Intent Based Networking - the technology

Jeff Tantsura
Head of Networking Strategy @Apstra
Chair IETF Routing and RIFT working groups, IAB
Why IBN?

Source: Twitter
Why IBN?

Source: Twitter
IBN Landscape

Source: Twitter
Intent Based Networking Maturity Levels

**Level 0**
- Basic Automation

**Level 1**
- Single Source of Truth

**Level 2**
- Real-time Change Validation

**Level 3**
- Self-Operation

**IBN Landscape**

- Multi Vendor
- Vendor Lock-in

- Scripting Tools
- NMS Tools
- White Box
- Hardware Vendor-Specific Management

- Intent Based Networking

Intent Based Networking Maturity Levels
IBN standardization – just the beginning

Outlines number of fundamental IBNS building blocks and their relationship:

- **SSoT**: Single Source of Truth - A functional block in an IBN system that normalizes users' intent and serves as the single source of data (normalized intended state) for every consumer.
- **IBA**: Intent Based Analytics - Analytics that are defined and derived from user's intent and used to validate the intended state.
- **PDP**: Policy Decision Point – part of intent definition, technology agnostic.
- **PEP**: Policy Enforcement Point – technology/device aware (e.g. ACL or FW rule).

↔ Has recently been adopted as the rg document
IBN – why DC is a good starting point?

Ability to reason about Intent is a fundamental property of an IBNS!
Complex systems fail in mysterious ways ;-) 

CLOS topologies are extremely regular/uniform and mutate (expand) in a very predictable way. PoD structure provides clear boundaries.

https://engineering.fb.com/data-center-engineering/16-minipack/
IBN – why DC is a good starting point?

Telco WAN evolution

Day 0

Day 2

https://www.travelportland.com/article/portland-japanese-garden/

https://www.pinterest.com/pin/300730812396461/
IBN – why DC is a good starting point?

IBN in Telco WAN – is there a hope?

IETF TEAS ACTN framework is a step in the right direction

Key Idea: introducing SDN controller hierarchies, and make use of abstraction techniques to provide multi-vendor, multi-domain solution

CNC: customer w/o network knowledge; representing application and service, to be understood by operators.

MDSC: bridges user to network.
- Customer Mapping/Translation;
- Virtual Service Coordination;
- Multi-domain Coordination;
- Abstraction/Virtualization;

PNC: Configures the Network.
- Control/Manage the NE;
- Monitoring the topology;

https://datatracker.ietf.org/meeting/103/materials/slides-edu- sesskan-ietf-traffic-engineering-overview-01
IBN – why DC is a good starting point?

IBN in Telco WAN – is there a hope? What about 5G?

E2E Network Slicing is too complex, transport part of it is a perfect candidate for IBN

Each controller/orchestrator performs:
1. Automation (aka creation)
2. Monitoring and analytics
3. Optimization

3GPP

AN NSSMF (i.e. RAN Slice Controller)

Transport NSSMF (i.e. Transport Slice Controller)

CN NSSMF (i.e. Core Slice Controller)

AN NSSI

Transport NSSI

CN NSSI

AN Automation

Transport Automation

CN Automation

AN Monitoring

Transport Monitoring

CN Monitoring

AN Optimization

Transport Optimization

CN Optimization

Cloud

Cloud

Cloud

https://datatracker.ietf.org/meeting/105/materials/slides-105-teasessa-03-5gtransportsliceconnectivityinterface-00
IBN – why DC is a good starting point?

IBN in Telco WAN – 5G NS

E2E Network Slicing is too complex, transport part of it is a perfect candidate for IBN
IBN Design Philosophy

Networks managed as a whole system, not individual components

Successful networks are defined by the outcomes produced by the whole system

**Intent Based Networking**

is about “**what**” not “**how**”

More details in: draft-irtf-nmrg-ibn-concepts-definitions
IBN life cycle

**Intent**
- consumption

**Design**
- modeling

**Build**
- instantiation

**Deploy**
- validation (continuous)

**Validate**
- that the network still does it as intended

**Tell me what you want (your intent)**
- Let me model/build the logical intent model
- Let me instantiate your intent (networking)
- Let me validate that the network still does it as intended
Day in the IBNS life

**Business Logic/ 3rd party**

**GUI**  
**REST API**

**DATASTORE**  
SSoT

- Device data model
- Service data model

**Normalized data**
- Expected operational state
- Operational/ derived state
- config/telemetry

**Vendor Y render**  
**Vendor Z render**

**NETWORK**

**Continuous Validation**

- Compliance
- Connectivity
- Performance
- Security
Programmable Network and Interfaces

A network is programmable when the control and data planes provide an interface that allows the state of the network to be modified and monitored through a machine readable data-driven API's.
Programmable Network and Interfaces

Interfaces

**Capabilities**: Provides information what can be programmed or controlled. This included schema’s and metadata, or rather information about how the control structures are organized, and how information is presented by network devices to the controller.

**Inventory**: Provides information about what devices are installed where in the network, potentially including any information about physical connections.

**Topology**: Provides information about the state of links connecting network devices. This includes all the artifacts of the topology described.

**Telemetry**: Includes operational state, counters, and other information about the current network state. This includes but not limited by: resources used/available, queue depths, delay, jitter, etc...

Source: Navigating Network Complexity: Book by Jeff Tantsura and Russ White
The need for data normalization

Fruitcake

Source: food.com
The need for data normalization

Source: Twitter
Architectural Goals of IBN

Problems to be solved:

- Composition/decomposition @scale

- Dealing with changes:
  - Planned change – can I achieve desired (future) state while preserving original intent (meeting SLO’s)
  - Unplanned change – impact of the change, difference between intended and operational states, how to get to intended state (remediation/notification)
Architectural Goals of IBN

Problems to be solved:

- Closed loop validation:
  - continuously validate outcomes against the intent to ensure that the composition is working as intended
  - extract more knowledge by collecting less data thru IBA (Intent Based Analytics)
  - highly optimized SNR (signal to noise ratio) in analytics
Dealing With Scale?
Function composition (computer science)

From Wikipedia, the free encyclopedia

Not to be confused with object composition.

