2023 32nd International Conference on Computer Communications and Networks (ICCCN) | 979-8-3503-3618-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCN58024.2023.10230112

Autonomous Network Assurance in Intent Based Networking: Vision and Challenges

Aris Leivadeas
Dpt. of Software Engineering and IT
École de Technologie Supérieure (ÉTS)
Montreal, Canada
aris.leivadeas@etsmtl.ca

Matthias Falkner
Global Service Provider Architecture
Cisco Systems Inc.
Ottawa, Canada
mfalkner@cisco.com

***Abstract*—Intent Based Networking (IBN) is a new paradigm that promises to create autonomous networks that can comply to high-level intents of network users and exhibit self-adaptation and self-optimization properties. The major IBN component to achieve this is network assurance. The assurance has as a goal to autonomously trigger corrective actions, whenever the conditions of the network do not allow to fulfill the performance requirements of its users. Accordingly, network assurance is expected to be largely based on monitoring and telemetry processes running conjointly with state of the art Artificial Intelligence techniques that will set off remedy solutions to bring the network into a compliance state. Given the importance of IBN and network assurance into the designing of autonomous networks, this paper introduces the main architectural components and technologies needed to achieve this visionary evolution of next generation networks. Particular emphasis is placed on how the network can interact with the end users and network operators in an easy way as mandated by the IBN premises. Finally, light is shed on the open challenges and future directions of this novel but unexplored research topic.**

***Index Terms*—Intent Based Networking, Network Assurance, Network Automation**

I. Introduction

Telecommunication networks have seen a tremendous evolution over the last two decades. A plethora of new applications and network services are continuously emerging, while new network technologies such as 5G and Internet of Things (IoT) are replacing the traditional service offering. In this new paradigm, the network is seen as a commodity bringing closer the users, the service providers and the network operators [1]. At the same time, applications can be very differentiated, with strict performance guarantees, and highly personalized in order to target individual users, group of users, enterprises or even vertical markets.

Nonetheless, this (r)evolution entails a significant increase of the network infrastructure with the addition of new network equipment from various vendors covering different network functionalities. Thus, network operators are finding themselves in a real predicament of how to efficiently and timely configure the infrastructure to support an unprecedented number of applications. Furthermore, given the fact that networks are dynamic in nature due to fluctuating communication conditions and varying user behavior, often reconfigurations and adaptations of the network are inevitable. Until now, this network refitting is being done through hectic and error-prone manual configurations using vendor specific command line orders and scripts. Hence, the new era of telecommunications cannot sustain such cumbersome configurations, laying the first stone for new autonomous network mechanisms.

It is to be noted that this is not a new realization but a scheduled shift of how networks should be operated and behave in the future, as introduced by IBM in the early 2000's [2]. Accordingly, the survival of the networks in the dawn of an ever increasing infrastructure with myriads of applications posing a vast range of requirements, relies on exhibiting the necessary self-* properties (i.e., self-optimization, self-healing, self-protection, etc.) [3]. Recent network softwarization technologies, such as Network Function Virtualization (NFV) and Software Defined Networking (SDN) made the first step towards automating the network configuration, nonetheless a lot of manual processes are still required for their operation [4].

Having automated processes is the first step towards reducing human intervention. However, to eliminate as much as possible the manual configuration, networks should move from an automated phase to an autonomous and self-governing one. Intent Based Networking (IBN) is such a paradigm that envisions creating a fully autonomous network able to self-adapt, facilitating the use of network and reducing the knowledge gap between simple end-users and network operators. To do so, IBN allows users to express what they want from the network through declarative and high-level requests that are called “intents”, instead of how to do it [5]. Following, through a set of controllers and orchestrators the intents are translated into technical policy configurations and finally deployed over the network fabric.

An inherent part of IBN and a building block towards the creation of an autonomous network is the “assurance” part. Intent assurance has as a goal to guarantee the correct behavior of the network for the whole lifecycle of the intent [6]. Specifically, the assurance can monitor the status of the network and the level of quality offered to the various deployed intents while responding to volatile communication conditions. For

This work was supported in part by the Cisco University Research Program Fund of Silicon Valley Community Foundation 2021-234759 (3696)

979-8-3503-3618-4/23/$31.00 ©2023 IEEE

the latter, corrective actions should be automatically triggered whenever an intent performance drift is noticed. The actions should be triggered either proactively or reactively, allowing the network to self-update and returning back to compliance [7].

To this end, in this paper, we aim to introduce a complete network assurance architecture in the context of IBN, identifying the main assurance components towards enabling an autonomous network that will be able to react to the changes of the network or the intent itself. Particular emphasis will be placed to the fact that the intent can be expressed or modified either from an end-user or from the operator, which can create conflicting conditions between the intent requirements and the network state.

The remainder of the paper is organised as follows. Section II provides the enabling technologies that constitute the IBN architecture. Section III introduces the principal components of the network assurance and how the IBN concept can be leveraged towards enabling network automation. The open challenges and future direction are discussed in Section IV. Finally, Section V concludes our paper.

II. Background and Enabling Technologies

A. *Intent-Based Networking*

A user intent, as defined by the Internet Engineering Task Force (IETF), refers to a set of operational goals and outcomes expressed in a declarative way without detailing how to implement them [5]. Accordingly, an intent-based network creates a unique opportunity for any user regardless their network knowledge to interact with it, and express their needs in a more human-friendly way. For instance, an enterprise user may request the following network service “All my outgoing traffic should be secured”. Similarly, a data center operator may express an intent as “Do not allow overloading of the data center servers”. More complex scenarios may arise when a vertical asks for an application i.e., “Provide a reliable and low latency communication for a telesurgery that will happen in room 52 tomorrow morning at 9am”.

