

Autonomic Networking

IETF ANIMA WG

An overview

Brian Carpenter

April 2017

English lesson

- Automatic
 - done as if by machine; self-acting or self-regulating mechanism
- Autonomous
 - without outside control; responding, reacting, or developing independently of the whole
- Autonomic
 - occurring involuntarily or spontaneously; occurring as a result of internal stimuli



Terminology (1)

- Autonomic Nervous System: a control system that acts largely unconsciously and regulates bodily functions such as heart rate.
- Autonomic Computing: self-managing distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity from operators and users (IBM, 2001).
- Autonomic Network: Self-managing (self-configuring, self-protecting, self-healing, self-optimizing) but allowing high-level guidance by a central entity ("Intent")
 - "Plug and play for the ISP" or "plug and play for the enterprise"

Terminology (2)

- Autonomic Function: A specific self-managing feature or function.
- Autonomic Service Agent (ASA): An agent that implements an autonomic function, in part (for a distributed function) or whole.
- Autonomic Node: A node that employs autonomic functions
- Autonomic Control Plane (ACP): Self-configuring fully secure virtual network used for all autonomic messaging.

For more details see RFC7575

Autonomic Networking Integrated Model and Approach (ANIMA) WG

- Initial work items
 - Bootstrapping & trust infrastructure
 - Secure Autonomic Control Plane (ACP)
 - Discovery for autonomic nodes
 - Negotiation & synchronisation for autonomic nodes
- Left for later
 - Intent (high level policy)
 - Defining the domain boundary
 - ASA life cycle, authorisation and coordination
 - Reporting
- Left for much later
 - Tie in to machine learning and other AI techniques

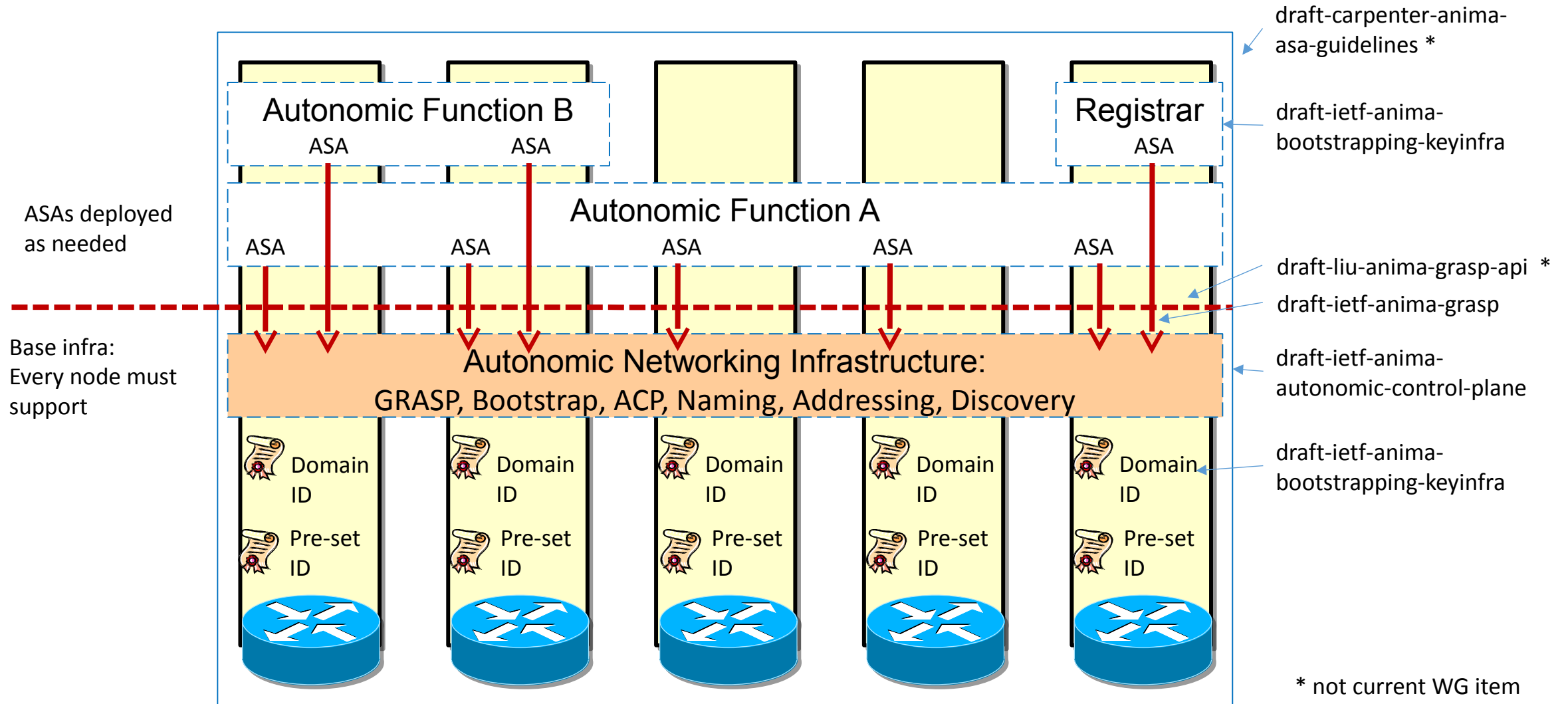
Bootstrap and ACP

- Secure bootstrap – all nodes must start (out of the box or after a factory reset) by using a registrar to authenticate themselves and obtain a domain certificate. This is coordinated with related work for IoT devices. No human intervention except to create the registrar.
- ACP – the ACP will bootstrap itself using only link-local IPv6 addresses and IPv6 Unique Local Address prefix. All links secured (IPsec). No human intervention except to define the domain boundary.

