

Internet Research Task Force
Internet-Draft
Intended status: Informational
Expires: December 29, 2014

M. Behringer
M. Pritikin
S. Bjarnason
A. Clemm
Cisco Systems
B. Carpenter
Univ. of Auckland
S. Jiang
Huawei Technologies Co., Ltd
L. Ciavaglia
Alcatel-Lucent
June 27, 2014

Autonomic Networking - Definitions and Design Goals
draft-irtf-nmrg-autonomic-network-definitions-01.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 high-level goal for an autonomic function is to have minimal dependencies on human administrators or centralized management systems. This usually implies distribution across network elements.

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 December 29, 2014.

Copyright Notice

Copyright (c) 2014 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 to Autonomic Networking	2
2. Definitions	3
3. Design Goals	4
3.1. Self-Management	4
3.2. By Default Secure	5
3.3. Decentralisation and Distribution	5
3.4. Simplification of the Northbound Interfaces	5
3.5. Abstraction	6
3.6. Autonomic Reporting	6
3.7. Modularity	6
3.8. Common Autonomic Networking Infrastructure	7
3.9. Independence of Function and Layer	7
3.10. Full Life Cycle Support	8
4. Non Design Goals	8
4.1. Eliminate human operators	8
4.2. Eliminate emergency fixes	8
4.3. Eliminate management control and central policy	9
4.4. Eliminate existing configuration tools	9
4.5. Eliminate existing network management systems	9
5. An Autonomic Reference Model	9
6. Security Considerations	11
7. Acknowledgements	11
8. Informative References	12
Authors' Addresses	12

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 elements 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 function 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 design 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 feature or 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: Functions do not require to be configured, but they configure themselves, based on self-knowledge, discovery, and intent. Discovery is the default way for an autonomic function to receive the information it needs to operate.
- o Self-healing: Autonomic functions adapt on their own to changes in the environment, and heal problems automatically.
- o Self-optimising: Autonomic functions automatically determine ways to optimise their behaviour.
- o Self-protection: Autonomic functions 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, a well-defined SDN system, including the controller elements, can be described over all as "autonomic", if the controller provides an interface to the administrator which has the same properties as mentioned above (high level, network-wide, etc).

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

management systems or controllers, as well as human operators. Self-managing functions on a node 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 functions 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. Common Autonomic Networking Infrastructure

[I-D.irtf-nmrg-an-gap-analysis] points out that there are already a number of fully or partially autonomic functions available today. However, they are largely independent, and each has its own methods and protocols to communicate, discover, define and distribute policy, etc.

The goal of the work on autonomic networking in the IETF is therefore not just to create autonomic functions, but to define a common infrastructure that autonomic functions can use. This autonomic networking infrastructure may contain common control and management functions such as messaging, service discovery, negotiation, intent distribution, self-monitoring and diagnostics, etc. A common approach to define and manage intent is also required.

Refer to the reference model below: All the components around the "autonomic service agents" should be common components, such that the autonomic service agents do not have to replicate common tasks individually.

3.9. 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 function on 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.

3.10. Full Life Cycle Support

An autonomic function 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 function must understand the full life cycle of the device it runs on, 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 function 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 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. 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 which implement autonomic functions. 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.

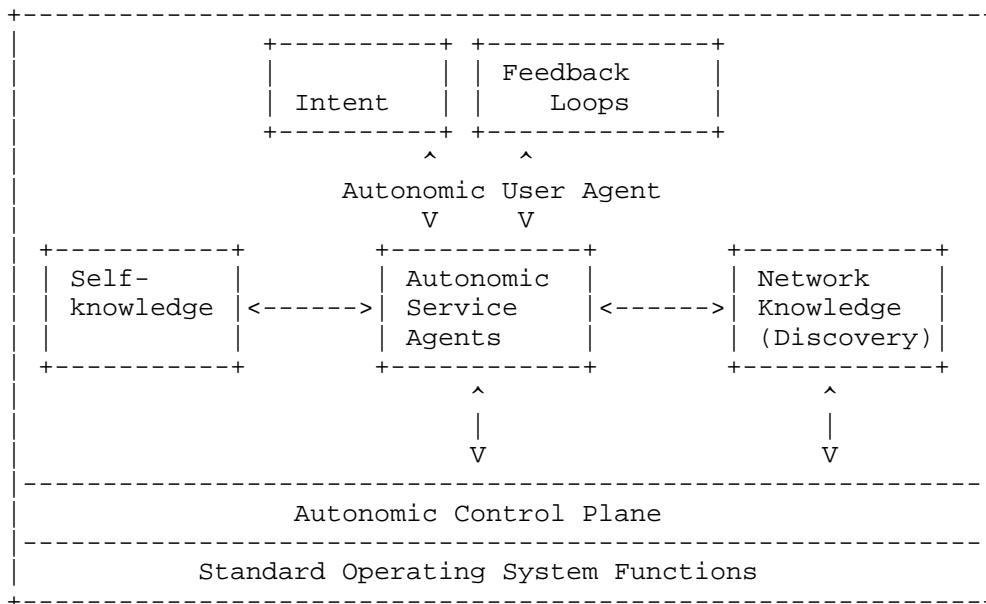


Figure 1

6. Security Considerations

This document provides definitions and design goals for autonomic networking. A full threat analysis will be required as part of the development of solutions, taking account of potential attacks from within the network as well as from outside.

7. 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.

8. 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>.
- [I-D.irtf-nmrg-an-gap-analysis] Behringer, M., Carpenter, B., and S. Jiang, "Gap Analysis for Autonomic Networking", draft-irtf-nmrg-an-gap-analysis-00 (work in progress), April 2014.
- [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

Network Management Research Group
Internet-Draft

Intended status: Informational Federal University of Rio Grande do Sul
Expires: December 22, 2014

J. Nobre
L. Granville
A. Clemm
A. Prieto
Cisco Systems
June 20, 2014

Autonomic Networking Use Case for Distributed Detection of SLA
Violations
draft-irtf-nmrg-autonomic-sla-violation-detection-00

Abstract

This document describes a use case for autonomic networking in distributed detection of SLA violations. It is one of a series of use cases intended to illustrate requirements for autonomic networking.

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 December 22, 2014.