From the above intents, it is easy to understand that the user may give different levels of details depending on the business domain, however all of the intents present a high-level and declarative description of what they expect from the network. Following, the intent needs to be somehow transformed into a representation that can allow the network to be configured according to the users’ requirements. For that and as shown in Fig. 1 an IBN System will entail the following five components: i) Intent Profiling, ii) Intent Translation, iii) Intent Resolution, iv) Intent Activation and v) Intent Assurance [8].

Intent Profiling refers to the way the users can issue their high-level intent. Several approaches exist, such as filling out a template, using natural language, or using an IBN specific language [9]. Intent translation emphasizes on the refinement of the intent and its transformation to a network policy that can include the technical details of the configuration [10]. Following, the intent resolution component tries to identify if two or more intents are conflicting with each other and how to resolve such conflicts [11]. After these first three steps, the translated intent would be ready to be activated by deploying the respective policies and services to the network [12]. Finally, the intent assurance part, as the most critical component will be responsible for monitoring and guaranteeing that the network will comply with the intent’s requirements throughout its lifetime [13].

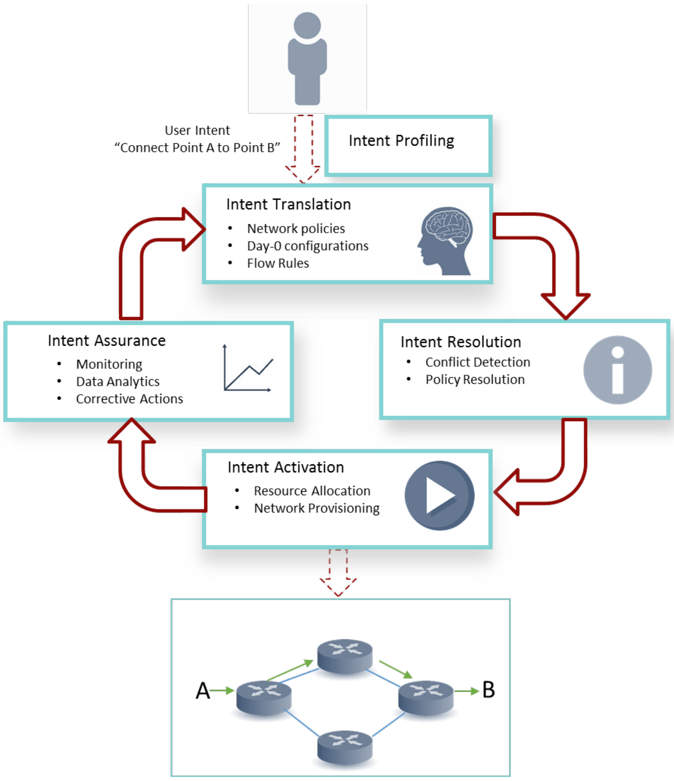

Fig. 1. IBN Architecture

Thus, a Closed-Loop Automation (CLA) is created that starts from accepting a high-level intent and finishes by continuously monitoring and assuring the deployed and translated intent. However, the component that manages to close this loop is the intent assurance, which promises to automatically detect any performance drift and autonomously find corrective actions.

B. *Network Softwarization*

IBN does not envision to create a complete new network paradigm. In contrast, it will be based on recent trends and efforts for network automation. The two most promising network technologies are expected to be NFV and SDN. Both these technologies have their own reference architectures and automation platforms that IBN could benefit from.

More specifically, SDN, through the separation of data and control plane, managed to aggregate the necessary intelligence into a centralized SDN controller that can enable network reprogrammability in an automated way. The SDN’s reference architecture consists of the infrastructure, control, and application layer [14] and enables the central controller to make intelligent forwarding decisions that are pushed to forwarding devices through SDN enabled interfaces and protocols. Hence, SDN can be seen as an indispensable tool for the automatic network management by providing the necessary agility and programmability. The SDN community has already made efforts to incorporate IBN capabilities, by adopting its north bound interface to accept intent expressions that can be later translated into flow rules and deployed into the forwarding devices [15].

Similarly, NFV leverages virtualization to softwarize network services by disassociating the network functionalities from the underlying hardware. Thus, through NFV, network functions or services (also called Service Function Chains) can be instantiated on demand and according to the user requirements, on top of generic servers and network endpoints [16]. NFV also comes with a reference architecture that comprises of the Virtual Network Functions (VNFs), NFV Infrastructure (NFVI), and NFV Management and Orchestration (MANO) [17]. The reference architecture can facilitate the automation of the configuration and provisioning of network services on top of the available infrastructure in a flexible and cost-efficient way. This is particular important for IBN, since intents can often be translated into network services that can be represented by VNFs.

C. Artificial Intelligence

Network Softwarization was the first approach of introducing automation aspects for the network management and orchestration. Nonetheless, it is clear that something is missing to have a successful transitioning from an automatic to an autonomous network, such as the one envisioned by IBN. At the same time, the increasing momentum and advances in the field of Artificial Intelligence (AI) and Machine Learning (ML) in every aspect of our social life, attracted the interest of the whole academic and industrial community, that sought to incorporate AI capabilities to a large variety of research domains.

Inevitably, AI has also attracted the interest of the information technology domain as well. In particular, AI and especially machine learning and machine reasoning through the analysis of network data could provide significant insights into the network performance, as well as into the detection of various networking problems [18]. Accordingly, ML techniques are expected to be one of the main building blocks of network assurance that will guarantee the autonomous behavior of the network by correlating the current state of the network with the performance levels offered to the end-users.

As stated in the Introduction, network conditions and workload generation may fluctuate according to the utilisation of the infrastructure and the behavior of the end-users. This could result in a deviation from what the intent wants from the network and what the network can actually give to the user as expressed in the intent. Hence, AI techniques could be used to analyze the dynamic nature of network conditions and how to guarantee the performance under a volatile environment [19]. Furthermore, they could be used to learn the user behavior through access patterns, statistical data, while associating the network performance with customized Quality of Experience (QoE) levels [20].

