ETSI GR NGP 006 V1.1.1 (2018-06)

Next Generation Protocols (NGP); Intelligence-Defined Network (IDN)

Disclaimer

The present document has been produced and approved by the Next Generation Protocols (NGP) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGR/NGP-006

Keywords
framework, next generation protocol

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Next Generation Protocols (NGP).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The scope of the present document is to specify the self-organizing control and management planes for the Next Generation Protocols (NGP), Industry Specific Group (ISG).

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] https://techcrunch.com/2016/03/24/microsoft-silences-its-new-a-i-bot-tay-after-twitter-users-teach-it-racism/.

[i.2] https://www.thesun.co.uk/tech/4141624/facebook-robots-speak-in-their-own-language/.

[i.3] Reed S, Akata Z, Yan X, et al.: "Generative adversarial text to image synthesis", in ICML 2016.

[i.4] Oord A, Dieleman S, Zen H, et al.: "Wavenet: A generative model for raw audio", arXiv:1609.03499, 2016.

[i.5] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton: "Deep learning", in Nature 521.7553 (2015): 436-444.

[i.6] Kingma D P, Welling M.: "Auto-encoding variational bayes", in ICLR 2014.

[i.7] Goodfellow, Ian, et al.: "Generative adversarial nets", in NIPS 2014.

[i.8] Cisco White Paper.

NOTE: Available at https://www.cisco.com/c/en/us/products/collateral/routers/wan-automation-engine/white_paper_c11-728552.html.

[i.9] https://arxiv.org/abs/1701.07274.

[i.10] ETSI TR 121 905: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

[i.11] ETSI TS 136 401: "LTE; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture description (3GPP TS 36.401)".

3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TR 121 905 [i.10] and the following apply to scenarios that include mobile network architectures:

3GPP™	3rd Generation Participation Project
AI	Artificial Intelligence
DHCP	Dynamic Host Configuration Protocol
E-W	East and West (direction)
IDN	Intelligence-Defined Network
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISG	Industry Specific Group
ML	Machine Learning
NE	Network Element
NGP	Next Generation Protocols
NMS	Network Management System
N-S	North and South (direction)
OAM	Operation And Management
OSPF	Open Shortest Path First
QoE	Quality of Experience

4 Overview

The Next Generation Protocols (NGP), ISG aims to review the future landscape of Internet Protocols, identify and document future requirements and trigger follow up activities to drive a vision of a considerably more efficient Internet that is far more attentive to user demand and more responsive whether towards humans, machines or things.

A measure of the success of NGP would be to remove historic sub-optimised IP protocol stacks and allow all next generation networks to inter-work in a way that accelerates a post-2020 connected world unencumbered by past developments.

The NGP ISG is foreseen as having a transitional nature that is a vehicle for the 5G community and other related communications markets to first gather their thoughts together and prepare the case for the Internet community's engagement in a complementary and synchronised modernisation effort.

Therefore NGP ISG aims to stimulate closer cooperation over standardisation efforts for generational changes in communications and networking technology.

The present document focuses on proposing a new Intelligence-Defined Network (IDN) architecture and a gap analysis of current architectures. The intelligence technologies can learn from historical data, and make predictions or decisions, rather than following strictly predetermined rules. On one hand, the IDN can dynamically adapt to a changing situation and enhance its own intelligence with by learning from new data. On the other hand, IDN can also aim at supporting human-based decision by pre-processing data and rendering insights to users through advanced user interfaces and visualisations. The integration with various network infrastructures, such as SDN, NFV&MANO, intelligence router, traditional router, is in the scope of the present document.

5 Background

5.1 Continuous Evolution of Network

The development of network is continuously evolving process. In different stages, the network faces to various and different complexity problems. Therefore, the operating and management methodologies are also various.

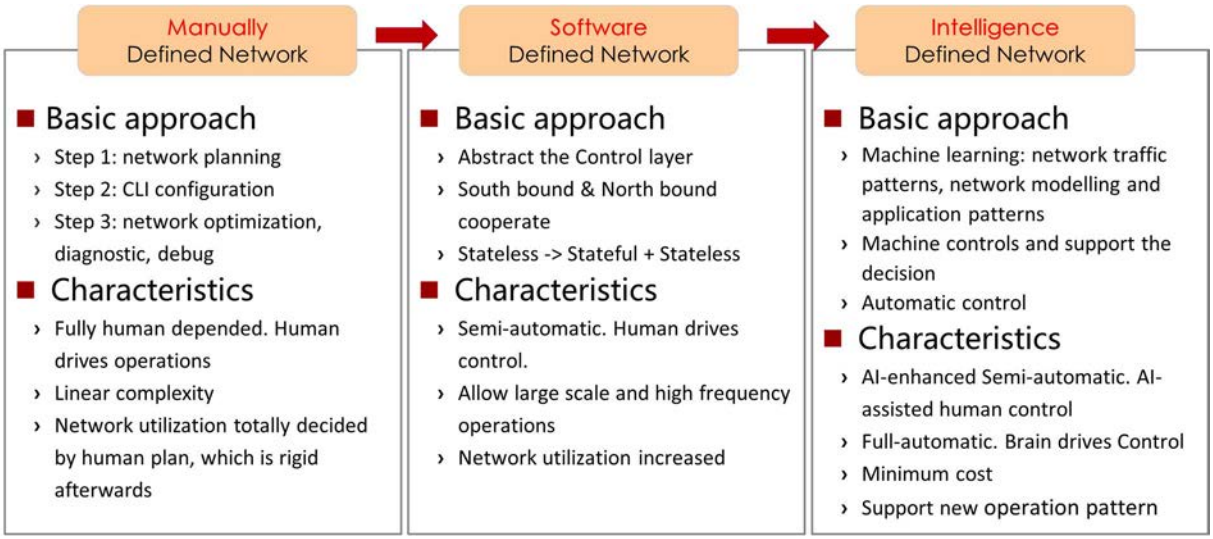

Figure 1: Three Stages Development

As Figure 1 shows, early networks are referred as Manually Defined Network. In such networks, the OAM basic approach is network planning, CLI configuration and network optimization. All the operations are fully human driven. Since the administrator needed to configure and control each device individually, the complexity and the cost of OAM was very high.

Along with the development, the scale of network and service became larger and larger. The OAM requirement has also increased. Due to the high degree of required control degree of requirement, a virtual control layer was developed. This layer supports batch operation of low layer devices, which improves the efficiency significantly. Because of the divide of control layer and forwarding layer, the configuration and controlling operation is implemented by south and north-bound cooperation. Southbound typically uses Netconf/YANG, OpenFlow, etc. to configure the network forwarding, policies, etc. Northbound abstracts the functionalities for application requirements, thus deriving the forwarding table and policies, etc. With this paradigm, the entire network was transformed to become semi-automatic. Many operations can be executed automatically and the administrator is only responsible for decision making.

Currently, the network undergoing a new transformation towards Intelligence-Defined Network (IDN). Since the network problems evolve to more complex, the traditional human decision-making can hardly support the requirements. Therefore, the AI methods, which can help for decision making and analysis, are introduced to solve OAM problems. The core of IDN is machine learning algorithms and models. The network, traffic and application patterns can be modelled by AI methods via learning from the existing data and experiences. It is expected as a full-autonomic system that can make decision itself, especially in the common and repeat events that do not need human to judge. This will decrease the OAM cost hugely in the future.

5.2 Functional and Systemic Requirement

IDN is seen as the next form of network evolution. Comparing with the current state, there are new requirements that declare the essential improvements of new approach.

The first part is functional requirements, which means the IDN approach should own the functions that the previous network approaches do not have. For IDN, some of the potential functional requirements are following:

- Real-time assessment. The IDN approach should provide a consolidated view of the current network status including traffic and running applications by providing aggregated and condensed insights.
- Prediction or inference. The IDN approach should have the ability to predict / infer the oncoming trend of network in multiple dimensions, such as inferring the QoS parameter according to the traffic matrix. This ability will support the intelligent system to implement proactive operations.
- Autonomic decision making. The network will not only execute the policies which produced by administrator but also autonomously make decision. This is one of the most important reasons that why AI technology is introduced into network.

- Verification. IDN is not only in charge of taking decisions but also (1) verifies that its own decision are properly applied and results in the expected states and (2) can be leveraged to verify that policies derived from multiple entities (concurrent IDN algorithms and even users) are coherent.

- Dynamical configuring ability. For stability consideration, typically operators try to minimize changes on the devices. However, one of the purposes of introducing AI technology is to modify the configuration so that adapt for the variation of network traffic and state.

The second part is systemic requirements, which means the IDN approach should own the system level abilities in the low layer (or say primordially) that the previous network approaches do not have or cannot easily complement. For IDN, the potential systemic requirements are following but not limited.

- Inherent data collection and orchestration. The current measure method is driven by external command. Namely, all the network data is a response (or feedback) of a specific command. The network devices do not widely support the actively data upload functions. This leads to at least two problems. The first one is the cost. When the measured data volume is large, there is nearly half of the signal messages and transfer time are wasted because one data feedback should be potentially triggered by one measure signal. This external trigger mechanism may not satisfy the requirements of huge network data collection. The second one is the complexity. The current measured factors are few (delay, jitter, loss). Even if in this case, it is hard to obtain the accurate data according to simple operations. The intelligent system may handle not only the existing factors but also other complex data types. Some of the factors may be hard to measure, such as if the queuing length is wanted to know. Furthermore, IDN decision algorithms could also rely on external data for which particular connectors are required. This potentially becomes one of the key systemic requirements.

- Data pre-processing. As multiple sources of data will be leveraged, normalisation techniques it its large sense should be used (including data alignment, sanitization). It also concerns the establishment of proper metrics (distance, similarities, and dissimilarities) which are in the core of ML algorithms whereas some collected data may not be easily mapper to a metric space by nature.

- Map algorithm to network. There is a huge gap between the current AI algorithms input/output and network policy. The former is pure mathematical expression while the latter one tends to be a kind of programmed language. If the intelligent system is seen as the mapping of physical network, it is very important to build up the "bridge" between the network semantics and algorithm semantics. Different with the process of data orchestration, the core problems here is how to generate and delivery the network policies based on the mathematical input/output of the algorithm.

5.3 Rapid Development of Machine Learning Technologies

Even though the use of Machine Learning technologies is still in its infancy in most fields of networking, it will become a much thought for opportunity to enhance network operations and performance in the coming years. This is mostly due to the rapid development of Machine Learning (ML) and associated Artificial Intelligence (AI) technologies in other fields.

ML/AI in picture/video/speech recognition as well as big-data analytics in areas such as e-commerce and search have evolved to a point where many of the methods and components of building solutions are well enough understood to apply them to novel fields - such as networking.

The ability of developers to rapidly build systems with ML/AI was vastly improved in the last few years through common tools such as TensorFlow that took most of the novel and unique complexity of building ML/AI solution into those expert built tools/libraries. The layers above those common libraries now become areas of development where more and more the subject matter experts (such as networking engineers) will be able to collaborate with data analysts to build those ML/AI solutions.

The performance of both AI/ML learning/training as well as the execution of the trained neural networks has been improved radically in the past years and it is expected lot more of these recent developments to proliferate into products.

GPUs (Graphic Processor Units) such as those from NVidia (as leader in the market) have evolved to be equal good high-performance parallel execution units for ML/AI training and inference. Algorithms to improve performance of execution by more than a factor of 1 000 have been developed in the past years.

Low-end ML/AI neural network inference hardware is now being released on products. Product means that these are hardware building block that can for example be added to existing low-end CPU chips such as ARM CPUs for cellphones and low-end network devices. This hardware can only execute neural networks (this is called inference), but not train those neural networks. These accelerators can do inference at minute fractions of the power needed in GPUs. The likely first big area where these will be used is speech recognition and translation on mobile phones.

6 Benefits of Introducing AI into Network

6.1 Towards Fully Autonomic Network

A fully autonomic network means that the network contains a closed-loop of "Measure-Analyse-Decide" which can realize the whole process autonomically. By means of AI-based learning and optimization techniques, the goal of IDN architecture is to learn about its behaviour, the fundamental relation between traffic load, network configuration and the resulting performance, understand the target policy set by the network administration, and configure that policy efficiently and fully autonomously. The advantage of a fully autonomic network is realizing the closed-loop of "Measure-Analyse-Decide", which will minimize the requirements for human administrators.

