ANIMA
Internet-Draft
Intended status: Informational
Expires: April 20, 2015

M. Behringer, Ed.
Cisco
B. Carpenter
Univ. of Auckland
T. Eckert
Cisco

October 17, 2014

A Reference Model for Autonomic Networking
draft-behringer-anima-reference-model-00

Abstract

This document describes a reference model for Autonomic Networking. The goal is to define how the various elements in an autonomic context work together, to describe their interfaces and relations.

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 April 20, 2015.

Copyright Notice

Table of Contents

1. Introduction

The document "Autonomic Networking - Definitions and Design Goals" [I-D.irtf-nmrg-autonomic-network-definitions] explains the fundamental concepts behind Autonomic Networking, and defines the relevant terms in this space. In section 5 it describes a high level reference model. This document defines this reference model with more detail, to allow for functional and protocol specifications to be developed in an architecturally consistent, non-overlapping manner.

As discussed in [I-D.irtf-nmrg-autonomic-network-definitions], the goal of this work is not to focus exclusively on fully autonomic

nodes or networks. In reality, most networks will run with some autonomic functions, while the rest of the network is traditionally managed. This reference model allows for this hybrid approach.

2. The Network View

This section describes the various elements in a network with autonomic functions, and how these entities work together, on a high level. Subsequent sections explain the detailed inside view for each of the autonomic network elements, as well as the network functions (or interfaces) between those elements.

Autonomic entities include:

- Network elements: A network element can be a fully or partially autonomic node. It runs autonomic functions, and interacts with other autonomic nodes.

- Registrar: Security is a fundamental requirement in an autonomic network. For nodes and services to securely interact without the need to provision shared secrets, a trust infrastructure must be in place. The registrar is the trust anchor in an autonomic domain.

- MASA: The MASA is service for devices of a particular vendor. It can validate the identity of devices that are to be used in an autonomic domain, assert which device is owned by which domain, etc.

3. Entities in an Autonomic Network

This section describes all the elements in an autonomic network, their function, internal organisation and architecture. In the network view in Section 2, this section describes the "boxes". The following sections describes how those boxes interact, and the necessary means to do so (addressing, routing, etc).

3.1. The Network Element

This section describes an autonomic network element and its internal

architecture. The reference model explained in [I-D.irtf-nmrg-autonomic-network-definitions] shows the sources of information that an autonomic service agent can leverage: Self-knowledge, network knowledge (through discovery), Intent, and feedback loops. Fundamentally, there are two levels inside an autonomic node: the level of autonomic service agents, which uses the functions of the autonomic networking infrastructure. Figure 1 illustrates this concept.