More particular, machine learning could be used for Service Level Agreement (SLA) violations prediction, resource utilisation forecast, anomaly detection, and root cause analysis, among others [21]. Additionally, in the context of IBN and Intent expression, AI through the form of Natural Language Processing (NLP) can be used to translate the user utterances into network configurations. Thus, various branches of AI can be used in many components of the IBN architecture. However, it is expected that intent assurance could be benefit the most from AI, in order to distill knowledge, gain operational insights, and participate in recommending corrective actions.

III. IBN-enabled Network Assurance Architecture

In this section, the main architectural components are presented along with their interactions towards providing the basis of the main functionalities of the intent assurance. Fig. 2, illustrates the envisioned architecture, which takes as input the translated intents from the IBN Manager. To not deviate the discussion from the assurance component and for reasons of simplicity, we assume that the IBN Manager will execute the intent translation, resolution and activation components. Interestingly, the first observation made by noticing Fig. 2 is that a second Closed Loop Automation is formed (the first one is depicted in Fig. 1) by continuously monitoring and intervening on the underlying infrastructure. In the following, more details are given regarding each identified component of this new CLA and the interaction between the intent assurance with the end-users and the network administrator.

A. Intent Expression

The issuer of the intent may be of different background, such as an enterprise employee, an application developer, a network operator/administrator etc. Thus, different level of details are expected in the intent. The intent translation component will be responsible for transforming the declarative intent into a low-level network policy. This policy will have the details of how the network should be configured, and from this policy the intent assurance will have to infer what exactly should be delivered to the user.

When an intent comes from an end-user it will usually contain a network service description/application, with some high-level qualitative parameters i.e., “I want a videoconferencing application with high QoS from 1pm to 2pm”. This could be translated into a tuple of specifying the IP address of the user as the source of the flow, the IP address of the video conferencing application as the destination of the flow, and some pre-specified thresholds of how the network perceives the high/medium/low levels of QoS. The thresholds could be specified by the network administrator according to the network capabilities and a long-term analysis of the network

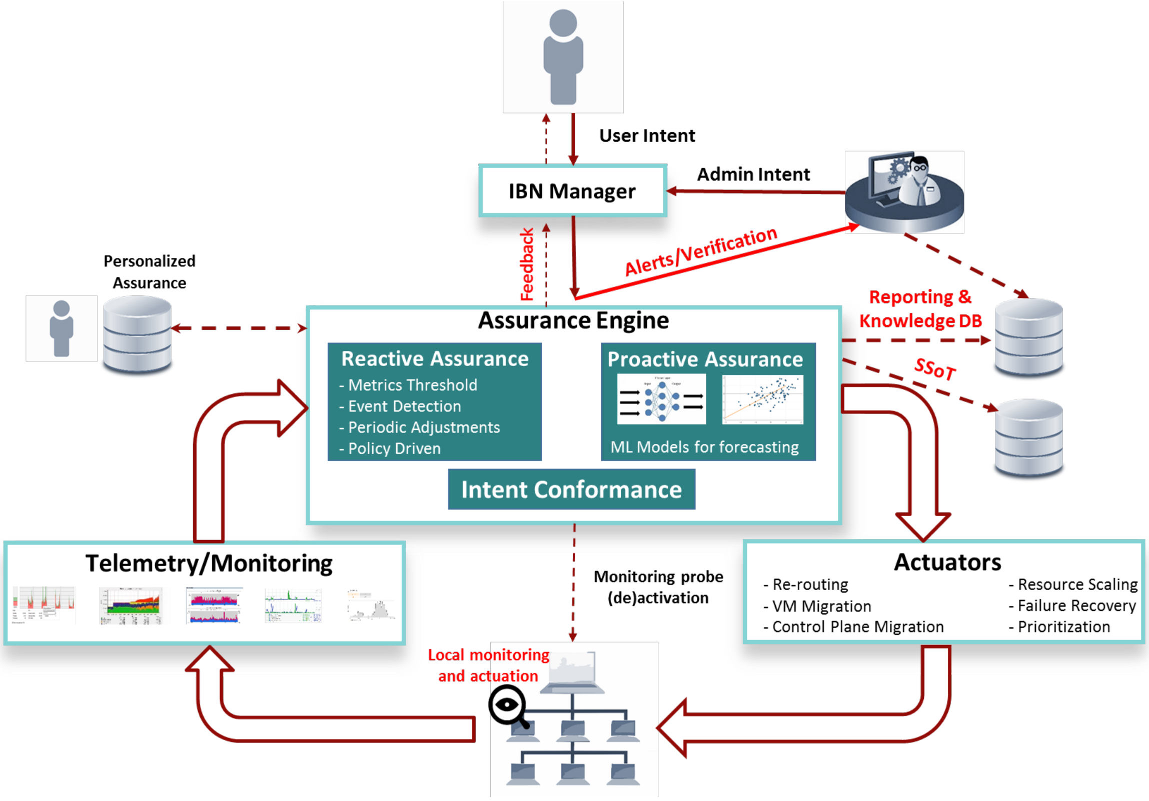

Fig. 2. Intent Assurance Architecture

behavior. However, since the network services and applications are consumed by humans there is a certain subjectivity involved. In other words, the intent assurance may validate that the performance is adequate (i.e. within some limits) but the user may perceive that the network configuration is adversely affecting its requested service. Hence, the intent assurance, may be equipped with a personalized interface that will allow the user to provide feedback on the level of satisfaction of the consumed service. At the same time, the feedback can be stored and provide different levels/thresholds of how the qualitative requirements can be more accurately translated into quantitative network configurations for particular users. This human intervention should not be mistaken as removing the autonomicity of the network. In contrast, it can be considered as an additional input to the intent assurance component that will allow it to be trained more correctly, minimizing any ambiguities that may arise.

