Self-Generated Intent-Based System

Mehdi Bezahaf - Lancaster University
Marco Hernandez, Lawrence Bardwell, Eleanor Davies, Matthew Broadbent, Daniel King, David Hutchison

10th International Conference on the Network of the Future (NoF’19)
Architecture Planes
Architecture Planes
Architecture Planes

Multi-sources data Collection
Architecture Planes

Multi-sources data Collection
Architecture Planes

Service Requester → Prediction → Knowledge Plane

Multi-sources data Collection
Architecture Planes

Service Requester -> Intent Plane -> Prediction -> Knowledge Plane

Multi-sources data Collection
Architecture Planes

Service Requester

Intent Plane

Prediction

Knowledge Plane

Management Plane

Multi-sources data Collection
Architecture Planes
Architecture Planes


Multi-sources data Collection
Architecture Planes

Service Requester

Prediction

Knowledge Plane

Intent Plane

Management Plane

Control/Data Planes

Orchestration Plane

Multi-sources
data Collection
Intent Plane

Service Requester

Intent Plane

Management Plane
Intent

- An *intent* expresses a requirement:
**Intent**

- An *intent* expresses a requirement:
  - Expressed from an external client/app
  - Qualitative
  - High level
  - *I want connectivity, Reserve me an audible connection…*
Intent

❖ An *intent* expresses a requirement:
  ❖ Expressed from an external client/app
    ❖ Qualitative
    ❖ High level
      ❖ *I want connectivity, Reserve me an audible connection…*
  ❖ Owned by the operator
    ❖ Quantitative
    ❖ Might be lower level and precise
      ❖ *Restrict the load to 50% max on a links…*
Quality of Service

- QoExperience and QoS are all what the end-users need
- It can be:
  - Jitter
  - Latency
  - Throughput
  - Bandwidth …
Resilience

- Resilience is the ability of the network to preserve all QoS and more face of various faults and challenges

- A system is Resilient, when it is:
  - Available:
    - Accessible when needed;
  - Reliable:
    - Able to provide the service when asked
Multi-layer Intent System
Intent Plane

Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself
Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself.

2- If it is from the client, it has to go through NLP/Mapping block.
Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself

2- If it is from the client, it has to go through NLP/Mapping block

3- We Inject Resilience and QoS to the original request
Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself

2- If it is from the client, it has to go through NLP/Mapping block

3- We Inject Resilience and QoS to the original request
Intent Plane

Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself

2- If it is from the client, it has to go through NLP/Mapping block

3- We Inject Resilience and QoS to the original request

Client/app Requester
Operator Requester
System itself Requester

NLP/Mapping

Reformulate with Resilience and QoS

Inject Resilience/QoS
1- Intent can be originated from client/app, operator or system itself

2- If it is from the client, it has to go through NLP/Mapping block

3- We Inject Resilience and QoS to the original request

4- The request now is mapped to lower, more technical level request through multiple layers
Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself

2- If it is from the client, it has to go through NLP/Mapping block

3- We Inject Resilience and QoS to the original request

4- The request now is mapped to lower, more technical level request through multiple layers

Client/app Requester

Operator Requester

System itself Requester

Intent Plane

NLP/Mapping

Reformulate with Resilience and QoS

Make it more feasible (actions)

Inject Resilience/QoS

Make it more feasible (actions)
Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself

2- If it is from the client, it has to go through NLP/Mapping block

3- We Inject Resilience and QoS to the original request

4- The request now is mapped to lower, more technical level request through multiple layers

5- Until we reach a certain simplification and technicality to handle it to the NM layer
Multi-layer Intent System

1- Intent can be originated from client/app, operator or system itself.

2- If it is from the client, it has to go through NLP/Mapping block.

3- We Inject Resilience and QoS to the original request.

4- The request now is mapped to lower, more technical level request through multiple layers.

