Intent-Based Networking for Zero-Touch Optical Networking

Sima Barzegar, Marc Ruiz, and Luis Velasco*

Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

**e-mail: luis.velasco@upc.edu*

2023 23rd International Conference on Transparent Optical Networks (ICTON) | 979-8-3503-0303-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICTON59386.2023.10207264

ABSTRACT

The Intent-Based Networking (IBN) paradigm targets at defining high-level abstractions, so network operators can define what are their desired outcomes without specifying how they would be achieved. The latter can be achieved by leveraging network programmability, monitoring and data analytics, as well as the key assurance component. In this paper, we introduce the IBN paradigm and its application to optical networking, highlighting the benefits that Machine Learning (ML) algorithms can provide to IBN. Illustrative examples of intent-based operation are presented for proactive self-configuration and cooperative intent operation.

Keywords: intent-based networking, network automation.

1. INTRODUCTION

Software Defined Networking (SDN) defines a centralized control plane architecture with global network vision. At the optical layer, the SDN controller can achieve optimal routing for optical connections (lightpaths) at provisioning time and during reconfiguration [1]. Placed besides the SDN controller, a Monitoring and Data Analytics (MDA) controller was proposed in [2] to collect monitoring data, analyze such data, and make decisions (control loop). Such data analysis can be based on Artificial Intelligence (AI) / Machine Learning (ML) algorithms [3], which enable network automation solutions, aiming at reducing operational costs. Among the large number of use cases for autonomous optical network operation, three major categories covering the entire lifecycle of optical connections are highlighted in [2]: *i*) automation of connectivity provisioning meeting some performance; *ii*) dynamic network adaptation, which entails monitoring one or more network entities (e.g., an optical connection) and making decisions to achieve some target performance; and *iii*) degradation detection and failure localization, where the performance to be achieved is related, e.g., to availability metrics.

However, the drawback is the proliferation of individual control loops, which brings also complexity to network management. In addition, defining how to achieve operational goals is very complex. In this scenario, Intent-Based Networking (IBN) proposes a different approach, where intents are defined as high-level abstractions that allow network operators to define *what* are their desired outcomes, without specifying *how* they would be achieved [4]. This strategy reduces human intervention and paves the way to the application of AI/ML techniques. In an IBN environment thus, operators provide intents as inputs to guide content-based systems to implement them without human intervention. Intents allow to define the goals and outcomes and provide: *i*) data abstraction to avoid users and operators to take care of specific device configuration; and *ii*) functional abstraction to avoid users and operators being concerned with how to achieve the goals.

This paper summarizes the tutorial on IBN for optical networking in [5] including examples of intent-based operation.

2. TOWARD NETWORK AUTOMATION

Network automation has been long time envisioned. In fact, the Telecommunications Management Network (TMN), defined by the International Telecommunication Union in [6], is a hierarchy of management layers (network element, network, service, and business management), where high-level operational goals propagate from upper to lower layers. In the way toward autonomic adaptation to changes, while hiding intrinsic complexity to operators and users, the Internet Engineering Task Force developed the concept of Policy-Based Network Management (PBNM) [7]. PBNM separates the rules governing the behavior of a system from its functionality. In PBNM, high-level management *policies* are broken down into low-level configurations and control logic (*policy rules*) to ensure that the network provides the required services. Policies can be defined as a set of simple *control loops*; each policy rule consists of a set of events and conditions and a corresponding set of actions, where each condition defines *when* the policy rule is applicable.

The most extended PBNM architecture consists of four systems (Fig. 1a): *i*) the policy management tool allows operators to define and update policies and it translates and validates policy rules; *ii*) the policy repository that stores the policies; *iii*) a set of policy decision points, which interprets the policies, translates them into a device-specific representation, and triggers the execution of the related actions whenever they satisfy the specified conditions; and *iv*) the policy enforcement points running on a policy-aware node that executes the policies. The drawback of PBNM is solving conflicts that might arise within or among policies; conflict resolution requires some external system or iterations with operators and/or users.