The intent may also be issued by a network administrator and most importantly it may express a network assurance policy. For instance, the administrator may request “I want to perform load balancing in the infrastructure” or it can be much more specific as “Monitor the CPU utilisation of the Web server and redirect new requests to the back up server when utilisation exceeds 80%”. Obviously, in this subset of intents we expect to see more details, and the translated policies will be less ambiguous for the intent assurance component. Additionally, in the last example, the administrator specified the type of correction that should be performed when a specific event happens. In other words, the intents of the administrator could look much closer to an Event-Condition-Action (ECA) policy [5]. However, sometimes the intent may not include the action. For example, the previous intent could be “The CPU utilisation of the web server should not exceed 80%”. In this case the intent assurance should automatically find a corrective action. Furthermore, the problem maybe extremely critical (i.e. a node failure) and the corrective action can significantly impact the already deployed intents. In this case the intent assurance component will have to send alerts to the network administrator and/or request the verification of the administrator before applying the necessary changes to the network.

B. Monitoring and Telemetry

Monitoring is one of the most important functionalities of the intent assurance. Through this network detection mechanism any performance drift or network anomaly could be revealed, triggering the rest of the functionalities to take actions. Due to the importance of knowing the status of the network, the monitoring module should continuously scan the network infrastructure [22].

Even though there are many monitoring tools available (i.e. Netflow, SFlow, SNMP, etc.) [23], the very first thing that the intent assurance should decide is what kind of monitoring probes will have to be activated and where. Otherwise, the network infrastructure will be overloaded with monitoring data, which in turn could affect the performance of the already deployed intents. Hence, the intent assurance will have to activate the appropriate monitoring probes to key points of present (PoP) of the network that will reflect where the intent should be assured. Similarly, the probes should be deactivated when the intent reaches its lifetime. However, a more static approach could be followed for some monitoring probes and for intents that do not follow a temporal behavior but in contrast should be constantly in place (i.e., intents that reflect permanent assurance policies of the network administrators).

Another important aspect is the way and frequency the monitoring data are collected. Usually, a publish-subscribe model is envisioned [4], where only relevant data from specific network elements in key PoPs are gathered. These data could follow a periodic, an event-driven, or a poll-based approach in order to collect a sufficient amount of knowledge of the network state. Finally, the way monitoring data can be collected can reflect how network intrusive this operation will be. Accordingly, there may be passive methods (i.e., tcpdump, ipfix, etc.), active but more bandwidth hungry methods (i.e., ping, one or two way active measurement protocol, etc.), or hybrid techniques to be followed [24].

With respect to the type of metrics to be monitored, a vast range of possibilities exists that depends on the type of the intent, the network/domain, and the capabilities of the infrastructure. For instance, in access networks, the intent assurance could collect information regarding user mobility, user density, interference, signal strength, signal to noise ratio (SNR), etc. [4]. For data center networks, monitoring may gather statistics of CPU and RAM utilization, storage, I/O, etc. [13]. Alternatively, for optical networks, data could be drawn reflecting the status of Quality of Transmission, optical power, optical SNR, etc. [25]. Nonetheless, there are more generic metrics that could be gathered that characterize the performance of the communication, such as throughput, delay, round trip time, jitter, packet delivery/error ratio, etc. [26], or the status of a network equipment, i.e., number of received or transmitted packets, bytes, and errors per Network Interface Card (NIC), size of queues, etc. [6].

C. Assurance Engine

After collecting the necessary data extracted by the monitoring tools, the following step is to analyze them in order to detect any kind of intent performance drift. Hence, the assurance engine component will be responsible on utilizing appropriate AI models to identify any anomalies and extract the necessary knowledge that will help on the corrective actions decision making. The way that the assurance engine will behave depends on whether it will wait a drift to occur and reactively detect it (reactive assurance) or if it will try to predict performance drifts through ML techniques before they happen (proactive assurance).

Regarding reactive assurance, the assurance engine may dictate certain actions to be taken, when a specific threshold is met (i.e., throughput or CPU/RAM exceeds a certain value), an event is detected (i.e., node/link failure) or a specific condition is met that is specified in the translated policy of an intent (i.e., "switch routing to a lower bandwidth path outside business hours to save operational costs"). Alternatively, policies could be updated in a periodic fashion according to the network state (i.e., "check resource utilisation from monitoring and redistribute load to achieve load balancing every hour").

Proactive assurance is more appropriate in dynamic scenarios, when the network state has frequent changes and the assurance engine will have to predict such variations that may lead to performance degradation. Thus, through a proactive strategy the assurance engine could ensure that the intent will indeed comply with the user's requirements throughout its lifetime. This strategy is expected to be heavily based on ML algorithms that can leverage the monitoring data and use them as input features for training and validation. Specifically, by monitoring and by keeping logs of the network's Key Performance Indicators (KPIs) could help creating time-series datasets that could be used to train various machine learning and deep learning (DL) methods to predict the trends of network changes and/or user behavior [20]. The ability to predict these changes, that could compromise an intent's performance, is of utmost importance in order to guarantee the service continuity that is an inherent requirement of modern applications. For instance, by predicting a link or VM overutilization could lead in planning ahead the corrective actions, in contrast to a threshold-based reactive approach that could entail several seconds to minutes to apply a remedy solution.

Nonetheless, it is difficult to have a prediction model that could forecast the future behavior of the network or an intent with a 100% accuracy. Thus, it is important to have a hybrid approach, where the proactive assurance will be complemented with a reactive approach to account for any mispredictions that may occur. Apart from that, another approach that could be integrated into the assurance engine is the intent conformance, where through periodic sanity checks the validity of the intent's behavior can be verified [27]. This could include polling active measurement protocols to test the performance of the communication or simply to ensure that the communication between the two endpoints specified in the intent is active.

D. Actuators

After detecting an intent performance drift, the next step is to actuate the corrective actions. Once more, the actions are tightly coupled with the type of the network and its capabilities. As an example, in an access network the remedy actions may involve a new scheduling decision, increasing/decreasing allocated resource blocks, initiation of handoff mechanisms, etc. In a Cloud environment, the assurance actuators may trigger VM/container migrations to a different server or data center, and/or appropriate scaling of the resources that could account for the dynamic change of the intent's requirements or the network conditions.