5- Until we reach a certain simplification and technicality to handle it to the NM layer.
Example – Parallel with RM-ODP model

Intent Plane

Management Plane
Example – Parallel with RM-ODP model

Intent Plane

I want connectivity

Management Plane
Example – Parallel with RM-ODP model


Intent Plane

I want connectivity

The need

I want connectivity now

action

when?
Example – Parallel with RM-ODP model


Information View what is about?

Intent Plane

I want connectivity

The need I want connectivity now

action

now

Client

with a good

bandwidth

Get connected

now

Management Plane
Example – Parallel with RM-ODP model


Information View: what is about?

Computational View: How does each bit work?

I want connectivity now

The need: action

when?

Get connected

with a good bandwidth

now

Get connected to the Internet with good bandwidth

now

Client

Internet

Intent Plane

Management Plane
Example – Parallel with RM-ODP model

**Entreprise View** what for? Why? Who? When?

**Information View** what is about?

**Computational View** How does each bit work?

**Engineering View** How does components work together?

Intent Plane

- I want connectivity
- The need action
- when?
- I want connectivity now
- Get connected
- with a good bandwidth
- now
- Get connected to the Internet with good bandwidth
- now
- In the next 5 secs
- Need public IP Xmbps min

Management Plane
**Example – Parallel with RM-ODP model**

- **Entreprise View** what for? Why? Who? When?
- **Information View** what is about?
- **Computational View** How does each bit work?
- **Engineering View** How does components work together?
- **Topology View** With what?

**Intent Plane**

- I want connectivity
- The need when?
- I want connectivity now
- action

With a good bandwidth

Get connected now

In the next 5 secs

Need public IP Xmbps min

**Management Plane**

- Client
- Internet

**Example Diagram**

- Get connected to the Internet with good bandwidth
- Get connected
- Need public IP Xmbps min
- Client
- Internet
Control/Data Planes

Legacy NMS

SDN Controller

Knowledge Plane

Collection

Societal behaviours

Weather events

Political decisions

... and more

Instrumentation – Cross-source data
Knowledge Plane

Multi-sources data Collection
Knowledge Plane

Multi-sources data Collection
Knowledge Plane

Intelligent Policy, Monitoring & Prediction
- Ressource Utilisation
- Fault Prediction
- Network Status Monitoring

Analytics
- Big Data Algorithm
- Machine Learning
- Reasoning

Data Collection
- Collection Adapter
- Data Management
- Southbound Interface

Multi-sources data Collection
Knowledge Plane

Intent Plane

Prediction

Intelligent Policy, Monitoring & Prediction
- Ressource Utilisation
- Fault Prediction
- Network Status Monitoring

Analytics
- Big Data Algorithm
- Machine Learning
- Reasoning

Data Collection
- Collection Adapter
- Data Management
- Southbound Interface

Multi-sources data Collection

Knowledge Plane
Management Plane

Intent Plane

Management Plane

Control/Data Planes
Management Plane

1- The Intent Plane sends the request to the management plane.
1- The Intent Plane sends the request to the management plane.

2- The network resources block manages the current resources.

Management Plane

Intent Plane

Control/Data Planes

Management Plane

Network Resources

Configuration & Resource Synchronisation

Resources Management

Report

Config

Network Element

Network Element
1- The Intent Plane sends the request to the management plane.

2- The network resources block manages the current resources.

3- The service manager using the management resource outputs try to answer the intent.
1- The Intent Plane sends the request to the management plane.

2- The network resources block manages the current resources.

3- The service manager using the management resource outputs try to answer the intent.
Two types of orchestration:

- Ressource orchestration
  - Governance of VNF instances sharing resources

- Service orchestration
  - Creation of E2E services composing different VNFs
Operational Loop