Currently, the process of measure, analysis and decision are mainly independent and the cooperation of such processes typically relies on humans. The limitation is caused by the lack of analysis ability of network, which is precisely AI technology performs really well. While in operation, the IDN architecture will react autonomously to relevant events (e.g. a failure, a spike in the traffic, etc.) and change the configuration accordingly. The core of AI technology is extracting the patterns (or knowledge) from complex data, in other words, discovering the rules then applying. In current, the forwarding process has achieved stateless or stateful full automation while most of the controlling process, such as the configuration and optimization, are still manual. The roadmap should be gradual, which starts from the local area autonomic to large area and finally to global. As if the development of self-driving, the automatic transformation is realized step by step, from such as auto-shift and auto-break. As yet, the AI technology is the one of the most possible ways to realize the whole process. During the development, the introducing of the AI technology gradually implement the closed-loop of network controlling so that reduce the unnecessary manual operation including coding, configuring, simple inference, etc. A fully autonomic network potentially decreases the cost of carriers during management and control. It will be benefit for the income in the long term.

6.2 Response to the challenge of complexity

AI and most notably Machine Learning (ML) techniques play a central role in the future architecture of networks. By means of ML mechanisms, the network behaviour can be learnt to obtain a ML-based model. This model can account for any arbitrary network characteristic of interest. As examples the models can characterize the energy consumption of the network or understand the relation between the traffic load and external factors such as popular sports events.

Traditionally network modelling has been done by means of simulation, however ML provides many advantages in this regard. First, ML scales very well with complexity and it is able to understand and model non-linear (complex) issues, indeed deep neural networks are able to account for multi-dimensional non-linear problems. On the contrary, simulations require costly development to model complex behaviour. Second, although training the neural networks is a CPU/GPU intensive process, once trained the neural network is very lightweight and fast, actually just a multi-dimensional function. However, both developing and running simulations is a costly process. This is relevant particularly when using network simulations to optimize the network performance, since each run is CPU intensive. And third, ML is able to understand the network (or parts of it) as a black-box and model behaviours even in the presence of hidden information. This provides important advantages in the simplification of the measurement process, since even in the presence of uncompleted information, ML can produce efficient models. On the other side, simulation cannot work with hidden information.

Along with the growth of scale, very large scale networks become unmanageable without intelligence. Because it is hard to formulate or design a rule universal for all.

Take the allocation of link resource as an example. According to the calculation with a traditional analytical model, the optimal solution for a specific question can be obtained with a long time computing. More often, the computing time is perhaps longer than transmitting the data via the worst link, which is meaningless. The Machine Learning (ML) based method can use the trained model and quickly output an approximate optimal solution according to various network parameters as input. This operation can continually adjust and optimize the solution. In general, any questions can be calculated and obtain a theoretical optimal solution by accurate and comprehensive calculation. However, the cost may be far smaller than the benefit, for example, the time for calculating an optimal route for a flow maybe far longer than the delay detouring few of congesting links. The birth of ML method is aiming to solve the complex issues according to analysing the huge amount of historic data and extracting the hidden rules behind the issues. The complexity actually lies on two aspects: the complexity of data to handle in terms of volume, heterogeneity, accessibility and even veracity and the complexity of problems to solve which are nowadays multi-faceted and try to accommodate multiple and sometimes partially antagonist objectives. The traditional network modelling and optimization techniques may perform unsatisfactorily since they are not inherently designed for such big data scenario. However, the introducing of AI technology potentially becomes a sharp weapon to deal with the complex network issues.

6.3 Response to the challenge of variation

The AI technology brings learning-based adaptability and flexibility. Actually, the machine learning based algorithm upgrade the controlling logic more adaptive. Comparing with experienced parameters from expert, the result of ML is more flexible, especially in real-time operation of the network. The supervised training can be used in decision-making or classification problems while the unsupervised training is good at extracting the patterns that might be hard for human to find. Meanwhile, the experienced parameters (also including decisions or policies) that obtained from experts is replaced with adaptive parameters which are controlled and updated by learning algorithm. The ML based intelligence technology can make the policy flexible. According to the specific environment, the administrator or managing system can train its own model so that to satisfy the diversity. Meanwhile, whether there are any changes in the hardware layer of the network, the requirement of user and service will change always, especially behaving in traffic character. Base on the machine learning algorithm, the intelligent system can capture data (both network data and content data) in real-time and then obtain the feature of the network by distributed or centralized training process. Finally, it will modify the parameters to match the requirement of current service. The adaptive ability makes network devices configuration match the distribution character of traffic and service so that to utilize the bandwidth flexible.

Not only to adapt to the change of network, the AI technology but also potentially bring more powerful ability to modify the network. For example, in the scenario that virtual topology is defined by software, the AI technology can control and modify the virtual network topology so that to satisfy the traffic change and user requirement. The under layer technology can be provided by slicing. Due to the introducing of AI technology, the network will be not only static but also dynamic, which enables more abilities for the network. Elasticity of networks promoted by SDN and NFV can be thus fully enabled by AI-based decisions.

6.4 Insights of the Network and Improve the Utilization

The rapid construction and change of the network makes itself complex than ever. The network is developing to a dynamically changing black box for the carriers. It is more and more difficult to learn what, where and when the fault happened, and eventually trace the root causes and responsible entities. This is paramount of importance since defining an appropriate responses, such as a counter-measures, would necessitate the most specific characterization (of the fault) to be efficient. The AI technologies might be helpful to this question. By data analysing and visualization, the intelligent system may help administrator to monitor and analyse the internal events and try to make them visual and comprehensible. Along with the increasing of network scale, the network administrator feels puzzled with the abnormal behaviour. Such a visualized system can help administrator to monitor, analyse and understand the internal events and even their relations with external indicators, which will improve the network insights dramatically.

The operation always faces to the fundamental trade-off between utilization and stability. Face to the uncontrollable and complex environment, because of the lack of methods for uncertain and latent elements, big amount of physical resource needs to be reserved to prevent the uncertain event happened. The redundancy depends on experience and risk control strategy. For example, for ensuring the service level agreement, carriers usually reserve an excess of bandwidth resource which may far larger than its real demand to prevent the accident. In other word, exchange utilization for stability. The intelligent system can deeply analyse the network in foundation level and utilize the knowledge to implement refine control and management. The learned knowledge is helpful for mining the latent factors that influence the network so that decrease the uncontrollability, which may help the administrator or decision system avoiding excessive waste. Along with the increasing of network scale, the intelligent system can essentially save the physical resource and improve the utilization of network and the mix between extending the network capacity with new dedicated hardware or accessing to shared cloud-based infrastructures. As an example, defeating link-flooding attacks can rely on increasing link capacity or buying a cloud-specific service, each of these solutions having their own costs depending on their utilization.

6.5 To Be Predictive

Prediction, may be the most important attribute that AI brings since it enables proactive behaviour. According to the model learnt from the network data, the prediction-based methods tend to foresee the evolution, such as the traffic load, link congestion, failure and all kinds of factors in advance. Take the risk of failure as an example. Whatever how precise and smart of calculation, including human brains, the deployed policy will face to the risk of failure. What is worse, the increasing number of network policies and their overlap will aggravate the risk. The predictability of network (e.g. the prediction of traffic) will enable the evaluation ability of network policy for network management system or decision system. By sensing the network state, the AI algorithm can derive the development of network and forecast the potential problems, such as traffic peak, congestion, device failure, etc. This will bring significant change for network management. The network administrator will always like to predict the problems and pre-process them, which will reduce the fault rate and save the repairing cost.

Another benefit of AI is that it can model the network and provide predictions of the performance before applying a particular configuration. This provides many advantages since configurations that reduce the performance and/or are unreliable can be avoided before applying it onto the real infrastructure.

ML methods have many applications in the field of prediction, since they are well-suited to model the dependencies of multi-dimensional non-linear behaviours. In this context, ML can help predicting important network parameters. Three relevant examples are listed following:

- Traffic characteristics. An ML-based algorithm can predict the traffic load in the network at different scales. First, different spaces can be considered: by individual links, by individual nodes (routers/switches), by end-point pairs (traffic matrices), by autonomous systems pairs, etc. The time dimension also affects the granularity (per hour, week, month, etc.). This can help preparing beforehand the network to account for such load and providing better service. With long-term prediction, this can help network operators to anticipate when the load will exceed the capacity and plan ahead future network hardware upgrades. This prediction can be based on external events, such as the weather, popular sports events, time-of-day, the load of certain services, etc. The same reasoning applies for other traffic characteristics such as latency or jitter.

- Failures or attacks. It is impossible to propose a system (hardware or software) to work without being faced with a failure or an attack. Predicting them is a valuable information to actually prepare proactive-plan by increasing redundancy among network equipment. For instance, predictive maintenance is thus also helpful to prepare hardware replacement in network. From a security perspective, analysing external indicators can be used to focus analysis on certain part of the network or services and thus do predictive security by adjusting in real time the security configurations.

- Prediction can also play a central role in low-level configurations parameters of the data-plane, for instance weights of routing protocols can be predicted to achieve optimal performance.

6.6 Potential Decision Efficiency

The introduction of AI technologies brings shorter response time and improve the decision efficiency. The increasing service volume has challenge the current decision and deployment system. The machine-assistant decision system is becoming the future pattern of network management. The machine learning algorithm can learn the historical decision and data then obtain a decision model. This model can provide recommendations or suggestions to the network administration regarding the operation of the infrastructure. For instance an AI-based algorithm can learn that a certain link is congested periodically and suggest a change in the configuration, or can recommend deploying a new RAN to improve the service to the wireless subscribers. On one hand, as mentioned in clause 6.4, these policies can be evaluated and own a risk index. When the risk is low enough (high reliable), the policy can be deployed automatically without delayed. When the risk is high, the decision system can output a set of solutions as pending suggestions for administrator. Essentially, the AI technology replaces the work of human labour in learning and mining the objective rules. The network administrator can pay more attention on dealing with the novel problems that machine cannot solve immediately or cannot judge accurately. On the other hand, the long-time training process can be executed off-line that independent with the network. The parallel training and execution design can avoid the timeliness problem of ML. Besides, interaction between humans and ML can be done in a seamless way such that benefit of each other's. Actually, a user can limit the searching space of a solution a ML tries to solve by adding her own expertise. In this way, the decision efficiency may be improve materially and then drives the improvement of network.

6.7 Potential Business Model

The introduction of AI technologies brings greater transmission ability to carriers. From the beginning to now, the Internet has experienced from point-to-point communication to asymmetric content delivery. The rise of CDN is the best illustration and response to this change. From the view of transmission, CDN enlarges the transmission ability for the huge amount of content, which decreases the load of core network and holds up the service requirement. Other similar cases are multicast and router cache. Both of them try to reduce the redundant transmission for same content. In recent years, live content is another emerging service, which distributes the content sources and challenges the network transmission ability and granularity sharply. The huge size of content data needs more power to transfer and the increasing distributed degree of users also need more efficient controlling system to drive. Whatever the change of business in the future, the general requirements will always orient to greater transmission ability. The network intellectualization will potentially force and support the requirements of new service pattern, which help the carriers quickly adapting to market changes.

7 Design Goals of IDN

7.1 Goal of IDN

The goal of IDN is introducing AI technology into network so that to further upgrade the automation level of the network and improve the automation, robustness, compatibility and universality.

Intelligent decision is one of the core technologies of IDN. Current network such as SDN also has some decision capability. However, that decision capability is very limited and cannot adaptive to diverse situations of network. SDN itself is more like a compiler than a decision maker.

The IDN should open its capabilities to up layer applications, possibly through a standardized interface. This is to allow users themselves or third-party developers to easily leverage the AI capability with customized requirements.

There could be different algorithms and mechanisms to deal with different problems. However, these algorithms and mechanisms should be integrated together as one universal platform for different scenarios and applications.

Ideally, the algorithms/mechanisms should also be de-coupled from specific applications as much as possible so that some common function modules could be reused by different tasks.

No matter how autonomic the networks could be, it is always a requirement that users/administrators need to intervene on the networks to make it running as expected, thus, IDN should reserve the user interface. The interface should be as simple and abstract as possible.

Another important principle is that using IDN should not assume users to be experts of relevant technologies (e.g. machine learning technology).

7.2 Deployment models: Centralized, distributed and Hybrid

There are various models how intelligent functions can be deployed. Three types, which are centralized, distributed and hybrid are considered. These are not absolute and clearly defined but describe the deployment approaches.

