

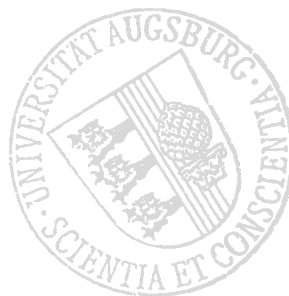
Seminar
Software Engineering for Distributed Systems
Sommersemester 2022

Key Challenges for Intent-Driven Mobile Network Management

Vladyslav Kolesnykov
Matriculation number: 2054429
vladyslav.kolesnykov@uni-a.de

Advisor: Tobias Foth
Software Methodologies for Distributed Systems (Prof. Bauer)
University of Augsburg

Abstract Intent-Driven Networking (IDN) is a promising concept that tends to automate conventional networking by applying a deeper level of intelligence. High-level intents offer an efficient and powerful way of expressing network requirements to manage its infrastructure, resources, and services without providing exact configurations. This paper provides an overview of the main challenges in applying IDN in mobile networks. It clarifies the definition of IDN, presents a generic architecture, and briefly covers an intent translation process. We also discuss available industrial products based on this concept and present the advantages of IDN over traditional networking.



Contents

1	Introduction	1
2	Overview of mobile networks	1
3	Intent-Driven Networking	2
3.1	Definition of Intent-Driven Networking	2
3.2	Intent-Driven System Architecture	3
3.3	Intent Translation	5
3.4	Industrial Products	6
3.5	Advantages of Intent-Driven Networking	6
4	Challenges of Intent-Driven Networking	7
4.1	Lack of standardization	7
4.2	Next-generation Mobile Networks	7
4.3	Application of Artificial Intelligence	8
4.4	Intent description and translation	9
4.5	Intent fulfillment and conflict resolution	9
4.6	Legacy networks	10
5	Conclusion	10
	References	11

1 Introduction

New customer-facing services, such as intelligent transport systems, smart cities, online gaming, and AR/VR create new challenges for the 5th Generation (5G) mobile networks. These include services with various quality of service (QoS) requirements, such as enhanced mobile broadband, ultra-reliable low-latency communications, and massive machine-type communications. However, the next-generation mobile networks (6G) will face even more QoS requirements such as less than 1-millisecond end-to-end latency, 1 Tb/sec throughput, massive machine-type communication, supporting a wide range of connected devices, and enabling real-time, highly reliable communication [1, 2]. Additionally, the traditional operation and maintenance management methods entail high labor costs, error probability, and low management efficiency. These challenges can only be addressed with the extensive adoption of network automation capabilities that minimize the need for human intervention.

Thus, future mobile networks should be self-configuring, self-managing, and self-optimizing. Intent-Driven Networking (IDN) is a promising network concept that has received great attention from industry and open source communities in recent years. Instead of performing manual network management, the network is provided with desired business intent that is used to express all requirements to be automatically fulfilled by a system. IDN has the potential to automate the network management process by providing simple and powerful tools to handle resources, infrastructure, and services [3]. However, the enabling techniques and frameworks of IDN are still in the further exploration phase.

The rest of this paper is organized as follows. Section 2 presents a short overview of mobile networks and enabling technologies for applying intelligence in networks. Section 3 introduces the definition of IDN, followed by generic architecture and intent translation process. In addition, the main advantages and IDN products are briefly covered. The main challenges of IDN that require further research are discussed in Section 4, followed by a conclusion in Section 5.

2 Overview of mobile networks

A mobile network is a communications network that provides a wireless connection to individual devices and enables access to the core network. Communication is achieved by transmitting data via radio waves from end-user devices to a transceiver. Currently, the evolution of mobile communications has the following technologies: GSM, GPRS/EDGE (2G), UMTS (3G), LTE (4G), and 5G. The advances made in the development of 5G cellular networks could have been achieved through various new techniques, such as softwarisation, virtualization, flexible and scalable management, as well as appropriate slicing of available network resources.[4]

Software-Defined Networking (SDN) is a concept that enables a dynamic and programmatically efficient network configuration in order to improve network performance and flexibility. The data plane, which is a layer where devices are interconnected through

wireless radio channels or wired cables, is decoupled from the control plane and managed by a centralized controller [5]. The SDN controller logically maintains the dynamic network, takes requests from the application layer, and manages the network devices. Simply put, the network is programmable through software applications running on top of the SDN controller that interacts with the underlying devices. It allows operators to manage the network and services from a centralized user interface. In [6], SDN was also proposed for the mobile networks to efficiently accommodate the increasing data traffic over the LTE or Wi-Fi accesses, considering the real-time network condition.

