none
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

S. Yan
Huawei
P. Martinez-Julia
NICT/Japan
A. Cabellos-Aparicio
Technical University of Catalonia

October 30, 2017

A General Considerations of Intelligence Driven Network
draft-yan-idn-consideration-00

Abstract

This document aims to pinpoint the work scope of Intelligence Driven Network (IDN) and mine the potential standardization work. Firstly, the problems and new requirements for the existing methods are analyzed. Numbers of high value use-cases are proposed as examples to instantiate them. A benchmark framework design is proposed, which is important during the machine learning and inference process. Finally, a reference model of IDN is proposed, based on which the potential standardization work is analyzed.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2018.

Table of Contents

1. Introduction

Recently, AI technology has made a great achievement and become more and more popular. The combination of AI and network is also a hot topic. The concept of Intelligence Driven Network (IDN) has been proposed. This concept is intended to describe the schemes that introducing AI into network and provide new solutions for the current and future network problems. There has been quite a lot of discussions about the AI application in the network in both academic and industrial area. However, the detail works, especially the potential standard points are still not clear.

In this document, we want to summerize the valuable content in the idnet maillist and make clear about the following.

- What are the requirements? In network area, what problems need AI to solve? It always makes misunderstanding that AI is almighty. But it is factual that AI has both advantages and disadvantages. The work scope and scenarios, which AI may be useful and perform well, will be discussed and analyzed.

- What are the gap when combining AI and network? The modern AI algorithms are proposed by image processing area but not network. Most of the algorithms cannot be migrated and used directly. Take the data format as an example. The input and output of the AI algorithm may be just numerical matrix or vector. The network data are not entirely formatted and regular. They need to be translated or converted before and after the algorithm. The gaps, like the data format, data orchestration and etc., will be analyzed.

- What are the potential and new standard points? The intruduction of AI will bring new requirements for the current network. For example, the AI engine may need high frequency and high accuracy data to feed. Moreover, these data needs to be captured and transmitted in real-time and continuously. What improvements should be accomplished for the existing protocols? Whether there are new protocol requirements? What communication processes are universal and what kinds of data format that can be utilized in most of the scenarios?

This document aims to become the blueprint for the future work. The structure is organized as following. Section 2 describes the work scope of idnet and summerize the use cases. Section 3 indicates the analysis of measurement and data format. Section 4 discusses about the benchmark of data. Section 5 abstracts the IDN architecture and gives a brief analysis of potential standard points. Section 6

points out the new security challenge which AI brings to the network. Section 7 to 9 are IANA, Acknowledgements and References.

TBD

2. Scope and use cases

TBD

2.1. Scope

A general description about what should be focused during the IETF work and what should not. Clarify the work boundary. TBD

2.2. High Value Use Cases

There are numbers of use cases, which have been discussed in the idnet mail list. Describe the scenarios that may be useful and valuable. A details analysis may be helpful for the data and protocol design.

2.2.1. Traffic Prediction

Collect the history traffic data and external data which may influence the traffic. Predict the traffic in short/long/specific term. Avoid the congestion or risk in previously.

The process, data format and message needs are:

Process: 1. Data collection (e.g. traffic sample of physical/logical port); 2. Training Model; 3. Real-time data capture and input; 4. Predication output; 5. Fix error and go back to 3.

Data Format:

Time : [Start, End, Unit, Number of Value, Sampling Period]

Position: [Device ID, Port ID]

Direction: IN / OUT

Route : [R1, R2, ..., RN] (might be useful for some scenarios)

Service : [Service ID, Priority, ...] (Not clear how to use it but seems useful)

Traffic: [T0, T1, T2, ..., TN]

Message :