The centralized model tends to aggregate the intelligent functions into fewer devices and centralize the computing, analysis and decision making. This model pushes the intelligent entity upward and uses master-slave relations to the controlled devices. Due to the high layer position and aggregating functions, the core intelligence point can own the best global view and controllability. It is easy and convenient to plan the global policy and optimization. The intelligent system is easy to deploy because most of the existing devices do not need to be upgraded. The development work is likely easier than in the other models. Less communications between devices needs to be implemented. Complex distributed algorithm issues can be avoided.

The disadvantages of centralized form are mainly from three aspects:

- The first aspect is heavy centralized load. The centralized intelligence will be responsible for all computing, analysing and decision making, which requires extensive storage and powerful computing resource.
- Single point of failure. Centralized designs may be more vulnerable to a single point of failure and attacks than a distributed system. The impact of a successful attack on a system with centralized design can be far greater as well.
- Latency. When control loops cannot be executed locally but have to run through a central system, such loops become slower. When such loops need to dynamically adjust actions to a particular set of conditions, the delay incurred can cause inaccuracy of those adjustments. This latency will even increase further if the aggregate data collected into the central intelligence or the aggregate amount of actions to be sent from the central intelligence are throttled by the network performance to/from that central location.

The distributed form tends to deploy the intelligent functions into each device. Each device may take part in the intelligent process and provide the computing and analysing ability as much as possible. Different with the centralized models, this framework owns more powerful local processing ability so that any complex problem can be divided into numbers of sub-problems and solved by local devices. The single point failure problem can be avoided possibly.

The disadvantages are from three aspects:

1) Firstly, many of the existing devices need to be upgraded. Otherwise, the new functions cannot be deployed distributed into the low layer devices.
2) Secondly, the intelligent system is complex. The intelligent functions are deployed into each possible devices. Therefore, a perfect distributed system, which can implement the communication, mission distributing and convergence, is needed. Otherwise, the distributed intelligent system can hardly run.
3) Finally, the deeply distributed design tends to fall into local optimization trap. It means that each local area produces its own optimal solutions and lack of the global view. The combination of several local solutions may perform badly, which has been proved timely in the past years.

The hybrid form tends to deploy the intelligent functions into both high layer and low layer devices. This form is the flexible solution, which aims to avoid the problems occurring in centralized and distributed design. For those functions, which requests less analysing and decision, it is suitable for deploying them into the low layer devices. For example, the simple traffic classification is suitable to deploy into the router. This will improve the classification efficiency and timeliness. However, the path planning work may not suitable to deploy into local devices. Because a mass of path information and traffic statistics may be needed and the optimization algorithm need huge CPU support. This work is too heavy for local devices.

The deployment policy will be not universal. It needs to be selected according to the actual situation. The overall principles are:

- High complexity function tends to high layer.
- Common function tends to low layer.
- Controlling function tends to be centralized.
- Data relative function tends to be distributed.

The classical example for the tendency to centralization is the evolution in traffic engineering from RSVP-TE to SR (Segment Routing). Initially, path calculation was performed in the distributed fashion by every edge-LSR using CSPF (Constrained Shorted Path First) and every LSR in the network needed to maintain state about the already reserved bandwidth so that these independently operating edge-LSR knew where in the network how much bandwidth was available. When it became clear that this approach did not allow to calculate good global optimizations (local optimization trap), centralized PCEs where introduced (Path Computation Engines). With only a single central entity being aware of the usage of bandwidth, the need for all LSR to know about allocated bandwidth also disappeared, and with it the need to use a protocol like RSVP-TE that did signal this allocation. This in result opened the door for lightweight forwarding plane operations with SR. It is quite likely that more intelligent algorithms in PCEs will be the next step of evolution of these concepts.

The opposite trend for distribution happens with enterprise networks evolved from leased-line hub & spoke topologies to any-to-any connectivity across the Internet, something that today many companies sell as so called SD-WAN (Software Defined WAN). Because traffic not passed through a central (hub) site anymore, many intelligent policies need to be executed distributed on every branch site, and depending on design of the solution, the determination of policy may be distributed or centralized (in which case the resulting solution is a hybrid one).

7.3 Wired and wireless consideration

Since fixed telecommunication networks (such as broad band service and traditional fixed voice service, etc.) and mobile telecommunication networks are usually separated systems, IDN fitting into fixed networks and mobile networks are introduced respectively.

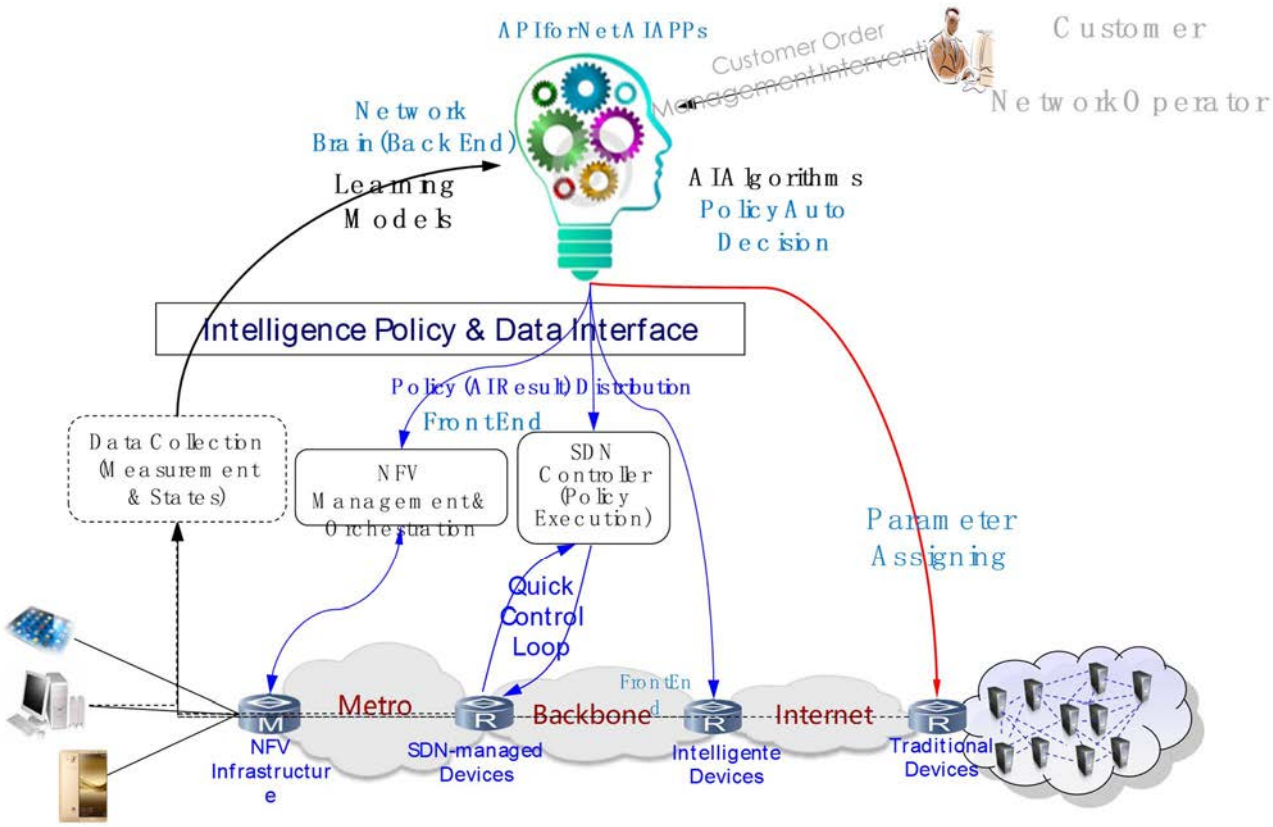

Figure 2: IDN in Fixed Network Scenarios

As the Figure 2 shown, the basic structure of the IDN is as the following:

- Upper layer: Applications and the "Network Brain"

  This layer is application-centric. The "Network Brain" normally does not directly interact with specific network devices; it provides data analysis/learning services to a variety of applications, and generate network-level policies to intervene the network to run as the applications expected.

The core of the Network Brain is mostly composed by a variety of AI relevant technologies such as machine learning algorithms, data-mining algorithms, expert systems, etc.

As described in clause 6.3, the Network Brain also represents an interface to applications so that developer could easily create specific tasks without handling the data analysis/mining/learning by themselves.

- Middle layer: Network Orchestration and Controlling

  This layer is to interpret the network-level policies generated by the Network Brain into device-level policies/configurations and deliver them to corresponding devices.

  SDN controller is an instance of the middle layer.

  Based on the Network Brain generated policies, this layer could make some simple decision by itself so that a quick control loop could be formed to control the devices behaviour in a much more efficient way.

- Under layer: Network Devices

There are different kinds of network devices: NFV infrastructure, SDN-managed devices, Intelligent Devices (which could directly interact with the Network Brain), and Traditional Devices.

As the Figure 3 shows, the wireless part of IDN is composed of a number of distinct mobile intelligent network decision entities. One centralized cross-domain IDN decision entity sits in upper layer. Below it in the network layer reside many distributed decision entities. Each of these entities is made up of four key components: wireless data collection, analysis & modelling, policy decision, and action verification and application, with these components an intelligent control loop can be realized.

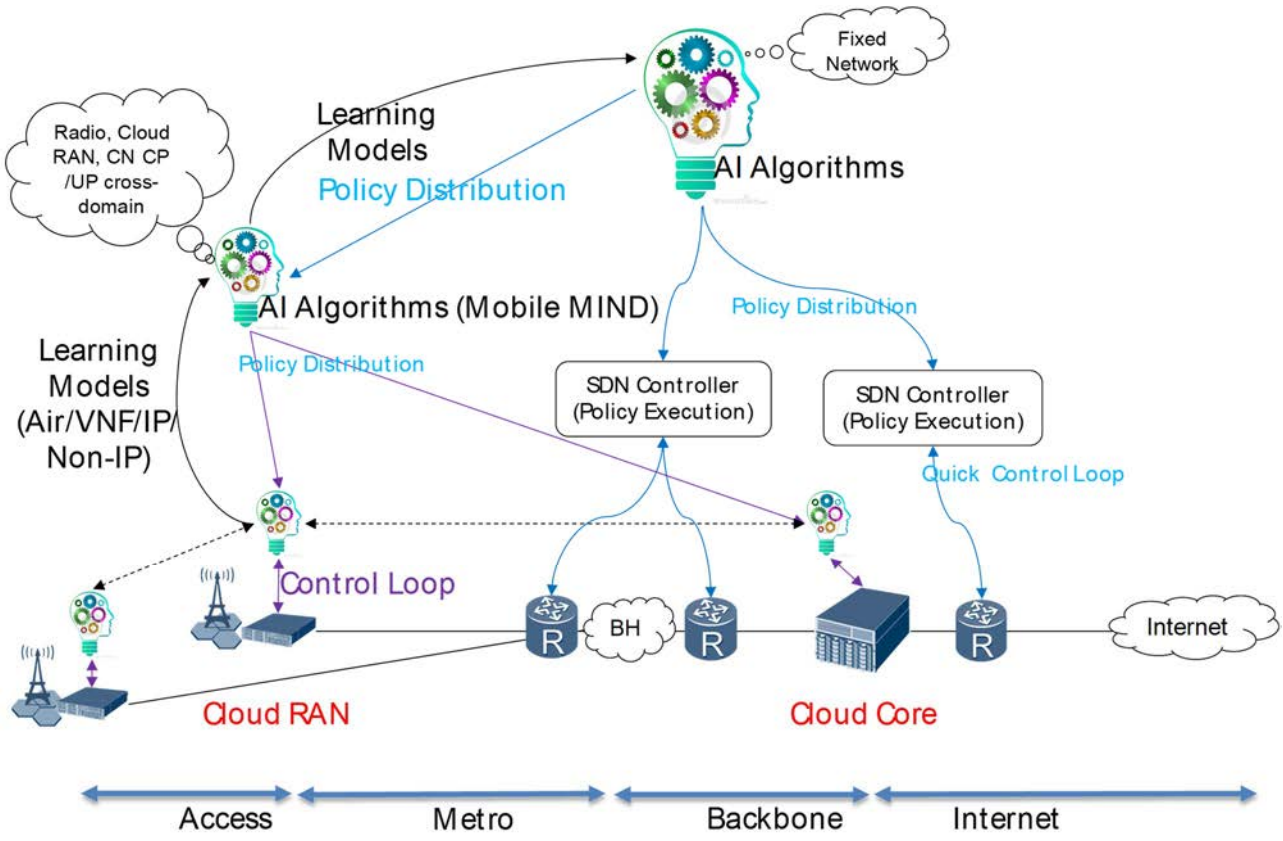

Figure 3: IDN in Wireless Network Scenarios

The IDN architecture will affect mobile network from context awareness, policy control architecture, intelligence decentralization and other more aspects, the IDN decision instances can be deployed in the radio access network as well as core network, in order to meet future wireless and 5G network diverse requirements. The IDN collects runtime network context as well as static parameters from virtualized or physical network functions and infrastructures.