The network management architecture has evolved with the development of the SDN concept that brings programmability to simplify configuration (it breaks down high-level service abstraction into lower-level device

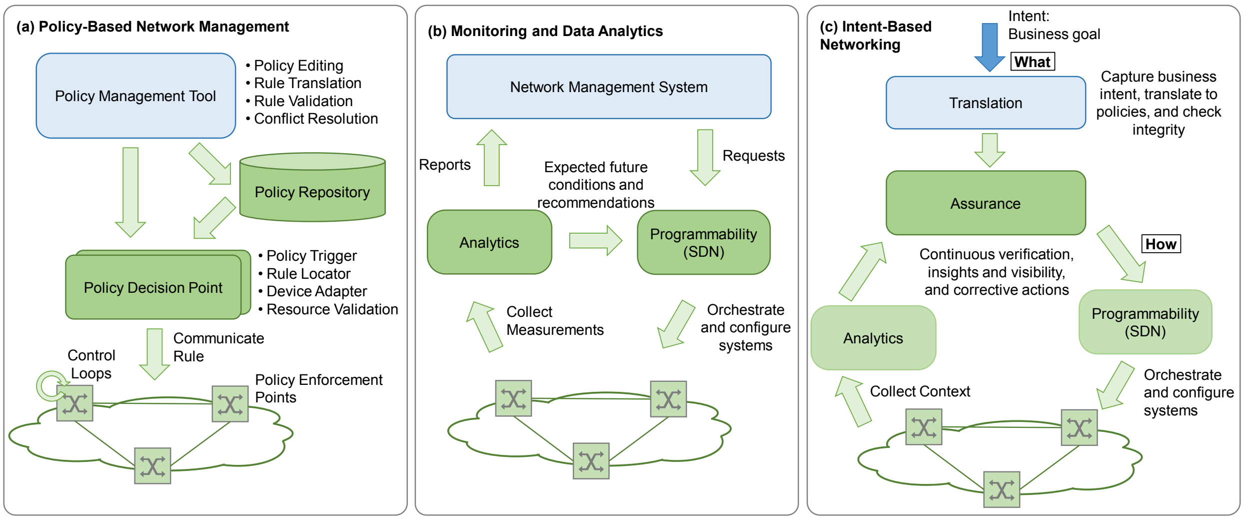

Figure 1. Steps toward network automation.

abstractions), orchestrates operation, and automatically reacts to changes or events. A data analytics system [2] can complement the SDN controller (Fig. 1b), so the network becomes proactive. Being proactive is of paramount importance, as the analytics system could anticipate anomalies and degradations before they cause major problems or become failures. Upon the detection, the analytics system can issue proper recommendations to the SDN controller, which can take the most appropriate action. Additionally, such analysis can be extended to forecasting network conditions that can be used to improve resource efficiency. In this architecture, control loops can be defined at various levels, from the device to the network, depending on the use case, as monitoring is collected and can be analyzed locally and/or network-wide. The drawback of this architecture is that the analytics system needs to combine information about services and the network itself, which, in practice, requires redesigning that and other control and management systems.

IBN complements SDN control and orchestration by allowing a declarative syntax while abstracting the operational process and focusing on behavior. Service definition can be based on templates to define resources and relationships for the service and allow specifying the Intent in terms of policy rules that guide the service behavior, specifying the applications, analytics and closed control loop events needed for the elastic management of the service.

A *translation* mechanism is needed to convert the intent into a network configuration to be automatically deployed within the network infrastructure and a set of policies that the IBN needs to verify that such policies can be executed (Fig. 1c). During the service lifecycle, the service *assurance* system makes sure that the network continues to deliver on that intent based on the specified design, analytics, and policies and with the help of ML algorithms. Intent-Based ML algorithms find the right knowledge and data to identify conditions with significant semantic value (insights) from raw telemetry, without being explicitly programmed. Actionable insights and rich context together with policy-driven closed loops can take automated actions whenever the network deviates from the intent. Reporting is intended to generate descriptive outputs, e.g., statistical summaries, as well as knowledge transfer of main key performance indicators of the service. Differentiated reports can be generated, so applications can reconfigure policies to adjust to service requirements and the network management can gather knowledge transferred for different services and processed jointly to improve actions [8], [9].