Request: ask for the data

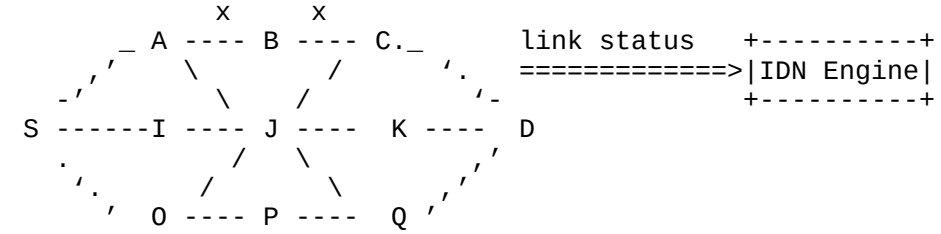
Reply: Data

Notice: For notification or others

Policy: Control policy

2.2.2. QoS management

It is worthy to predict the traffic change for avoiding the congestion and ensuring QoS. As the following figure shown, the AI system continuously collects link status data from the network. This AI system is responsible for two things. One is monitoring and predicting the traffic on each link and the other one is calculating the usable route for any pair of nodes according to the prediction and current link status. Assume that there is a VPN named VPN_S_D from node S to D which pass through S-A-B-C-D. According to the prediction, there will be a huge traffic flow from node A to C in the future 10 min. The traffic will increase the end-to-end delay from S to D so that the QoS will not be ensured.



There are at least two solutions. one is modifying the object's configuration to avoid the potential congestion. For example, we modify the VPN_S_D route from S-A-B-C-D to S-I-J-K-D. The other one is restricting non-object's transmission so that to protect the object's QoS. For example, we increase the reserved bandwidth of VPN_S_D or modify the route of non-object flows from S-A-B-C-D to S-I-J-K-D therefore most of the traffic will not affect VPN_S_D.

Here we may have some challenges. Challenge 1 is the AI prediction and autonomic decision should be a quick response. The whole process must be finished before the congestion happens meanwhile the AI system is meaningless. The question is how to implement such quick response? Challenge 2 is whether there is existing protocols which can support high frequency measurement? Because AI system needs to be fed with continuous link status data. And the real-time data need

to be captured frequently otherwise the route change will be worthless. I think the protocols that support high frequency measurement and data collection may become one of our focus point.

The process, data format and message needs are:

Process: 1. Data capture (e.g. traffic sample of physical/logical port); 2. Training Model; 3. Real-time data capture and input; 4. Output percentages; 5. Fix error and go back to 3.

Data Format:

Time : [Timestamp, Value type (Delay/Packet Loss/...), Unit, Number of Value, Sampling Period]

Position: [Link ID, Device ID]

Value: [V0, V1, V2, ..., VN]

Message :

Request: ask for the data

Reply: Data

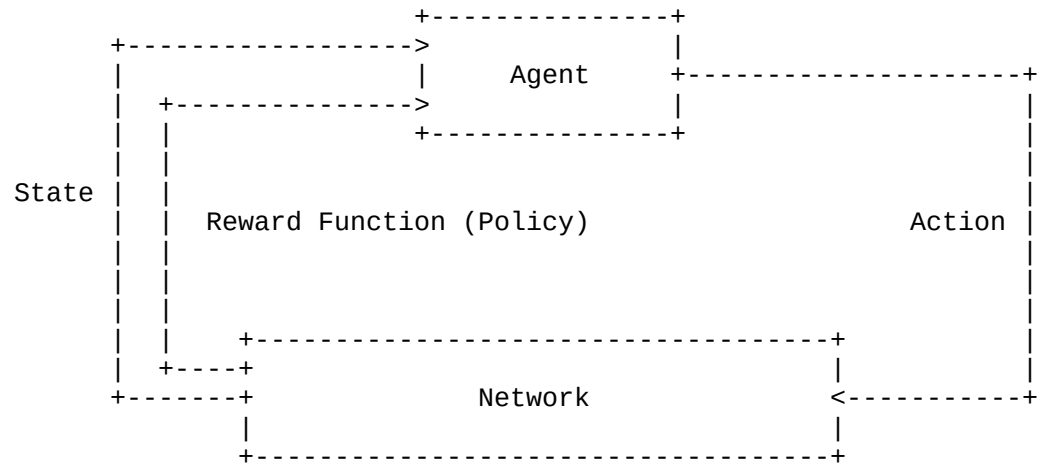
Notice: For notification or others

Policy: Control policy

2.2.3. Deep Reinforcement-Learning Control of the Network

Recently important breakthroughs have been achieved in the area Deep-Reinforcement Learning (DRL) [REF1] architectures where agents can be trained online to operate complex environments and achieve quasi-optimal configurations. In this context, a DRL can be used to control the routing of the network and achieve the target policy set by the administrators (e.g., [REF2, REF3, REF4]).