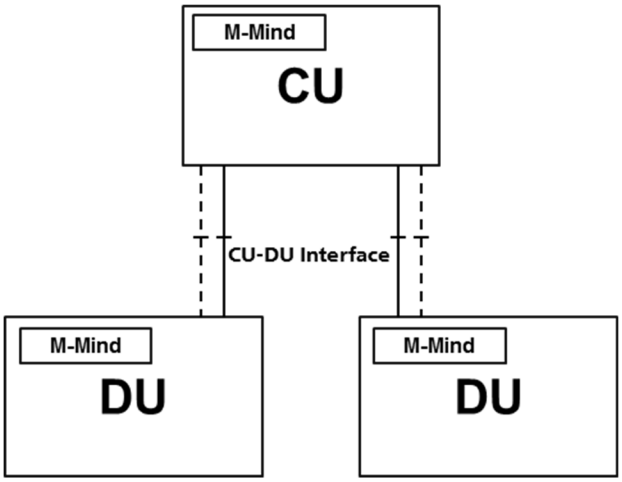

Figure 4: New RAN architecture with CU and DU separated

As Figure 4 shows, in New RAN architecture gNodeB will be divided into Central Unit (CU) and Distributed Unit (DU), CU and DU can be defined as follows:

- Central Unit (CU): a logical node that includes the RAN functions excepting those functions allocated exclusively to the DU. CU controls the operation of DUs, and is supposed runs on commodity hardware.
- Distributed Unit (DU): a logical node that includes, depending on the functional split option, a subset of the gNodeB functions. The operation of DU is controlled by the CU, and is supposed runs on proprietary hardware.

Wireless IDN should be either independent entity or be integrated in both CU and DU to enhance those functions accomplishment; RAN internal functional split is still under discussion, such Functions similar to E-UTRAN as listed in ETSI TS 136 401 [i.11].

7.4 Security and Privacy Considerations

When security relevant decisions are made based on the use of intelligent analytics or automated intelligent decision making, care should 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 analysed how that potentially enables attackers to easier feed data that derails the intelligent system ability to distinguish good from bad behaviour. As [i.1] and [i.2] reported, many companies have been puzzled by the unprecedented "attack" that the legal and safe operations and data cause unexpected result. The traditional security problems are caused by the bugs in design or implementation. These two may become the typical examples of potential new security challenges. The future system should include the ability that can deal with the following problems but not limited.

- Negative Data: the training system should have the ability to recognize data which may induce the system to become in an unexpected form on purpose. This problem will be serious in every data-fed system and it never happened in the past design.
- Conflict Data: the training system should have the ability that can recognize the data which may cause confliction to the current known strategies or states. This problem may happen when user or device execute its private rules in a large and share area unintentionally or intentionally.

Content data transmitted through network contains private data about users. Whereas AI techniques are powerful tools to automate network management functions, systematizing the large collection and processing of data presents some risk regarding the privacy of users. Indeed, predicting the location of a mobile users can clearly help in allocating resources in a RAN but it is also a privacy breaches. A future AI-based system should take in considerations the two following:

- Limitation of private data. Privacy can be leaked out from collected or post-processed data. Decisions algorithms may need it. Thus, any algorithms should clearly specify the mention of information it uses to get as input/output and clearly limits according to it. Therefore, acquiring non used data should be excluded. As a result, this will prevent to gather potential private information which is not use and also improve the scalability of the collection process. Such a limitation has to be also compatible with legislation.

- Data security. The IDN architecture should include the necessary mechanisms to avoid unfortunate data leakage. First, collected data should be accessible by only specific algorithms that use them in an appropriate way. Second, collected data could have a maximum lifetime to be then discarded when being meaningless for further processing. Third, the IDN architecture should prevent any algorithms (for instance provided by a third party) aiming at discretely supporting private data exfiltration.

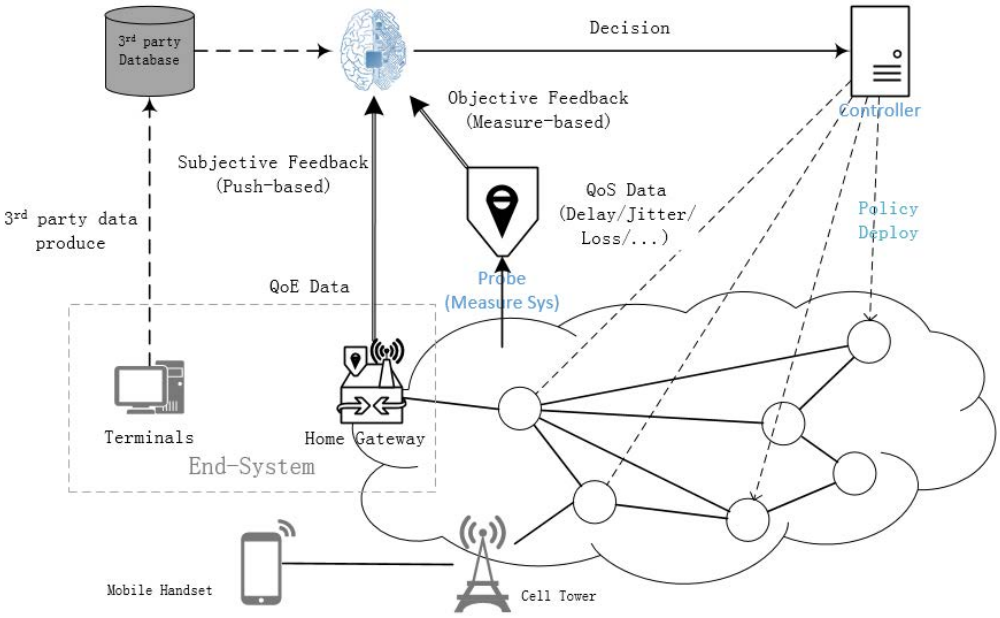

Figure 5: Information Flow

7.5 Multi-objectives Resolution

Intelligent techniques relying on IA, ML or optimization techniques are able to provide good results only if objectives are well scoped and defined. For instance, reducing the network load or latency is a standard objective of traffic engineering techniques. However, a network operator usually expects to satisfy several and parallel objectives like security and performance. Those can be in fact antagonist and compete to each other. As more individual problems to solve are introduced, the global "network brain" may suffer to apply incoherent policies making so impossible to reach states expected by AI algorithms. They can continue by refining their decision with reinforcement learning but this will continue in the wrong direction. The problem to address in that case is the conjunction of multiple decision algorithms which actually needs to be aware of each other. Such awareness can take several form. Algorithms can interact together in a pair-wise manner or through a central controller, the brain. The latter can also play the role of an AI orchestrator by providing feedback to the algorithms regarding what has been really applied in the network compared to what they expect. Hence, continuous learning can be based on valuable data.

The next question is how to balance priorities between algorithms, e.g. security or performance first. Such a question is hardly addressable by fully automated method but one could imagine a global algorithm taking a single objective such as reducing costs or increasing the benefit to automatically balance the risk between losing some customers due to performance issues or security issues. Legal penalties could be even then included in such a decision algorithm. Such a problem is very large and paves the way of opportunities for IA in network as well. This is all the more true with multi-tenant scenarios where operators may provide their infrastructures to others who want to build AI-added services on top of it.

8 The proposed IDN Architecture

8.1 Reference Architecture

In this clause, a reference Intelligence-Defined Network architecture has been proposed. This architecture can cover, explain and support most of the current use cases and scenarios.

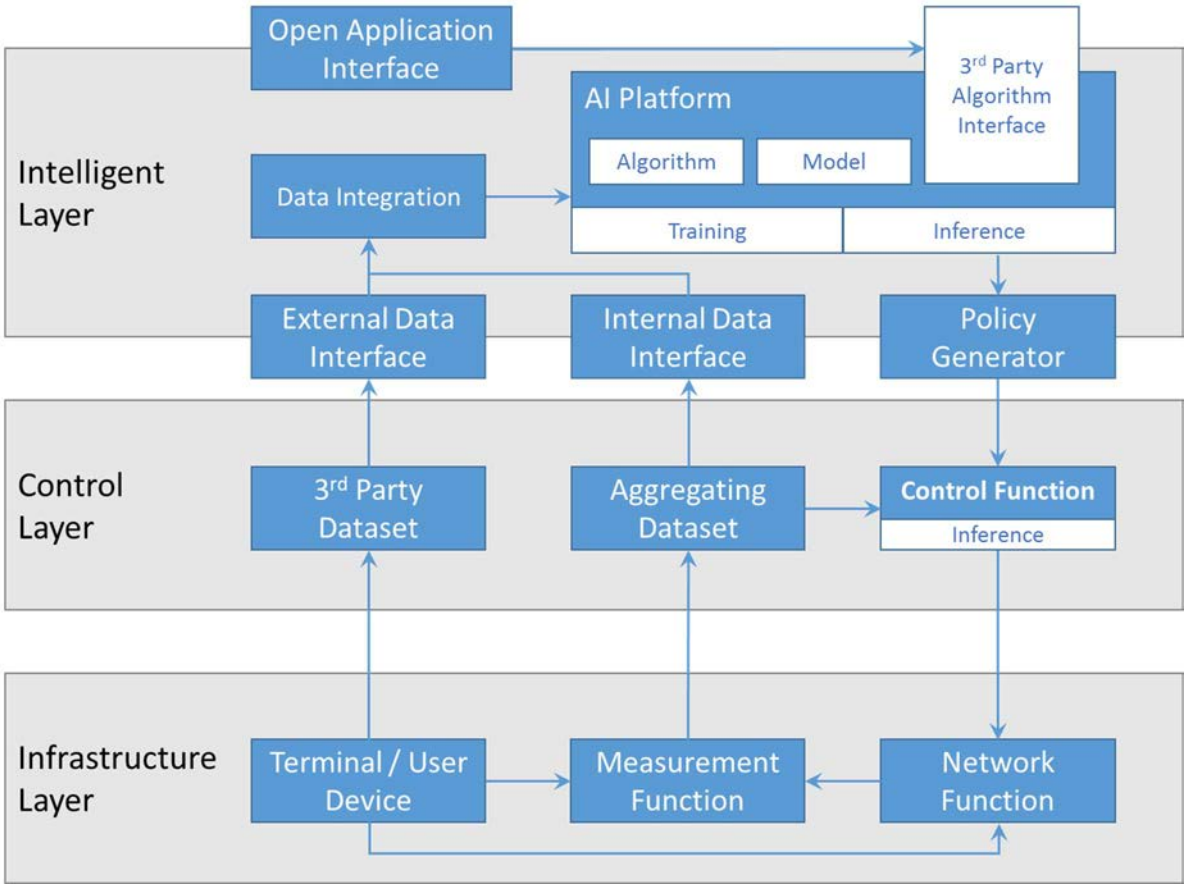

Figure 6: The reference IDN Architecture

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 common approach for the measurement function are probe systems, which are deployed among the network in a distributed manner. 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 Table 1. The Terminal/User Device stands for the device that produces and consumes data, which may include PC, smart phone, datacentre, content storage server, cloud, etc. Some of the data produced by terminal/user devices is measurable. This type of data will be captured by the measurement function. In-network measurement can be done in an active or passive manner. In the passive manner, data are directly derived from observations without acting in the network. Active measure would request some operations to be achieved such as testing routes or performance. Active measurement itself can be controlled by the AI loop to provide enough accurate results to support efficient decision making processes. 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. Such data can help to consolidate network insights, as for example regarding current threats, but also the future configuration of the network, by pre-allocating resources to services gaining in popularity. What is more, the Terminal/User Device may also directly control the network function via User Network Interface, which is describe as the UNI loop in the following.

Table 1: Measured Information

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	……

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 orchestrators are control functions. Traditional network devices such as routers integrate the forwarding and control functions (although as of today not with many instances of intelligent control functions). Traditional routers therefore include functions from two layers. 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, cleaning 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.