3. USE CASES

Autonomous network operation reduces human intervention related to the configuration of the network. Such operation requires collecting performance measurements from the network and developing intelligent algorithms that make decisions proactively to reach some performance defined for each network entity. A related concept is that of independent operation vs coordinated operation. Independent operation occurs when the decisions that are made on a network entity are based on measurements collected for the same entity. However, since in a network infrastructure many entities are sharing the set of common resources, pure independent operation is rare, as it can lead to overall suboptimal resource utilization and even to result in poor performance because of the natural competence for resources. Therefore, some kind of coordination among entities should be devised.

Proactive Self-configuration

Autonomous network operation can be *reactive* (i.e., in response to events) or *proactive* (i.e., acting ahead of time). Let us illustrate the difference with an example, where a packet connection (*PkC*) is established and conveys a traffic flow with unknown traffic characteristics. Our target here is to allocate just enough capacity to ensure the required performance, which would optimize resource utilization. However, every different PkC supports services with different operational goals in terms of delay and throughput (e.g., keeping the total delay below a given maximum, or minimizing the capacity while ensuring zero packet losses, etc.), and so, the tailored capacity dimensioning is required.

Imagine that a policy-based management based on a fixed threshold (e.g., defined in terms of the ratio traffic volume over capacity) is set to operate the capacity of a PkC. Note that such operation can be highly reliable and it is based on a specific rule that is easily understood by human operators. However, deciding the value of the threshold requires knowledge of the traffic: *i*) a high threshold value (e.g., 90%) would result into poor performance coming from high delay, and it can be worse when the variability of the traffic is high; and *ii*) a low threshold value (e.g., 60%) would result into poor resource utilization. Therefore, some traffic analysis would be required. Further, since traffic characteristics can change over time, such analysis need to be continuously performed to change the operating model, when needed.

When PkCs are routed on top of virtual networks, where virtual links (*vLink*) are supported by the optical layer, capacity might not be instantly allocated. Let us illustrate this problem with an example. Fig. 2a shows two PkCs (DC1-DC4 and DC2-DC3) that are established on top of a virtual network. Packet nodes are connected through vLinks, each supported by lightpaths on the optical layer. To minimize overprovisioning, such capacity is dynamically adjusted, thus enabling the dynamic vLink capacity management, e.g., by establishing and releasing parallel lightpaths between the end packet nodes.

Note that modifying the capacity of a PkC entails programming some rules in packet nodes and new capacity becomes immediately available. In contrast, adding more capacity to the vLink entails establishing a new lightpath, which requires some time (e.g., one minute). Therefore, vLink intents must make decisions with enough time to guarantee capacity availability. Such time depends, among others, of the packet traffic variation and thus, the value of the configured threshold could result into high delay and packet loss. The inner graph for PkC DC2-DC3 in Fig. 2a shows the capacity adjustments performed assuming that the operational goal of the PkC is to minimize the allocated capacity to reduce connectivity costs, by following as close as possible the input traffic, while avoiding traffic loss. Fig. 2b-c present two alternative approaches to operate the capacity of the PkCs, based on a simple threshold rule or based on an intelligent ML-based algorithm. Every connection (PkC or vLink) intent agent collects the amount of input traffic that is injected to the connection, as well as some other measurements, like packet loss and delay, and it determines the capacity of the connection that will be needed to meet the given operational goals for the next period (e.g., one minute). Such capacity can be used to program some rules in the packet nodes not only to increment the capacity but also, e.g., to adjust the amount of buffer at the input of the connection.