Other actions that could be followed in a more generic networking environment is flow migration or rerouting. These actions could be applied in any type of network (carrier, enterprise, optical, cloud, etc.), when the performance of a link or a path is or expected to be affected as deduced by the assurance engine. Other network problems related to fault recovery may yield similar actions as for example node/link failures, which can trigger VM/flow migration or even control-plane migration (when a network controller faces an issue) [8].

The intent assurance could also dictate the re-evaluation of priorities between the intents by reclassifying less-important intents and provide remedy solution only to intents that include high reliability requirements [18]. This could be considered as a last resort, when not all affected intents can be assured. In this case, the intent assurance component should generate alarms to notify the users and the network administrator.

Until now, the CLA of the assurance has been presented as a centralized and global approach that collects, analyzes, and acts upon a network domain. However, to increase local efficiency, smaller and local CLAs could be applied on a network element level or on a group of elements in a distributed fashion. For instance, critical network elements that could be proven to be vital for the whole network operation (i.e., single-points of failure) may be enabled with local monitoring and actuation to reduce any significant delays of a central decision making process.

Finally, by observing this newly created CLA depicted in Fig. 2, that consists of monitoring, analyzing and triggering actions, on a high level we could deduce that the intent assurance resembles to an IoT system. In other words, the whole network operation is digitized, creating through the intents a direct communication channel between humans and the network itself, while allowing the network operators to create business insights by offering intelligent network services and taking smarter decisions. For the latter, a knowledge database could be created enhanced by reporting services as it will presented in the following subsection.

E. Reporting & Knowledge Database

Whenever an action is made from the intent assurance component, a report should be generated that can be stored in a database (DB). Along with the report, a snapshot of the network conditions before and after of the intent drift could be also stored for future reference. This information could be used later on from the network administrators for further analysis that could help them refine their intents and policies. This can help them gain more knowledge and insights of the proper network operation, how new intents could lead to less disruptive actions, and how to bring closer the user intents with the network capabilities.

The reporting & knowledge DB could also assist with Root Cause Analysis by finding the actual problem that led the network to self-adapt. This information can be used to retrain AI models to provide faster identifications and prompt application of remedy solutions before non-reversible problems occur. In other words, the network state and root cause can be correlated generating alarms and triggering remedial actions beforehand. Additionally, the problem identification could be classified according to its criticality, in which part of the network appeared, for how long, and what was the impact on the intent's performance drift.

The reports could also facilitate to create long-term statistics of the utilisation of the network by providing average, cumulative distribution functions, standard deviation and ranges of key network resources [19]. The particular knowledge could be used for intent recommendation, network optimization and network planning. For example, the assurance may reveal that some intents are infeasible i.e., "I want ultra low latency and strong encryption". The IBN manager may perceive that the particular intent can be feasible and translate the requirements into a communication delay of $1ms$ and the use of an IPSec function. However, the use of an IPSec may yield sufficient delays that cannot meet the $1ms$ requirements. In this case, the assurance could recommend the use of lighter encryption methods, while the reporting of these actions could facilitate the translation mechanism or providing alternative recommendations when a user express an intent that is infeasible.

Furthermore, the long-term statistics could reveal information regarding traffic growth and the correlation of expected workload with the resource availability. This could lead to change the objectives of the intent activation to perform a better optimization of the resource usage to accommodate more intents at the same time. Additionally, if this is deemed to be impossible, the intent assurance component could recommend appropriate network planning actions to the administrators (i.e. addition of resources or network equipment) for brownfield network development and planning [28].

F. Single Source of Truth

A final functionality that the intent assurance may be equipped with is the Single Source of Truth (SSoT). SSoT has been recently included in the definition document of IBN by the IETF. Specifically, according to IETF, SSoT includes the set of validated intent expressions and the records of their operational states [5]. Hence, SSoT could be used for the intent assurance in order to compare any deviation between the current state and the desired state. Following, if any difference is noticed between those two states, a performance drift could be detected triggering the necessary actions.

Another way that the SSoT could be used is in more critical scenarios. For example, when a newly deployed intent significantly and adversely affects the network performance, the system could consult what was the SSoT before the activation of the intent and perform a rollback [18]. In more extreme cases, when there is a critical communication problem and the assurance engine cannot find a possible and quick action to take, the intent assurance component could try to self-heal the network by reverting into the SSoT state [29]. Hence, the SSOT DB could contain several validated operational states of the network and their configurations, allowing the intent assurance to rollback to the state that best describes the intents' collective requirements.

IV. Open Challenges

A. Intent Expression and Assurance

The intent is expected to contain a certain level of ambiguity and especially for users with lower knowledge, the interpretation of the intent from an IBN system can be equivocal. This can make the intent assurance a difficult process. For example, when a performance drift is detected, the selected corrective actions may not necessarily guarantee the user intentions. Alternatively, the intent assurance may trigger corrective actions when they are not actually needed. Additionally, a wrong interpretation of the desired outcome may lead into the activation of the wrong monitoring probes, creating thus a chain of network misconfigurations. As stated in the previous section, this could be partly avoided by the use of a personalized assurance, where assurance policies can be associated and customized according to specific users and their prior intents. Additionally, a bidirectional communication channel could be established between the user and the assurance component, allowing the assurance to send recommendations and different possible solutions and letting the user to select the best course of actions.

B. Lack of Data

The intent assurance is expected to largely rely on AI and ML algorithms. Thus, the heart of the AI techniques and by extension, the assurance itself will be various datasets. Unfortunately, there is a lack of rich and well organized datasets, which could lead to lower prediction accuracy, and thus inadequate assurance actions [4]. At the same time, since many new applications and network services are constantly emerging it makes it even worse to find up-to-date datasets to test or transfer learning to IBN systems. This may lead the AI techniques to have to treat previously unseen data or insufficient amount of data, affecting once more the prediction accuracy [19]. Normally, as IBN and network automation in general tend to be the new norm in networking, more complete datasets are expected in the near future [12]. In the mean time, Generative Adversial Networks (GAN) could be a viable solution towards the lack of datasets by generating artificial data that match the statistics of an existing dataset [30].

