Autonomic Networking - Definitions and Design Goals
draft-irtf-nmrg-autonomic-network-definitions-00.txt

Abstract

   Autonomic systems were first described in 2001. The fundamental goal
   is self-management, including self-configuration, self-optimization,
   self-healing and self-protection.

   This document applies the concepts of autonomic systems to a network,
   and describes the definitions and design goals of Autonomic
   Networking. The goal is a network where nodes have minimal
   dependencies on human administrators or centralized management
   systems.

Table of Contents

1. Introduction to Autonomic Networking

   Autonomic systems were first described in a manifesto by IBM in 2001
   [Kephart]. The fundamental concept involves eliminating external
   systems from a system's control loops and closing of control loops
   within the autonomic system itself, with the goal of providing the

autonomic system with self-management capabilities, including self-
configuration, self-optimization, self-healing and self-protection.

IP networking was initially designed with similar properties in mind.
An IP network should be distributed and redundant to withstand
outages in any part of the network. A routing protocol such as OSPF
or ISIS exhibits properties of self-management, and can thus be
considered autonomic in the definition of this document.

However, as IP networking evolved, the ever increasing intelligence
of network element was often not put into protocols to follow this
paradigm, but into configuration. This configuration made network
elements highly dependent on some process that manages them, either a
human, or a network management system.

Autonomic Networking aims at putting the intelligence of today's
operations back into algorithms at the node level, to minimize
dependency on human administrators and central management systems.
Some information an autonomic node requires however cannot be
discovered; where input from some central intelligence is required,
it is provided in a highly abstract, network wide form.

This document provides the definitions and gesign goals for Autonomic
Networking.

2. Definitions

Autonomic: Self-managing (self-configuring, self-protecting, self-
healing and self-optimizing); however, allowing high-level guidance
by a central entity, through intent.

Intent: An abstract, high level policy used to operate the network
autonomically. Its scope is an autonomic domain, such as an
enterprise network. It does not contain configuration or information
for a specific node. It may contain information pertaining to nodes
with a specific role.

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

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

Autonomic Service Agent: An agent implemented on an autonomic node
which implements an autonomic function, either in part (in the case
of a distributed function) or whole.

Autonomic Node: A node which employs autonomic functions. It may
operate on any layer of the networking stack. Examples are routers,
switches, personal computers, call managers, etc.

Fully Autonomic Node: A node which employs exclusively autonomic
functions. It requires no configuration.

Autonomic Network: A network containing autonomic nodes.

Fully Autonomic Network: A network consisting of exclusively fully
autonomic nodes.

3. Design Goals

This section explains the high level goals of Autonomic Networking,
independent of any specific solutions.

3.1. Self-Management

The original design goals of autonomic systems as described in
[Kephart] also apply to Autonomic Networks. The over-arching goal is
self-management, which is comprised of several self-* properties.
The most commonly cited are:

o Self-configuration: The nodes do not require to be configured, but
   they configure themselves, based on self-knowledge, discovery, and
   intent. Discovery is the default way for a node to receive the
   information it needs to operate.

o Self-healing: The nodes adapt on their own to changes in the
   environment, and heal problems automatically.

o Self-optimising: The nodes automatically determine ways to
   optimise their behaviour.

o Self-protection: The nodes automatically secure themselves against
   potential attacks.

Almost any network can be described as "self-managing", as long as
the definition of "self" is large enough. For example, to a
residential user, the service provider network she connects to could
be considered "autonomic", because the user only specifies a very
high level policy such as "Internet access" and is not exposed to any
internals of the network.

For the work in the IETF and IRTF we define the "self" properties on
the node level. It is the design goal to make network nodes self-
managing, in other words, minimally dependent on management systems

or controllers, as well as human operators. Self-managing nodes
might need to exchange information with other nodes in order to
achieve the required goals.

[3.2](#). By Default Secure

All autonomic interactions should be by default secure. This
requires that any member of an autonomic domain can assert its
membership using a domain identity, for example a certificate issued
by a domain certification authority. This domain identity is used
for nodes to learn about their neighbouring nodes, to determine the
boundaries of the domain, and to cryptographically secure
interactions within the domain. Nodes from different domains can
also mutually verify their identity and secure interactions as long
as they have a common trust anchor.