The high layer, which is also the main body of IDN, is the Intelligence Layer. This layer is commonly deployed in the datacentre, 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. A data integration component is defined to emphasize the need to adopt format and structure of various types of collected information to the needs of the AI Platform.

The core of the AI Platform are algorithm and model. The AI platform 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 a 3rd party Algorithm Interface is defined to provide an adaptive developing ability. The user (or a 3rd party) may develop his/her own algorithms and upload then onto the AI Platform via a northbound Open Application Interface. Additional Northbound Open Application interfaces can also be used to connect other software platforms to the AI Platform to create a cooperation between multiple systems (not shown).

The output of AI Platform 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 clauses.

Figure 7 shows some more example details of the back end (northbound) processing that could go along with the above shown AI platform. The core component is network policy task management module. Face to the upper side, it can interact with network administrator via the Intelligence Network Task Description API. The network tasks are deployed automatically or manually. Face to the underside, it can accept data from the outside or deploy policy to the low layer devices via the intelligence Policy & Data Interface (IPDI). The input data will be processed after IPDI and then transport to the AI platform. The purple blocks, such as Task Modelling, AI Algorithms and AI Platform, are mainly relative with AI technology. The AI platform is mainly responsible for training the model. The task modelling and AI algorithms are responsible for storing the trained model. The management module can call the corresponding function in these modules to generate the output decision. What is more, the 3rd-party algorithms are also accepted. Network clients can upload their own algorithm model into the engine.

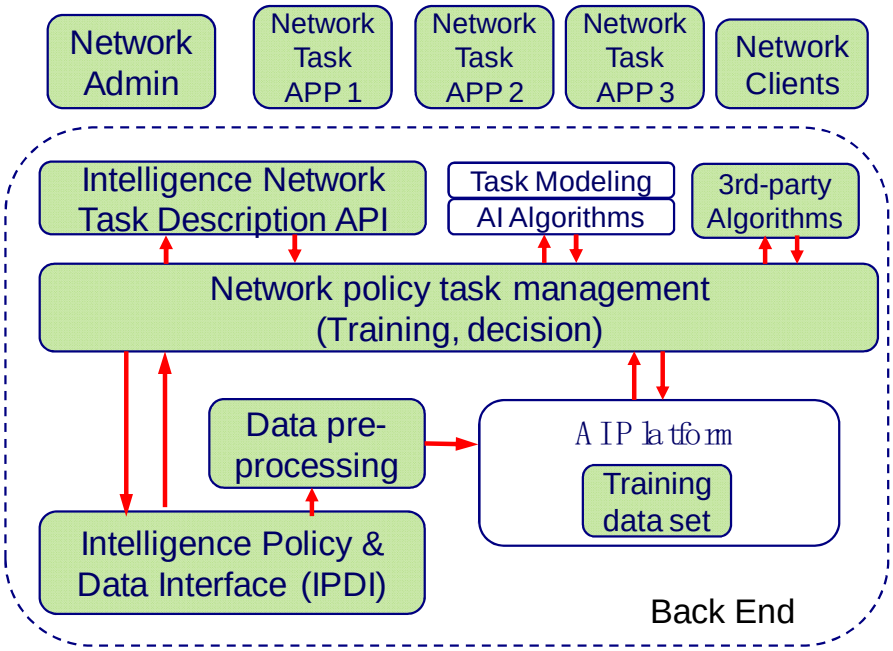

Figure 7: Back End Components of IDN

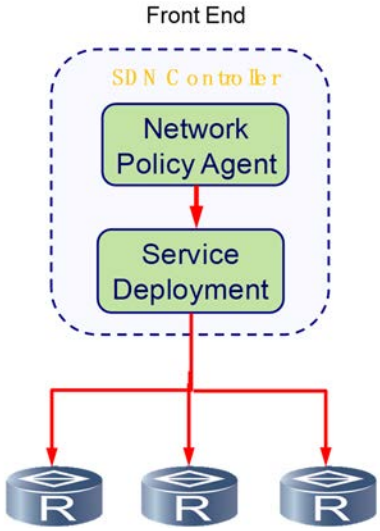

Figure 8: Front End Components of IDN

Figure 8 shows the Front End Components of IDN. From this view, the low layer devices are SDN-enable routers, which can be fully controlled by the SDN controller. The controller is divided into two parts. To the upper layer, the Network Policy Agent module is responsible for receiving the policy from the IDN engine. Then, the agent will generate the details configuration information and push to the Service Deployment module. And finally, the SDN-enable routers will be configured by the controller to execute the policy.

Figure 9 describes the information flow from the network view with the example of residential users. This user accesses the network via the home gateway when using in-home networks (wired, wireless) and the cell-tower when using 3G/4G/5G - e.g. from a mobile handset/tablet. The terminal and home gateway combine the end-system. The ISP deployed the probe functions in the network (collect the data of network, such as QoS data) and the home gateway (collect the data of user, such as the QoE data), respectively. The former one is objective feedback which is measure-based and collect the data periodically and the latter one is subjective feedback which is push-based and collect the data randomly.

Both of the two sources of data are seen as internal data and will be upload to the intelligent system (brain system). Besides that, there may be several 3rd party databases that can provide external data, which is produced by various applications. The AI algorithm and model will aggregate these internal and external data then make an optimizing policy to improve its service. This policy is translated into the format that controller can read and execute. After the policy deployment, the configuration of network devices (routers) will be modified and the route between terminal and content server is optimized.

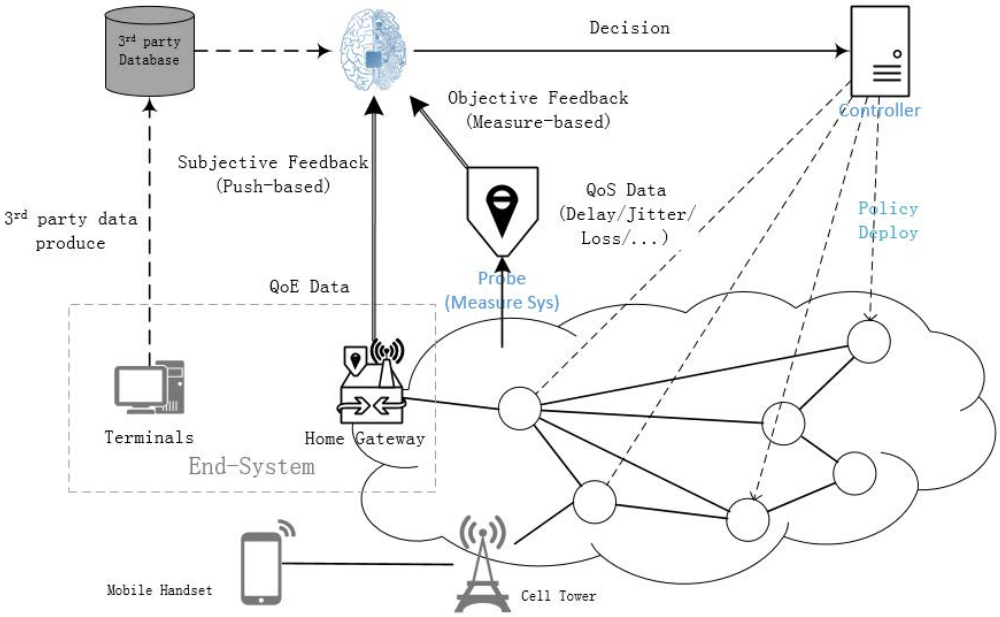

Figure 9: Information Flow

8.2 Comparing System design

8.2.1 Overview

Any network faces to the function deployment problem, which means how to allocate the functions and deploy them into different layers and devices. For example, which layer should the routing planning function be deployed? The IP network typically deploys the routing algorithm in the forwarding layer so that each router can learn and plan the routes via its local information. This is efficient but faces to the lack of optimization in the global area. The SDN network tends to deploy the route planning function on the high layer controller to solve the global optimization problem but potentially introduces the massive load to the single point. Fortunately, these is no right or wrong answer for this problem but only design consideration according to the different scenarios and requirements. In IDN, three key indicators have been proposed to describe the function deployment.

Table 2: Key Indicators

Indicator	Property
N-S Communication Density (C) E-W Communication Density (D) Function Divergence (H)	[0, 1] C + D + H = 1

The N-S Communication Density means that how many N-S direction context will be executed in once decision making process. The larger of C means the function tends to be deployed upper layer and centralized. This is because if the function is deployed and integrated in high layer, the computing and control devices will be isolated from the information by the low layer devices. Therefore, it potentially needs more N-S direction information exchange during the decision making.

The E-W Communication Density means that how many E-W direction context will be executed in once decision making process. The larger of D means the function tends to be deployed upper layer and centralized. This is because if the function is deployed and separated in low layer, the low layer devices need to exchange more information each other so that it can obtain the necessary information to make the decision. Take the IP routing algorithm as an example.

The function is deployed among all routers in the network. During the route calculation, each router will exchange the information with at least all the neighbour routers but no any context with the manage devices (N-S direction devices) whatever there is or not. This function can be call fully distributed but not centralized. Take the SDN flow table as another example. If the SDN switch want to obtain a next hop for a series of packets, it can only ask the high layer controller to calculate and then wait for the configuration. During this process, no any information needs to exchange between SDN switches. This function can be call fully centralized but not distributed.

The Function Divergence means that how many types of devices, which are function independent to each other, will join in once decision making process. The larger of H means the function tends to be deployed multiple layer and hybrid. This concept may be novel but important. This concept describes the complexity of a function which centralization and distribution cannot. Take the route optimization in Figure 9 as an example. If the ISP wants to optimize the route service for an end-system, there at least six types of devices in three layers need to take part in. It is really difficult to define the optimization function is centralized controlled or distributed controlled. However, the hybrid degree describe this well. It is highly hybrid.

It is necessary to note that the centralized and distributed is not opposite but negative correlative. Any function at least needs two independent devices to take part in the process. It is hardly to define a function to be distributed or centralized and it is harder in a network. Here the Decision-Making Triangle describes the three relationships well. The IDN functions can be also deployed into multiple forms according to the requirements.

These three key indicators can be expressed in visual three dimensions graph. As Figure 10 shown, there are three dimensions to evaluate the design trend of IDN architecture, which are N-S Communication Density (marked as C), E-W Communication Density (marked as D) and Function Divergence (marked as H). A Decision-Making Triangle is proposed to describe the network function deployment form.