C. AI Algorithms

The merging of intent assurance and AI is not an easy task. Apart from the lack of datasets, another challenge is the selection of the appropriate AI and ML algorithms. Various network problems may be detected with different AI models. Additionally, it cannot be expected that an IBN system will be able to include all possible AI techniques into the intent assurance component. Furthermore, an AI technique that has proven to be a good solution in a network domain (i.e. RAN) does not necessarily mean that will provide the same level of assurance in another domain (i.e. Cloud), due to the different statistical properties of the data generated or the varying behavior of the users issuing the intent. To overcome such challenges, a possible solution could be the use of “AI as a Service”, where the assurance engine could outsource a specific assurance service, such as root cause analysis, to a third party organization [1]. Moreover, the selection of the appropriate AI technique could be resolved by the use of AutoML techniques that could assist into the selection of the appropriate AI model, its proper hyperparameter tuning and training optimization [31].

D. Monitoring

Monitoring is another crucial functionality of the intent assurance, as it generates the input to the rest of the assurance components. Thus, it is easy to understand that according to the type of the intent and its KPI requirements a poor monitoring can cause significant threats in the proper assurance of the intent. A first question that has to be answered is the time granularity of collecting data. Several monitoring tools may provide different periods of data monitoring. For example, collecting information in the order of several minutes may result into missing some short and sudden network abnormalities, which nonetheless may cause significant performance drifts and SLA violations for critical intents. Additionally, some intents are expected to be short-lived and a sparse sampling rate may not be the best way to assure them. On the other hand, a frequent data collection rate may create scalability issues of how to exchange, store and manage the data [32]. At the same time, monitoring should be adapted according to the size and the heterogeneity of the infrastructure. A large network containing different types of nodes and links and thus, accommodating a high number of heterogeneous intents, may create the need for additional monitoring of data that may not be provided by the existing monitoring tools. This could lead into deploying multiple monitoring platforms that could increase the added overhead. Overall, it becomes a significant challenge to collect high-quality data in an efficient, elastic, and timely way [24].

Hence, the monitoring platforms used for the intent assurance should be flexible enough to allow customizing the sampling rate and extendable enough to add new metrics and types of monitoring probes. Furthermore, and as described in the previous section, to reduce any unnecessary data generation, the monitoring probes should be activated and deactivated in a dynamic fashion according to the assurance requirements and the state of the network. For example, when the conditions of the network change in such a way that multiple intents need to be reconfigured, more data may need to be gathered to better understand the problem and to propose more efficient corrective actions. Accordingly, the monitoring itself should be more intelligent allowing to be self-adapted or changed on-demand in order to provide a high-quality intent assurance.

E. Energy and Resources

The intent assurance through the continuous collection of large volume of monitoring data and the execution of computational intensive AI algorithms, will create several scalability issues. Simultaneously, the expansion of the network infrastructure to accommodate all the new and upcoming application and communication paradigms is expected to exponentially increase the complexity of providing the necessary assurance. First of all, the amount of storage resources needed to store the monitoring data, the various operational snapshots of the network state, and the numerous reports and logs will significantly stress the storage capabilities of a network domain. Secondly, AI algorithms can be resource voracious, entailing the consumption of many GPU, CPU and memory resources. Finally, another challenge that is often counterbalanced by the advantages that AI will bring, is the energy consumption. Specifically, the training and execution of the AI modules and the exigencies of high computational resources could potentially result in significant energetic consumption with a severe impact on the environmental and operational costs. Thus, appropriate resource allocation techniques leveraging the offloading of data and AI execution to more powerful infrastructures could be explored. Additionally, lightweight ML techniques should be pursued that could try to find an optimal trade-off between accuracy, inference time, and resource and energy consumption.

F. Multi-Domain Assurance

Often, the intents will not only be deployed in a single domain (e.g. an enterprise network, a data center network, etc.). They will rather be activated and assured in a multi-domain context. In this case, each domain could include an intent assurance component with the main functionalities presented above and by making the appropriate adjustments (i.e. to account for the heterogeneity and different capabilities of each domain). The main challenges arising herein, is how the domains will be synchronized and managed, how and if they will share knowledge, and how the corrective actions could propagate across the domains.

The most typical example of a multi-domain intent will be the activation of a 5G/6G network slice [21] that will require a cross-domain visibility for its assurance (i.e. access, transport, core). In such circumstances, each domain will manage its resources and the assurance of the partial intent autonomously [33]. The application of intent assurance individually in each domain could be an option, however local assurance will not necessarily result into the global assurance of the intent. Hence, a collaboration between domains should be enabled. This collaboration could be in the form of a centralized approach, with a central IBN Orchestrator that will be able to synchronize the IBN managers and by extension the intent assurance of each domain, or through a distributed approach where some domains could create a direct assurance communication channel.

In both cases, the knowledge transfer can become an issue, since some domains would not want to divulge operational data of their infrastructure, which can be used by their competitors to create strategical advantages. The latter may not be encountered in every multi-domain case. For example, a multi-domain enterprise which can manage multiple domains (i.e. enterprise branches, wide access networks, virtual private clouds, etc.) may follow a centralized approach. In this case, an intent-enabled Software Defined Wide Access Network (SD-WAN) could be used, having a detailed view of the capabilities of each domain and centrally managing the intent assurance through transfer learning techniques. In any other case, a flexible communication between the domains should be established while respecting their privacy. A possible solution to achieve this is the use of Federated Learning (FL) approaches. In particular, FL could enable a distributed learning approach, minimizing the volume of exchanged data, and solve any privacy issues assuring the intent performance in an end-to-end fashion [20]. In any case, the multi-domain assurance should be able to identify which domain may cause the intent's performance drift and avoid redundant and iterative triggering of corrective actions that could propagate erroneous assurance decisions among the different domains.