Intent Plane

Management Plane

Knowledge Plane

Control/Data Planes

Orchestration Plane
Operational Loop

Human Intent

Intent Plane

Knowledge Plane

Resources Mng

Management Plane

Orchestration Plane

Control/Data Planes
Operational Loop

Human Intent

Map it with Resilience/Qos

Knowledge Plane

Resources Mng

Orchestration Plane

Management Plane

Control/Data Planes
Operational Loop

Human Intent

Map it with Resilience/QoS

Intent Plane

Management Plane

Check against resources

Resources Mng

Knowledge Plane

Orchestration Plane

Control/Data Planes

Map it with Resilience/QoS

Check against resources
Operational Loop

- Human Intent
  - Map it with Resilience/Qos

- Intent Plane
- Knowledge Plane
- Resources Mng

- Orchestration Plane
- Management Plane
- Control/Data Planes

Inject new config
Operational Loop

Human Intent
Intent Plane
Map it with Resilience/Qos
Check against resources
Management Plane
Inject new config
Orchestration Plane

Collect Data
Knowledge Plane
Resources Mng
Control/Data Planes
Operational Loop

1. Human Intent
2. Map it with Resilience/Qos
3. Check against resources
4. Inject new config
5. Collect Data
6. DL/ML
7. Resources Mng
8. Orchestration Plane
9. Intent Plane
10. Management Plane
11. Control/Data Planes
Operational Loop

- Human Intent
  - Map it with Resilience/Qos
- Intent Plane
- Management Plane
  - Check against resources
- Resources Mng
  - Inject new config
  - DL/ML
- Collect Data
  - Knowledge Plane
  - System Self-Intent
  - Resources Mng
- Orchestration Plane
- Control/Data Planes
Technologies

- **mininet**, a popular network emulator using SDN and OpenFlow
  - Real-time
  - Full-stack
  - Uses OVS for networking
- **ONOS** OpenFlow controller used to control the network forwarding
- Modified *iperf* for traffic generation
- Service Manager Agent and Digital Asset Agents in *Scala*
- Anomaly Detection in *R*
Scenario

- Simple network topology
  - 4 hosts
  - 3 network nodes
- Observe a QoS feature of the network that can be improved
  - In this case, throughput
- Identify a possible improvement in this feature
- React to this change by modifying the network in some way
Scenario
Scenario
Scenario

Network Controller

Collect and inject new config

Topology

Network Device

Network Device

...
Scenario

Collect and inject new config
Collect data for each device
Scenario

Service Manager Agent

Digital Asset Agent

Digital Asset Agent

Network Controller

Collect and inject new config

Collect data for each device

Report to the Service Manager Agent

Topology

Network Device

Network Device

Mininet
Scenario

Service Manager Agent

Anomaly Detection Module

Digital Asset Agent

Digital Asset Agent

Network Controller

Collect and inject new config

Collect data for each device

Report to the Service Manager Agent

Topology

Network Device

Network Device

Mininet
Scenario

- **Service Manager Agent**
  - Self-Intent formulation
  - Anomaly Detection Module
  - Anomaly Detection Module

- **Digital Asset Agent**
  - Collect and inject new config
  - Collect data for each device

- **Network Controller**
  - Report to the Service Manager Agent
  - Collect and inject new config

- **Network Device**
  - Collect data for each device

- **Topology**
  - Mininet

**Flow**
- Inject new Intent
- Self-Intent formulation
- Anomaly Detection Module
- Service Manager Agent
- Collect and inject new config
- Network Device
- Collect data for each device
- Network Controller
- Report to the Service Manager Agent
- Network Device
Traffic

- Generate varying “UDP” traffic between the hosts to exercise the anomaly detection
  - From a low level of sustained throughput
  - Up to a high level of sustained throughput
Online Failure Detection
Online Failure Detection
Base Topology

H1

1Gbps links

H2

N3

H3

N1

N2

H4
User-Intent

Host-to-host intent
Host-to-host intent
User-Intent

Host-to-host intent
Busy link
Self-intent reaction
Self-intent reaction
Future Work

- The conditions in which a change is detected could be anything:
  - Device temperature
  - Costs changing
  - Multiple features
- Measure the impact of instrumentation (agents’ cost)
- The response could also be tailored to more complex intents!
- Can we *predict* when a failure may occur and react accordingly?
- Can we *recommend* a series of remediation strategies and have a human choose?