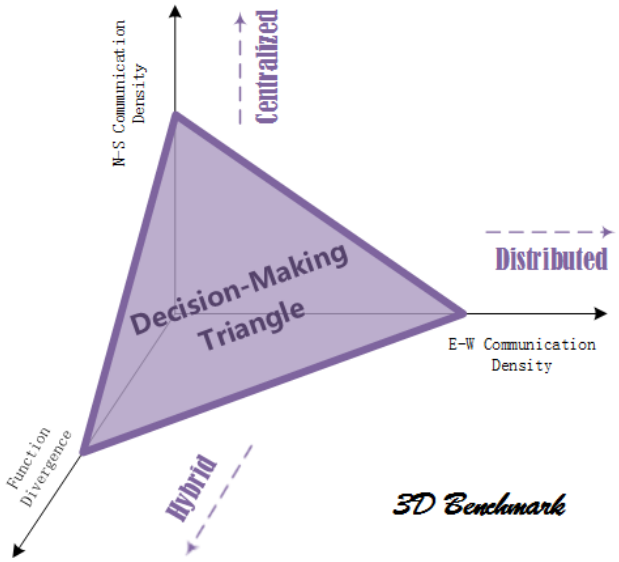

Figure 10: Visual Expression of Function Deployment

8.2.2 Distributed Architecture

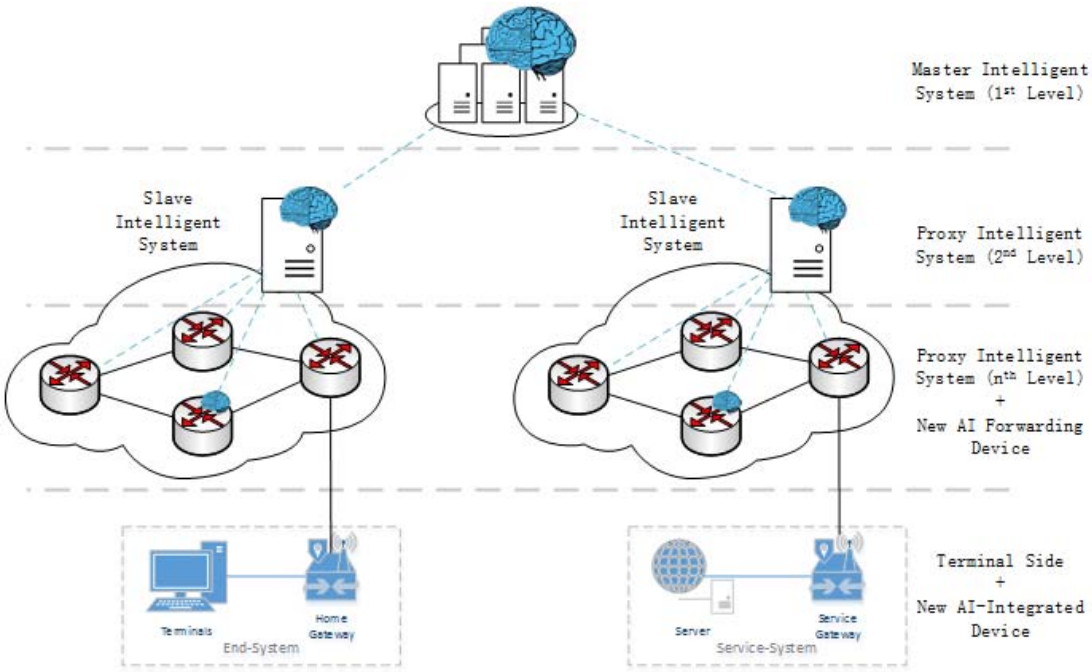

Figure 11: IDN Distributed Architecture

Figure 11 shows the schematic diagram of distributed architecture. The intelligent functions are deployed separately in multiple layers. The decision function are designed as a layered and tree structure. The lower intelligent system plays a slave role and becomes the local proxy of the higher layer intelligent system. In this architecture, the data tends to exchange among the devices of same layer or same type. The decision tends to be made in local and downward layer. The upward layer intelligent system play two roles. One is managing and organizing its slave systems, including the initialization and maintenance work. The other is to agent the communication between downward layer's devices.

The key advantage of the distributed architecture is the high scalability. Any complex or huge scale problems can be divided to multiple devices and each device takes on part of the problems and less pressure. This design potentially faces to light single point failure problem along with the increasing of requirement. The disadvantage is the lack of global view. Since each part of the network owns the local intelligent system and decision-making ability, the combination of several local optimized solution will not be the global optimized solution and this issue can be hardly solved.

Due to the feature of distributed design, the functions, which may have high traffic and repeat works and require low global optimization support, are suitable employing this architecture.

8.2.3 Centralized Architecture

Figure 12 shows the schematic diagram of centralized architecture. The intelligent functions are deployed centralize in few of layers and devices. The decision function are designed as a client-server structure. The lower devices plays a client role and simply becomes the data supporter and policy executor of the higher layer intelligent system. In this architecture, the data tends to aggregate to the devices of upward layer or different type. The decision tends to be made in upward layer and then sent to the downward layer for execution. The upward layer intelligent system plays the server role. On one hand, it collects and maintains the important data from the low layer devices. On the other hand, the intelligent system provides calculating and analysing ability to the low layer devices. Make the decision and direct other devices to execute.

The key advantage of the centralized architecture is the dominant force. Any complex or huge scale problems can be solved by the global view and full information of whole network. The deployment is easier because most of the functions are integrated into few of devices. However, this design potentially faces to the heavy single point failure problem along with the increasing of requirement. Since most of the calculating work is aggregated to the centralized intelligent system device, the high efficiency brings the heavy work load.

Due to the feature of centralized design, the functions, which may have some traffic and calculating works and require high global optimization support, are suitable employing this architecture.

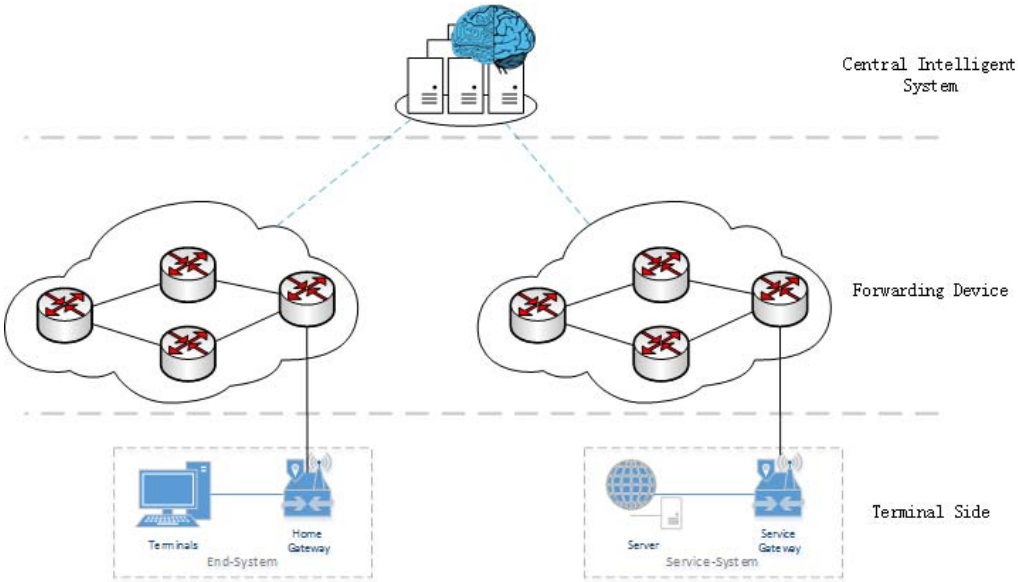

Figure 12: IDN Centralized Architecture

8.2.4 Hybrid Architecture

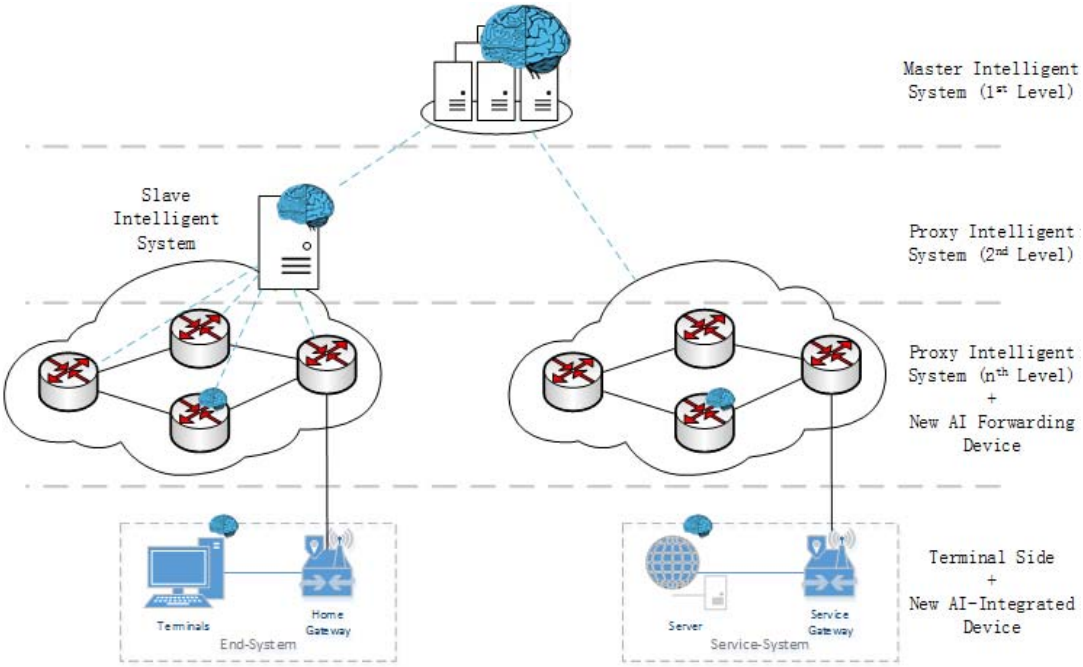

Figure 13: Hybrid Architecture

Figure 13 shows the schematic diagram of hybrid architecture. The intelligent functions are deployed separately in multiple layers and multiple types of device. Moreover, these device's function are independent to each other. The decision function are designed as a layered and pipeline structure. In each link of the pipeline, the architecture may be design to centralized or distributed. Each link gets the result from the upstream link and output the result to the downstream link. The responsibility of each link are highly independent to each other. In general, the AI system in each layer mainly play the role of backbone that means to integrate the result and organized the decision flow to run around in the network.

The key advantage of the hybrid architecture is the high flexibility. The AI system is open and easy to access. The whole system can be organized as toy block. Every function can be deployed according to its feature and requirement. However, this kind of system will be hard to build because massive of data format, interfaces and protocols are needed to design. Another serious lack is serious. Since the sub-system input and output only the result from and to the upstream and downstream, it is really hard to share the data content between two sub-systems. The hidden relationship among the massive data is potentially lost.

Due to the feature of hybrid design, the functions, which may have high independence and private works and require flexible optimization support, are suitable employing this architecture.

8.3 Controlling Loop

8.3.1 AI-Enhanced Close Loop

Figure 14 shows the controlling loop of IDN. One of the aims is to build up a fully automatic decision system that can analyse plan and execute the decision, which the close loop ring implements. The data is produced by the device of infrastructure layer and transmitted to the dataset. After analysis, the formatted data is transmitted to the intelligent system and finally arrived at the AI platform. The AI model and algorithm will deal with the input data and calculate a result for current state. The output decision may be translated into a specific data model/format before deploying. After the controller execute the decision, the network device configuration will be modified and the network layer is optimized. In the whole process, every step is executed automatic and there is no human action taking part in.

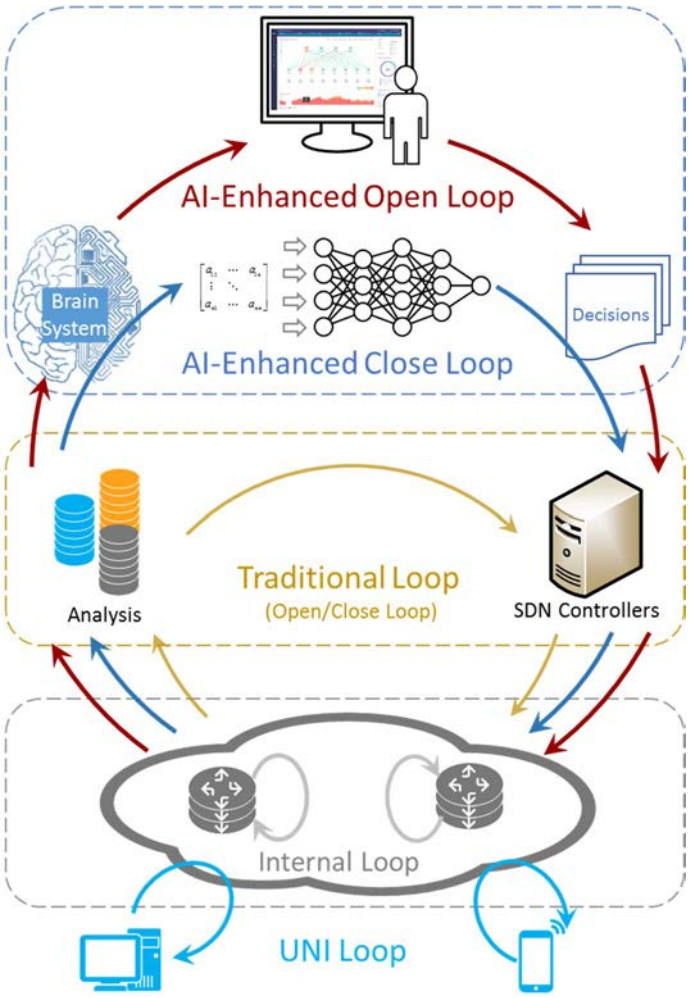

Figure 14: The Controlling Loop Model

Figure 15 has shown an example Use-Cased: Deep Reinforcement Learning-based Routing. An architecture of a fully autonomic routing system has been proposed. The system is based on Deep-Reinforcement Learning (DRL) techniques [i.9]. DRL uses an agent that interacts with by changing the environment through actions, such actions change the state of the environment. The goal of the agent is find the set of actions that puts the environment in a state that maximizes a certain reward. In the context of computer networks, the infrastructure can be understood as the environment where the agent acts.