The following figure describes a common architecture of a DRL operating a network. The agent acts upon the network (action) by changing the configuration, this results in the network changing its fundamental state (e.g, different per-link utilization and a different traffic load). Finally, the reward function is defined by the operator and represents the target performance (e.g., load-balance the traffic in the network). The agent will learn how to act upon the network to maximize the expected reward function.



The main operational advantages of DRL agents with respect to existing optimization techniques are:

1. DRL are able to learn and generalize from past experience to provide solutions to unseen scenarios. This is not possible using existing optimization techniques that do not learn from the past.

2. Once trained, either offline or online, DRL agents can optimize in one single step. On the contrary, existing optimization techniques require to run iteratively each time a new scenario is found, for instance when a link goes down or the traffic changes in a significant way. It is worth noting that a common practice is to run such techniques in advance of common scenarios and store their resulting configurations, however it is very complex to consider all the potential scenarios.

3. DRL agents see the network as a black-box and do no need any prior assumption about the system. However heuristics, very commonly used in optimization strategies, are tailored for the problem they are trying to optimize. However, an operator only needs to change the reward function to implement a different target network policy.

In what follows we describe the process, data format and messages needed assuming a DRL agent that seeks to load-balance the traffic of the network that is, to minimize the maximum loaded link. This is a very common optimization strategy.

Process: 1.- Act upon the network by changing the routing configuration, for instance using a standard mechanism. 2.- Receive

the state of the network, this is the per-link delay and the current traffic load. 3.- Compute the reward function as a function of the state. 4.- Deep Reinforcement Learning training. 5.- Go back to step 1.

Data Format

(state) Per-Link Utilization: [link id, utilization, averaging time]

(action) Change on the routing configuration. This can be done through the SDN controller and/or other standard mechanisms.

(reward) This is an algorithm that has as input the state and as output a value that represents how close we are to the target policy set by the operator. More about this can be found in the next section.

Messages:

State: Measure the per-link utilization

Action: Change the routing configuration

2.2.3.1. The Reward Function as the Network Policy

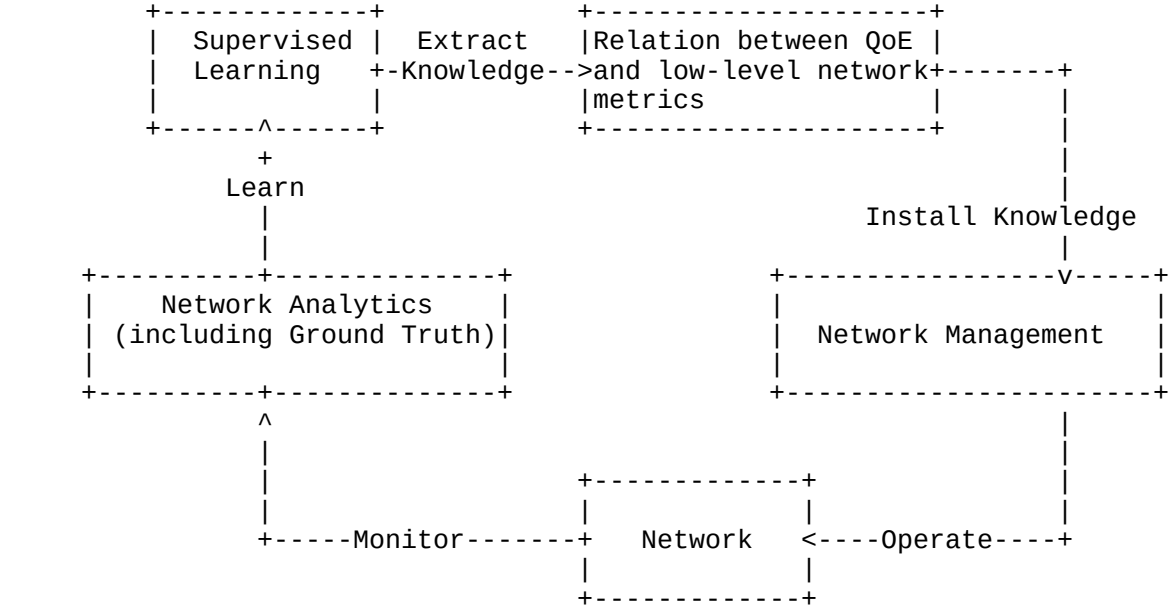
The agent seek to maximize the expected reward function and it represents the target policy that the agent will aim to achieve and configure on the network. In this context the reward function is the mathematical representation of the target network policy. However, the entire architecture includes a set of different pieces that may come from different vendors but must interoperate, the pieces are: the agent itself, the reward function and the state. This requires the following standardization efforts:

