Knowledge Defined Networking: State of the Art and Research Challenges

Adnane Mzibri

July 20, 2022

Knowledge Defined Networking: *State of the Art and research challenges*

Adnane Mzibri[1]

[1]ANISSE, Faculty of Sciences, Mohammed V University Rabat, Avenue des Nations Unies, Agdal, 10000 Rabat, Morocco

ABSTRACT
Recently, combining a software-defined network (SDN) with a machine learning (ML) technique to improve network performance has become an even more attractive topic for the research community, thanks to the availability of processing and storage capabilities.

Knowledge-Defined Networking (KDN) is a new paradigm that combines SDN, Network Telemetry, network analysis (NA) and ML algorithms.

SDN provides a complete view of the network through a logically centralized controller. This allows the collection of a large amount of data describing the network behaviour. Network telemetry gathers data generated in the network, such as SNMP, sFlow [4], NetFlow [5] and Syslog data. NA analyses and structures the collected data. Finally, ML algorithms process the structured data, develop knowledge, and exploit this to control the network.

This paper presents a brief overview of the KDN paradigm; also, it describes the concepts covered by its architecture and reports on several works done in this field.

Keywords: *SDN, Machine Learning, Network Analytics, Knowledge-Defined Networking, Network Telemetry.*

1. INTRODUCTION

Network and business Applications are playing an increasingly critical role for the company. They are no longer just there to support the business (ERP, e-commerce portals, etc.). They are at the heart of each business organization, which is thinking more than ever about its "digitalization" strategy.

Disruptive models using innovative applications challenge the «traditional» businesses; for example, Uber and Taxify compete for standard taxis, Airbnb and Booking are challenging hotels, YouTube is broadcasting football matches and rivalling television channels...

Each organization is taking its "digitalization" more seriously that why the implementation of applications is evolving much more rapidly.
This pressure has become too much for network's administrators that why SDN has been taken seriously in recent years

SDN decouples control from the data plane. This provides a logically centralized control plane, which simplifies network control and makes it more agile.
To make the network intelligent and more efficient, the research community has associated SDN networks with the machine learning approach in several works. This has motivated different universities and research centres (UC Berkeley, UPC, CISCO, HP Enterprise, Brocade, and CA Labs) to propose in 2016 [2] a new paradigm based on Knowledge Plane[1], Machine Learning and Network Analytics called Knowledge Defined Networking (KDN) [3].

The KDN offers a simple but brilliant idea: Estimate a Network behaviour without knowing a network's configuration or topology, based only on some observations of its behaviour.
In the KDN context, network Data Plane elements (routers and switches) are equipped with improved computing and storage capabilities. This has enabled a

new breed of network monitoring techniques, commonly referred to as network telemetry [7, 8].
Such techniques provide real-time packet and flow-granularity information, as well as configuration and network state monitoring data, to a centralized Network Analytics platform [12, 13].
The remainder of this paper includes fundamental concepts that we must absolutely know to assimilate the KDN paradigm structured as follows:
We start with the definition of SDN, its architecture, and its main characteristics in section 2. Then, section 3 describes the Machine Learning approach applied to Networks, while section 4 defines the Networks Analytics approach, in section 5 we introduce KDN recent paradigm, which brings together all previously cited domains and gives some of his applications, section 6 discus the difference between combining SDN with ML and KDN paradigm. Finally, we conclude this paper.

2. SDN ARCHITECTURE

The SDN paradigm based on separating the control plane from the data plane and placing it in a logically centralized point called the controller. Its architecture based on three planes: the data plane, the control plane and the application plane (Fig. 1).

This concept is developed in 2008 at the American universities of Berkeley and Sandford, introduces a new paradigm in network, servers and storage architectures. From now on, the IT infrastructure adapts to the applications and not the other way around.