Copyright Notice

Copyright (c) 2014 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. Problem Statement	3
3. Benefits of an Autonomic Solution	4
4. Intended User and Administrator Experience	5
5. Analysis of Parameters and Information Involved	5
5.1. Device Based Self-Knowledge and Decisions	5
5.2. Interaction with other devices	5
5.3. Information needed from Intent	6
5.4. Monitoring, diagnostics and reporting	6
6. Comparison with current solutions	6
7. Related IETF Work	6
8. Acknowledgements	7
9. IANA Considerations	7
10. Security Considerations	7
11. References	7
11.1. Normative References	7
11.2. Informative References	8
Authors' Addresses	8

1. Introduction

The Internet has been improving dramatically in terms of size and capacity, and accessibility in the last years. Besides that, the communication requirements of distributed services and applications running on top of the Internet have become increasingly accurate. Performance issues caused by violations on these requirements usually present significant financial loss to organizations and end users. Thus, the service level requirements of critical networked services provided have become a critical concern for network administrators. To ensure that SLAs are not being violated, which would usually incur in costly penalties, service levels need to be constantly monitored at the network infrastructure layer. To that end, network measurements must take place. Network measurement mechanisms are performed through either active or passive measurement techniques. In passive measurement, network conditions are said to be checked in a non intrusive way because no monitoring traffic is created by the measurement process itself. In the context of IP Flow Information EXport (IPFIX) WG, several documents were produced to define passive measurement mechanisms (e.g., flow records specification [RFC3954]). Active measurement, on the other hand, is intrusive because it injects synthetic traffic into the network to measure the network performance. The IP Performance Metrics (IPPM) WG produced documents

that describe active measurement mechanisms, such as: One-Way Active Measurement Protocol (OWAMP) [RFC4656], Two-Way Active Measurement Protocol (TWAMP) [RFC5357], and Cisco Service Level Assurance Protocol (SLA) [RFC6812]. Active measurement mechanisms usually offer better accuracy and privacy than passive measurement mechanisms. Furthermore, active measurement mechanisms are able to detect end-to-end network performance problems in a fine-grained way. As a result, active is preferred over passive measurement for SLA monitoring. Measurement probes must be hosted and activated in network devices to compute the current network metrics (e.g., considering those described in [RFC4148]). This activation should be dynamic in order to follow changes in network conditions, such as those related with routes being added or new customer demands.

2. Problem Statement

The activation of active measurement probes (sender and responder considering the architecture described by Cisco [RFC6812]) is expensive in terms of the resource consumption, e.g., CPU cycle and memory footprint, which could be useful for primary network functions (e.g., routing and switching). Besides that, the probes also increase the network load because of the injected traffic. The resources required and traffic generated by the measurement probes are a function of the number of measured network destinations, i.e., with more destinations the larger will be the resources and the traffic needed to deploy the probes. Thus, to have a better monitoring coverage it is necessary to deploy more probes what consequently turns increases consumed resources. Otherwise, enabling the observation of just a small subset of all network flows can lead to an insufficient coverage. The current best practice in feasible deployments of active measurement solutions to distribute the available measurement probes along the network consists in relying entirely on the human administrator expertise to infer which would be the best location to activate the probes. This is done through several steps. First, it is necessary to collect traffic information in order to grasp the traffic matrix. Then, the administrator uses this information to infer which are the best destinations for measurement probes. After that, the administrator activates probes on the chosen subset of destinations considering the available resources. This practice, however, does not scale well because it is still labor intensive and error-prone for the administrator to compute which probes should be activated given the set of critical flows that needs to be measured. Even worse, this practice completely fails in networks whose critical flows are too short in time and dynamic in terms of traversing network path, like in modern cloud environments. That is so because fast reactions are necessary to reconfigure the probes and administrators are not just enough in computing and activating the new set of probes required every time

the network traffic pattern changes. Finally, the current active measurements practice usually covers only a fraction of the network flows that should be observed, which invariably leads to the damaging consequence of undetected SLA violations. Management software can be embedded inside network devices to control the deployment of active measurement mechanisms. In fact, this is done by some network equipment vendors, specially to avoid the starvation of the network devices (e.g., due to configuration errors and lack of experience from human administrators). However, the current approach do not enhance the active measurement capabilities in important terms, such as scalability and efficiency. For example, the number of local available measurements (and, consequently, detected SLA violations) is still bounded by the number of deployed probes. Thus, if the number of SLA violation is greater than the number of available probes, only a fraction of the violations will be observed. Also, devices cannot share resources and knowledge about the networking infrastructures in order to take advantage of remote management information (e.g., measurement results).

3. Benefits of an Autonomic Solution

The use case considered here is distributed autonomic detection of SLA violations. The use of Autonomic Networking (AN) properties can help the activation of measurement probes [P2PBNM-Nobre-2012]. Peer-to-Peer (P2P) technology can be embedded in network devices in order to improve the probe activation decisions using autonomic loops. Thus, it would be possible to coordinate the probe activation and to share measurement results among different network devices. The problem to be solved by AN in the present use case is how to steer the process of measurement probe activation by a complete solution that sets all necessary parameters for this activation to operate efficiently, reliably and securely, with minimal human intervention and without the need for. An autonomic solution for the distributed detection of SLA violations can provide several benefits. First, this solution could optimize the resource consumption and avoid resource starvation on the network devices. This optimization comes from different sources: sharing of measurement results, better efficiency in the probe activation decisions, etc. Second, the number of detected SLA violations could be increased. This increase is related with a better coverage of the network. Third, the solution could decrease the time necessary to detect SLA violations. Adaptivity features of an autonomic loop could capture faster the network dynamics than an human administrator. Finally, the solution could help to reduce the workload of human administrator, or, at least, to avoid their need to perform operational tasks. The active measurement model assumes that a typical infrastructure will have multiple network segments and Autonomous Systems (ASs), and a reasonably large number of several of routers and hosts. It also

considers that multiple Service Level Objectives (SLOs) can be in place in a given time. Since interoperability in a heterogenous network is a goal, features found on different active measurement mechanisms (e.g. OWAMP, TWAMP, and IPSLA) and programability interfaces (e.g., Cisco's EEM and onePK) could be used for the implementation. The autonomic solution should include and/or reference specific algorithms, protocols, metrics and technologies for the implementation of distributed detection of SLA violations as a whole.

4. Intended User and Administrator Experience

The autonomic solution should avoid the human intervention in the distributed detection of SLA violations. Besides that, it could enable the control of SLA monitoring by less experienced human administrators. However, some information is necessary from the human administrator. For example, the human administrator should provide the SLOs regarding the SLA being monitored. The configuration and bootstrapping of network devices using the autonomic solution should be minimal for the human administrator. Probably it would be necessary just to inform the address of a device which is already using the solution and the devices themselves could exchange configuration data.

5. Analysis of Parameters and Information Involved

5.1. Device Based Self-Knowledge and Decisions

Each device has self-knowledge about the local SLA monitoring. This could be in the form of historical measurement data and SLOs. Besides that, the devices would have algorithms that could decide which probes should be activated in a given time. The choice of which algorithm is better for a specific situation would be also autonomic.

5.2. Interaction with other devices

Network devices could share information about service level measurement results. This information could speed up the detection of SLA violations and increase the number of detected SLA violations. In any case, it is necessary to assure that the results from remote devices have local relevancy. The definition of network devices that exchange measurement data, i.e., management peers, creates a new topology. Different approaches could be used to define this topology (e.g., correlated peers [P2PBNM-Nobre-2012]). To bootstrap peer selection, each device could use its known endpoints neighbors (e.g., FIB and RIB tables) as the initial seed to get possible peers.

5.3. Information needed from Intent

TBD

5.4. Monitoring, diagnostics and reporting

TBD

6. Comparison with current solutions

There is no standartized solution for distributed autonomic detection of SLA violations. Current solutions are restricted to ad hoc scripts running on a per node fashion to automate some administrator's actions. There some proposals for passive probe activation (e.g., DECON and CSAMP), but without the focus on autonomic features. It is also mentioning a proposal from Barford et al. to detect and localize links which cause anomalies along a network path.