1. The reward function and its translation from the human-readable target network policy. The operators may want to use different vendor DRL agents that need to understand the reward function. Please note that the reward function depends on the representation of the state.

2. The state includes monitoring information about the network, such as the per-link utilization or the traffic load. Since the state is an input of the agent and is used in the reward function, there is a need for standard representation so that the different pieces can interoperate.

2.2.4. QoE Management via Supervised Learning

Networks can measure low-level metrics, such as delay, jitter and losses. However users perceive the performance of the network based on QoE metrics, such as Mean Opinion Scores. Unfortunately, QoE metrics cannot be typically directly measured over the wire and as such, need the subjective views of the users. The challenge is then to operate the network based on low-level metrics while fulfilling non-measurable QoE metrics. One of the main reason behind this challenge is that the relationship between the low-level and the QoE metrics are very complex, i.e. multi-dimensional and non-lineal.



For this a well-established technique (e.g., see [REF5] and the references therein) is to follow the architecture depicted in the following figure. First the network low-level metrics are measured using telemetry, this information is stored in the Network Analytics platform. In addition to this users and or applications are polled to obtain QoE metrics of the network. The data-set containing both the low-level metrics and the QoE metrics is considered the ground truth.

By means of supervised learning (e.g., deep neural networks) we aim to learn the relation between the low-level and the QoE metrics. As an example we aim to learn the relation between the amounts of losses in different wireless links, the SNR and the utilization with the perceived MoS. Typically it has been shown that such relationship is

non-lineal and multi-dimensional and as such, can be understood by a neural network. This relationship is the knowledge that we extract from the ground truth and it is used by the Network Management (NM) module. By means of this knowledge, the NM can understand how to operate the network based on low-level metrics (e.g., keep losses below a certain threshold) to fulfill QoE requirements.

2.2.5. TBD

3. Measurement and Data Format

TBD

3.1. Measurement Tools and Methods

The modern AI algorithms are mostly based on data-driven, which means that the AI engine needs quite plenty of data to feed and upgrade. In other words, higher frequency and accuracy data is required. The high scalability requirement needs distributed measurement tools to provide such abilities. The traditional methods and improvements may hardly support.

Firstly, the current measurement methods mostly orient to the service. For example, the voice service requires the end to end delay and jitter in a low level. Besides that, the AI engine may need more data from both network and other sources. For example, the QoE and identity information may influence the AI engine to make different decisions. The current measurement tools and data model cannot support this ability. Thus, the potential usable tools and methods, such as high frequency, high precision, new KPIs and so on, may need to develop.

Secondly, the current measurement methods mostly cannot support high frequency measurement. Even though it can, the data feedback scheme is commonly closed. The word "closed" means that the measured data is commonly sent to the device which launches the measure action rather than the data demander (AI Engine). The future measurement tools require more programmability, especially in the data feedback scheme.

TBD.

3.2. Data Format Analysis

There is huge gap between the current network data and algorithm data. The network data, such as IP address, delay, link utilization and etc., is mostly semantic. It means that each data actually describe a specific physical or logical entity. For example, one IP

address means a certain location or a certain host in the network. However, the input and output data of an algorithm is usually non-semantic, which means it is not responding to a specific concept/action/device that can be found in the network. This depends on the fundamental design of AI algorithm and is hardly changed in the short term.

Another issue is that the AI engine potentially needs to obtain data from external sources. For the data that can be provided one-off, it is easily solved according to the application. For the data that needs to be provided continuously (e.g. the real-time external data), it is required to define the data format that satisfy the algorithm. Similarly, the output of algorithm may need to be translated into specific format that the next step devices can run and execute. Otherwise, it is hard to build up the full autonomic close loop of the network management. In other words, the data aggregation process is important and it is valuable to build the bridge between the network data and algorithm data.