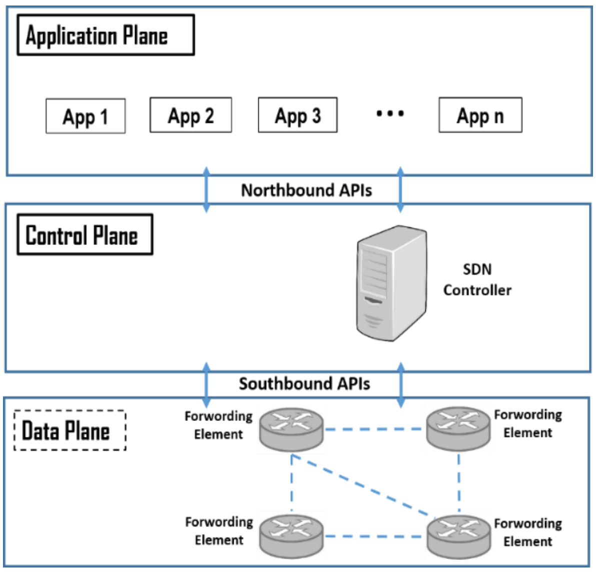

Figure 1 The Software Defined Networking Architecture.

The data plane (sometimes called the user plane, routing plane, bearer plane or carrier plane) is responsible for carrying the traffic. While the role of the control plane is to implement rules on the nodes of the SDN network. The rules can concern specific network services processing such as routing, load balancing, high availability, QoS, etc.

The application plane contains the typical network applications or functions like network monitoring, intrusion detection systems, load balancing or firewalls...

The flow of information between the control plane and the other plans is done through Northbound and Southbound APIs. Based on an open-source platform Northbound API is the software interface between the controller and the SDN applications. While the Southbound API [5] is the Protocol required for the SDN controller to control and manage the interface between the various devices on the network. In the case of a multi-controller architecture, another interface called East-West is necessary to monitor the interactions between the different controllers.

3. MACHINE LEARNING FOR NETWORKING

ML is a set of methods used to automatically detect patterns from data.

The discovered patterns will be used to predict future data to facilitate decision-making processes. ML can be divided into supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning (RL).
In the context of networks, certain possibilities arise from the use of ML such as prediction of behaviour in the network [27], anomaly detection [28], traffic identification and flow classification [29]. For more information on the applicability of ML in the network area, we refer the reader to Ayoubi et al [25] and Boutaba et al [26].
Figure 2 shows the machine learning workflow applied to the network domain following the six steps below:

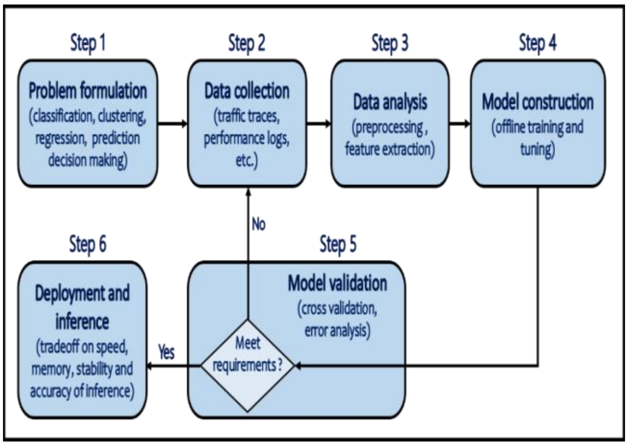

Figure 2 The typical workflow of machine learning for networking.

Problem Formulation: The first step in machine learning is deciding what you want to predict, also known as a "label" or "target response".

Data Collection: The purpose of this step is to collect a large set of representative network data (DATASET) without bias [7].

Data analysis: Each network problem has its characteristics and is influenced by many factors, but only a few factors (i.e. feature) have the more effect on the target network performance.

Before providing the features to a machine-learning algorithm, it is recommended to preprocess and clean raw data, through processes such as normalization, discretization, and missing value completion, to identify possible problems and learn more about them because the predictive power of the model is proportional to the quality of the data provided.