7. Related IETF Work

The following paragraphs discuss related IETF work and are provided for reference. This section is not exhaustive, rather it provides an overview of the various initiatives and how they relate to autonomic distributed detection of SLA violations. 1. [LMAP]: The Large-Scale Measurement of Broadband Performance Working Group aims at the standards for performance management. Since their mechanisms also consist in deploying measurement probes the autonomic solution could be relevant for LMAP specially considering SLA violation screening. Besides that, a solution to decrease the workload of human administrators in service providers is probably highly desirable. 2. [IPFIX]: IP Flow Information EXport (IPFIX) aims at the process of standardization of IP flows (i.e., netflows). IPFIX uses measurement probes (i.e., metering exporters) to gather flow data. In this context, the autonomic solution for the activation of active measurement probes could be possibly extended to address also passive measurement probes. Besides that, flow information could be used in the decision making of probe activation. 3. [ALTO]: The Application Layer Traffic Optimization Working Group aims to provide topological information at a higher abstraction layer, which can be based upon network policy, and with application-relevant service functions located in it. Their work could be leveraged for the definition of the topology regarding the network devices which exchange measurement data.

8. Acknowledgements

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from Bruno Klauser, Eric Voig, and Hanlin Fang.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

The bootstrapping of a new device follows the approach of homenet [draft-autonomic-homenet], thus in order to exchange data a device should register first. This registration could be performed by a "Registrar" device or a cloud service provided by the organization to facilitate autonomic mechanisms. The new device sends its own credentials to the Registrar, and after successful authentication, receives domain information, to enable subsequent enrolment to the domain. The Registrar sends all required information: a device name, domain name, plus some parameters for the operation. Measurement data should be exchanged signed and encrypted among devices since these data could carry sensible information about network infrastructures. Some attacks should be considering when analyzing the security of the autonomic solution Denial of service (DoS) attacks could be performed if the solution be tempered to active more local probe than the available resources allow. Besides that, results could be forged by a device (attacker) in order to this device be considered peer of a specific device (target). This could be done to gain information about a network.

11. References

11.1. Normative References

- [P2PBNM-Nobre-2012]
Nobre, J., Granville, L., Clemm, A., and A. Prieto, "Decentralized Detection of SLA Violations Using P2P Technology, 8th International Conference Network and Service Management (CNSM)", 2012, <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6379997>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC6812] Chiba, M., Clemm, A., Medley, S., Salowey, J., Thombare, S., and E. Yedavalli, "Cisco Service-Level Assurance Protocol", RFC 6812, January 2013.
- [draft-autonomic-homenet]
Behringer, M., Pritikin, M., and S. Bjarnason, "draft-behringer-homenet-trust-bootstrap", draft-behringer-homenet-trust-bootstrap-02 (work in progress), February 2014.

11.2. Informative References

- [RFC3954] Claise, B., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, October 2004.
- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.

Authors' Addresses

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

Email: jcnobre@inf.ufrgs.br

Lisandro Zambenedetti Granville
Federal University of Rio Grande do Sul
Porto Alegre
Brazil

Email: granville@inf.ufrgs.br

Alexander Clemm
Cisco Systems
San Jose
USA

Email: alex@cisco.com

Alberto Gonzalez Prieto
Cisco Systems
San Jose
USA

Email: albertgo@cisco.com

Network Management Research Group
Internet-Draft
Intended Status: Informational
Expires: January 4, 2015

O. Festor
Inria
A. Lahmadi
University of Lorraine - LORIA
R. Hofstede
A. Pras
University of Twente
July 3, 2014

Information Elements for IPFIX Metering Process Location
draft-irtf-nmrg-location-ipfix-01

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire on January 4, 2015.

Copyright Notice

Copyright (c) 2014 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.

Abstract

This document defines a set of Information Elements for the IP Flow Information Export (IPFIX) protocol for exporting location information of any device (both fixed and mobile) that acts as an IPFIX Flow Exporter. The specified Information Elements support both geospatial and civic location data.

Table of Contents

1. Introduction	3
1.1. Motivation	3
1.2. Terminology	4
2. Relationships with GEOPRIV	4
3. Location Information	4
3.1. Geospatial Location Information	4
3.2. Civic Location Information	5
4. Location Information Elements	5
4.1. geospatialLocationCRSCode	5
4.2. geospatialLocationLat	6
4.3. geospatialLocationLng	6
4.4. geospatialLocationAlt	6
4.5. geospatialLocationRadius	6
4.6. civicLocationType	7
4.7. civicLocationValue	7
4.8. locationMethod	7
4.9. locationTime	7
4.10. deviceId	7
5. Guidelines for Using Location Information Elements	8
6. Recommended Templates for Geospatial and Civic Location Export	8
6.1. Geospatial Point Location Template	9
6.2. Geospatial Circle Location Template	10
6.3. Geospatial List Template	11
6.4. Civic Location Template	13
6.5. Compound Location Template	14
7. Security Considerations	15
8. IANA Considerations	16
8.1. locationMethod Sub-Registry	16
9. Acknowledgements	16
10. References	16
10.1. Normative References	16
10.2. Informative References	17
Appendix A. Example Implementation	18
Authors' Addresses	19

1. Introduction

The importance of geographic location information in the Internet is growing rapidly. It can be used for business advertisements, admission control and security analysis, for example. Most mobile devices, such as smart phones, tablets and sensors, have capabilities for determining and exposing their geographic location. Besides that, they are accountable for an increasing share of the overall network traffic. In contrast to fixed devices, which usually have their physical location configured in a static manner, mobile devices can exploit several location systems for obtaining their location. This type of information is already used by a wide range of applications and services, such as navigation systems and friend finder services. Relating the location information of a device to this network traffic can be beneficial to many network management and measurement applications, including traffic profiling, anomaly detection and provider-independent network measurements. Hence, exporting location information associated to traffic Flows is desirable in various situations.

The IPFIX protocol [RFC7011] has been designed for the purpose of exporting IP traffic Flows based on Information Elements. This document defines a set of IPFIX Information Elements that provide a means for Metering Processes to encapsulate location information within exported Flows. This will be done by relying on existing location information formats, as they have been developed in other standardization areas for encoding civic locations, geographic coordinates, etc. In summary, this document defines the IPFIX Information Elements that are suitable for encapsulating pre-existing location information data.

1.1. Motivation

A typical IPFIX Metering Process is used for aggregating IP traffic and related measurement data into Flow Records at a fixed Observation Point. After expiration, Flow Records are sent to a Flow Collector for storage and analysis. The collected information is typically represented in a purely time-based manner, which means that Flow Records provide an aggregated view on network traffic over time. However, when Metering Processes are running on devices with a (frequently) changing physical location, data analysis applications may need to be aware of these movements since they are likely to affect the behavior of the network in terms of routing, throughput, etc. An example scenario is a virtualized environment, where virtual machines change location during migration from one server to another, or even between data centers. Thus, a location-aware metering process will be able to associate their Flows to their current locations.

In fact, we are not dealing anymore with Flows associated to a fixed Observation Point, but with a multitude of sub-Flows for which the Observation Point locations have to be reported. To facilitate this, location information needs to be obtained and processed by the Metering Process in an IPFIX Flow Exporter. In the end, it will be beneficial when network management applications are able to relate service quality parameters to location changes, instead of assuming a single location for all observed parameters.