A strong, cryptographically verifiable domain identity is a
fundamental cornerstone in autonomic networking. It can be leveraged
to secure all communications, and allows thus automatic security
without traditional configuration, for example pre-shared keys.

Autonomic nodes must be able to adapt their behaviour depending on
the domain of the node they are interacting with.

[3.3](#). Decentralisation and Distribution

The goal of Autonomic Networking is to minimise dependencies on
central elements; therefore, de-centralisation and distribution are
fundamental to the concept. If a problem can be solved in a
distributed manner, it should not be centralised.

In certain cases it is today operationally preferable to keep a
central repository of information, for example a user database on a
AAA server. An autonomic network must also be able to use such
central systems, in order to be deployable. However, it is possible
to distribute such databases as well, and such efforts should be at
least considered.

[3.4](#). Simplification of the Northbound Interfaces

Even in a decentralised solution, certain information flows with
central entities are required. Examples are the definition of intent
or high level service definitions, as well as network status requests
and aggregated reporting.

Therefore, also elements in an autonomic network require a northbound
interface. However, the design goal is to maintain this interface as
simple and high level as possible.

3.5. Abstraction

An administrator or autonomic management system interacts with an
autonomic network on a high level of abstraction. Intent is defined
at a level of abstraction that is much higher than that of typical
configuration parameters, for example, "optimize my network for
energy efficiency". Intent must not be used to convey low-level
commands or concepts, since those are on a different abstraction
level. The administrator should not even be exposed to the version
of the IP protocol running in the network.

Also on the reporting and feedback side an autonomic network
abstracts information and provides high-level messages such as "the
link between node X and Y is down".

3.6. Autonomic Reporting

An autonomic network, while minimizing the need for user
intervention, still needs to provide users with visibility like in
traditional networks. However, in an autonomic network reporting
should happen on a network wide basis. Information about the network
should be collected and aggregated by the network itself, presented
in consolidated fashion to the administrator.

The layers of abstraction that are provided via intent need to be
supported for reporting functions as well, in order to give users an
indication about the effectiveness of their intent. For example, in
order to assess how effective the network performs with regards to
the intent "optimize my network for energy efficiency", the network
should provide aggregate information about the number of ports that
were able to be shut down while validating current service levels are
on aggregate still met.

Autonomic network events should concern the autonomic network as a
whole, not individual systems in isolation. For example, the same
failure symptom should not be reported from every system that
observes it, but only once for the autonomic network as a whole.
Ultimately, the autonomic network should support exception based
management, in which only events that truly require user attention
are actually notified. This requires capabilities that allow systems
within the network to compare information and apply special
algorithms to determine what should be reported.

3.7. Modularity

It is unrealistic to expect a fully autonomic network in complex
environments for many years to come. While simple networks may

become autonomic in one single step, a phased approach is required
for most of today's networks.

Autonomic functions can be implemented in a modular way. For
example, the internal routing algorithm in many networks today is
already mostly autonomic. Other modules can be made autonomic step
by step.

3.8. Independence of Function and Layer

Today's autonomic functions may reside on any layer in the networking
stack. For example, layer 2 switching today is already relatively
autonomic in many environments; routing functions can be autonomic.
"Autonomic" in the context of this framework is a property of a node.
This node can be a switch, router, server, or call manager.
Autonomic functionality is independent of the function of a node.
Even application layer functionality such as unified communications
can be autonomic.

An Autonomic Network requires an overall control plane for autonomic
nodes to communicate. As in general IP networking, IP is the layer
that binds all those elements together; autonomic functions in the
context of this framework should therefore operate at the IP layer.
This concerns neighbour discovery protocols and other autonomic
control plane functions.

3.9. Full Life Cycle Support

An autonomic node does not depend on external input to operate; it
needs to understand its current situation and surrounding, and
operate according to its current state. Therefore, an autonomic node
must understand its full life cycle, from first manufacturing testing
through deployment, testing, troubleshooting, up to decommissioning.

The state of the life-cycle of an autonomic node is reflected in a
state model. The behaviour of an autonomic node may be different for
different deployment states.

4. Non Design Goals

This section identifies various items which are explicitly not design
goals for autonomic networks, which are mentioned to avoid
misunderstandings of the general intention.