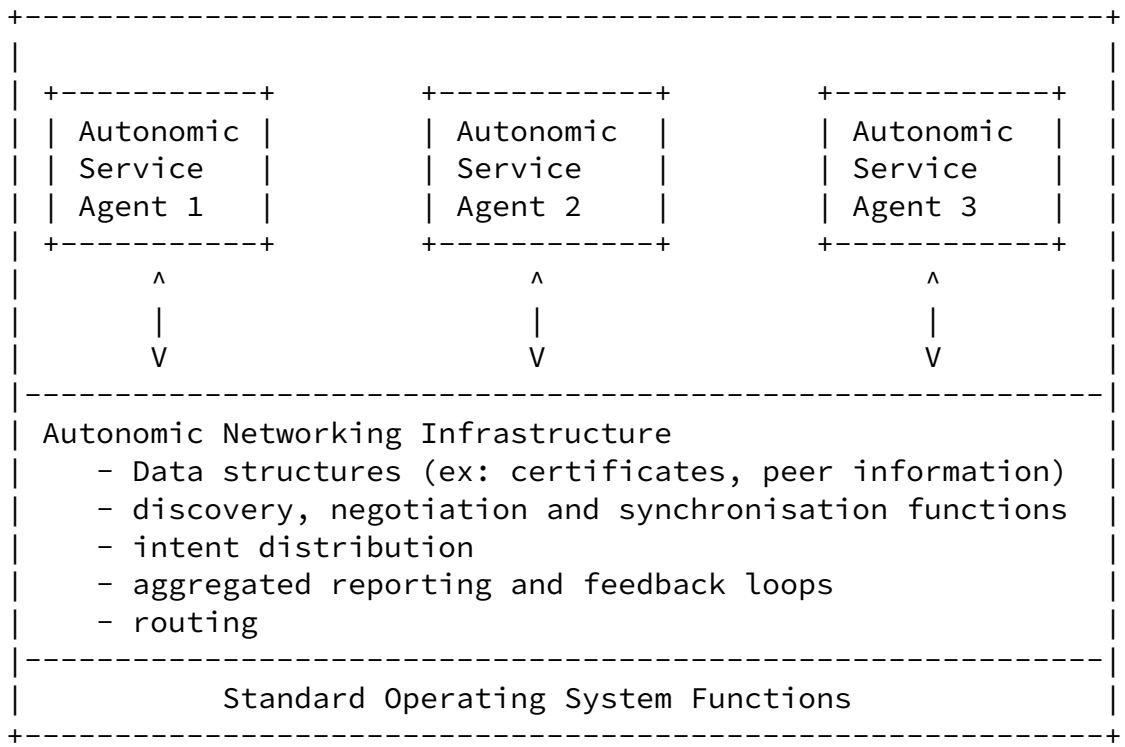

Figure 1

The Autonomic Networking Infrastructure (lower part of Figure 1) contains node specific data structures, for example trust information about itself and its peers, as well as a generic set of functions, independent of a particular usage. This infrastructure should be generic, and support a variety of Autonomic Service Agents (upper part of Figure 1). The Autonomic Control Plane is the summary of all

interactions of the Autonomic Networking Infrastructure with other nodes and services.

The use cases of "Autonomics" such as self-management, self-optimisation, etc, are implemented as Autonomic Service Agents. They use the services and data structures of the underlying autonomic networking infrastructure. The underlying Autonomic Networking Infrastructure should itself be self-managing.

3.2. The Registrar Element

This section describes the registrar function in an autonomic network. It explains the tasks of a registrar element, and how registrars are placed in a network, redundancy between several, etc. [tbc]

3.3. The MASA

tbc

4. Naming

Inside a domain, each autonomic device needs a domain specific identifier. [tbc]

5. Addressing

tbc

6. Trust Infrastructure

Autonomic nodes have direct interactions between themselves, which must be secured. Since an autonomic network does not rely on configuration, it is not an option to configure for example pre-shared keys. A trust infrastructure such as a PKI infrastructure must be in place. This section describes the principles of this trust infrastructure.

A completely autonomic way to automatically and securely deploy such a trust infrastructure is to set up a trust anchor for the domain, and then use an approach as in the document "Bootstrapping Key Infrastructures" [I-D.pritikin-bootstrapping-keyinfrastructures].

7. Autonomic Control Plane

This section describes how autonomic nodes interact. In the network view in Section 2 this section describes the "lines" and "arrows" between nodes. The summary of autonomic interactions forms the "Autonomic Control Plane". This control plane can be either implemented in the global routing table of a node, such as IGPs in today's networks; or it can be provided as an overlay network, as described in [I-D.behringer-autonomic-control-plane]. This section describes the function of the autonomic control plane, independent of its implementation.

7.1. Discovery

Traditionally, most of the information a node requires is provided through configuration or northbound interfaces. An autonomic function should only minimally rely on such northbound interfaces, therefore it needs to discover resources in the network. This section describes various discovery functions in an autonomic network.

Discovering nodes and their properties: A core function to establish an autonomic domain is the discovery of autonomic nodes, primarily adjacent nodes. This may either leverage existing neigbhour discovery mechanisms, or new mechanisms.

Discovering services: Network services such as AAA should also be discovered and not configured. Service discovery is required for such tasks. An autonomic network can either leverage existing service discovery functions, or build a new approach.