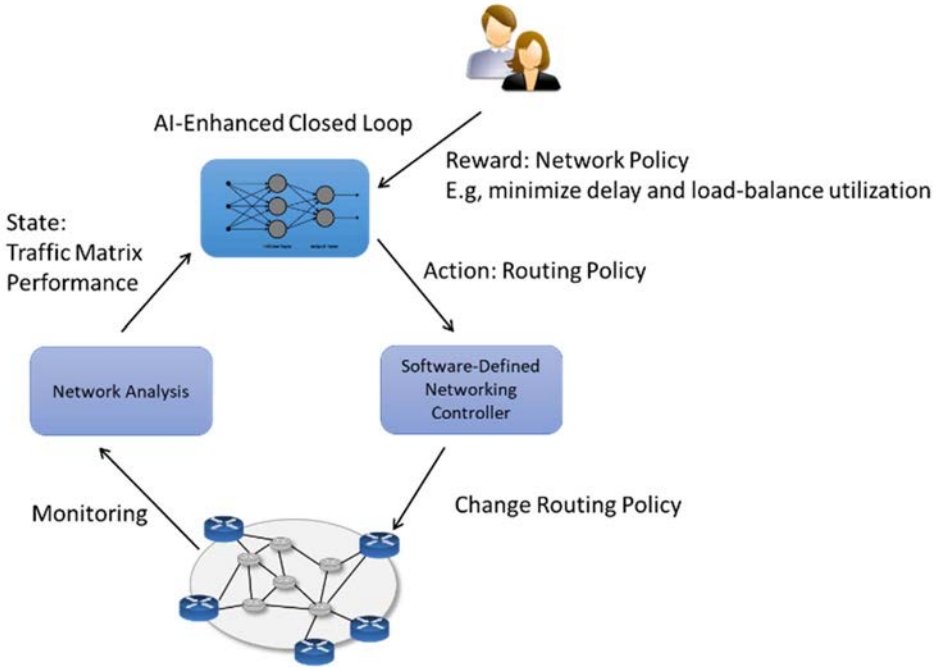

Figure 15: Use-Case: DRL-based Routing

Operation:

In this context the DRL loop (AI-Enhanced Open Loop) operates as follows:

1) The agent acts upon the network by changing the network routing policy, this can be the weights of the links (similarly to OSPF) or the OpenFlow configuration for flow-based routing. This action is applied through the SDN controller that transforms the output of the DRL agent into imperative commands that are understood by network data-plane elements (e.g. CLI, OpenFlow, etc.).

2) The network reacts to this change by routing flows through different paths. This fundamentally changes the state of the network and results in a different performance.

3) By means of the monitoring infrastructure, the agent receives the new state of the network. In this particular example this is represented by the performance of the network (e.g. per-link utilization and QoS metrics). In addition to that the agent is informed about the current Traffic Matrix that the network is forwarding.

4) During the training phase, the agent has learnt which are the set of actions that will maximize the reward. In this case the reward can be understood as the network policy, as an example: 'minimize the delay of the flows while load-balance the utilization of the network'. With this, the agent will choose the set of actions that will match the performance of the reward function, autonomically managing the routing policy while achieving the high-level goals set by the network administrators.

This loop is repeated for each new Traffic Matrix, where the agent will pick the routing configuration that successfully implements the reward (network policy).

Training:

With DRL, training of the agent is achieved by means of exploration. This means that the agent will pick random actions to test which is the effect on the environment and the reward. Through this process, the agent builds internally a model of the network to maximize the reward. After a certain number of steps, the agent can then operate the network autonomically.

Since the exploration phase (training) may result in the agent applying routing configurations that may render the infrastructure down, several approaches can be followed for training:

1) Simulator/Model: In this case, the agent is first trained with a simulation or model of the real network, once training is finished then the agent is deployed online to operate the network.

2) Expert: In this case the agent is trained by means of an expert, this expert can be a traditional network optimization system or a human network administrator. The agent monitors the actions taken by the expert to build the internal model, again after training the agent is deployed online.

Advantages:

Such approach provides several important advantages with respect to traditional optimization algorithms:

- Real-time operation: The DRL agent is capable of operating autonomously the infrastructure in real time. On the contrary, traditional optimization algorithms require lengthy (and costly) iterative searches to find the optimal configuration. This is why typically they do not operate in real-time.

- Black-box optimization: Existing network optimization techniques are tailored towards a particular goal (e.g. load-balance traffic). This means that changing the optimization goal often requires a new network optimization technique. With DRL changing the goal just means changing the reward function, but not the software of the agent. It is worth noting that a change in the reward function requires re-training the agent.

- Model-free: Typically network optimization algorithms operate on top of a model of the network, for instance a simulator. However, DRL can be trained with the real-infrastructure or, if this is found problematic, by means of an expert.

8.3.2 AI-Enhanced Open Loop

Even though the intelligent system could implement the fully autonomic controlling, there is still necessary to reserve the open interface for human intervention. The information from both External and Internal will influence the decision. The open loop may have at least three modes:

- Firstly, when the machine learning algorithm cannot make a certain optimized decision, the AI platform should output several pending solutions for the administrator as choices. The network administrator can simply choose one of them or modify some of the parameters before execution.

- Secondly, the administrator can modify the network configuration when necessary whatever the processing is reasonable or not. That means, human's volition is high than machine.

- Thirdly, the 3rd party application should have an interface to access to the controlling system. As mentioned above, the whole intelligent system should be open to the permissible user and service. This can implement the system to system communication and cooperation.

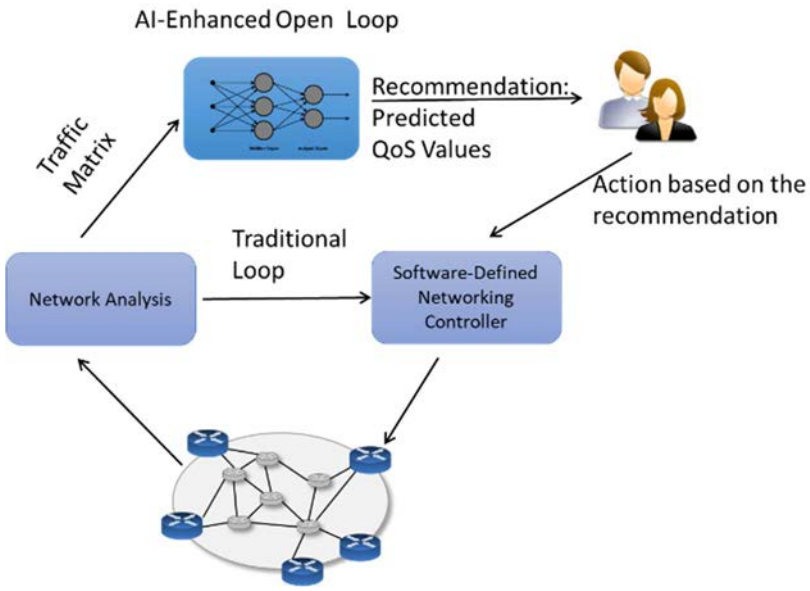

Figure 16: Use-Case: QoS Monitoring and Recommendation

Figure 16 is a use-case that exemplifies the AI-Enhanced open loop, in this case Machine Learning techniques are used to provide valuable recommendations to the network administrators to improve the overall performance of the network.

In this case a Deep Generative Model (DGM) (for instance a Generative Adversarial Networks) is used to predict the performance of the network given the current traffic matrix. This means that the DGM model is trained to predict the QoS values (delay, jitter and losses) of the network when loaded with a particular traffic matrix.

Operation:

The system operates as follows:

1) The Network Analysis platform is monitoring the load of the network by sampling the amount of traffic in each of the ingress/egress routers. With this builds periodically the current Traffic Matrix.

2) The Traffic Matrix is used as input for the DGM, the model has been trained to predict which will be the resulting QoS performance of the network for when loaded with a particular Traffic Matrix.

3) If the predicted QoS parameters are not met (for instance they are above a certain threshold) then the network administrator is alerted and recommended to change the network configuration to avoid the QoS-violation.

Training:

In order to train the DGM a dataset of tuples <Traffic Matrix, QoS parameters> is constructed, that is a set of examples of the performance of the network (QoS) for different set of loads (Traffic Matrices). Machine Learning techniques require that such dataset is representative enough to build accurate models.

In this case training can occur offline using historical values. Since changes on the Traffic Matrix naturally occur while the network is in operation, the monitoring system can store such matrices. In addition and for each matrix, the monitoring system has to measure the corresponding QoS parameters, once the historical archive is representative enough the DGM can be trained. It is worth noting that each model is built for a particular network topology and routing configuration, if one of these characteristics changes the model has to be re-trained.

Advantages:

Existing techniques for modelling the performance of a network are based either on analytical or simulation techniques. Analytical techniques cannot model complex network behaviours, for instance they can model the utilization of the network but it is very hard to build a model for the delay in a real network. Simulations can actually model such complex behaviour, but this comes at a high development and computational costs. On the contrary, DGM can model complex multi-dimensional non-linear behaviours and at the same time once trained, are fast and lightweight. As such, DGM are lightweight and accurate techniques that can be effective used to predict the QoS of the network.

The cost associated to DGM models are in the training phase, which is typically expensive in terms of computing costs. However and for this use-case such costs are incurred offline (training is offline) and not while in operation.

8.3.3 Traditional Loop

IDN should be compatible with traditional network control logic and protocols so that the traditional control loop should be kept. For example, the current SDN architecture is centralized control and distributed forwarding. The controller will collect the key parameters from the under layer network and make a high layer policy. This process should be kept in the future IDN architecture.

8.3.4 Internal Loop

For efficiency, some of the low layer network functions quickly iterate in local area without long distance interaction. For example, the quick reward information between devices may not pass through the high layer devices. That means some of the simple intelligent functions will be implemented in the forwarding layer devices. The future intelligent router may run the AI algorithms in local and make optimizing decision without the support of high layer system.

8.3.5 UNI Loop

The UNI Loop stands for User Network Interface Loop, which is mainly responsible for the interaction between the user model and network model. In the future, the terminal will not only play the role of data producer, but also join in the network control process via UNI.

8.4 Core Support Technologies

8.4.1 Network modelling

Generative model is a kind of model that often used to model a joint probability distribution over observations in the real world. Intuitively, it can generate (high- dimensional) data samples following the underlying unknown data distribution, which is learned from a group of collected data samples in the real world. In recent years, with the increasing power of deep learning, the generative model develops fast and becomes one of the hottest topic (termed the "deep generative model" (DGM)) in the research areas of AI. Due to the fast development of effective training methods (including both advanced algorithms and specialized GPU hardware) of ultra-deep neural network from big data, DGM exhibits great performance gains in multiple domains. For example, a typical deep generative model has been successively applied to generate data samples of highly-complex structures with state-of-art performance, such as generating high-quality images [i.3] and human-level speech [i.4] from given text description. Traditional algorithm typically requires data modelling and feature extraction by human analysis, which leads to both large overhead of human efforts and inaccurate data models. Compared with previous statistics modelling method, the surprisingly high performance gains from DGM should be contributed to the high effectiveness of deep neural network on representation learning and feature extraction from large amounts of data [i.5].

DGM typically include the VAE (Variational AutoEncoder) [i.6] and GAN (Generative Adversarial Network) model [i.7]. VAE model is built by combining both the power of traditional Bayesian variational inference and deep learning, while GAN model is developed by combining the power of game theory and deep learning. By allowing additional inputs, they can be extended to the conditional version, named by CVAE and CGAN, respectively. In the computer network, a DGM can be used to model some data distribution that are quite challenge for traditional statistic methods, such as estimating the traffic demand matrix and path delay. A conditional DGM can further learn how such complex data distribution is related to the network measurements.