1.2. 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 [RFC2119].

2. Relationships with GEOPRIV

Associating geographic location information with network traffic on the Internet has been addressed by the GEOPRIV working group. There, a Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) option containing civic address information has been specified in [RFC4776]. A similar option for geospatial information has been defined in [RFC6225]. The group has also defined a set of requirements to be respected when collecting and using Location Objects related to a specific user [RFC3693]. These requirements include usage policies and privacy preferences associated to the Location Object as expressed by a user. All the security and privacy requirements defined in [RFC3693] concern location data collection, and usage MAY be applied to the IPFIX protocol when conveying location information. The GEOPRIV working group has extended the XML-based Presence Information Data Format in [RFC5491], to allow the encapsulation of location information within a presence document.

3. Location Information

The location of a device can generally be defined in two ways, namely by geospatial location coordinates and civic location information. Geospatial location coordinates are made up of latitude, longitude and altitude coordinates, while civic location information encompasses abstract notions of a location, such as "in the kitchen", "in Bakerstreet" or "in a train approaching Nancy, France". The usage of these two types of location representations are addressed by the GEOPRIV group in [RFC5491] and [RFC5139], respectively. This document assumes that devices use one or more existing mechanisms for the purpose of retrieving location information and therefore does not define any new mechanisms for location retrieval.

3.1. Geospatial Location Information

To obtain geospatial location information, one needs to rely on a numeric coordinate system. Such systems provide location information either in two dimensions (latitude and longitude) or three dimensions (latitude, longitude and altitude). Relying on a single point of location is normally not considered sufficient, since an area or volume of uncertainty SHALL be specified. In theory, this area or volume represents a coverage in which the device has a high probability of being found, and the point is the centroid for the area or volume. In [GeoShape] a set of geometric areas and volumes has been specified to define a location with uncertainty. A standard set of Coordinate Reference Systems (CRS) and units of measure are also specified in [GeoShape]. Implementations MUST specify distances and heights in meters as defined in EPSG 9001. Angular measures MUST be specified using degrees as identified by the EPSG 9102 code. The values of EPSG codes can be resolved by using the CRS Registry Service operated by the Oil and Gas Producers Association [OGP].

3.2. Civic Location Information

In contrast to geospatial location information, which relies on numeric data formats, the civic location format conveys pure textual information. It is applicable to device locations in buildings, for example. It MAY be a civic address closely related to a postal address, commonly used by local postal services for delivering mail. It MAY also be some approximated information, such as "living room", "Office 123 in Building 2". The civic location information format has been addressed in [RFC4776], where a set of parameters are provided to describe civic locations. In contrast to geospatial location information, which is the geospatial location of the device as a set of latitude, longitude and altitude coordinates represented by a CRS, civic location information can often be interpreted even if incomplete. For example, while geospatial information is not available inside buildings, civic location information can still provide an estimation of a device's location.

4. Location Information Elements

The following Information Elements can be used for exporting location-related information of a Metering Process. They SHALL be used for exporting geospatial and civic location, together with IPFIX Information Elements already defined for exporting IP traffic Flows.

4.1. geospatialLocationCRSCode

Description: Denotes the Coordinate Reference System (CRS) codes according to which the location coordinates are organized and related to the real world, as specified in [GEOSHAPE]. In this document we mandate the use of the World Geodetic System 1984

(WGS84) [WGS84] coordinate reference system and the usage of the European petroleum survey group (EPSG) code 4326 for two-dimensional (2D) shape representations and EPSG 4979 for three-dimensional (3D) volume representations.

Data Type: unsigned16
Data Type Semantics: identifier
PEN (provisional): 12559 (Inria)
ElementId: 401

4.2. geospatialLocationLat

Description: Denotes the coordinate information value of the latitude.

Data Type: float64
PEN (provisional): 12559 (Inria)
ElementId (provisional): 402

4.3. geospatialLocationLng

Description: Denotes the coordinate information value of the longitude.

Data Type: float64
PEN (provisional): 12559 (Inria)
ElementId (provisional): 403

4.4. geospatialLocationAlt

Description: Denotes the coordinate information value of the altitude.

Data Type: float64
PEN (provisional): 12559 (Inria)
ElementId (provisional): 404

4.5. geospatialLocationRadius

Description: Denotes a radius value (in meters) of a location described using a circular area in a two-dimensional CRS or a sphere shape in a three-dimensional CRS.

Data Type: float32
Data Type Semantics: quantity
PEN (provisional): 12559 (Inria)
ElementId (provisional): 405

4.6. civicLocationType

Description: Denotes the civic location information type as specified in [RFC4776].

Data Type: unsigned8

PEN (provisional): 12559 (Inria)

ElementId (provisional): 406

4.7. civicLocationValue

Description: Denotes a civic location information element that MUST be encoded as a UTF-8 string. The location information MAY be a civic address as specified in [RFC4776] or information on proximity to known objects.

Data Type: string

PEN (provisional): 12559 (Inria)

ElementId (provisional): 407

4.8. locationMethod

Description: Denotes the way in which the location information has been obtained. The locationMethod sub-registry is defined in Section 8.1.

Data Type: unsigned8

Data Type Semantics: identifier

PEN (provisional): 12559 (Inria)

ElementId (provisional): 408

4.9. locationTime

Description: Denotes the time when the location information is obtained on a device acting as an IPFIX Flow Exporter. The time is expressed in seconds since January 1, 1970, 00:00:00 UTC.

Data Type: dateTimeSeconds

Data Type Semantics: quantity

PEN (provisional): 12559 (Inria)

ElementId (provisional): 409

4.10. deviceId

Description: Denotes an identifier of a physical device acting as an IPFIX Flow Exporter. The Exporting Process uses this identifier to uniquely identify the device where Flows were metered. The identifier is unique per device. This Information Element can

be used when an IPFIX Flow Exporter is behind a NAT.

Data Type: unsigned64
Data Type Semantics: identifier
PEN (provisional): 12559 (Inria)
ElementId (provisional): 410

5. Guidelines for Using Location Information Elements

The specified location Information Elements in this document SHALL be used by a Metering Process for constructing an IPFIX location Template with respect to the following conventions.

Guideline #1: Location Information Elements MUST describe a discrete location defined as a place, point or area in which a Metering Process (i.e., IPFIX Flow Exporter) can be found.

Guideline #2: In situations where a discrete location can be described in multiple ways, each location SHOULD be described by means of a separate Template. A compound Template containing a subTemplateMultiList field [RFC6313] SHOULD be used in which each top-level element corresponds to a different location Template. For example, the location of a device being at the fifth floor of a particular building can be described using both a geospatial point (the location of the building) and civic information (fifth floor of a building).

Guideline #3: Exporting more than one location in a Flow Record MUST only be done if the different location descriptions refer to different places.

Guideline #4: A Metering Process MAY apply time-based Flow expiration policies as described in Section 5.1.1 of [RFC5470], or location-/distance-based expiration policies. For example, a Metering Process MAY expire current Flows when the device moves from one room to another.

Guideline #5: When another type of location data is available and needed to be sent, the Flow Exporter MUST send the template of the new location format.