G. Standardization

IBN has attracted the interest of major standardization communities that have already tried to provide a proper definition. For instance, IETF regularly updates its concepts and definitions of IBN [5] and the European Telecommunication Standards Institute (ETSI), through the Experiential Network Intelligence (ENI) [34] initiative, tries to provide various design options for the IBN paradigm. Both these efforts underline the importance of the intent assurance. Additionally, there are many other standardization organizations and sub-groups that attempt to surface the importance of the assurance in the context of autonomous and Zero Touch Network and Service Management (ZSM) [4]. This is normal in such a preliminary stage that IBN and network automation is found, since there is no coordination between the different standard bodies. Nonetheless, it is important to create a more unified and collaborative initiative that will guide the design of the next generation of autonomous networks. The particular synergy could also establish the specifications of the hardware and software that the new network equipment should be equipped with to enable the CLA that IBN envisions. This could also enable the vendor interoperability, and the standardization of interfacing under a single and multi-domain scenario.

H. Security

Security always plays a primordial role in telecommunications. Accordingly, it cannot be excluded from the intent assurance component. Regardless of whether the security requirements are absent or explicitly stated in the intent, the intent assurance should have at least some default security mechanisms in place to guarantee the robustness of the intent against possible attacks. This stems from the fact, that as the infrastructure of modern networks grows, so does the surface of menace. Additionally, the type of the attack and the form it takes is constantly evolving. Thus, the security assurance should be able to rapidly adapt to this evolution of the attacks and possibly proactively detect them, in order to prevent any security breaches. A possible solution could be the use of transfer learning, where existing security and anomaly detection models could be adapted to efficiently detect any new form of threat [20]. Finally, we should not neglect the fact that the intent assurance would be largely based on data processing and the involved AI techniques. Accordingly, new security issues may arise, when the attack targets to compromise the data integrity collected from the monitoring process, which in result could affect the operation of the AI algorithms and lastly the action proposed by the assurance [30]. Hence, it may be important the monitoring platforms to be enhanced with authentication and authorization mechanisms and add extra security functionalities to the platforms that host the AI algorithms.

I. Diversified Assurance

When thinking about the network assurance in current and legacy networks, typical KPIs are coming into mind, such as throughput, latency, SNR, etc. However, future networks will go beyond these metrics and the intents may contain information of different business domains. We should not also forget that IBN is not restricted to be used in traditional networks or to be used only by network familiar users. The Internet is now a commodity and every type of business, enterprise and production environment relies on networks for their goods and services. The most typical example is Industry 4.0, where the IoT and 5G technologies are fueling an industrial revolution by converging the information technology (IT) and operational technology (OT). This means, that the intent may require guarantees on the level of production, the health protection of the workers, machine failure prevention, etc. This is a significant challenge, and requires a new tuning of the assurance process, where typical IT KPIs have to be correlated with up until unrelated and unknown OT KPIs [35].

J. IBN 2.0

Future autonomous networks envision the reduction if not the elimination of the human intervention. The reason and as stated in the Introduction, is that manual configurations are error-prone and time consuming. This can be proven critical in the context of intent assurance, where any non conformity of the performance should be immediately detected and acted upon. However, a question that may arise is if the network administrator should be completely removed from the intent assurance process or not. In other words, when the intent assurance autonomously propose and trigger a corrective action, how can someone be sure that these changes would be acceptable from an operator [18]? It is normal for the network operators to develop some kind of mistrust. To this end, and as a first stage, the administrator may want to keep a certain control over the infrastructure, while being involved in the decision making. This could be considered as a first version or a non-standalone phase of the IBN. Accordingly, in Section III, we have included a direct communication channel between the network administrator and the assurance engine to account for this needed supervision, especially for critical events. However, after building the necessary trust on the AI-enabled parts of IBN, a second phase or a standalone IBN should be expected. In either case, networks will remain user-centric, while users and administrators will be the main regulators of their input and output, creating a human to AI to human loop.

V. Conclusion

Intent Based Networking is an inherent part of the design of future autonomous and zero-touch networks that envision creating a closed loop automation process. One important component of this closed loop is the intent assurance. Through this assurance, the network should automatically be self-adapted and optimized to guarantee that the user requirements expressed in the intent will be satisfied throughout its lifetime. To this end, in this paper, we have introduced a visionary architecture for providing the necessary assurance in an IBN context. In particular, the proposed intent assurance created a secondary closed loop automation by monitoring the network state, analyzing the data through an AI-enabled assurance engine, and finally trigerring corrective actions via appropriate actuators. The introduced intent assurance also provided a feedback to the issuer of the intent allowing them to get notified or to participate in the assurance process. The paper concluded with some interesting open challenges that can shape the future directions of the intent assurance process.

References

[1] M. Gramaglia, M. Kajo, C. Mannweiler, O. Bulakci, and Q. Wei, “A unified service-based capability exposure framework for closed-loop network automation,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 11, p. e4598, 2022.

[2] P. Horn, “Autonomic Computing: IBM’s Perspective on the State of Information Technology,” Tech. Rep., 2001.

[3] J. Kephart and D. Chess, “The vision of autonomic computing,” *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[4] M. Liyanage, Q.-V. Pham, K. Dev, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, and G. Yenduri, “A survey on Zero touch network and Service Management (ZSM) for 5G and beyond networks,” *Journal of Network and Computer Applications*, vol. 203, p. 103362, 2022.

[5] A. Clemm, L. Ciavaglia, L. Granville, and J. Tantsura, “Intent-Based Networking - Concepts and Definitions,” Intent Research Task Force (IRTF), RFC 9315, Tech. Rep., 2022.