TBD.

4. Benchmarking Framework

A standard benchmarking framework is required to assess the quality of an AI mechanism when it is used to resolve a specific problem in the network management and control area. It comprises a reference set of procedures, methods, models, and boundary values that *must* be enforced to the benchmarked mechanism, so that its operation can be comparable to other mechanisms and users can easily understand what to expect from each one.

Moreover, both the metrics included as a reference within the benchmarking framework and the results obtained from its application to a new mechanism must follow a standard format. Therefore, the standard formats must be enforced to all data, either being introduced to the benchmarking application or system (consumed), or obtained from its application (produced).

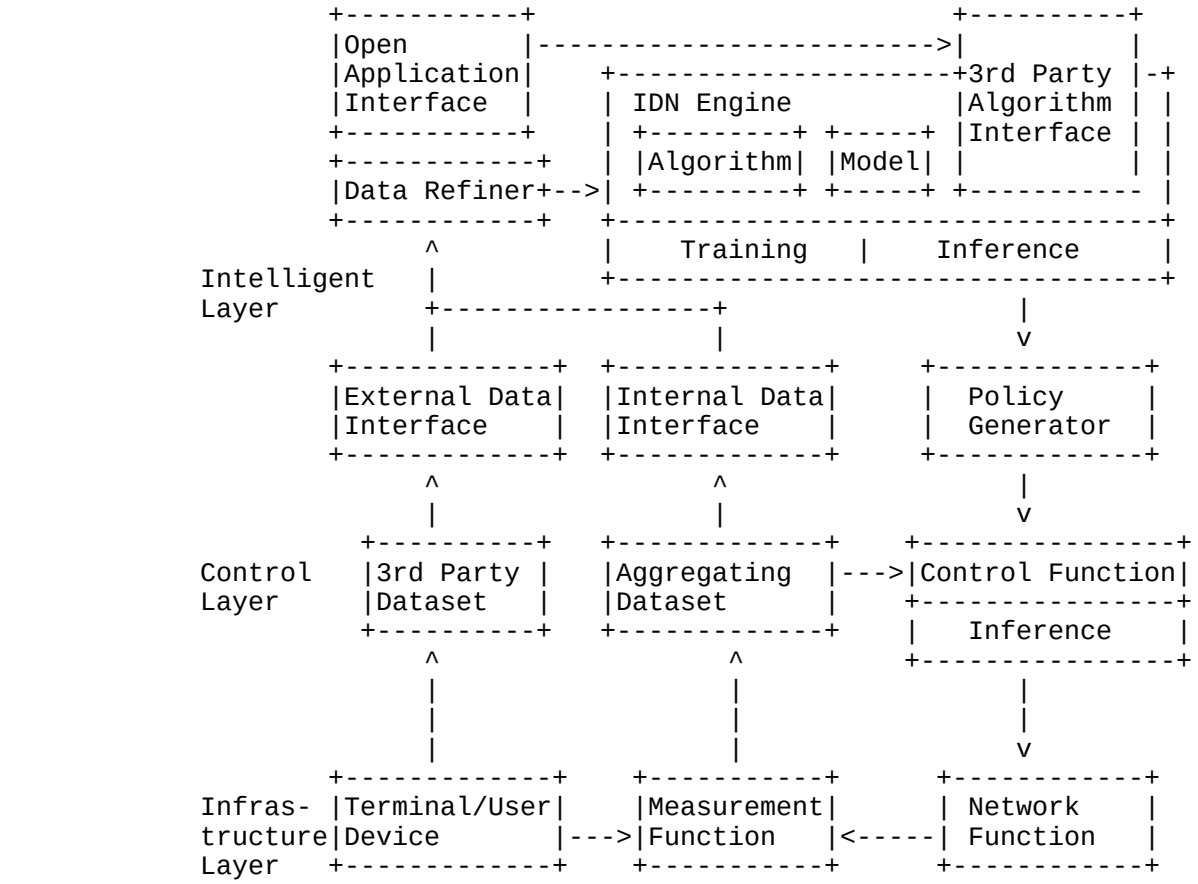
A common and decentralized "data market" can (and would) arise from the inclusion, dependency, and the general relation of all data, considering it is represented using the same concepts (ontology) and the standard format mentioned here. As a reference, it is worth to mention that a similar approach has been already applied to genome and protein data to build standardized and easily transferable data banks [PMJ1][PMJ2] [PMJ3], and they have demonstrated to be key enablers in their respective work areas.