6. Recommended Templates for Geospatial and Civic Location Export

The following Templates are defined as recommended Templates for exporting geospatial and civic location information. The geospatial templates are related to a point, circle or area shapes. The definition and usage of the shapes is covered in [GeoSHAPE]. Civic locations can be exported using a Template containing a

subTemplateList [RFC6313], where each element of the list corresponds to a Template.

6.1. Geospatial Point Location Template

The point shape is the simplest form of a geospatial location, which SHOULD be used when there is no known uncertainty. The following Template is defined for exporting a 2D geospatial point location:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 2										Length = 28																													
Template ID = 300										Field Count = 2																													
locationMethod = 408										Field Length = 1																													
locationTime = 409										Field Length = 4																													
geospatialLocationCRSCode=401										Field Length = 2																													
geospatialLocationLat = 402										Field Length = 8																													
geospatialLocationLng = 403										Field Length = 8																													

Figure 1: Template for exporting a 2D point-based geospatial location

For illustration, the following presents an example Data Record to export a 2D geospatial point location:

```

      0              1              2              3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Set ID = 300      |      Length = 28      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| locMethod = 3 |      locationTime = 123455555      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ... octet 4 |geospatialLocationCRSCode=4326 |geospatial ... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      ... LocationLat = 48.690855      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ... octet 6 - 8      |geospatial ... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      LocationLng = 6.172851      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ... octet 6 - 8      | Padding (opt) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 2: Data Record of a geospatial 2D point location

6.2. Geospatial Circle Location Template

The circle Template is suitable for exporting the location of a flow observed within a circle shape where its center is represented using a geospatial point position and its radius represents the uncertainty.

```

Template Record for Geospatial Circle (ID = 301)
| locationMethod(408)[1]
| locationTime(409)[4]
| geospatialLocationCRSCode(401)[2]
| geospatialLocationRadius(405)[4]
| geospatialLocationLat(402)[8]
| geospatialLocationLng(403)[8]

```

Figure 3: Template for exporting a circle-based geospatial location

The following presents an example of a Data Record carrying a circle-based geospatial location:


```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Set ID = 301      |      Length = 32      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| locMethod = 3 |      locationTime = 123455555      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ... octet 4 |geospatialLocationCRSCode=4326 | geospatial ...|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      ... LocationRadius = 850.24      | geospatial ...|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      ... LocationPosLat =      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      42.5463      | geospatial ...|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      ... LocationLng =      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      -73.2512      | Padding (opt) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 4: Data Record of a circle-based geospatial location

6.3. Geospatial List Template

The list locations Template is suitable for exporting a variable-length list of different geospatial point positions of a single flow. For example, it could be used to export the start and the end locations of a flow. The template relies on a subTemplateList data type to export the list of geospatial point-based positions. This template requires [RFC6313] compliant Exporting and Collecting Processes. Figure 5 depicts an example of such a subTemplate for exporting each element of the list.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Set ID = 2      |      Length = 20      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Template ID = 302      |      Field Count = 2      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| locationTime = 409      |      Field Length = 4      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| geospatialLocationLat = 402 |      Field Length = 8      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| geospatialLocationLng = 403 |      Field Length = 8      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 5: Template for exporting a geospatial 2D point-based position

```

Template Record for Geospatial List (ID = 303)
| locationMethod(408)[1]
| geospatialLocationCRSCode(401)[2]
+-subTemplateList(292)[0xFFFF]
  +-Geospatial 2D Point position Template Record(302)[16]

```

Figure 6: Template for exporting a geospatial list of locations

The following presents an example Data Record carrying a list of two geospatial point positions. Each point-based position is defined as an element of a subTemplateList Information Element with semantic "allof".

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Set ID = 303          |          Length = 53          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| locMethod = 3 |geospatialLocationCRSCode=4326 |          255          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|Geospatial Point List length=43 |semantic=allof| Template ID = |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... 302          |          locationTime = ...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|... 1234555555 |          geospatialLocationLat1 = ...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          43.311          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... octet 8          |          geospatialLocationPostLng1 = ...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          -73.422          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... octet 8          |          locationTime = ...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|... 1234555555 |          geospatialLocationLat2 =          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          43.111          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... octet 8          |          geospatialLocationLng2 =          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          -73.322          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... octet 8          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 7: Data Record of a geospatial list of point-based locations

6.4. Civic Location Template

A civic-based location Data Record consists of a tuple of (civicLocationType, civicLocationValue) Information Elements. Each tuple is defined as an element of a subTemplateList Information Element with semantic "allOf". This template requires [RFC6313] compliant Exporting and Collecting Processes.

```

Template Record for Civic location (ID = 304)
| locationMethod(408)[1]
| locationTime(409)[4]
+-subTemplateList (292)[0xFFFF]
  +-Civic element Template Record (ID = 305)
    | civicLocationType(406)[1]
    | civicLocationValue(407)[v]

```

Figure 8: Template for exporting a civic location

The "Civic element" Template Record, as shown in Figure 8, MUST be defined for each tuple. For the purpose of illustration, we consider exporting the civic location "Inria Nancy-Grand Est, Building B, Office 123" obtained through DHCP. Using the Template described in Figure 8, the resulting Data Record is as follows:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Set ID = 304										Length = 62																													
locMethod = 3										locationTime = 123455555																													
... octet 4										255										Civic elements list length = 50																			
semantic=allOf										Civic element TemplateID = 305										CivicType=21																			
21										CivicValue = Inria Nancy-Grand																													
										Est ...										CivicType=25																			
10										CivicValue = Building																													
										B ...										CivicType=28																			
10										CivicValue = Office																													
										123 ...																													

Figure 9: Data Record of a civic location

Note that the values of the `civicLocationType` are defined in [RFC4776].

6.5. Compound Location Template

A compound location is used to describe a location, represented by a composite of both civic and geospatial information. An example situation is a two-dimensional geospatial 2D point position (latitude, longitude) describing a location of a building, and a civic element representing the floor in that building. A `subTemplateMultiList` [RFC6313] SHOULD be used to export a Template for both geospatial and civic information. To represent the above example, the following Template is defined:

```
Template Record for Compound Location (ID = 306)
| locationTime(409)[4]
+-subTemplateMultiList(293)[0xFFFF]
  +-Geospatial Template Record (ID = 307)
    | locationMethod(408)[1]
    | geospatialLocationCRSCode(401)[2]
    | geospatialLocationLat(402)[8]
    | geospatialLocationLng(403)[8]
  +-Civic location Template Record (ID = 308)
    | locationMethod(408)[1]
    | civicLocationType(406)[1]
    | civicLocationValue(407)[v]
```

Figure 10: Template for exporting a compound location

A data Record encoded using the Template shown in Figure 11 is represented as follows:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Set ID = 311           |           Length = 64           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           locationTime = 1234555555555           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           255           | Attributes List Length = 53 | semantic=allOf|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Template ID = 312           | Geospatial Attr Length = 19 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| locMethod = 3 |geospatialLocationCRSCode=4326 |geospatial ... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           ... LocationLat1 =           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           -34.407           |geospatial ... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           ... LocationLng1 =           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           150.8883           | Template ID = |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ... 313           | Civic location Attr length=25 | locMethod=3 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| CivicType = 21|           21           | CivicValue = Inria ... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Nancy-Grand Grand Est ...           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 11: Data Record of a compound location

7. Security Considerations

The use of location information on the Internet has been discussed in "GeoPriv Requirements" [RFC3693], while the threats facing Internet protocols that carry location information are detailed in [RFC3694]. IPFIX messages exchanged between the Metering and Collecting Processes that carry location information should be signed and encrypted to provide protection, as defined in [RFC3694]. Support for Flow Record anonymization, as expressed in [RFC6235], is strongly recommended, since the dissemination of Flow Records including location information raises greater privacy issues than the dissemination of regular Flow Records. The applicability and analysis of these security requirements for the IPFIX protocol - especially in the case where location information is conveyed - is however outside of the scope of this document. This document only specifies the new IPFIX Information Elements for exporting location information. Otherwise, the same security considerations as those defined for the IPFIX protocol and the IPFIX information model apply.