Network Function Virtualization (NFV) is a complementary technology to the SDN, which can provide the infrastructure on which SDN can run. Network virtualization is an approach that decouples the network functions from their supported hardware and allows them to be implemented as software. NFV makes it possible to programmatically create, provision, and manage networks in software while using the underlying physical network. Virtualization decouples network services, such as routers, firewalls, and load balances, from the specific hardware and allows virtual provisioning of an entire network. The combination of SDN and NFV can address the challenges of dynamic resource management and intelligent service orchestration. [5]

IDN takes traditional networking management one step further. It builds upon SDN and NFV technologies enabling a higher level of intelligence and automation in networks. IDN allows configuring a network to address the desired business goal by specifying just what the resulting state of the system shall be instead of how the system has to reach it.

3 Intent-Driven Networking

This section presents the formal definition of IDN, followed by generic architecture and intent translation process that uses machine learning techniques. Furthermore, it briefly surveys available industrial products and lists the advantages of IDN over regular networking.

3.1 Definition of Intent-Driven Networking

In order to define the term IDN, it is essential to understand what intent is. Though there is no standard term intent, all definitions presented by various research institutions are pretty similar. The intent can be described as a declarative way to define the desired state of the system that can be translated into an advanced policy. It abstracts the components and capabilities of the network from a requirements perspective. The intent does not define any specific commands on how to reach the resulting state. Instead, it only specifies the desired outcome of the network. For example, *"a cell should have a range of 1 km with a data rate of 100 Mbit/s."* It means that the network should be able to understand the intent and map it into a specific device-level configurations. The system automatically observes its current state and adjusts itself to achieve the desired outcome if there are any deviations. [7, p. 22867]

Intent-Driven Networking is a technology concept that tends to automate network administrative tasks by applying a deeper level of intelligence. IDN captures, translates, and applies business intent across the network. The network is able to automatically deploy, configure, verify and optimize itself to achieve a target state defined by intent. The network performs continuous monitoring and adjustment to ensure alignment, which is achieved through a closed-loop system. Eventually, IDN provides full-lifecycle management for the network. [7]

IDN management provides four main functionalities: [8, p. 271]

1. Translation and verification. The IDN captures and translates business intent into the necessary system configuration, which is then automatically applied across the network. The system then validates the correctness of the resulting policy.
2. Automatic deployment. The IDN installs translated and verified policies across physical and virtual network infrastructure through network automation and network orchestration.
3. Network state awareness. The state of the network may constantly change, which can lead to some inconsistencies between implementation and the desired outcome. The system needs to continuously monitor and manage the network to guarantee fulfillment of the business intent at any point in time.
4. Accurate diagnosis and dynamic optimization. The ultimate goal is for the network to continuously monitor and adjust network performance to ensure the desired business outcome. If expected intent is not achieved, the system takes corrective actions.

3.2 Intent-Driven System Architecture

Different research groups have various concepts and design solutions for the IDN architecture. Figure 1 illustrates a generalized architecture for IDNs presented in [9, p. 3]. Its core elements are discussed below.

Intent Processor (IP) is the core component of IDN architecture that accepts high-level intent provided by users or various applications through a corresponding interface. Intents can be given in different formats, such as simple text, executable commands via a Command-Line Interface (CLI), interaction with a graphical user interface (GUI), or audio interface as speech. After new intent is provided, the translator module performs mapping into a machine-readable representation for the required modules or services. The IP is also responsible for intent assurance. Based on the information from State Observer, IP takes corrective action and assists in the reconfiguration of the underlying infrastructure to preserve the required intents.