Model construction: Model building involves model selection, training, and tuning. To do it we split a dataset into a training dataset (DTrain) and test dataset (DTest), typically 70-80% for training and 20-30% for test.

Model Validation: The validation of the model consists of evaluating whether the learning algorithm works sufficiently well. During this step, cross-validation is typically used to test the overall accuracy of the model to show if the model is over-fitted or under-fitted.

Deployment and Inference: in this step, we must have a compromise between precision and overload because it has an impact on the performance of the practical network system.

3. NETWORKS ANALYTICS

Network Analytics refers to analyzing network data to identify and understand patterns occurring on the network. Applying an analytics solution required having a holistic view of the network; because, if we only examine part of the network, the results of the data analysis will be useful only for the analyzed part of the network that is why it is interesting to use network analytics with SDN, which provides a full network view.

There are several open-source Network Analytics framework for example:

NOESIS [22,23] an open-source framework for network-based data mining. It features a large number of techniques and methods for the analysis of structural network properties, network visualization, community detection, link scoring, and link prediction. This framework is available under a BSD open-source software license.

NFStream[24]is a Python framework providing fast, flexible, and expressive data structures designed to make working with online or offline network data both easy and intuitive. Additionally, it has the broader goal of becoming a common network data analytics framework for researchers providing data reproducibility across experiments.

KDN proposes a new paradigm that integrates and leverages SDN, network analysis, and ML techniques in a single architecture. As a result, it presents the operational loop described in Figure 3.

There are two different ways to control the operation of the network (the closed-loop mode and the open-loop mode):

- The closed-loop mode aims to automate the network operation tasks (i.e., recognize and act). This includes automatic decision-making and real-time optimization of the network configuration (e.g., routing, security policy).
- The open-loop mode focuses on implementing recommender systems that enable network administrators to make better decisions (i.e., recognize-explain-suggest). This includes performing what-if analysis, estimating performance metrics (e.g., loss, delay), or validating a configuration before implementing it in the network.

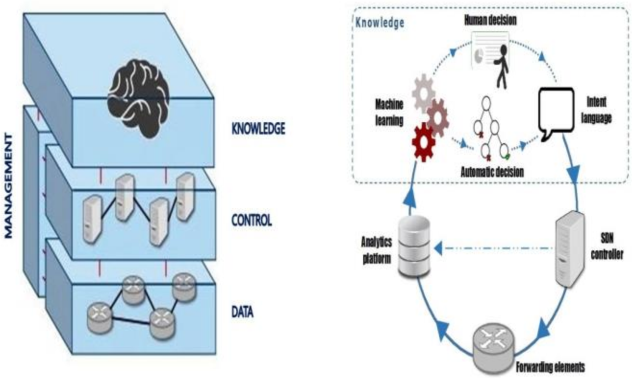

Figure 3 The KDN Planes and operational loop [2].
Source: A. Mestreset al., "Knowledge-Defined Networking"ACM SIGCOMM Computer Communication Review, vol. 47, no. 3, pp. 2-10, 2017

KDN comprises four planes: Data Plane, Control Plane, Management Plane, and Knowledge Plane.

Data Plane: The Data Plane is responsible for storing, forwarding, and processing data packets.

Control Plane: The Control Plane provides the interfaces to receive the instructions from the KP; then the Controller transmits the instructions to forwarding devices. Furthermore, the Control Plane sends metadata to the Management Plane about the network state. Besides, The Control Plane exchanges operational state to update the data plane matching and processing rules. In an SDN network, this role is assigned to the –logically centralized– SDN controller that programs SDN data plane forwarding elements via a southbound interface, typically using an imperative language. While the data plane operates at packet time scales, the control plane is slower and typically operates at flow time scales.