8. IANA Considerations

This document specifies several new IPFIX Information Elements and types that need to be registered with IANA.

8.1. locationMethod Sub-Registry

The values of the location methods are enumerated within an IANA registry [RFC4119]. However, integer identifiers for these methods need to be registered with IANA as described below.

Number	Method	Description
0	GPS	Global Positioning System
1	A-GPS	GPS with assistance
2	Manual	Entered manually by a user
3	DHCP	Provided by DHCP [RFC5985]
4	Triangulation	Triangulated from time-of-arrival, signal strength or similar measurement
5	Cell	Location of the cellular radio antenna
6	802.11	IEEE 802.11 access point location

9. Acknowledgements

The authors were partly funded by FLAMINGO, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme, and the EIT ICT Labs activity "Smart Networks at the Edge".

10. References

10.1. Normative References

- [GeoShape] Thomson, M. and C. Reed, "GML 3.1.1 PIDF-LO Shape Application Schema for use by the Internet Engineering Task Force (IETF)", Candidate OpenGIS Implementation Specification 06-142r1, Version: 1.0, April 2007.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6313] Claise, B., Dhandapani, G., Aitken, P., and S. Yates, "Export of Structured Data in IP Flow Information Export (IPFIX)", RFC 6313, July 2011.

- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.
- [RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, March 2009.
- [RFC4776] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", RFC 4776, November 2006.
- [RFC6225] Polk, J., Linsner, M., Thomson, M., and B. Aboba, Ed., "Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information", RFC 6225, July 2011.
- [RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", RFC 5139, February 2008.
- [RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.
- [RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, "Geopriv Requirements", RFC 3693, February 2004.
- [RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization Support", RFC 6235, May 2011.

10.2. Informative References

- [NFDUMP] Haag, P., "NFDUMP", <http://nfdump.sourceforge.net>, May 2013.
- [NFSSEN] Haag, P., "NfSen", <http://nfsen.sourceforge.net>, January 2012.
- [SURFMAP] Hofstede, R., Fioreze, T., "SURFmap: A Network Monitoring Tool Based on the Google Maps API", Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, 2009, June 2009.
- [OGP] Oil and Gas Producers Association, "EPSG Geodetic

Parameter Registry", <http://www.epsg-registry.org>, August 2011.

[RFC5513] Farrel, A., "IANA Considerations for Three Letter Acronyms", RFC 5513, April 1 2009.

Appendix A. Example Implementation

This appendix is intended to show an example application that relies on the set of IPFIX Information Elements described in this document. This application, named SURFmap, is a network monitoring tool based on the Google Maps API and uses Flow data to visualize network Flows on a map [SURFMAP]. By default, geolocation databases are used for retrieving the (estimated) physical location associated to an IP address. The Information Elements described in this document, however, will allow SURFmap to use the absolute location information exported for Flows.

SURFmap has been developed in the past as a plugin to NfSen [NFSEN]. NfSen provides a Web-frontend to nfdump [NFDUMP], which is a set of tools for flow data collection and processing, among others. To support collection and processing of Flow Records containing any of the new Information Elements (e.g. by SURFmap), an extension to nfdump has been developed.

The following presents a set of Flow Records that have been exported by a mobile Flow Exporter. Several fields, such as destination IP address and port number, location timestamp and location method have been left out for the sake of space. It is clear that the mobile device has moved while exporting Flow Records, as the latitude and longitude coordinates have changed over time.

Start time	Src IP Addr:Port	Pkts	Bytes	Latitude	Longitude
20:19:21.852	173.194.40.113:443	9	2730	48.690855	6.172851
20:21:42.307	91.202.200.229:80	13	9137	48.690855	6.172851
20:21:42.307	10.21.20.232:59521	15	1547	48.690855	6.172851
20:22:38.084	73.194.40.113:80	8	1799	48.690855	6.172851
20:22:38.084	10.21.20.232:34056	9	877	48.690855	6.172851
21:17:13.498	173.194.45.80:443	12	2830	48.713145	6.17526
21:17:13.498	10.21.20.232:49233	15	2301	48.713145	6.17526
21:17:16.919	10.21.20.232:15572	1	72	48.744506	6.154815
21:17:16.919	172.20.2.39:53	1	257	48.744506	6.15481

Authors' Addresses

Olivier Festor
Inria
615 rue du Jardin Botanique
54600 Villers-les-Nancy
France

Phone: +33 3 83 59 30 66
Email: Olivier.Festor@inria.fr

Abdelkader Lahmadi
University of Lorraine - LORIA
615 rue du Jardin Botanique
54600 Villers-les-Nancy
France

Phone: +33 3 83 59 30 00
Email: Abdelkader.Lahmadi@loria.fr

Rick Hofstede
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Phone: +31 53 489 2013
Email: r.j.hofstede@utwente.nl

Aiko Pras
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Phone: +31 53 489 3778
Email: a.pras@utwente.nl

Network Management Research Group
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

M. Behringer
Cisco Systems
B. Carpenter
Univ. of Auckland
S. Jiang
Huawei Technologies Co., Ltd
February 14, 2014

Gap Analysis for Autonomic Networking
draft-jiang-nmrg-an-gap-analysis-00

Abstract

This document summarises a problem statement for an IP-based autonomic network that is mainly based on distributed network devices. The document reviews the history and current status of autonomic aspects of IP networks. It then reviews the current network management style, which is still heavily depending on human administrators. Finally the document describes the general gaps between the ideal autonomic network concept and the current network abilities.

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, 2014.

Copyright Notice

Copyright (c) 2014 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. Terminology	3
3. Current Status of Autonomic Aspects of IP Networks	3
3.1. IP Address Management and DNS	3
3.2. Routing	4
3.3. Configuration of Default Router	4
3.4. Hostname Lookup	5
3.5. User Authentication and Accounting	5
3.6. Security	5
3.7. Miscellaneous	6
4. Current Non-Autonomic Behaviors	6
4.1. Network Establishment	7
4.2. Network Maintenance & Management	7
4.3. Troubleshooting and Recovery	8
5. Approach toward Autonomy	9
5.1. More Coordination among Devices or Network Partitions	9
5.2. Benefit from Knowledge	9
6. Security Considerations	10
7. IANA Considerations	10
8. Acknowledgements	10
9. Informative References	10
Authors' Addresses	12

1. Introduction

The general goals and relevant definitions for autonomic networking are discussed in [I-D.irtf-nmrg-autonomic-network-definitions]. In summary, the fundamental goal of an autonomic network is self-management, including self-configuration, self-optimization, self-healing and self-protection. Whereas interior gateway routing protocols such as OSPF and IS-IS largely exhibit these properties, most other aspects of networking require top-down configuration often involving human administrators and a considerable degree of centralisation. In essence Autonomous Networking is putting all network configuration onto the same footing as routing, limiting manual or database-driven configuration to an essential minimum. It should be noted that this is highly unlikely to eliminate the need for human administrators, because many of their essential tasks will remain. The idea is to eliminate tedious and error-prone tasks, for

example manual calculations, cross-checking between two different configuration files, or tedious data entry. Higher level operational tasks, and trouble-shooting, will remain to be done in any case.