Optimization Decision Maker (ODM) is responsible for intent verification optimization and decision making. After receiving as input a mid-level representation of intent, ODM tries to find a feasible and the most optimal solution that is then returned to IP. Based on AI- and ML techniques, the AI Engine (AIE) of ODM learns from

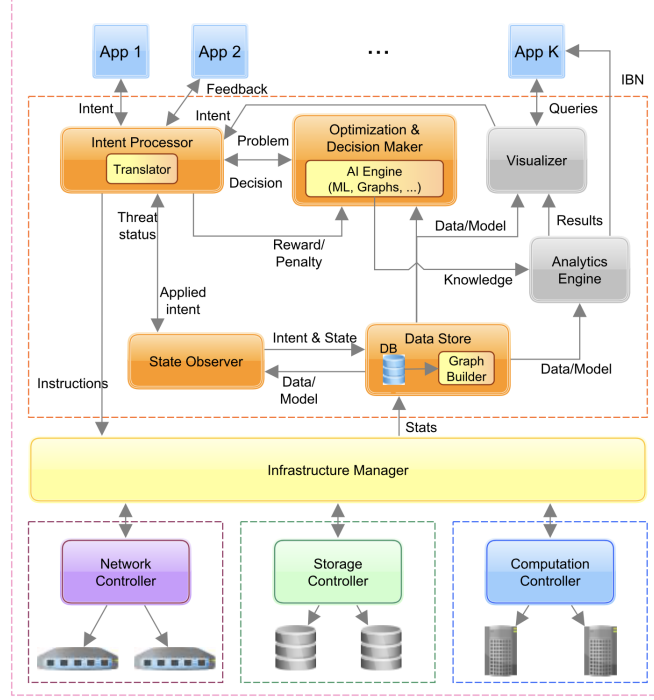


Figure 1: Basic Architecture of IDN. [9]

past intents to make better decisions in terms of feasibility and intent installation. In addition, this module may be trained on rewards and penalties provided by IP.

Data Store (DS) may store all relevant data collected from the network in a centralized or distributed manner. It provides an API so that other components can store and retrieve data from the DS. Additionally, DS may have a Graph Builder (GB) module that provides a graph view of the underlying network. The graph model can be utilized for optimization, decision making, generating analytics, and network visualization.

State Observer (SO) is a crucial component for intent assurance because it is responsible for monitoring the network. Based on data retrieved from DS it observes the current state and informs IP if there is any threat or inconsistency against installed intent. For instance, a node can be overloaded with network traffic leading to failure in serving a critical service. In this case, the SO identifies the threat and informs the Intent Processor.

The following modules, such as **Visualizer** and **Analytics Engine (AE)** are optional components of IDN but still provide some useful functionality. The Visualizer displays information about the state and performance of the network at a glance using a graphical user interface (GUI). It ensures total visibility of the entire network by providing graphical representations of network devices, network metrics, and data flows. AE is used to process queries related to network performance, based on raw monitoring data and graph model. Third-party applications may access AE using its API to perform network monitoring. In IDN, **applications** are transparent to the underlying system and only interact via appropriate APIs, which hide the underlying system's complexities.

The underlying resource layer provides the physical network infrastructure with various device entities. It contains multiple controllers like **Network (NC)**, **Storage (SC)**, and **Computation Controller (CC)** that configure the network devices, such as routers, switches, disks, and servers, according to a required intent. The NC can be an SDN controller or network management system in a traditional network. The **Infrastructure Manager (IM)** represents an interface between the IP and the different controllers. IM translates feasible intents into domain-specific instructions and also collects various statistics from the controllers.

In comparison, the authors in [10, p. 13] present another IDN architecture, which is more detailed and contains four following layers: application, intent, network management and orchestration, and resource layers. This survey gives a comprehensive overview of the functions and features of each layer with its key enabling technologies.

3.3 Intent Translation

The paper [11] introduces an intent-refinement process that uses machine learning and feedback from the operator to translate intents into network configurations. The proposed refinement method extracts an intermediate representation of intent from natural language before compiling it into SDN commands.