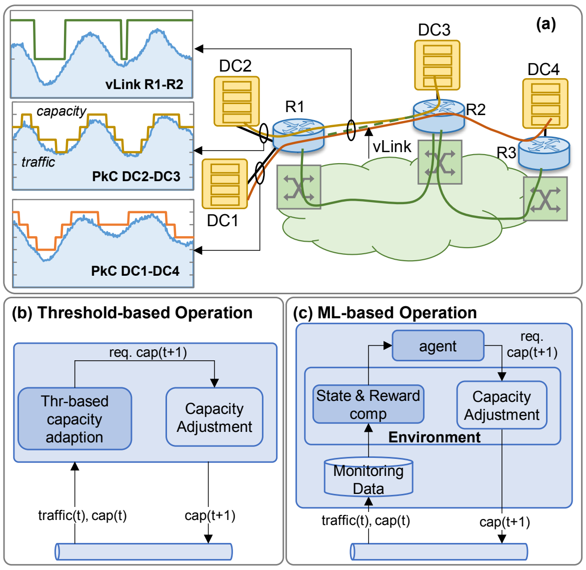

Figure 2. Capacity operation of PkCs and vLinks.

Cooperative Intent Operation and Transfer Knowledge

Although PkCs and related vLinks can work independently, making decisions based on the observed input traffic, some coordination might facilitate the overall operation. For instance, as a result of the capacity required by the PkCs, the capacity of the vLink needs to be reconfigured, as observed in Fig. 2a. Nonetheless, if the available capacity of the vLink is exhausted, competition for the available capacity of the vLink would lead to poor performance for both PkCs.

A possible solution to avoid conflicts and countereffects between intent agents competing for common resources is to consider cooperation among them to ensure that they can achieve their operational goals. To illustrate such coordination, let us consider the multilayer scenario in Fig. 2a. We assume that PkCs have different objectives. On the one hand, PkC DC1-DC4 requires that the maximum end-to-end delay is not violated, whereas PkC DC2-DC3 requires minimize overprovisioning. In spite of the subtle difference in the plots in Fig. 2a between both PkCs, the capacity of DC1-DC4 is always large enough with respect the input traffic to ensure that the delay added by the time spent in the queues is under the given maximum. Note that the capacity of DC2-DC3 is kept closer to the actual traffic. Considering the capacity requirements from PkCs, vLinks can be easily managed; the capacity of vLink R1-R2 varies after adding or releasing one lightpath to adapt its aggregated capacity to the PkCs requirements, which motivates intent coordination.

To manage the capacity of the entities, the architecture in Fig. 3 supports a hierarchy of intents, where each intent agent is in line with that in Fig. 2b-c. In the case of vLink intent agents, they receive as input the aggregated amount of input traffic in the vLink, its actual capacity, as well as the total capacity that PkCs will require for the next period, and are in charge of managing the vLinks capacity by establishing and tearing down lightpaths. Besides, there is some knowledge that can be transferred from PkC intents to vLink intents, which cannot be anticipated by means of monitoring the (aggregated) traffic in the vLink. Knowledge that can be transferred include: *i*) traffic models for the PkC; *ii*) sudden capacity increase due to customer operational decisions (e.g., a pre-planned increase of productivity of a factory can lead to data traffic increase); or *iii*) PkC rerouting requiring new connectivity to be supported by the underlying network. This knowledge could be used by vLink intents to increase the capacity or, on the contrary, reject the request if no resources are available. Note that such rejection would be informed back to PkC intents, which will use knowledge to reformulate their decisions and for finding alternatives to achieve the operational objectives.

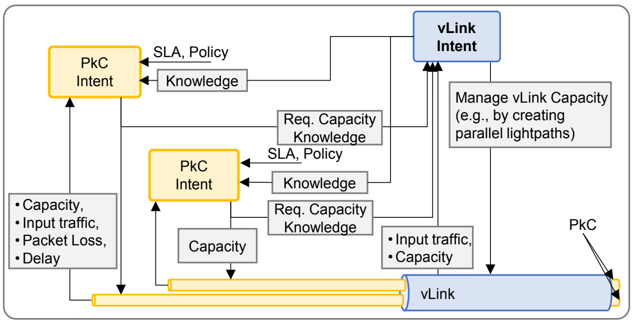

Fig. 3. Intent cooperation and transfer knowledge.