Note in draft: This is a preliminary version. It certainly lacks information about current status, and it lacks many external references. Especially the final section (Section 5) is very preliminary. Comments and suggestions are very welcome.

2. Terminology

The terminology defined in [I-D.irtf-nmrg-autonomic-network-definitions] is used in this document. Additional terms include:

- o Automatic: A process that occurs without human intervention, with step-by-step execution of rules. However it relies on humans defining the sequence of rules, so is not Autonomic in the full sense. For example, a start-up script is automatic but not autonomic.

3. Current Status of Autonomic Aspects of IP Networks

This section discusses the history and current status of autonomy in various aspects of network configuration, in order to establish a baseline for the gap analysis. In one particular area, routing protocols, autonomic information exchange and decision is a well established mechanism. The question is how to extend autonomy to cover all kinds of network management objectives.

3.1. IP Address Management and DNS

Originally there was no alternative to completely manual and static management of IP addresses. Once a site had received an IPv4 address assignment (usually a Class C /24 or Class B /16, and rarely a Class A /8) it was a matter of paper-and-pencil design of the subnet plan (if relevant) and the addressing plan itself. Subnet prefixes were manually configured into routers, and /32 addresses were assigned administratively to individual host computers, and configured manually by system administrators. Records were typically kept in a plain text file or a simple spreadsheet.

Clearly this method was clumsy and error-prone as soon as a site had more than a few tens of hosts, but it had to be used until DHCP [RFC2131] became a viable solution during the second half of the 1990s. DHCP made it possible to avoid manual configuration of individual hosts (except, in many deployments, for a small number of servers configured with static addresses).

In terms of management, it is difficult to separate IP address management from DNS management. At roughly the same time as DHCP came into widespread use, it became very laborious to manually maintain DNS source files in step with IP address assignments. Because of reverse DNS lookup, it also became necessary to synthesise DNS names even for hosts that only played the role of clients. Therefore, it became necessary to synchronise DHCP server tables with forward and reverse DNS. For this reason, Internet Protocol address management tools emerged. These are, however, a centralised and far from autonomic type of solution.

A related issue is prefix delegation, especially in IPv6 when more than one prefix may be delegated to the same physical subnet. DHCPv6 Prefix Delegation [RFC3633] is a useful solution, but how this topic is to be handled in home networks is still an open question. Still further away is automated assignment and delegation of IPv4 subnet prefixes.

Another complication is the possibility of Dynamic DNS Update [RFC2136]. With appropriate security, this is an autonomic approach, where no human intervention is required to create the DNS records for a host. Also, there are coexistence issues with a traditional DNS setup.

3.2. Routing

Since a very early stage, it has been a goal that Internet routing should be self-healing when there is a failure of some kind in the routing system (i.e. a link or a router goes wrong). Also, the problem of finding optimal routes through a network was identified many years ago as a problem in mathematical graph theory, for which well known algorithms were discovered (the Dijkstra and Bellman-Ford algorithms). Thus routing protocols became largely autonomic in the 1980s, as soon as the network was big enough for manual configuration of routing tables to become difficult.

IGP routers do need some initial configuration data to start up the autonomic routing protocol. Also, BGP-4 routers need static configuration of routing policy data. So far, this policy configuration has not been made autonomic at all.

3.3. Configuration of Default Router

Originally this was a manual operation. Since the deployment of DHCP, this has been automatic as far as most IPv4 end systems are concerned, but the DHCP server must be appropriately configured. In simple environments such as a home network, the DHCP server resides in the same box as the default router, so this configuration is also

automatic. In more complex environments, where an independent DHCP server or a local DHCP relay is used, configuration is more complex and not automatic.

In IPv6 networks, the default router is provided by Router Advertisement messages [RFC4861] from the router itself, and all IPv6 hosts make use of it. The router may also provide more complex Route Information Options. The process is automatic as far as all IPv6 end systems are concerned, and DHCPv6 is not involved. However there are still open issues when more than one prefix is in use on a subnet and more than one first-hop router may be available as a result.

3.4. Hostname Lookup

Originally host names were looked up in a static table, often referred to as /etc/hosts from its traditional file path in Unix systems. When the DNS was deployed during the 1980s, all hosts needed DNS resolver code, and needed to be configured with the IP addresses (not the names) of suitable DNS servers. Like the default router, these were originally manually configured. Today, they are provided automatically via DHCP or DHCPv6 [RFC3315]. For IPv6 end systems, there is also a way for them to be provided automatically via a Router Advertisement option. However, the DHCP or DHCPv6 server, or the IPv6 router, need to be configured with the appropriate DNS server addresses.

3.5. User Authentication and Accounting

Originally, user authentication and accounting are mainly based on the physical connectivities. Network operators charged based on the set up of dedicated physical links with users. Autonomic user authentication are introduced by Point-to-Point Protocol [RFC1661], [RFC1994] and RADIUS protocol [RFC2865], [RFC2866] in early 1990s. As long as a user complete online authentication through RADIUS protocol, the accounting for that user starts on AAA server autonomically. This mechanism enables charging business model based on the usage of users, either traffic based or time based. However, the management for user authentication information remains manual by network administrators.

3.6. Security

Security has many aspects that need configuration and are therefore candidates to become autonomic. On the other hand, it is essential that a network's central policy should be applied strictly for all security configuration. As a result security has largely been based on centrally imposed configurations.

Many aspects of security depend on policy, for example firewall policies. Policies are by definition human made and will therefore also persist in an autonomic environment. However, policies are becoming more high-level, abstracting for example addressing, and focusing on the user or application. The methods to manage, distribute and apply policy, and to monitor compliance and violations could be autonomic.

Today, many security mechanisms show some autonomic properties. For example user authentication via 802.1x allows automatic mapping of users after authentication into logical contexts (typically VLANs). While today configuration is still very important, the overall mechanism displays signs of self-adaption to changing situations.

BGP Flowspec [RFC5575] allows a partially autonomic threat defense mechanism, where threats are identified, the flow information is automatically distributed, and counter-actions can be applied. Today typically a human operator is still in the loop to check correctness, but over time such mechanisms can become more autonomic.

Negotiation capabilities, present in many security protocols, also display simple autonomic behaviours. In this case a security policy about algorithm strength can be configured into servers but will propagate automatically to clients. A proposal has been made recently for automatic bootstrapping of trust in a network [I-D.behringer-default-secure]. Solutions for opportunistic encryption have been defined [RFC4322], [I-D.farrelll-mppls-opportunistic-encrypt], but these do not adhere to a central policy.