The intent is collected through a chat interface, and a neural network learning model is used to extract entities from natural language. Examples of such entities are the network endpoints, middleboxes, and temporal configurations for the policy. A previously trained sequence-to-sequence learning model based on a Recurrent Neural Network (RNN) with Long Short-Term Memory is used to translate extracted entities to structured intents written in the Nile language. Thus, the operator can build powerful yet simple intents. For example, an input *"Every day from 09:00 to 18:00, the cell should have a range of 1 km with a data rate of 100 Mbit/s, latency under 10 milliseconds and frequency band 2000 MHz"* can be represented as in Listing 1. The extracted Nile intent acts as an abstraction layer for lower-level configuration and policy languages. Authors claim the Nile's grammar is expressive enough to represent most network intents.

Listing 1: Nile intent example. [11]

```
define intent qosIntent:
  from      endpoint("cell")
  with      range("equal", "1000m"),
            throughput("more or equal", "100mbps"),
            latency("less", "10ms"),
            frequency("equal", "2000MHz")
  start     hour("09:00")
  end       hour("18:00")
```

The structured intent definition generated by the translation model is then presented to the network operator for confirmation. The operator may either confirm the cor-

rectness of the intent program or make adjustments if necessary. Based on the user's response, the translation model is then trained, ensuring that the results improve every time the operator provides new intent.

Finally, having a structured intermediate intent representation verified by the user, the Intent Deployer can compile and deploy it into configuration commands using SONATA-NFV, which is an emulation platform that deploys network functions as Docker containers. Decoupling provided by intermediate intent representation ensures the reusability of the proposed solution by allowing compilation of the intents to other existing network configurations and policy languages, such as Janus, PGA, and Kinectic.

However, intent translation is not limited to this implementation. The same paper [11, p. 20] also briefly covers other existing intent languages, frameworks, and compilers, discussing their goals and shortcomings.

3.4 Industrial Products

In recent years, the concept of IDN has received great attention from various standardization organizations, open-source communities, and industry. This section presents some available IDN solutions.

The European Telecommunications Standards Institute (ETSI) has worked on Experiential Networked Intelligence (ENI), which is a Network Management architecture that uses AI techniques and context-aware policies to address heterogeneous service requirements based on user needs, environmental conditions, and business goals. In terms of open-source solutions, the Open Network Operating System (ONOS) was designed to help network service providers build software-defined networks. ONOS has a component called Intent Framework that allows applications to specify their network control requirements in form of intents. Another open-source project called OpenDayLight also has the Intent Component that enables the controller to manage network services and resources based on generalized and abstracted policy semantics. Also, several industrial companies have developed their IDN products. Cisco has created a digital network architecture (DNA) that is an IDN management platform. Another product Apstra operating system (AOS) is aimed to automate the entire lifecycle of network infrastructure and services. Huawei has announced its IDN solution that will enable SDN-based networks to evolve into IDN-based. [7]

3.5 Advantages of Intent-Driven Networking

Intent-Driven Networking has clear advantages over regular networking. Most of these advantages lead to significant time savings and improve the agility of the entire system [12]. The main benefits of IDNs are listed below: [13, p. 11]

Simplifies and automates network operations. IDN reduces the complexity of the management and maintenance of the entire infrastructure. It facilitates work associated with the traditional network configuration and simplifies the deployment of additional network services.

Automates troubleshooting and resolution. Through its closed-loop design, an intent-based network reduces complex troubleshooting scenarios. As assurance processes verify the alignment of network configuration with intent, IDN can recognize potential problems before they occur and immediately perform correcting actions.

Enables automatic learning and optimization. Using machine learning techniques, IDN may predict any violations of the expressed intent before changes will be applied, forecast trends, identify anomalies, predict and validate system-level performance. IDN may also learn from past decisions to determine the most optimal solution.

Improves network security capabilities. Part of the monitoring is that IDN is always looking for threats in the network data. Security violations are immediately identified and prevented.

Continuous alignment of the network with the business objectives. IDN ensures that the network always stays in compliance with the business goals by constantly performing intent assurance and implementation.