Management Plane: The Management Plane ensures the correct operation and performance of the network in the long term. It defines the network topology and handles the provision and configuration of network devices. The management plane is also responsible for monitoring the network to provide critical network analytics.

To this end, it collects telemetry information from the data plane while keeping a historical record of the network state and events.

Knowledge Plane: the heart of the knowledge plane is its ability to integrate behavioral models and reasoning processes oriented to decision making into an SDN network. In the KDN paradigm, the KP takes advantage of the control and management planes to obtain a rich view and control over the network. It is responsible for learning the behavior of the network and, in some cases, automatically operate the network accordingly.

Fundamentally, the KP processes the network analytics collected by the management plane, transforms them into knowledge via ML, and uses that knowledge to make decisions (either automatically or through human intervention). While parsing the information and learning

Table 1. Some KDN Applications.

Authors	Applications	Ref	ML algorithms	Performance analysis
Saptarshi Ghosh	Routing optimization	[14]	RNN with LSTM	pre-calculates all possible paths between each pair of network nodes and provides self-healing with constant-time convergence to have a self-organized cognitive routing framework supporting URLLC[1]
Alex M. R. Ruelas& Christian E.Rothenberg	load balancing	[15]	ANN	predicting the network performance according to traffic parameters paths using the ANN Algorithm to proposes a load balancing method
Quang Tran Anh Pham et al.	QoS-aware Routing	[17]	DRL & CNN	extract the mutual impacts between the flows in the networks by exploiting a DRL agent with convolutional neural networks in the context of KDN to improve the performance of QoS-aware routing against two metrics latency and packet loss rate
Jonghwan Hyun et al.	propose a KDN architecture for self-driving network	[19]	--	Propose an architecture for self-driving network and also present network monitoring system implementation on SDN controller using INT, and discuss its limitations.
Edwin FerneyCastillo et al.	propose a KDN architecture that keeps Control Channel Overhead (CCO) and the Extra CPU Usage of the Controller (CUC)within thresholds	[20]	RL	proposes an architecture called ipro based on knowledge Defined networking paradigm, and reinforcement learning algorithm to determine the probing interval which maintains the overload of the control channel (CCO) and the Extra CPU Usage of the Controller (CUC) within thresholds

from it is typically a slow process, using such knowledge automatically can be done at a time-scale close to those of the control and management planes.

3. DISCCUTION

Can we say that we are in the KDN context when we simply use ML Algorithms on data stored on the SDN controller to improve its performance?

The answer is obviously no because we still need to add the management plan that communicates with all the other plans in the KDN, and add telemetry to the data plan.

In-band Network Telemetry (INT) is a Framework for collecting and reporting network state by the data plane called metadata (e g. of metadata, Switch ID, timestamp, Ingress/Egress Port ID, Link Utilization, Hop Latency, Egress Queue Occupancy, Egress Queue Congestion Status), without requiring intervention or work by the control plane. This technology enables switches to provide detailed information on network load and use it to provide congestion control mechanisms and therefore better manage high-speed network stability.
Such techniques provide real-time packet and flow-granularity information, as well as configuration and network state monitoring data, to a management plan.

The data plane of KDN is responsible for the storage, routing and processing of data packets. In SDN networks, the data plane is only responsible for routing packets. They operate without regard to the rest of the network and depend on the others planes to populate their routing tables and update their configuration.

The management plan ensures the well working and performance of the network in the long term. It defines the network topology and manages the provisioning and configuration of network devices. In SDN Networks, the SDN controller usually handles this task.

The management plane is also responsible for monitoring the network to provide critical network analysis. To this end, it collects telemetry information from the data plane while maintaining a historical record of the network state and characteristics.

The core of the knowledge plane is its ability to integrate behavioral models and decision-oriented reasoning processes. In the KDN paradigm, the KP takes advantage of the control and management planes to gain a rich view and control over the network. It is responsible for learning the network behavior and, in some cases, automatically operating the network accordingly. The KP processes the network analytical data collected by the management plan, transforms it into knowledge through ML, and finally make decisions (automatically or provides recommendations that require human validation).

