Overview of autonomic networking ideas and the need, scope and criteria for use cases
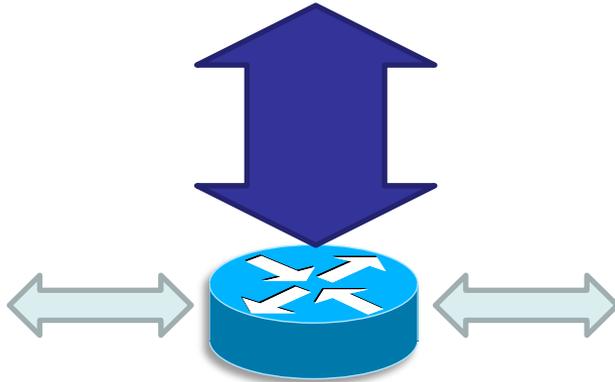
UCAN BoF

90th IETF, 23 July 2014

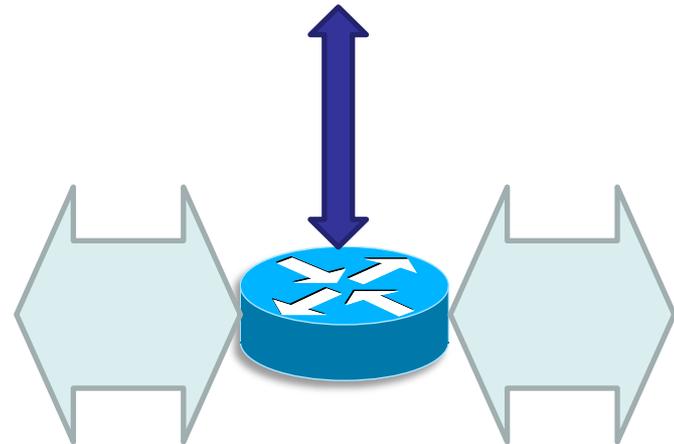**Michael Behringer, Brian Carpenter, Sheng Jiang**

Traditional

- Configuration
- Monitoring, Reporting
- Troubleshooting

- Routing

Autonomic

- Policy and Service Orchestration
- Aggregated Reporting
- Simplified troubleshooting

- Routing
- Discovery
- Autonomic interactions (self-*)

Autonomic Networking means:
→**Minimize operator interventions**
→**Minimize NMS dependencies**

Overall Goal

- **Distribute what you can, centralise what you must**
- **Create a common infrastructure for autonomic functions**

Example:
Control and Management Plane Security

- **Today, control plane and management plane is secured by:**
  - **IGP routing authentication (OSPF, ISIS, etc)**
  - **BFD authentication, BGP routing authentication, SSH, etc…**
- **Each function defines it's own**
  - **Key material**
  - **Algorithms**
- **Idea: Have a single way to secure all these functions**
  - **Based on one common trust anchor**
  - **Protocols still use their own methods, but based on shared trust model**
  - **→ Protocols become "secure by default"**

draft-irtf-nmrg-autonomic-network-definitions

[…]

Gap Analysis for Autonomic Networking

draft-irtf-nmrg-an-gap-analysis-00

Introduction

- **Goals and definitions are from draft-irtf-nmrg-autonomic-network-definitions.**

- **This draft aims to identify status of autonomic behaviors and outline what is missing.**

- **Reviews status for address management, DNS, routing, security & AAA.**

- **Then reviews non-autonomic behaviors and gaps.**

Autonomic Networking - Overview

Non-autonomic behaviors (1)

- **Network establishment, including:**
  - **analyze the requirements of the new network**
  - **design network architecture and topology**
  - **decide device locations and capacities**
  - **security bootstrap***
  - **transplant initial network management policies/behaviors from other networks and localizing them***

- **Network Maintenance & Management:**
  - **Configuration updates after installing (or removing) devices***
  - **Adjust the network into the best possible situation.***
  - ***candidates for autonomic operation**

Non-autonomic behaviors (2)

- **Troubleshooting and Recovery:**
  - **Overload of central or human management during major failures.***
  - **Associating warnings from multiple devices***
  - **Correcting software failures and configuration errors***
  - **Predicting failures or overloads before they occur***

  - ***candidates for autonomic operation**

Approach to autonomy: what's missing? (1)

- **More Coordination among Devices or Network Partitions**
  - **Exchange knowledge between components**
  - **Horizontal as well as vertical information exchange**
  - **Detect and correct inconsistencies where they arise**
- **Don't rely on a superior intelligence except for general policy intent.**
  - **Do not wait for instructions before correcting or improving configuration.**

10

Approach to autonomy: what's missing? (2)

- **Forecasting and Dry Runs**
  - **In a conventional network, configuration changes have to be designed theoretically.**
  - **There is a real risk that applying the changes to the running network will cause a failure.**
  - **An autonomic network could fill this gap with a "dry run" mode.**

- **Benefit from knowledge**
  - **Historic knowledge, knowledge transplanted from other networks, and relationship between network events and configuration may help network to configure and stabilise itself.**

11

Scope and Criteria for Use Cases

Criteria for Use Cases

A function is a good candidate for autonomic work at the IETF if:

- **There is operator interest**
- **Distribution**
  - **The function cannot be completely centralised, or**
  - **There are advantages with distribution**
- **Sharing infrastructure**
  - **There are other functions that require similar functionality or benefit from the proposed function**
- **Abstraction**
  - **There is an easy way to describe the function, high-level, network wide**
- **Simplicity to implement**
  - **Makes it easier to get started**