Fast implementation of business goals into network configurations. Users can quickly provide high-level business intents, which are then translated into optimal network policies. The configuration of the network components is fully automated.

4 Challenges of Intent-Driven Networking

IDN improves the automation of networks and abstracts complex problems, but it also faces some challenges. This section covers several issues that require further research.

4.1 Lack of standardization

Different research groups, open-source communities, and industrial companies independently explore and develop IDN solutions. Existing IDN frameworks are based on heterogeneous architectures and underlying techniques that present different functional characteristics. Each component and architectural layer heavily relies on API, and the current unified standard has not been formed. Thus, the absence of a standardized format for specifying intents and common architecture is one of the main challenges of IDN. [7]

4.2 Next-generation Mobile Networks

Autonomy and ubiquitousness are two characteristics of next-generation mobile networks that should support enhanced mobile broadband, and highly reliable, low-latency, long-distance, and high-mobility communication. 6G must provide ultra-massive machine-type communications that support connectivity for the Internet of Things (IoT), fully autonomous vehicles and Internet of Bodies with various smart wearable devices. Additionally, 6G will evolve into a large-dimensional space-air-ground-underwater network to provide full coverage and unlimited wireless connection. Eventually, 6G will provide connectivity for the most possible needs of all mankind and industries. [14, p. 29]

However, current mobile network capabilities cannot meet the demands of extremely wide coverage and ubiquitous connectivity. The evolution of network standards and frequency bands in mobile networking presents diversification and isomerization that increases the complexity and the cost of network operation and maintenance. Intelligentization, powered by AI, is the key to achieving fully autonomous network management. 6G networks are expected to be zero-touch and intent-based to automate network operations and provide flexibility for diverse user and business requirements. The combination of quantum theory and AI can create more powerful and efficient algorithms that will significantly enhance the automation and intelligence of next-generation wireless networks.[14]

4.3 Application of Artificial Intelligence

The increasing complexity of the mobile networks requires IDNs to be able to translate users' business intents and other requirements into the network configuration, operation, and maintenance policies, and perform the self-configuration, self-optimization, and self-healing of the network. These tasks can be accomplished by AI technologies to ensure network performance while reducing operational costs and improving network robustness. AI can replace fields in wireless networks that require expert knowledge by automating network operations. Thus, mobile networks would greatly benefit from advances in the domains of AI and Machine Learning (ML). The operations such as planning, configuration, monitoring, optimization, healing of network, and security assurance are some potential areas that require the integration of AI in the IDN. [8]

Cellular network planning is a challenging process that involves resolving a high number of complex issues, such as uncertainty of input parameters, lack of accurate network details, and complexity of the network. For example, a typical 5G node could have thousands of configuration parameters. Standard network planning is an iterative process based on approximation techniques with fixed pre-defined inputs. AI techniques like Simulated Annealing, Tabu Search, and Genetic Algorithms can be used to solve complex network design problems, by making more accurate predictions.[15]

The self-organizing features including self-configuration, self-optimization, and self-healing play a critical role in the automation of complex networks. An AI-enabled network can dynamically orchestrate its components and network slices. For instance, it can choose various radio access technologies to provide better connectivity. AI techniques like Transfer Learning, Reinforcement Learning, and Dynamic Programming may be utilized to address configuration management with a highly automated approach.[15]

Network monitoring could be challenging in new generation mobile networks due to the dynamic and heterogeneous nature of the cellular environment. With AI-powered diagnostic analytics, the network can quickly and accurately detect problems and resolve them, even before they occur. Based on data from various sources, algorithms like Logistic Regression (LR), Support Vector Machine (SVM), and Hidden Markov Model (HMM) could be utilized for network anomaly detection. [15]

In mobile networks, AI can dynamically manage multiple connections and select optimal mobile cells to ensure connection continuity. Based on traffic forecasting, optimal

user scheduling can be selected to improve network performance and radio resource efficiency. In addition, using AI, nodes can optimize parameters to achieve load balancing and reduce retransmission overheads to enhance the reliability of data transmission. Finally, by extracting useful information from network traffic and learning routing strategies, the network can provide optimal paths for data transmission. [14]

The research on the integration of IDNs and AI approaches into mobile networks has just started in recent years. The current studies focus mostly on the core networks, and Intent-Driven mobile networks need further exploration before IDN makes a breakthrough in wireless communication technology, achieving a novel standard in mobile networks.