[6] B. Martini, M. Gharbaoui, and P. Castoldi, “Intent-based network slicing for sdn vertical services with assurance: Context, design and preliminary experiments,” *Future Generation Computer Systems*, vol. 142, pp. 101–116, 2023.

[7] P. Lingga, J. J. Kim, and J. P. Jeong, “Intent-based network management in 6g core networks,” in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, 2022, pp. 760–762.

[8] A. Leivadeas and M. Falkner, “A survey on intent-based networking,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 625–655, 2023.

[9] M. Kiran, E. Pouyoul, A. Mercian, B. Tierney, C. Guok, and I. Monga, “Enabling intent to configure scientific networks for high performance demands,” *Future Generation Computer Systems*, vol. 79, pp. 205–214, 2018.

[10] A. S. Jacobs, R. J. Pfitscher, R. A. Ferreira, and L. Z. Granville, “Refining network intents for self-driving networks,” in *Proceedings of the Afternoon Workshop on Self-Driving Networks*, ser. SelfDN 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 15–21.

[11] X. Zheng, A. Leivadeas, and M. Falkner, “Intent based networking management with conflict detection and policy resolution in an enterprise network,” *Computer Networks*, vol. 219, p. 109457, 2022.

[12] A. Leivadeas and M. Falkner, “VNF Placement Problem: A Multi-Tenant Intent-Based Networking Approach,” in *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2021, pp. 143–150.

[13] X. Zheng and A. Leivadeas, “Network assurance in intent-based networking data centers with machine learning techniques,” in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 14–20.

[14] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, “A survey on software-defined networking,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.

[15] D. Sanvito, D. Moro, M. Gullì, I. Filippini, A. Capone, and A. Campanella, “Onos intent monitor and reroute service: enabling plug & play routing logic,” in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 272–276.

[16] A. Leivadeas, M. Falkner, I. Lambadaris, and G. Kesidis, “Dynamic traffic steering of multi-tenant virtualized network functions in sdn enabled data centers,” in *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, 2016, pp. 65–70.

[17] ETSI, “Network Functions Virtualisation (NFV); Architectural Framework,” European Telecommunication Standards Institute (ETSI), ETSI GS NFV 002, Tech. Rep., 2014.

[18] M. Falkner and J. Apostolopoulos, “Intent-based networking for the enterprise: A modern network architecture,” *Commun. ACM*, vol. 65, no. 11, p. 108–117, oct 2022.

[19] K. Samdanis, A. N. Abbou, J. Song, and T. Taleb, “Ai/ml service enablers & model maintenance for beyond 5g networks,” *IEEE Network*, pp. 1–10, 2023.

[20] J. Wang, J. Liu, J. Li, and N. Kato, “Artificial intelligence-assisted network slicing: Network assurance and service provisioning in 6g,” *IEEE Vehicular Technology Magazine*, vol. 18, no. 1, pp. 49–58, 2023.

[21] N. F. S. de Sousa, M. T. Islam, R. U. Mustafa, D. A. L. Perez, C. E. Rothenberg, and P. H. Gomes, “Machine learning-assisted closed-control loops for beyond 5g multi-domain zero-touch networks,” *Journal of Network and Systems Management*, vol. 30, no. 46, 2022.

[22] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, “Ensemble learning-based network data analytics for network slice orchestration and management: An intent-based networking mechanism,” in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, pp. 1–5.

[23] M. Jovanovic, M. Cabarkapa, D. Budimir, and D. Budimir, “Network service assurance and telemetry optimisation using heuristics,” in *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, 2022, pp. 264–267.

[24] H. Song, F. Qin, M.-J. P., and A. Wang, “Network Telemetry Framework,” Intent Engineering Task Force (IETF), RFC 9232, Tech. Rep., 2022.

[25] A. Pagès, F. Agraz, and S. Spadaro, “Sdn-based band-adaptive quality assurance scheme in support of heterogenous b5g services over sliceable multi-band optical networks,” *Optical Switching and Networking*, vol. 47, p. 100721, 2023.

[26] A. Hameed, J. Violos, A. Leivadeas, N. Santi, R. Grünblatt, and N. Mitton, “Toward qos prediction based on temporal transformers for iot applications,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4010–4027, 2022.

[27] K. Tanabe, T. Fukuda, and T. Kuroda, “Automated performance evaluation of intent-based virtual network systems,” in *2020 16th International Conference on Network and Service Management (CNSM)*, 2020, pp. 1–7.

[28] R. Mohamed *et al.*, “Automatic feasibility restoration for 5g cloud gaming,” in *IEEE International Conference on Communications (ICC)*, 2023, pp. 1–6.

[29] M. Jain *et al.*, “Intent-based, voice-assisted, self-healing sdn framework,” *Journal of Network Communications and Emerging Technologies (JNCET)*, vol. 10, no. 2, pp. 1–9, 2020.

[30] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, “Machine learning-based zero-touch network and service management: a survey,” *Digital Communications and Networks*, vol. 8, no. 2, pp. 105–123, 2022.

[31] A. Collet, A. Banchs, and M. Fiore, “Lossleap: Learning to predict for intent-based networking,” in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 2138–2147.

[32] N. Andriolli *et al.*, “Optical networks management and control: A review and recent challenges,” *Optical Switching and Networking*, vol. 44, p. 100652, 2022.

[33] V. S. Mai, R. J. La, T. Zhang, and A. Battou, “End-to-end quality-of-service assurance with autonomous systems: 5g/6g case study,” in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, p. 644–651.

[34] ETSI, “Experiential Networked Intelligence (ENI); InTent Aware Network Autonomicity (ITANA),” European Telecommunication Standards Institute (ETSI), ETSI GR ENI 008, Tech. Rep., 2021.

[35] A. Kattepur, A. R. Nair, M. Saimler, and Y. Donmez, “Industrial 5g service quality assurance via markov decision process mapping,” in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–8.