4. BACKGROUND ON ADVANCED ML TECHNIQUES

Many intent-based solutions need from ML techniques as a way to implement proactive approaches. In this section, we just summarize advanced ML techniques that can be used in the applications presented in Section 3.

Time series forecasting covers those methodologies that predict future events as a function of previous observations. Traditionally, Autoregressive Integrated Moving Average (ARIMA) models have been proposed for time series forecasting, due to several key characteristics, such as easiness of interpretability and the ability to provide probability distributions of the predicted events. Deep learning techniques can be applied to predict complex future events without considering strongly limiting assumptions. In particular, the use of feed-forward neural networks (FFNN) allows considering complex nonlinear relations among input features and the predicted future event. Although FFNN can be designed and trained to predict time series events, they were devised for applications that do not depend on time. On the contrary, Recurrent Neural Networks (RNN) have been proposed specifically to deal with time series events, since they can explicitly manage the ordering among inputs. RNNs implement knowledge persistence, so it can be used for predictions. However, in general, this memory is short and knowledge vanishes with time. To improve RNNs, Long Short-Term Memory (LSTM) networks were proposed to expand temporal dependence learning. LSTM units consist of a set of different complex gates, namely input, output, and forget gates and the coefficients of the network are dynamically managed to keep long term memory. LSTMs provide accurate prediction of time series with complex temporal correlation, e.g., periodical sharp changes. Reinforcement Learning (RL) considers the paradigm of an intelligent *agent* that takes actions in an *environment* (as in Fig. 2c). At every time step t, with a given state s, the agent selects action a with respect to a policy, and it receives from the environment a reward r and the new state s'. The objective is to find the optimal policy that maximizes a cumulative reward function. RL fits perfectly as part of intent agents, as the related problems can be usually stated in the form of a Markov decision process.

5. CONCLUSION

The IBN paradigm was introduced focused on its application to optical networking. IBN allows network operators to define *what* are their desired outcomes without specifying *how* such outcomes would be achieved. IBN can be fueled by the use of ML algorithms. Illustrative examples of intent-based operation that use ML techniques were introduced.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the H2020 B5G-OPEN (G.A. 101016663) and the AEI IBON (PID2020-114135RB-I00) projects and from the ICREA institution.

REFERENCES

[1] L. Velasco *et al.*, "In-operation network planning," *IEEE Communications Magazine*, vol. 52, 2014.
[2] L. Velasco *et al.*, "Monitoring and data analytics for optical networking: Benefits, architectures, and use cases," *IEEE Network Magazine*, vol. 33, 2019.
[3] D. Rafique and L. Velasco, "Machine learning for optical network automation: Overview, architecture and applications," *J. of Optical Comm. and Networking*, vol. 10. 2018.
[4] A. Clemm *et al.*, "Intent-Based Networking - Concepts and Definitions," IRTF draft work-in-progress, 2021.
[5] L. Velasco *et al.*, "Intent-based networking for optical networks," *J. of Optical Comm. and Networking*, vol. 14, 2022.
[6] "Principles for a Telecommunications Management Network," ITU-T, Rec. M.3010, 2000.
[7] J. Strassner, *Policy-Based Network Management: Solutions for the Next Generation*, Morgan Kaufmann, 2004.
[8] M. Ruiz *et al.*, "Knowledge management in optical networks: Architecture, methods and use cases [Invited]," *J. of Optical Comm. and Networking*, vol. 12, 2020.
[9] F. Tabatabaeimehr *et al.*, "Cooperative learning for disaggregated delay modeling in multidomain networks," *IEEE Transactions on Network and Service Management*, vol. 18, 2021.