3. CONCLUSION

Applying the power of Machine Learning in networking allows the network to be more intelligent; consequently, it helps to make optimal decisions. In addition, ML technics promises a solution to several native problems of traditional networks, such as manageability, configuration, scalability, and elasticity.

Knowledge Defined Networking heralds a significant change in networks in the coming years. Networks will see their architecture evolve profoundly, facilitating new uses. All this will be possible thanks to programmability of Networks, data analysis and machine learning. No area seems to be spared: WAN, data centers, campus, security... The challenge for network administrators is to accompany this new stage to take advantage of these new capabilities.

REFERENCES

[1] E.M. Clarke, E.A. Emerson, Design and synthesis of synchronization skeletons using branching time temporal logic, in: D. Kozen (Eds.), Workshop on Logics of Programs, Lecture Notes in Computer Science, vol. 131, Springer, Berlin, Heidelberg, 1981, pp. 52–71. DOI: https://doi.org/10.1007/BFb0025774

[2] J.P. Queille, J. Sifakis, Specification and verification of concurrent systems in CESAR, in: M. Dezani-Ciancaglini and U. Montanari (Eds.), Proceedings of the 5th International Symposium on Programming, Lecture Notes in Computer Science, vol. 137, Springer, Berlin, Heidelberg, 1982, pp. 337–351. DOI: https://doi.org/10.1007/3-540-11494-7_22

[3] C. Baier, J-P. Katoen, Principles of Model Checking, MIT Press, 2008.

[4] M. Kwiatkowska, G. Norman, D. Parker, Stochastic model checking, in: M. Bernardo, J. Hillston (Eds.), Proceedings of the Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM), Springer, Berlin, Heidelberg, 2007, pp. 220–270. DOI: https://doi.org/10.1007/978-3-540-72522-0_6

[1] David D. Clark et al.: A knowledge plane for the internet. In: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications.ACM. 3–10(2003)

[2] Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet, P., Alarc´on, E., Meyer, D., Barkai, S., Maino, F., Ermagan, V., Coras, F., Latapie, H., Cassar, C., Evans, J., Muntes, V., Walrand, J., Cabellos, A.:Knowledge-Defined Networking. ACM CCR. (2016)

[3] Mestres, Albert et al.: Knowledge-Defined Networking. in arXiv:1606.06222, (2016); and in ACM SIGCOMM Computer Communication Review(2017)

[4] Albert Cabellos and Albert Mestres.KDN database (public datasets for ML), http://knowledgedefinednetworking.org/

[5] Rodriguez-Natal A., et al.: LISP: a southbound SDN protocol?, IEEE Communications Magazine, vol. 53, no. 7, pp. 201-207, (2015)

[6] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang.: Machine learning for networking: Workflow, advances and opportunities. IEEE Network. 32, 2 92–99(2018)

[7] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, L. J. Wobker, and B. Networks.: In-band Network Telemetry via Programmable Dataplanes, SOSR, pp. 2–3(2015)

[8] Y. Zhu, N. Kang, J. Cao, A. Greenberg, G. Lu, R. Mahajan, D. Maltz, L. Yuan, M. Zhang, B. Y. Zhao, and H. Zheng.: Packet-Level Telemetry in Large Datacenter Networks. in Proceedings of the ACM SIGCOMM, pp. 479–491(2015)

[9] P. Bosshart, G. Varghese, D. Walker, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, and A. Vahdat.: P4: Programming Protocol-Independent Packet Processors. ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 87–95(2014)

[10] Bruno Nunes Astuto et al.: A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. In:IEEECommunicationsSurveys and Tutorials vol.16, pp. 1617–1634 (2014)

[11] Diego Kreutz et al.: Software-Defined Networking: A Comprehensive Survey. InProceedings of the IEEE.Vol.103, no.1,pp. 14–76(2015),