4.1. Eliminate human operators

The problem targeted by autonomic networking is the error-prone and
hard to scale model of individual configuration of network elements,
traditionally by manual commands but today mainly by scripting and/or
configuration management databases. This does not, however, imply
the elimination of skilled human operators, who will still be needed
for oversight, policy management, diagnosis, reaction to help desk
tickets, etc. etc. The main impact on operators should be less
tedious detailed work and more high-level work. (They should become
more like doctors and nurses than hospital orderlies.)

4.2. Eliminate emergency fixes

However good the autonomous mechanisms, sometimes there will be fault
conditions etc. that they cannot deal with correctly. At this point
skilled operator interventions will be needed to correct or work
around the problem. Hopefully this can be done by high-level
mechanisms (adapting the policy database in some way) but in some
cases direct intervention at device level may be unavoidable. This
is obviously the case for hardware failures, even if the autonomic
network has bypassed the fault for the time being. Truck rolls will
not be eliminated when faulty equipment needs to be replaced.
However, this may be less urgent if the autonomic system
automatically reconfigures to minimise the operational impact.

4.3. Eliminate management control and central policy

Senior management might fear loss of control of an autonomic network.
In fact this is no more likely than with a traditional network; the
emphasis on automatically applying general policy and security rules
might even provide more management control.

4.4. Eliminate existing configuration tools

While autonomic networks will rarely need manual intervention, there
is no expectation that traditional top-down configuration tools will
vanish immediately. Autonomic techniques will have to co-exist with
them, and they will survive for as long as they are useful.
Initially they will certainly play a part in confidence-building in
the autonomic method, and they will be held in reserve for emergency
use for a long time.

4.5. Eliminate existing network management systems

Existing monitoring and reporting systems will continue to be needed,
and as just noted existing configuration mechanisms will not vanish.
Therefore, it is to be expected that the existing NMS will be

retained in parallel with autonomic mechanisms, and will be adapted
as necessary. Some aspects of the autonomic mechanism (e.g.
aggregated reporting, exception reporting) should indeed be
integrated with the existing NMS as far as possible.

5. Guidelines for Case Studies

[This section is work in progress.]

6. An Autonomic Reference Model

An Autonomic Network consists of Autonomic Nodes. Those nodes
communicate with each other through an Autonomic Control Plane which
provides a robust and secure communications overlay. The Autonomic
Control Plane is self-organizing and autonomic itself.

An Autonomic Node contains various elements, such as autonomic
service agents. Figure 1 shows a reference model of an autonomic
node. The elements and their interaction are:

o Autonomic Service Agents, which implement the autonomic behaviour
   of a specific service or function.

o Self-knowledge: An autonomic node knows its own properties and
   capabilities

o Network Knowledge (Discovery): An autonomic service agent may
   require various discovery functions in the network, such as
   service discovery.

o Intent: Network wide high level policy. Autonomic Service Agents
   use an intent interpretation engine to locally instantiate the
   global intent. This may involve coordination with other Autonomic
   Nodes.

o Feedback Loops: Control elements outside the node may interact
   with autonomic nodes through feedback loops.

o An Autonomic User Agent, providing a front-end to external users
   (administrators and management applications) through which they
   can communicate intent, receive reports, and monitor the Autonomic
   Network.

