A Mechanism for Intent Driven Adaptive Policy Decision Making

Joseph McNamara*, Liam Fallon†, and Enda Fallon*
*Software Research Centre, Athlone Institute of Technology, Athlone, Ireland
joseph.mcnamara@ericsson.com efallon@ait.ie
†Network Management Lab, Ericsson, Athlone, Ireland
liam.fallon@est.tech

***Abstract*—This demonstration focuses on the comparison and identification of conflicts between independent goals described through intent.**

Our approach, implemented in the Adaptive Policy EXecution system, is presented in 3 stages: 1) Intent generation, describing the structure of an intent 2) Intent comparability, detailing the requirements allowing for the effective comparison of goals and 3) Conflict resolution, detailing the process of identifying appropriate responses in accordance with already established intent goals. This demo will showcase each of the stages highlighting both the positive aspects and limitations of the approach.

***Index Terms*—Adaptive Policy, Intent, Conflict Mitigation, Recursive Structures**

I. Introduction

Intent based networking has been proposed as an approach to reduce complexity in network configurations [1]. The concept takes a modeling language and uses it to describe an abstract view of an intended network model [2]. Intent is currently topical in research, viewed as a mechanism to enhance network flexibility and management, however there lacks a unified definition for Intent Driven Networks [3]. This suggests intent requires more maturity before a consensus can be achieved and it can progress towards standardization. The authors of [4] indicated that intent based networking has not evolved since 2015 in regard to frameworks, platforms and tools however advances in artificial intelligence such as Natural Language Understanding are expected to increase its adaption in the future. Adopting Natural Language Understanding within an intent based system can produce abstract rules for the structuring of intent information. This would enable flexible representations of intent but is not a core requirement for the realization of these systems. Policy has played a core role in the realization of many intent driven mechanisms, often used to create actionable responses from abstract network statements [5]. This has been seen at multiple levels of the network from intent driven forwarding rules for programmable switches [6] to virtual network management platforms [7] and the orchestration of dynamic service chaining of Virtual Network Function [8]. Our demonstration adopts intent as a driving mechanism for a network configuration usage scenario. The usage scenario describes a network emulation tool influenced through policy. Intent events received by policy and trigger a validation cycle where active intents are compared and influence new network configurations. The objective of the demonstration is to identify the requirements of an intent driven approach and provide processes to meet these requirements. In §II we detail the structuring of the intent information. In §III we discuss the granularity required in intent information to produce actionable responses and identify achieved goals. In §IV we describe the mechanism responsible for identifying and mitigating conflicts between distinct intents. In §V the usage scenario is described for this paper.

978-3-903176-31-7 © 2020 IFIP

II. Intent Generation

The structuring of intent for this system was influenced by two factors. The maintaining of readability for humans and machine while not limiting the level of descriptive detail possible in the intent. From a generic policy perspective, allowing for a high level of detail is important as information stored within the event can be used to provide context to the situation requiring a decision. In this case our policy is less concerned with highly detailed events given the concept of intent based networking, however the intent must provide a condition to be met with the detail necessary to identify the violation of the condition. Our approach builds intent as a recursive structure with the keywords Who, What, When, Where and How. *Who*: Can describe the agent responsible for the intent creation or the module or component to be affected. *What*: Can describe the job to be undertaken or the goal to be achieved. *When*: Can describe a time frame for the intent to be enforced as a once off or as a reoccurring goal. *Where*: Can describe the components to be affected or a condition / situation where the intent is to be implemented. *How*: Can describe conditions on to which the goal is achieved or can define done in regard to the intent goal.

The *Usage Scenario Intent* (Fig.1) describes a KPI enforcement goal with a time frame condition. The intent describes who generated it along with a general description of what the intent is. The goals and conditions are then described lower in the intent structure.

III. Intent Comparison

On receiving an intent the policy engine triggers an intent validation cycle. This cycle is used to ensure that newly received intents are in accordance with already active intents

```
{
        "Who":  "NetworkAdministrator",
        "What":  "KPIEnforcement",
        "When":  [{
                "What":  "Start Time",
                "When":  "08:00"
            },
            {
                "What":  "End Time",
                "When":  "12:00"
            }
        ],
        "Where":  {
            "What":  "Bandwidth",
            "How":  "<1MBs"
        },
        "How":  "null"
}
```

Fig. 1: Usage Scenario Intent

```
/who/NetworkAdministrator
/what/KPIEnforcement
/when/what/startTime
/when/when/08:00
/when/what/endTime
/when/when/12:00
/where/what/Bandwidth
/where/how/<1MBs
/how/null
```

Fig. 2: Intent Path Collection

in the policy engine. The first step is the translation of the new intent event into a common internal structure. The use of an internal common structure allows the policy to map the intent to a supported framework without introducing additional dependencies to external systems. As a result formatting issues are avoided and adjusting to new industry formats can be done through a small policy update. A recursive function is used to navigate the intent parsing the data into a collection of path like statements similar to a folder directory. This collection of statements produces a tree structure which, through the keywords, can be navigated to identify the depth of the tree and the associations between values.

