy

The distribution of intent can be done by using GRASP [I-D.ietf-anima-grasp] and ACP [I-D.ietf-anima-autonomic-control-plane]. The operator can issue a new intent or modify an intent through any authorized nodes in the autonomic network. After that, the intent will be flooded to all the nodes in the autonomic network. Another scenario is that when a new node joins into an autonomic domain, it may receive an intent from its neighbor.

For example, GRASP can be used to communicate version number of the intent, and meanwhile, a URL where to find it.

{Editor Notes: other distribution methods are also possible. }

7. Management of ANIMA Intent Policy

Every Autonomic Node in the Autonomic domain should own an intent with the same version. Any updating of intent will cause the change of the intent version number. To ensure all the nodes own the same intent, the nodes should be able to communicate with neighbors in the domain about the version of the intent. If its neighbor has a newer version of intent, it can request an intent update.

If the operator issues a new intent or modify intents, it will trigger a domain level updating of intent. Nodes in the Autonomic Network should be aware which domain it belongs to, and accept intent for that domain.

{Editor Notes: talk about the questions as follows. When/on which triggers are intents generated, updated? How the domain(s) are defined and recognized (if I am an AFA, how do I know I am part of domain x, y or z...?). }

8. Interpretation of ANIMA Intent Policy

After receiving an intent, the Autonomic Node should confirm whether it is acceptable, according to the domain name information, intent version, signature, and so on. If it passes the validation, an intent interpretation module will be involved to decide which ASAs will be involved in. Coordination of intents may be needed before the execution of the policies interpreted from the intent.

{Editor Notes: talk about the questions as follows. How the AFAs receive, understand and react to an intent? }

{Editor Notes: how the splitting (step 5 in the Life Cycle section) happens here can be explained more here. It would be better that an example can be introduced here.}

9. Uniform Format of the ANIMA Intent Policy

{Editor Notes: Format of Intent is FFS. It is suggested to contain the following information.}

This section proposes a uniform intent format. It uses the tag-based format.

Autonomic intent: The root tag for the Autonomic Network Intent.

Intent type: It indicates the intent type, which is associated with a specific Autonomic Function.

Autonomic domain: It indicates the domain of the Autonomic Network. It is also the scope of the Autonomic Network Intent.

Intent version: It indicates the version of the ANIMA Intent Policy. This is an important feature for synchronization.

Model version: The version of the model used to define the intent.

Name: The name of the intent which describes the intent for human operators.

Signature: The signature is used as a security mechanism to provide authentication, integrity, and non-repudiation.

Timestamp: The timestamp of the creation of the intent using the format supported by the IETF [TBC].

Lifetime: The lifetime in which the intent may be observed. A special case of the lifetime is the definition of permanent intents.

Content: It contains the main information of the intent. It may include objects, policies, goals and configuration data. The detailed contents and formats should be defined under their specific situations by documents that specifies the Autonomic Service Agent. Within the content, there may be sub_intents.

10. Security Considerations

Relevant security issues are discussed in [I-D.ietf-anima-grasp]. The ANIMA Intent Policy requires strong security environment from the start, because it would be great risk if the ANIMA Intent Policy had been maliciously tampered. The Autonomic Intent should employ a signature scheme to provide authentication, integrity, and non-repudiation.

11. IANA Considerations

This document defines one new format. The IANA is requested to establish a new assigned list for it.

12. Acknowledgements

The authors of this draft would like to thank the following persons for their valuable feedback and comments: Bing Liu, Brian Carpenter, Michael Richardson, Joel Halpern, John Strassner, and Jason Coleman.

This document was produced using the xml2rfc tool [RFC2629].

13. Change log [RFC Editor: Please remove]

draft-du-anima-an-intent-00: original version, 2015-06-11.

draft-du-anima-an-intent-01: add intent use case section, add some elements for the format section, and coauthor Jeferson Campos Nobre and Laurent Ciavaglia, 2015-07-06.

draft-du-anima-an-intent-02: add the intent concept section, and some other sections, 2015-10-14.

draft-du-anima-an-intent-03: modify the use case section, and add some other contents, 2016-03-17.

draft-du-anima-an-intent-04: modify the use case section, add the procedure section, and reorganize contents, 2016-07-08.

draft-du-anima-an-intent-05: modify the use case section, and delete some sections, 2017-02-15.

14. References

[I-D.ietf-anima-autonomic-control-plane]
Behringer, M., Eckert, T., and S. Bjarnason, "An Autonomic Control Plane", draft-ietf-anima-autonomic-control-plane-05 (work in progress), January 2017.

- [I-D.ietf-anima-grasp]
Bormann, C., Carpenter, B., and B. Liu, "A Generic
Autonomic Signaling Protocol (GRASP)", draft-ietf-anima-
grasp-09 (work in progress), December 2016.
- [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-02 (work in progress), July 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629,
DOI 10.17487/RFC2629, June 1999,
<<http://www.rfc-editor.org/info/rfc2629>>.
- [RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A.,
Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic
Networking: Definitions and Design Goals", RFC 7575,
DOI 10.17487/RFC7575, June 2015,
<<http://www.rfc-editor.org/info/rfc7575>>.

Authors' Addresses

Zongpeng Du
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: duzongpeng@huawei.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: jiangsheng@huawei.com

Jeferson Campos Nobre
Federal University of Rio Grande do Sul
Porto Alegre
Brazil

Email: jcnobre@inf.ufrgs.br

Laurent Ciavaglia
Alcatel Lucent
Route de Villejust
Nozay 91620
France

Email: laurent.ciavaglia@alcatel-lucent.com

Michael Behringer
Cisco Systems
Building D, 45 Allee des Ormes
Mougins 06250
France

Email: mbehring@cisco.com