The initial scope of input/output data would be the datasets, but also the new knowledge items that are stated as a result of applying the benchmarking procedures defined by the framework, which can be collected together to build a database of benchmark results, or just contrasted with other existing entries in the database to know the position of the solution just evaluated. This increases the usefulness of IDNET.

5. References Model and Potential Standardization Points

5.1. References Model

A three layers reference model of IDN has been proposed as follow. This architecture can cover, explain and support most of the current use cases and scenarios.



The under layer is Infrastructure layer, which contains network function, measurement function and terminal/user device. The network function stands for the traditional routers, switches and other network devices, which are responsible for constructing the network foundations and forwarding data. The Measurement function stands for devices that can collect information from the network and various devices. A popular option are probe system, which is deployed distributed among the network. Besides that, some of the network devices integrate the measure function and play two roles. The information may involve but not limited the content listed in following table. The Terminal/User Device stands for the device that produces and consumes data, which may include PC, smart phone, datacenter, content storage server, cloud and etc. Some of the data produced by terminal/user devices is measurable. This type of data will be captured by the measurement function. Other types of data that cannot be measured directly by network measurement functions is represented as 3rd party datasets, which hopefully can be utilized in the future via 3rd party integration at the intelligence layer.

Type	Content
Network Data	Delay, Jitter, Packet Lose Rate, Link Utilization, ...
Device Data	Device Configuration, VPN Configuration, Slicing Configuration, ...
User Data	QoE Feedback, User Information, ...
Data Packet	Packet Sample, Packet Character, ...
Other Type	TBD

The middle layer is Control Layer, which contains Control Function, Dataset Aggregation (Function) and 3rd Party Dataset. The control function stands for entities that can control, configure and operate devices, especially network devices. In SDN, controller and orchestrator are control functions. Classical network devices such as routers integrate the forwarding and control functions (although as of today not with many instances of intelligent control functions). Classical routers therefore include functions from two layers. We foresee that the control function will most likely only perform intelligent inference, but not learn. For example, to execute neural networks, but do not train them. This is only an assumption at this time though and may prove to be wrong in the future when training becomes something easier defined into the control layer.

The aggregated dataset function owns the ability to gather and tidy the data. The database or database cluster is the typical example. Some of the control devices, such as SDN controller, integrate this function. Distributed instances aggregate data have also been defined. The network data can be directly sent back to the control function in support of network policies. For example, the controller can adjust the flow table according to the local cache which collects the network data periodically from the devices in its controlled area. The 3rd party dataset involves the data that may be provided by all kinds of applications or services. For example, the content provider may own social contact data and the map service provider may own the geographic data. This information does not belong to the network but could be very helpful for intelligent analytics and decision making in the network - which is why we device in the architecture the ability to communicate it between 3rd parties and the network.

The high layer, which is also the main body of IDN, is the Intelligence Layer. This layer is commonly deployed in the datacenter, or large scale computing centre that can support massive storage and computing resources. To the south direction, there are two interfaces which provides external data (3rd party data oriented) and internal data (network data oriented) access. We define a data refiner component to emphasize the need to adopt format and structure of various types of collected information to the needs of the IDN Engine.

The core of the IDN Engine are algorithm and model. The IDN Engine can be built based on the result of the large body of research and platform development work that already exists (albeit mostly developed for and deployed with non-network data). The platform should be agile extensible for future services, therefore we define a 3rd party Algorithm Interface to provide an adaptive developing ability. The user (or a 3rd party) may develop his/her own algorithms and upload then onto the IDN Engine via a northbound Open Application Interface. Additional Northbound Open Application interfaces can also be used to connect other software platforms to the IDN Engine to create a cooperation between multiple systems (not shown).

The output of IDN Engine is transmitted to the Policy Generator. Since the policy language might be machine readable or unreadable, the Policy Generator is responsible for generating the executable commands and connect to the control devices. This process refers to the interactions of northbound interface of control devices - which is what often gets standardized. Therefore, some of the potential standardization points will be mentioned in the following.