The *Intent Path Collection* (Fig.2) shows the parsed intent for our usage scenario shown in Fig.1. Navigating the intent path collection is straightforward allowing new intents and current network state information to be evaluated quickly. Building intent as a recursive structure provides a number of important features for the system. When parsing the intent, the structure can be broken down into a collection of paths which are easily navigated and stored by the APEX system. This allows for the direct comparison of different intents within the policy framework.

IV. Intent Resolution

Direct comparison of independent intent path collections can only identify commonality between intents based on the overlapping of keywords. The system can identify that two independent intents are referring to a similar condition, however the system has no context to what the condition is or how it can be impacted. Our solution was to implement a dictionary designed with our policies role in mind. With a dictionary the system can map the values of the intent

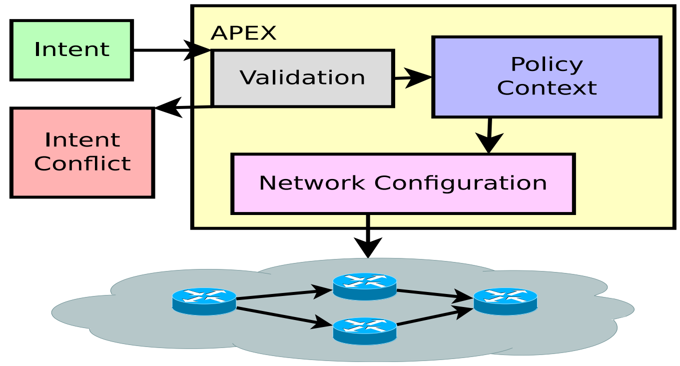

Fig. 3: Architecture

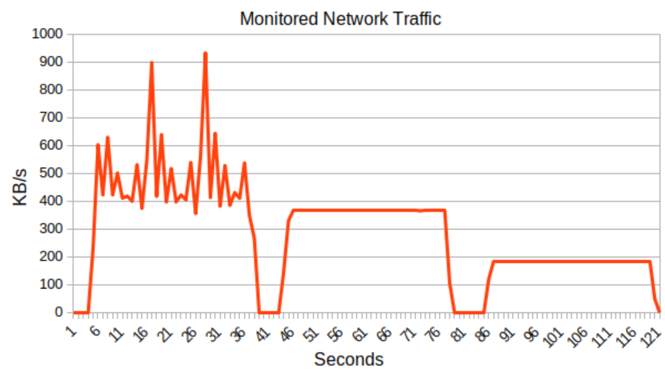

Fig. 4: Network Traffic Data

to a predefined structure. This structure provide attributes to recognized values, allowing them to be compared on a common level given they share comparable features An ontological approach introduces a dependency into the system, the modeling of intent values, however the standardization of these values internally provides the meta data required to enable informed comparisons. Without informed comparisons the policy cannot resolve independent intents impacting common network attributes.

V. Usage Scenario

The proposed demo shows our intent driven policy approach to generate network configurations for the Mininet network emulation tool. The architecture shown in Fig.3 describes the relationships between our components. Intents are generated and sent to the APEX engine. The intent received by the APEX system is shown in Fig.1. These intents are processed and compared, as a result a new Mininet configuration is generated. After the generation of the Mininet configuration a new intent is introduced to the system, triggering a new validation attempt. During this validation attempt new intents are processed into the path collection format shown in Fig.2. Using the dictionary, value identifiers are mapped to the corresponding structures. Newly mapped values are then compared, generating values that share validity across stored intents. These values are used in the generation of a network configuration file.

The network traffic displayed in Fig.4 show the realization of three intents. The first intent is described in Fig.1, the second intent is the same as the first except reducing bandwidth to 366KB/s and the third reduces bandwidth further to 184KB/s. In the event that a newly received intent directly conflicts with an existing intent a notification event is generated containing information on the owners of the conflicting intent and the values responsible for the conflict.

ACKNOWLEDGMENTS

This work is partly funded by the European Commission via the ARCFIRE project (Grant 687871) under the H2020 program. This work was also supported in part by Irish Research Council Enterprise Partnership Scheme Postgraduate Scholarship 2017 under Project EPSPG/2017/275.

REFERENCES

[1] Y. Tsuzaki and Y. Okabe, "Reactive configuration updating for intent-based networking," in *2017 International Conference on Information Networking (ICOIN)*, 2017, pp. 97–102.

[2] J. Augé and M. Enguehard, "A network protocol for distributed orchestration using intent-based forwarding," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 718–719.

[3] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, "A survey on intent-driven networks," *IEEE Access*, vol. 8, pp. 22 862–22 873, 2020.

[4] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[5] B. E. Ujcich, A. Bates, and W. H. Sanders, "Provenance for intent-based networking," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, 2020, pp. 195–199.

[6] B. Lewis, L. Fawcett, M. Broadbent, and N. Race, "Using p4 to enable scalable intents in software defined networks," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, 2018, pp. 442–443.

[7] Y. Han, J. Li, D. Hoang, J. Yoo, and J. W. Hong, "An intent-based network virtualization platform for sdn," in *2016 12th International Conference on Network and Service Management (CNSM)*, 2016, pp. 353–358.

[8] F. Callegati, W. Cerroni, C. Contoli, and F. Foresta, "Performance of intent-based virtualized network infrastructure management," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.