In computer science, function composition is an act or mechanism to combine simple functions to build more complicated ones. Like the usual composition of functions in mathematics, the result of each function is passed as the argument of the next, and the result of the last one is the result of the whole.

Programmers frequently apply functions to results of other functions, and almost all programming languages allow it. In some cases, the composition of functions is interesting as a function in its own right, to be used later. Such a function can always be defined but languages with first-class functions make it easier.

The ability to easily compose functions encourages factoring (breaking apart) functions for maintainability and code reuse. More generally, big systems might be built by composing whole programs.
Why model a graph?

- Networks are intuitively the connected set of nodes and relationships
- As network requirements change the model can be easily extended
- Efficiently run queries that were not anticipated at model design time

*Hint*: you will not know all the queries at model definition time
Intent-> Graph composition
Function composition
Resulting Model
Query: Links that carry “A2” traffic
Decomposition (computer science)

Decomposition in computer science, also known as factoring, is breaking a complex problem or system into parts that are easier to conceive, understand, program, and maintain.
DECOMPOSITION

Great, Big Problem

Break down into smaller, logical parts

Part 1 of problem

Further break down into even smaller, logical parts

Sub-problem 1

Sub-problem 2

Part 2 of problem

Further break down into even smaller, logical parts

Sub-problem 3

Sub-problem 4
Decomposition: walking the graph
Query:

```camel
match(
  node("system", role="spine")
  .out()
  .node("interface")
  .out()
  .node("link")
  .in()
  .node("interface")
  .in()
  .node("system", role="leaf")
)
```
Query:

```
match(
    node("system", role="spine")
      .out()
    .node("interface")
      .out()
    .node("link")
      .in()
    .node("interface")
      .in()
    .node("system", role="leaf")
)
```

Steps

start 0 1 2 3 4 5 6 7 8

<FindNodeAction type="system" role="spine">
Query:
```
match(
    node("system", role="spine")
    .out()
    .node("interface")
    .out()
    .node("link")
    .in()
    .node("interface")
    .in()
    .node("system", role="leaf")
)
```

Execute Query
Close

Query:
```
match( node("system", role="spine") .out()
    .node("interface") .out() .node("link") .in()
    .node("interface") .in() .node("system", role="leaf") )
```

Steps
```
start 0 1 2 3 4 5 6 7 8
```

<RelationshipTargetAction index> 1 type=interface

Paths (12)
Query:

```cypher
match(
  node("system", role="spine")
  .out()
  .node("interface")
  .out()
  .node("link")
  .in()
  .node("interface")
  .in()
  .node("system", role="leaf")
)
```

Steps

```
<RelationshipSourceAction index=7 type="system" role="leaf">
```

Paths (6)
Query:
```
match(
    node("system", role="spine"), out()
    .node("interface"), out()
    .node("link"), in()
    .node("interface"), in()
    .node("system", role="leaf")
)
```

Steps
```
<RelationshipSourceAction index=7 type="system" role="leaf">
```

Paths (8)
Query:

```cypher
match(
    node("system", role="spine")
    .out()
    .node("interface")
    .out()
    .node("link")
    .in()
    .node("interface")
    .in()
    .node("system", role="leaf")
)
```

Steps

```
<relationship source="action index">7 type="system"
role="leaf"
```

Paths (8)
Query:

```
match {
  .node("system", role="spine") .out()
  .node("interface") .out()
  .node("link") .in()
  .node("interface") .in()
  .node("system", role="leaf")
}
```

Steps

```
<RelationshipSourceAction> index=7 type="system" role="leaf"
```

Paths (6)
Query:
```
match(
  node("system", role="spine")
  .out()
  .node("interface")
  .out()
  .node("link")
  .in()
  .node("interface")
  .in()
  .node("system", role="leaf")
)
```

Execute Query
Close

Query:
```
match (node("system", role="spine"), out())
  .node("interface") .out() .node("link") .in()
  .node("interface") .in() .node("system", role="leaf")
```

Steps
```
start 0 1 2 3 4 5 6 7 8
```

Paths (8)
Intent Based Analytics

Extract more knowledge by collecting less data (orders of magnitude less)
Was I looking for something?
Gathering high def telemetry
For all my leaf1 interfaces
For all my leafs
So that I have insight

Question: Is my fabric ECMP imbalanced?
IBA : ECMP fabric health (load sharing across fabric links)
Would ML be helpful?

Holistic view:
Collect “all” counters all the time.

Macroscopic view:
Catch interesting state changes. Dim.-Redux, Cluster.

Microscopic view:
Choose counters which best describe the state change.

Source: BRKOPS-3825, Frank Brockners
IBA – context aware analytics

Declaratively specified, definition is de-coupled from instantiation
Once specified, is in constant sync with intent
Extracts knowledge out of the raw telemetry – context drives the content
New telemetry is “wired-in”
Conclusion

- Basic automation, while hot topic - is the first and easiest step in the IBN journey

- Single source of truth is mandatory for an IBN system to be able to reason about any change

- Day 2 operations @scale:
  - context aware continues validation
  - dealing with changes
  - configuration drift
  - remediation

is the most complicated area of technologies to deal with!
Thank You!

www.apstra.com

@ApstraInc
https://www.linkedin.com/company/apstra
https://www.facebook.com/apstrainc/