5.2. Measurement

In IDN, the intelligent system (or database) needs frequent and repeat measurement to obtain the link information. A fast measure and feedback protocol is needed to meet the requirement of measurement and data collecting. It may be based on SNMP or an absolutely new protocol. The intelligent system needs massive data to feed and support to formulate the policy and decision. Therefore, the measurement must be satisfy the data requirement of IDN. Firstly, there may be higher-level requirement for the existing measuring technology. The high timeliness is one of the potential point. The IDN's control function needs accurate, global and highly real-time network data support. The current measure technology can only satisfy at least two characters of the three. Secondly, the IDN may need more kinds of data type to measure. Not only the delay, jitter and packet loss rate, but also the link utilization and other necessary parameters.

5.3. Data representation, transport and aggregation

The data representation is significant. Most of the current AI algorithms were born in the pattern recognition area, especially the image processing. The advantage of these algorithms is that they are very good at dealing with complex problems, especially mining and modeling the hidden relationship among the non-semantic data. One of the disadvantages is that almost all the algorithms require the training data has a high concordance. Fortunately, the image file instinctively owns this character. All the images can be expressed as uniform binary vectors or can be easily transformed into uniform format. But this condition is hardly satisfied in network area.

A uniform data format is required, which can implement the justification, correlation and affiliation of the data. Which may obtain the best performance of AI algorithm to mine the valid pattern hidden in the data. Since the intelligent system is data-driven, and the data resources are from different kind of vendors and device types, the data representation SHALL be consistent so that the intelligent system could merge the data and do the analysis/learning. Also, the data collection interface might also need to be standardized so that the interface is able to get the data the intelligent system needs.

Moreover, it is significant to standard the policy representation. Since there may multiply SDN controller system, a readable and uniform policy representation is valuable to improve the policy deploying efficiency and simplify the communication between controllers on the East-West direction.

5.4. Legacy Device Route control

Similar with IPv4/IPv6 transition, the IDN potentially faces to the legacy problem, which means that the new devices and functions will co-work with the legacy devices. Therefore, it is potentially required to design the control protocols to solve the transition problems.

5.5. TBD

TBD

6. Security Considerations

When security relevant decisions are made based on the use of intelligent analytics or automated intelligent decision making, care must be taken to understand the new security challenges. When for example more intelligent decisions are enabled through the collection of ever more data, it needs to be analyzed how that potentially enables attackers to easier feed data that derails the intelligent system ability to distinguish good from bad behavior.

TBD

7. IANA Considerations

There is no IANA action required by this document.

8. Acknowledgements

TBD

9. References

9.1. Normative References

[ISO_IEC10589] "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO/IEC 10589:2002, Second Edition.", Nov 2002.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5301] McPherson, D. and N. Shen, "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, DOI 10.17487/RFC5301, October 2008, <https://www.rfc-editor.org/info/rfc5301>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

9.2. Informative References

[PMJ1] , <https://www.ncbi.nlm.nih.gov/genome/>.

[PMJ2] , <https://www.ncbi.nlm.nih.gov/genbank/>.

[PMJ3] , <https://www.rcsb.org/pdb/home/home.do>.

[REF1] "Human-level control through deep reinforcement learning. Nature, 518(7540), pp.529-533.", 2015.

[REF2] "A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization. arXiv preprint arXiv:1709.07080.", September 2017.

[REF3] "A roadmap for traffic engineering in SDN-OpenFlow networks. Computer Networks, 71(C):1–30", October 2014.

[REF4] "Packet routing in dynamically changing networks: A reinforcement learning approach. In Advances in neural information processing systems, pages 671–678,", 1994.

[REF5] "A machine learning approach to classifying YouTube QoE based on encrypted network traffic. Multimedia Tools and Applications", January 2017.

Authors' Addresses

Shen Yan
Huawei
Beiqing
Beijing, Haidian 100095
China

Email: yanshen@huawei.com

Pedro Martinez-Julia
NICT/Japan

Email: pedro@nict.go.jp

Albert Cabellos-Aparicio
Technical University of Catalonia

Email: albert.cabellos@gmail.com