DGM is useful for optimizing the traffic engineering. For example, existing traffic engineering typically adopts the strategy of adjusting the flow routing paths to achieve load balancing. Since it is important to ensure the path delay does not exceed some SLA criterion (e.g. 90 % end-to-end delay should be less than 100 ms), one key problem involved is how to predict the end-to-end delay of specific paths when given a new flow allocation onto them. If this problem is well-solved, it becomes easy for a strategy to know a flow should choose which path to go, since the resulted path delay under different choices have been well estimated. Traditional methods on this problem requires careful mathematical modelling of path delay with queuing theory. However, to make it easy to analyse, such methods need some ideal assumptions for the traffic model (e.g. assuming the flow arrives following Poisson distribution), which may not hold in real networks. To overcome this issue, the DGM can learn the complex relationship between path delay and traffic loads directly from measurement data. Specifically, with the VAE model, one can first measure and collect pairs of traffic loads and corresponding path delay in the network. Next, a VAE model can be setup with the input as three parts: path delay, traffic loads and a high-dimensional Gaussian variable, while the output is the distribution of the path delay. Between the input and output, VAE consists of two neural networks (named the encoder and decoder). Then the traditional stochastic gradient descent (SGD) algorithm is employed to train the VAE model using the collected delay and traffic data. After training, only the decoder of VAE is used for inference. The decoder has an input of the traffic load and a high-dimensional Gaussian variable, while its output is the distribution of the path delay. When given any specific traffic load to the input of the VAE decoder, the different samples of the high-dimensional Gaussian variable are drawn into the decoder and the decoder output is calculated as samples of path delay. In this way, a probability distribution of path delay is obtained and the value of 90 % end-to-end path delay can be easily marked. Other values like mean and standard deviation can also be obtained.

DGM is also a natural application to generate high-dimensional data with highly-complex structures in computer network, such as the traffic demand matrix. Traffic demand matrix estimation from several measurements of link loads is a key problem in existing core network planning [i.8]. Since it is hard for existing network to directly measure the entire traffic demand matrix, current approach requires the estimation based on link measurements (or called the "interface traffic") and also full expert knowledge on the network topology/routing rules. This traditional estimation process generally requires solving an implicit integer programming problem, and may take quite a long computation time when the network scale is large and the routing rules are highly-complex. With the tool of DGM (e.g. GAN), it is possible to learn the mapping from a set of link measurements to the distribution of traffic demand matrix. It indicates that the DGM not only outputs the estimated traffic demand matrix, but can also display the whole probability distribution of the traffic demand matrix due to the inaccurate and incompleteness of the link measurements. After training, DGM can do a fast traffic demand inference for a given set of link measurements, which requires only once feed-forward computation of the neural network. Specifically, some known traffic demands with corresponding to the link loads in the network can be tested firstly. A typical GAN model consists of two neural networks (generator and discriminator). The generator in the GAN model is setup with the input of link loads and a high-dimensional Gaussian variable, while the output is the generated traffic demand matrix. The discriminator in the GAN model is setup with input of the link loads and the generated traffic demand matrix, while the output is the probability of whether the generated traffic demand matrix comes from the real traffic distribution. By collecting pairs of the traffic demand matrix and corresponding measured link loads, the GAN model can be trained with SGD algorithm. Similar as the VAE model, after training, only the generator of GAN is used for traffic demand inference. Since the GAN model is well-known to successively learn the inner hidden structure of the generated high-dimensional data in real world [i.3], [i.4], it is expected to learn good insights into generating high-dimensional traffic demand matrix following real traffic distribution using the clues of link measurements.

8.4.2 Measurement and Data Orchestration

This block is responsible for obtaining and storing relevant information for further processing by ML algorithms. Actually, this block can be divided into the following sub-blocks:

- Data acquisition: this consists in gathering data from network or from a third party. Data sources are thus very various and can come from the hardware or software infrastructure. This block mainly relies on traditional monitoring protocols as well as newly developed ones, when necessary. Depending on the use-case the block should monitor the network at packet-level, flow-level or coarser levels. In addition, the configuration of the network includes key information to learn the behaviour of the network as well as its elements.

- Data cleaning: because data can be polluted by badly collected information, invaluable noise due to error or fault in the monitoring process itself. Besides, extreme cases can bias the learning process. Such inputs should be discarded before being processed for supporting decisions. AI driven algorithms can also be used in this task such as outlier detection techniques.

- Data sanitization and anonymization: to make the approach compatible with privacy concerns, such a step might be required. For example, assuming that the ML application is provided by a third party, the operator could not be allowed to reveal sensitive or client-specific information.

- Data transformation, normalization and scaling: data collected from multiple locations, multiple sources or from multiple configurations of probes cannot be seen on the same scale. For example, different sampling rates can be applied from different flow collectors. Percentage based values may be more significant than absolute values but such a statement is not always true and highly dependent of the use case. Although ML algorithms can require specific inputs such as real values, integers, positives or bounded values, produced inputs by raw data acquisition could not respect such conditions. Data transformation is thus necessary and, again, AI technique can be already applied at this level. For example, methods from Natural Language Processing can be leveraged to transform text based data into numerical vectors which represent the most used types of inputs in ML.

- Data aggregation: pre-processing data to combine multiple input instances into a consolidate insight before real processing for decisions have multiple advantages. First, it reduces the overhead induces by transmission of heavy load of data which can occur when measurement processing and decision algorithms are not collocated. Second, the ML process that will analyse consolidated data needs thus to process less data. Learning is thus sped up.

A central aspect of this module is how it represents the data for the upstream functional blocks for processing, learning and optimization, in AI this is known as feature engineering. Feature engineering is the process of creating data representations (features) of the computer network data (e.g. regarding traffic, configuration, etc.) that makes ML algorithms work. Without proper representations of the data, such algorithms do not operate efficiently. Although some automatic techniques exist (e.g. auto-encoders), this process is typically carried by experts in the field. Feature engineering actually may span over multiple sub-blocks described above. For example, if feature engineering is guided by a human, this impact from the first step that defines the data to be collected. In case of an automated selection of features, the rationale is to collect as much as possible of initial data which are then transformed into a limited sets of features.

Other well-established applications of Machine Learning have already developed features to represent their data, consequently an important effort is needed to carry this work in the computer network domain, finding ways to represent traffic flows, routing configuration, network policies, etc. Standardizing such features is central for the interoperability of different implementations of the IDN architecture.

9 Potential Standardization Works

9.1 Overview

In this clause, the potential standardization opportunities in IDN have been analyzed. The following points not only service to IDN but also available in other area. Figure 17 shows the overview of potential standardization points.

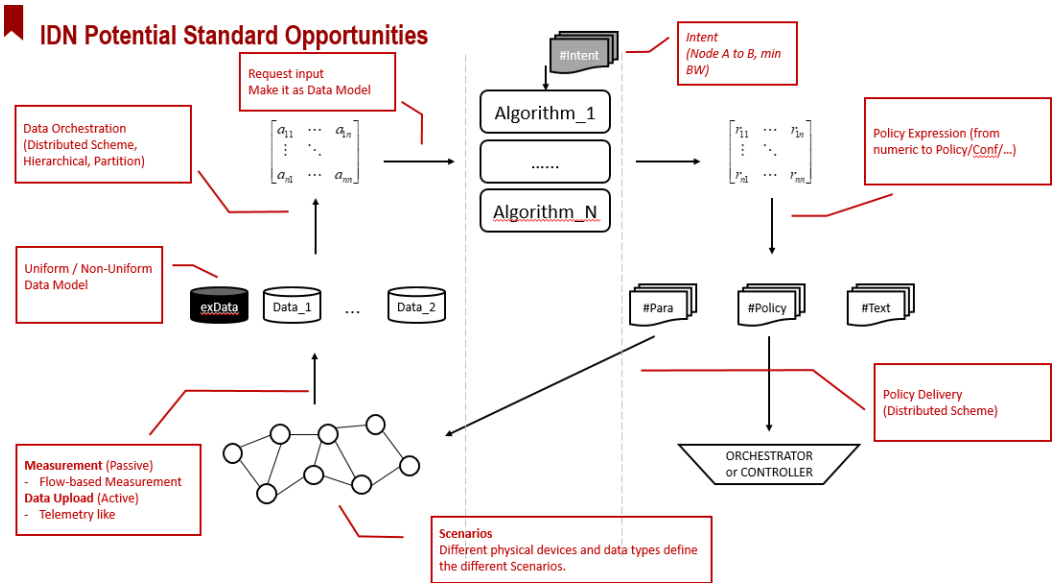

Figure 17: Potential Standardization Points Overview

9.2 Measurement

The intelligent system needs massive data to feed and support to formulate the policy and decision. Therefore, the measurement should 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.

Secondly, the IDN may need more kinds of data type to measure. Not only the service-oriented data, such as delay, jitter and packet loss rate, but also other new necessary parameters. For example, is it possible to measure the QoE of user? Or is it possible to measure the flow continuity? So far as to measure how many 10 Mb/s links can be provide from a specific router to the destination and how many slicing resource can be provided? The flow-based parameter measurement may become important in the future.

Thirdly, the current measuring method may become the alternate way for the legacy device, which cannot provide the proactive data upload ability. This will be a complement solution during the transition period.

9.3 Data Centric standards

Not all data can be used in AI system training or execution. The data should be formatted, program readable, explainable, labelled (sometimes not), aligned, semantic (perhaps) and with statistical significance. This provides several standardization opportunities:

- Data representation. The data representation is significant. Most of the current AI algorithms were born in the pattern recognition area, especially the image processing. All the images can be expressed as uniform binary vectors or can be easily transformed into uniform format. Unfortunately, this 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, especially when complex relationships in high dimensions data are focused. YANG model is a potential solution for this problems. The data listed in Table 1 should be covered.

- Data orchestration. The data may come from different source. For example, the delay, jitter and bandwidth can be captured from the network in real-time, which is call internal data and describe the properties of network. However, the other types of data, e.g. temperature, traffic information, which is not relevant with network but indeed influences the network policy, may come from external source. Such data may have various format, precision, size and so on. This will not satisfy the requirement of AI algorithm because most of the current (may be same in the future) algorithms require the data format uniform very strictly. Therefore, the data, whatever captured or measured from the network, should be organized or arranged before input to the algorithm. Currently, most of the data arrangement is implemented in centralized database. The scalability will be challenged when the data volume grows higher. A distributed data collection and orchestration method may be needed so that to build up the bridge between network data and algorithm. Assume that there are three types of devices in the network: data producer, data consumer and controller. When the data producer accesses to the network, the controller will send an index ID of a specific algorithm to the producer via the option of some protocols. This ID indicate the positions that the data will be placed in the specific algorithm. When the data consumer request data from the producer, it will indicate the algorithm ID in the option. After the data producer receives this request, it will send back the requested data to the consumer with both the index ID and algorithm ID. Thus, when the data arrives to the consumer, it will be easily recognized that which position of this data should be placed. This method can distribute the data arrangement into each data producer, which improve the scalability of data collection.

- Data transport. When the data producer is far from the data consumer, or the structure is too complex, it is necessary to simplify the data transport process. For example, a data proxy may organize the local area data as a whole and then transport it to the data consumer. Similar with the bone router in an AS, according to the negotiation, a header data device will play the slave role to communicate with the data consumer so that to simplify the whole communication. Another example maybe the data relay. A relay may stand for all devices behind to organize and feedback the data to the consumer.

- Inherent data upload. Instead of the network measurement, it is necessary to explore an inherent data upload scheme that allow the network device subscribe/push data proactively.

9.4 Control Centric standards

Around the control, there are numbers of potential standard points:

- Device access. If the intelligent layer is seen as an overlay of physical network, the intelligent device actually accesses to an intelligent system when it accesses into the physical network. Just like the DHCP, a fully autonomic system should own the ability that auto-configuring the new device. For example, to allocate the manage device of current area.

- Policy representation. Different with the data representation, the policy representation needs to translate the uniform format data (algorithm output) into various policies that the controller or managing device can read and execute. So YANG model may be not sufficient to implement this function. Another reason is that 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 E-W direction. What is more, some of the policies are sent to controller or orchestrator for execution while others may directly sent to the intelligent device as parameters, which needs different representations.

- Election and hierarchy. When the network scale is very large, the centralized and flat manage scheme tends to low efficient. The hierarchical scheme is one of the potential solutions for the large scale network scenario. Thus, it may be necessary to explore the negotiation method among the local devices, such as to poll a leader data device standing for others. This may support the data transport mentioned above.

Annex A:
Authors & contributors

The following people have contributed to the present document:

Rapporteur:
Dr. Sheng Jiang, Huawei Technologies Co. Ltd.

Other key contributors:
Bing Liu, Huawei Technologies Co. Ltd.

Zhibo Gong, Huawei Technologies Co. Ltd.

Qiaoling Wang, Huawei Technologies Co. Ltd.

Yixu Xu, Huawei Technologies Co. Ltd.

Shen Yan, Huawei Technologies Co. Ltd.

Albert Cabellos, Universitat Politècnica de Catalunya

Toerless Eckert, Huawei Technologies Co. Ltd.

Jérôme François, INRIA Nancy - Grand Est.

History

Document history		
V1.1.1	June 2018	Publication