Autonomic Network Intent and Format

draft-du-anima-an-intent-04

Zongpeng Du
Sheng Jiang
Jeferson Campos Nobre
(new) Michael Behringer
Laurent Ciavaglia

Changes in version 04

- New section on Intent Life Cycle (section 4)
  - Initial contribution from Michael B.
  - Tentative description of Intent “flow”
  - Lots of comments/feedback on the mailing list

Intent Life Cycle (1/2)

Step 1, 2: Business goals formalized as machine readable Intents, encoded as file(s) and "given to the network“.

Step 3: Intent file is "ingested“ on a node and then needs to be distributed.

Step 4: Intent file is flooded to all nodes in a network. Each node re-distributes the original Intent files, without modification nor interpretation. Every node has a copy of the original Intent file(s).

Step 5: Intent is split into sections, one for the ANI itself, others for specific Autonomic Functions. ASAs are notified if there is new Intent for them. Some intent sections may not apply to a particular node. Now each component of a node (ANI, all ASAs) know their respective Intent.

Step 6: The ANI as well as all ASAs on a node interpret their respective Intent section(s). It gets translated into a "target configuration". For this translation, it may be necessary for ASAs to communicate with ASAs on other nodes. All such communications may be triggered by Intent, but the communications themselves are not Intent.

Intent Life Cycle (2/2)

Step 7: The target configuration resulting from Intent has the lowest priority; any other man agement method (CLI, NETCONF, etc.) overrides Intent.

Step 8: Each autonomic function needs to register with a "conflict resolution function" whic h parameters it modifies; in case of conflict, the conflict resolution function takes a decisi on and feeds that back to the autonomic functions. This may modify the target configura tion.

Step 9: Applying the target configuration.

Step 10: The NOC needs to know about certain conditions, such as conflicts with non-auton omic management. Not all conflicts can be resolved automatically, so they may require N OC actions. Undesirable states (deviations from expected default behavior) may have to b e communicated too. To some extent, Intent itself can specify which conditions should tri gger feedback loops to the NOC. Feedback loops may happen at other phases as well (ex: 8).

Current definitions

- An abstract, declarative, high-level policy used to operate an aut onomic domain
  (as per draft-ietf-anima-reference-model-02 and RFC7575)

- One Autonomic Network = Multiple Intents

- One Intent = Multiple Outputs

- Network operators/administrators writes Intents

- Autonomic Functions define what Intents they understand

Open Issues

- Conflicts between Intents and between outputs of Intents should be managed
  - coordination or conflict resolution mechanism(s)
- Heterogeneity in format and semantics of what AFs understand
  - translation mechanism(s), standardization of format and semantics?
- Definition of intents for the ANI (functions)?
- Mode(s) of distribution of Intent
  - Flooding, other approaches?

Future of the work…

- A high-level, non prescriptive section is present in the reference model document.
- Intent specifications is not a chartered work item.
- However, an AN cannot operate without guidance from the network operator.
- Suggestions?

Thank you

Questions

1-Who writes intents?

2-How many intents?

3-How many domains?

4-What are the intent levels/hierarchy?

5-Where/by what is intent processed/compiled?

6-Flooding: what are the requirements?

7-How is intent understood by node/ASA?

8-Can an ASA write an intent for another ASA?

Examples

A-Do the right thing

B-Freeze network enrollment

C-Arrange VM guest distribution so that (CPU) utilization is < 70%

D-Assign prefixes to RAN nodes

E-Protect premium users traffic

F-Maximize energy savings