o Autonomic Control Plane: Allows the node to communicate with other
   autonomic nodes. Autonomic functions such as intent distribution,
   feedback loops, discovery mechanisms, etc, use the autonomic
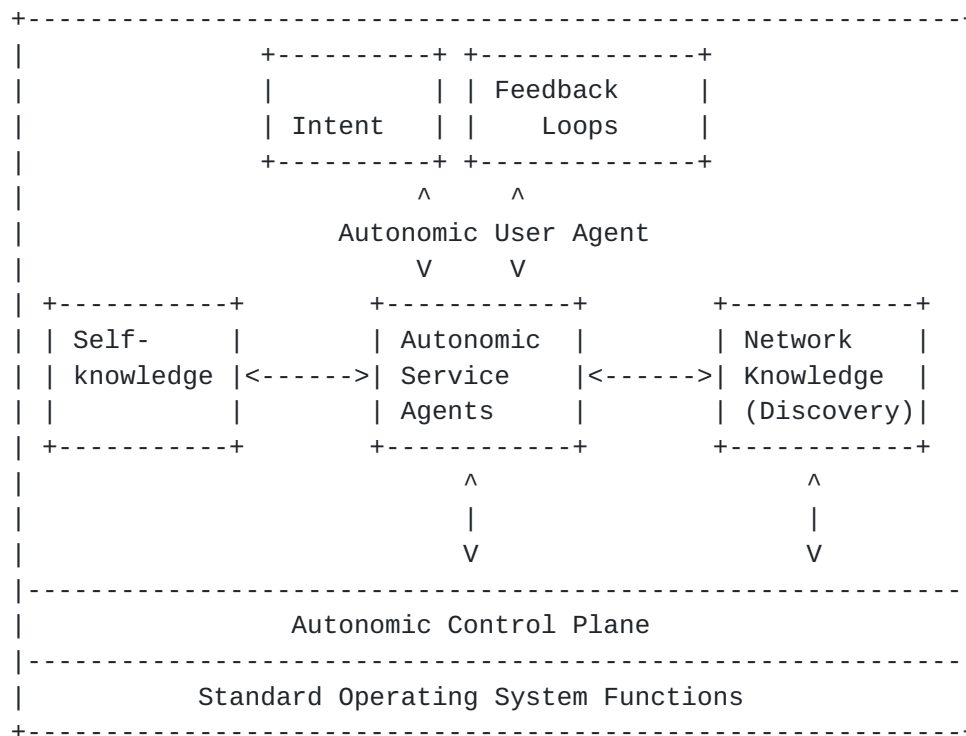   control plane.

```
+-------------------------------------------------------------+
|             +----------+ +--------------+                   |
|             |          | | Feedback     |                   |
|             | Intent   | |    Loops     |                   |
|             +----------+ +--------------+                   |
|                  ^         ^                                |
|                 Autonomic User Agent                       |
|                  V         V                               |
| +-----------+       +------------+      +------------+  |
| | Self-     |       | Autonomic  |      | Network    |  |
| | knowledge |<----->| Service    |<----->| Knowledge  |  |
| |           |       | Agents     |      | (Discovery)|  |
| +-----------+       +------------+      +------------+  |
|                          ^                    ^           |
|                          |                    |           |
|                          V                    V           |
|-------------------------------------------------------------|
|             Autonomic Control Plane                        |
|-------------------------------------------------------------|
|           Standard Operating System Functions              |
+-------------------------------------------------------------+
```

                            Figure 1

7. Security Considerations

   This document specifies a framework. Security is an integral part of
   this framework.

8. Acknowledgements

   The work on Autonomic Networking is the result of a large team
   project at Cisco Systems. In alphabetical order: Ignas Bagdonas,
   Parag Bhide, Balaji BL, Toerless Eckert, Yves Hertoghs, Bruno
   Klauser.

   The ETSI working group AFI (http://portal.etsi.org/afi) defines a
   similar framework for autonomic networking in the "General Autonomic
   Network Architecture" [GANA]. Many concepts explained in this
   document can be mapped to the GANA framework. The mapping is outside
   the scope of this document. Special thanks to Ranganai Chaparadza
   for his comments and help on this document.

9. Informative References

   [GANA]   ETSI GS AFI 002, , "Autonomic network engineering for the
               self-managing Future Internet (AFI): GANA Architectural
               Reference Model for Autonomic Networking, Cognitive
               Networking and Self-Management.", April 2013,
               <http://www.etsi.org/deliver/etsi_gs/AFI/001_099/002/
               01.01.01_60/gs_afi002v010101p.pdf>.

   [Kephart]   Kephart, J. and D. Chess, "The Vision of Autonomic
               Computing", IEEE Computer vol. 36, no. 1, pp. 41-50,
               January 2003.

Authors' Addresses

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

   Email: mbehring@cisco.com

   Max Pritikin
   Cisco Systems

   Email: pritikin@cisco.com

   Steinthor Bjarnason
   Cisco Systems

   Email: sbjarnas@cisco.com

   Alex Clemm
   Cisco Systems

   Email: alex@cisco.com

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

   Email: brian.e.carpenter@gmail.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

   Laurent Ciavaglia
   Alcatel-Lucent

   Email: Laurent.Ciavaglia@alcatel-lucent.com