3.7. Miscellaneous

There are innumerable other properties of network devices and end systems that today need to be configured either manually or using a management protocol such as SNMP [RFC1157] or NETCONF [RFC6241]. In a truly autonomic network, all of these would need to either have satisfactory default values or be configured automatically. Some examples are parameters for tunnels of various kinds, flows (in an SDN context), quality of service, service function chaining, energy management, system identification, NTP configuration etc. Even one undefined parameter would be sufficient to prevent fully autonomic operation.

4. Current Non-Autonomic Behaviors

In the current networks, many operations are still heavily depending on human intelligence and decision, or on centralised top-down network management systems. These operations are the targets of

Autonomic Network technologies. The ultimate goal of Autonomic Network is to replace tedious human operations by autonomic functions, so that the networks can independently run without having to ask human support for routine details, while it remains possible to restore human intervention when unavoidable. Of course, there would still be the absolute minimum of human input required, particularly during the network establishment stage, and during difficult trouble-shooting.

This section analyzes the existing human and central dependencies in the current networks.

4.1. Network Establishment

Network establishment requires network operators to analyze the requirements of the new network, design a network architecture and topology, decide device locations and capacities, set up hardware, design network services, choose and enable required protocols, configure each device and each protocol, set up user authentication and accounting policies and databases, design and deploy security mechanisms, etc.

Overall, these jobs are quite complex work that cannot become fully autonomic in the foreseeable future. However, part of these jobs may be able to become autonomic, such as device and protocol configurations and database population. The initial network management policies/behaviors may also be transplanted from other networks and automatically localized.

4.2. Network Maintenance & Management

The network maintenance and management are very different for ISP networks and enterprise networks. ISP networks have to change much more frequently than enterprise networks, given the fact that ISP networks have to serve a large number of customers who have very diversified requirements. The current rigid model is that network administrators design a limited number of services for customers to order. New requirements of network services may not be able to be met quickly by human management. Given a real-time request, the response must be autonomic, in order to be flexible and quickly deployed. However, behind the interface, describing abstracted network information and user authorization management may have to depend on human intelligence from network administrators in the foreseeable future. User identification integration/consolidation among networks or network services are another challenge for autonomic network access. Currently, the end users have to manually manage their user accounts and authentication information when they switch among networks or network services.

Classical network maintenance and management mainly manages the configuration of network devices. Tools have developed to enable remote management and make the management easier. However, the decision of each configuration depends either on human intelligence or rigid templates. This is the source of most network configuration errors. It is also the barrier to increase the utility of network resources because the human management cannot respond quickly enough to network events, such as traffic bursts, etc. For example, currently, a light load is normally assumed in network design because there is no mechanism to properly handle a sudden traffic flood. It is actually normal to avoid network crashes caused by traffic overload by wasting a huge amount of resources.

Autonomic decision processes of configuration would enable dynamic management of network resources (by managing resource relevant configuration). Self-adapting network configuration would adjust the network into the best possible situation, which also prevents configuration errors from having lasting impact.

4.3. Troubleshooting and Recovery

The current networks are suffering difficulties in locating the cause of network failures. Although the network devices may issue many warnings during running, most of them are not sufficiently precise to be identified as errors. Some of them are early warnings that would not develop into real errors. Others are in effect random noise. For many scenarios, human experience is vital to identify real issues and locate them. This situation may be improved by associating warnings from multiple network devices together. Also, introducing automated learning techniques (comparing current warnings with historical relationships between warnings and actual faults) could increase the possibility and success rate of autonomic network diagnoses and troubleshooting.

Depending on the network errors, some of them may require human interventions, particularly for hardware failures. Meanwhile, some network management behavior may help to reduce the impact from errors, such as switching traffic flows around. Today this is usually manual. Software failures and configuration errors (including to roll back software versions and to reboot hardware) currently depend on humans. Such problems could be autonomically corrected if there were diagnostics and recovery functions defined in advance for them. This would fulfill the concept of self-healing.

5. Approach toward Autonomy

The task of autonomic networking is to build up individual autonomic decision processes that could properly combine to respond to every type of network event. This section (when complete) will outline what needs to be developed.

5.1. More Coordination among Devices or Network Partitions

Events in networks are normally not independent. They are associated with each other. But most of current response functions are based on independent processes. The network events that may naturally happen distributed should be associated in the autonomic processes.

In order to make right or good decisions autonomically, the network devices need to know more information than just reachability (routing) information from the relevant or neighbor devices. There are dependencies between such information and configurations. Currently, most of these configurations currently require manual coordination by network administrators.

There are therefore increased requirements for horizontal information exchanging in the networks. Particularly, negotiation among network devices are needed for autonomic decision.

[I-D.jiang-config-negotiation-ps] analyzes such requirements. Although there are many existing protocols with negotiation ability, each of them are only serve a specific and narrow purpose.

[I-D.jiang-config-negotiation-protocol] is one of the attempts to create a generic negotiation platform, which would support different negotiation objectives.

5.2. Benefit from Knowledge

The more knowledge we have, the more intelligent we are. It is the same for networks and network management. It is when one component in the network lacks knowledge that affects what it should do, and another component has that knowledge, that we usually rely on a human operator or a centralised management tool to convey the knowledge.

Up to now, most available network knowledge is only the current network status, either inside a device or relevant data from other devices.

However, historic knowledge is very helpful to make correct decisions, in particular to reducing network oscillation or to manage network resources over time. Transplantable knowledge from other networks can be helpful to initially set up a new network or new network devices. Knowledge of relationship between network events

and network configuration may help network to decide the best parameters according to real performance feedback.

6. Security Considerations

This document is focussed on what is missing to allow autonomic network configuration, including of course security settings. Therefore, it does not itself create any new security issues. It is worth underlining that autonomic technology must be designed with strong security properties from the start, since a network with vulnerable autonomic functions would be at great risk.

7. IANA Considerations

This memo includes no request to IANA.

8. Acknowledgements

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

9. Informative References

[I-D.behringer-default-secure]

Behringer, M., Pritikin, M., and S. Bjarnason, "Making The Internet Secure By Default", draft-behringer-default-secure-00 (work in progress), January 2014.

[I-D.farre111-mpls-opportunistic-encrypt]

Farrel, A. and S. Farrell, "Opportunistic Encryption in MPLS Networks", draft-farre111-mpls-opportunistic-encrypt-02 (work in progress), February 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-00 (work in progress), December 2013.

[I-D.jiang-config-negotiation-protocol]

Jiang, S., Carpenter, B., Liu, B., and Y. Yin, "Configuration Negotiation Protocol for Network Devices", draft-jiang-config-negotiation-protocol-00 (work in progress), October 2013.

- [I-D.jiang-config-negotiation-ps]
Jiang, S., Yin, Y., and B. Carpenter, "Network Configuration Negotiation Problem Statement and Requirements", draft-jiang-config-negotiation-ps-02 (work in progress), January 2014.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", STD 15, RFC 1157, May 1990.
- [RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.
- [RFC1994] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.
- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC4322] Richardson, M. and D. Redelmeier, "Opportunistic Encryption using the Internet Key Exchange (IKE)", RFC 4322, December 2005.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, August 2009.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

Authors' Addresses

Michael H. Behringer
Cisco Systems

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

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