GeneRic Autonomic Signaling Protocol (GRASP)

- GRASP will be used for signaling between ASAs
 - That includes the special-purpose ASAs that support both secure bootstrap & ACP creation
 - After that, GRASP runs over the ACP to guarantee security
- GRASP provides discovery, flooding, synchronization and negotiation for the technical objectives supported by ASAs
 - Based on CBOR (Concise Binary Object Representation)
 - Objectives can be expressed in JSON or Python-like syntax & semantics

Reference Model – High Level View



Network with autonomic functions

More about a GRASP Objective

- A configurable parameter:
 - a logical, numerical or string value, or a more complex data structure.
 - used in Discovery, Negotiation, Flooding and Synchronization.
 - semantics depend on the autonomic function concerned, and are built into the code of each ASA.
- Example for IPv6 prefix management:

```
["PrefixManager", flags, loop_count,  
                                [PD_support, prefix_length, prefix]]
```

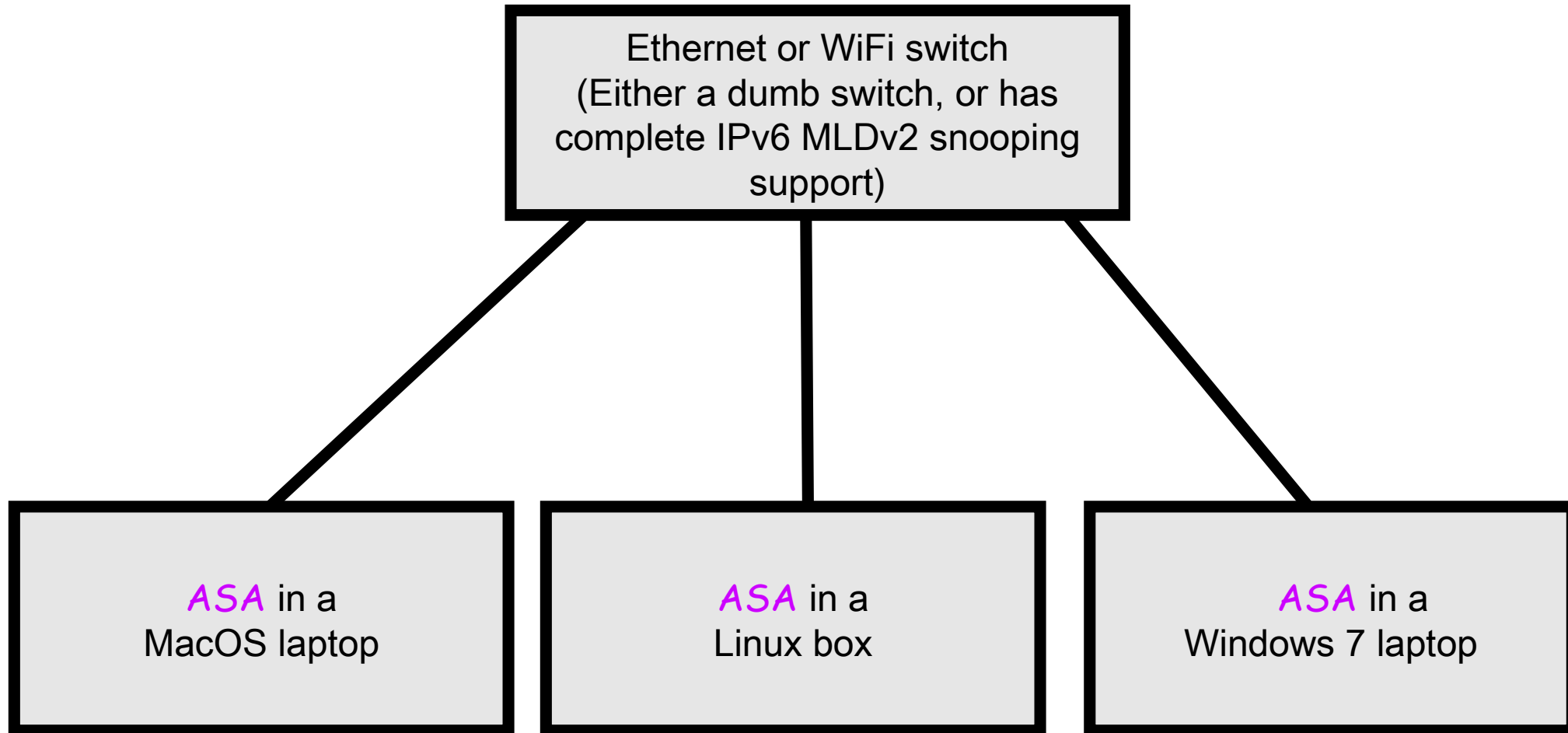
GRASP Messages

- Discovery (multicast)
Discovery Response
- Request Synchronization
Synchronization
- Flood Synchronization (multicast)
- Request Negotiation
Negotiation
Confirm Waiting
Negotiation End

GRASP Prototype

- A Python 3 implementation of GRASP as a module **grasp.py**
- About 1800 lines of code
- A test suite to exercise as many code paths as possible
- Various toy ASAs to test "real" operation across the network
 - bank/client negotiation
 - model of secure bootstrap process
 - model of IPv6 prefix management
- Some documentation

Tests



More...

- RFC 7575
- RFC 7576
- <https://datatracker.ietf.org/wg/anima/documents/>
- <https://www.cs.auckland.ac.nz/~brian/graspy/>