7.2. Negotiation and Synchronisation

Autonomic nodes must negotiate and/or synchronise parameters, etc. The document "A Generic Discovery and Negotiation Protocol for Autonomic Networking" [I-D.carpenter-anima-gdn-protocol] explains

requirements for negotiation and synchronisation in an autonomic network, and a protocol for this purpose.

7.3. Intent Distribution

The distribution of intent is also a function of the Autonomic Control Plane. Various methods can be used to distribute intent across an autonomic domain.

7.4. Reporting

An autonomic network offers through the autonomic control plane the possibility to aggregate information inside the network, before sending it to the admin of the network. While this can be seen or implemented as a specific form of negotiation, the use case is different and therefore mentioned here explicitly.

7.5. Feedback Loops

Feedback loops are required in an autonomic network to allow administrator intervention, while maintaining a default behaviour. Through a feedback loop an administrator can be prompted with a default action, and has the possibility to acknowledge or override the proposed default action.

7.6. Routing

All autonomic nodes in a domain must be able to communicate with each other, and with autonomic nodes outside their own domain. Therefore, an Autonomic Control Plane relies on a routing function.

8. Hybrid Approach with Non-Autonomic Functions

This section explains how autonomic functions can co-exist with non-autonomic functions, and how a potential overlap is managed. (tbc)

9. Security Considerations

9.1. Threat Analysis

This is a preliminary outline of a threat analysis, to be expanded and made more specific as the various Autonomic Networking specifications evolve.

Since AN will hand over responsibility for network configuration from humans or centrally established management systems to fully distributed devices, the threat environment is also fully distributed. On the one hand, that means there is no single point of failure to act as an attractive target for bad actors. On the other hand, it means that potentially a single misbehaving autonomic device could launch a widespread attack, by misusing the distributed AN mechanisms. For example, a resource exhaustion attack could be launched by a single device requesting large amounts of that resource from all its peers, on behalf of a non-existent traffic load. Alternatively it could simply send false information to its peers, for example by announcing resource exhaustion when this was not the case. If security properties are managed autonomically, a misbehaving device could attempt a distributed attack by requesting all its peers to reduce security protections in some way. In general, since autonomic devices run without supervision, almost any kind of undesirable management action could in theory by attempted by a misbehaving device.

If it is possible for an unauthorised device to act as an autonomic device, or for a malicious third party to inject messages appearing to come from an autonomic device, all these same risks would apply.

If AN messages can be observed by a third party, they might reveal valuable information about network configuration, security precautions in use, individual users, and their traffic patterns. If encrypted, AN messages might still reveal some information via traffic analysis, but this would be quite limited (for example, this would be highly unlikely to reveal any specific information about user traffic). AN messages are liable to be exposed to third parties on any unprotected Layer 2 link, and to insider attacks even on protected Layer 2 links.

10. IANA Considerations

This document requests no action by IANA.

11. Acknowledgements

tbc

12. Change log [RFC Editor: Please remove]

00: Initial version.

13. References

[I-D.behringer-autonomic-control-plane]
Behringer, M., Bjarnason, S., BL, B., and T. Eckert, "An Autonomic Control Plane", draft-behringer-autonomic-control-plane-00 (work in progress), June 2014.

[I-D.carpenter-anima-gdn-protocol]
Carpenter, B., Jiang, S., and B. Liu, "A Generic Discovery and Negotiation Protocol for Autonomic Networking", draft-carpenter-anima-gdn-protocol-00 (work in progress), October 2014.

[I-D.irtf-nmrg-autonomic-network-definitions]
Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking - Definitions and Design Goals", draft-irtf-nmrg-autonomic-network-definitions-04 (work in progress), October 2014.

[I-D.pritikin-bootstrapping-keyinfrastructures]
Pritikin, M., Behringer, M., and S. Bjarnason, "Bootstrapping Key Infrastructures", draft-pritikin-bootstrapping-keyinfrastructures-01 (work in progress), September 2014.

Authors' Addresses

Michael H. Behringer (editor)
Cisco

Email: mbehring@cisco.com

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Toerless Eckert
Cisco

Email: eckert@cisco.com