[12] Oliner, A., et al.: Advances and challenges in log analysis. Communications of the ACM. vol. 55, no.2, pp. 55-61(2012)

[13] Clemm, A., et al.: DNA: An SDN framework for distributed network analytics. Integrated Network Management (IM), IFIP/IEEE International Symposium on. IEEE, (2015)

[14] Ghosh, Saptarshi.: A Cognitive Routing Framework for Reliable IoT Communication in 6G-SON. DOI :10.13140/RG.2.2.34212.50568 (2020)

[15] Alex M.R. Ruelas, Christian E. Rothenberg.: Load balancing method for KDN-based data center using neural network. Universidad de Lima, 87-97(2019)

[16] Open Networking Foundation .: Software-Defined Networking: The New Norm for Networks. ONF white paper, (2012)

[17] Q. T. A. Pham, Y.Hadjadj-Aoul and A. Outtagarts.:Deep reinforcement learning based QoS-aware routing in knowledge-defined networking. Proc. Qshine EAI Int. Conf. Heterogeneous Netw. Qual. Rel. Security Robustness, pp. 1-13 (2018).

[18] Xiaoqin Pan, Shaofei Tang, ZuqingZhu.:Multilayer Network Monitoring and Data Analytics over Encrypted Telemetry Data. IEEE 20th International Conference on Communication Technology (ICCT), pp. 1577-1581(2020)

[19] J. Hyun, N. Van Tu and J. W. Hong.: Towards knowledge-defined networking using in-band network telemetry. IEEE/IFIP Network Operations and Management Symposium (NOMS) , pp. 1-7,doi: 10.1109/NOMS.2018.8406169. Taipei, (2018)

[20] Edwin Ferney Castillo, Oscar Mauricio Caicedo Rendon, Armando Ordonez, LisandroZambenedetti Granville.: IPro: An approach for intelligent SDN monitoring. Computer Networks. Vol.170, 107108, ISSN 1389-1286 (2020)

[21] W. Lu, L. Liang, B. Kong, B. Li and Z. Zhu.: AI-assisted knowledge-defined network orchestration for energy-efficient data center networks. IEEE Commun. Mag., vol. 58, no. 1, pp. 86-92 (2020)

[22] V. Martínez, F. Berzal and J. Cubero.: The NOESIS open source framework for network data mining. 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), Lisbon, pp. 316-321(2015)

[23] V. Martínez, F. Berzal and J. Cubero.: NOESIS: A Framework for Complex Network Data Analysis. pp, 1–14(2019). doi:10.1155/2019/1439415

[24] NFStreamhttps://www.nfstream.org/docs/

[25] Ayoubi, S.; Limam, N.; Salahuddin, M.A.; Shahriar, N.; Boutaba, R.; Estrada-Solano, F.; Caicedo, O.M.Machine Learning for Cognitive Network Management.IEEE Commun. Mag.2018,56, 158–165.

[26] Boutaba, R.; Salahuddin, M.A.; Limam, N.; Ayoubi, S.; Shahriar, N.; Estrada-Solano, F.; Caicedo, O.M.A comprehensive survey on machine learning for networking: Evolution, applications and researchopportunities.J. Internet Serv. Appl.2018,9, 16.

[27] Chen, Z.; Wen, J.; Geng, Y. Predicting future traffic using Hidden Markov Models. In Proceedings of theIEEE 24th International Conference on Network Protocols, Singapore, 8–11 November 2016; pp. 1–6.

[28] Niyaz, Q.; Sun, W.; Javaid, A.Y. A Deep Learning Based DDoS Detection System in Software-DefinedNetworking (SDN).arXiv2016, arXiv:1611.07400.

[29] Zhang, J.; Chen, X.; Xiang, Y.; Zhou, W.; Wu, J. Robust network traffic classification.IEEE/ACM Trans. Netw.2015,23, 1257–1270.