4.4 Intent description and translation

In IDNs, it is expected for the intents to be human-readable at the application level and more machine understandable as the intent flows through the whole system towards the resource layer. The machine-readable format as a mid-level representation is required for efficient intent processing and implementation. However, a human-readable format is preferable for users since arbitrary complex goals can be expressed in a high-level and flexible way. The translation of abstract intents into concrete network configurations is one of the biggest challenge and design goals in IDN systems.

Several different approaches have been proposed for intent representation, such as ontology-based, unrestricted language, vocabulary-restricted, and other hybrid methods. The intent translation is aided by various Natural Language Processing (NLP) technologies. For instance, NLP can be used for understanding the intent structure and its translation through Sequence-to-Sequence (seq2seq) learning models such as recurrent neural networks (RNN, Long-Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)) and advanced models including attention, Bidirectional Encoder Representations from Transformers (BERT). The context and declarative parameter retrieval leading to the appropriate network configuration can be achieved through a Convolutional Neural Network (CNN). However, the capabilities of these approaches are still limited compared to naturally spoken language. The intent representation and translation problems still have to be explored to provide scalable solutions for human-readable intent description models. [10, p. 18]

4.5 Intent fulfillment and conflict resolution

To ensure the correct fulfillment of provided intent, IDN should perform its syntactical and semantical validation. Valid intents should still be checked for technical feasibility, in order to avoid conflicts that might occur due to inconsistency, missing network functionalities, or conflicts with existing intents. Infeasible intents should be rejected and the system has to provide an operator with appropriate feedback. Thus, it is necessary to monitor, collect and store network state data and capabilities to perform feasibility checking.

The survey [3] presents two methods for fulfilling user-specified intents. The first approach is based on Intent Logic Units (ILU), which may be considered as a wrapper around command sequences or logic that need to be executed to achieve a specific goal. ILUs are stored in an Intent Logic Library (ILL). Intent-Logic Execution Platform (ILEP) retrieves appropriate ILU from the ILL library and schedules its execution. The second method uses Intent-Driven Network Automation Function Orchestrator, which identifies if the intent is valid and executable by Cognitive Automation Network (CAN). Both methods allow the fulfillment of intents but need to be extended in the future to support more features.

IDN is expected to be provided multiple active intents that should be handled at the same time in an efficient way, leading to various control actions utilizing a shared set of resources. The possibility of conflicts among active intents is unavoidable and requires intent coordination and conflict determination, prevention, and resolution. In order to solve conflicts in network control and management, simple forms of coordination, such as prioritization, partial fulfillment, or queuing were proposed. There are also abstract approaches that use the tools of formal languages and first-order logic to handle conflicts. In addition, the dynamic environment of mobile networks, like the changing number of users and traffic demand, makes this task even more complicated. Therefore, the detection and resolution of intent conflicts is a major challenge in IDN and needs further research. [3]

4.6 Legacy networks

Industries and enterprises tend to move away from legacy network technologies as more efficient solutions become available and the nature of consumer demand changes. However, the legacy network equipment and systems are still often in use. The aspect of migrating them to IDN architecture has to be considered. [9]

5 Conclusion

This paper presents the challenges of Intent-Driven Networking which is the next step toward an autonomous network management system. It gets vast attention from open source communities and industry in recent years. The use of high-level intents simplifies complex network management by offering an efficient and powerful way to manage network infrastructure, resources, and services.

Firstly, we clarified a definition of IDN and presented the corresponding general architecture with its main enabling components. We also covered an intent-refinement process that uses machine learning techniques to translate intents into network configurations. Then we presented available industrial products based on the IDN concept and identified its advantages over regular networking. In addition to the lack of a standardized format for specifying intents and common architecture, IDN faces some challenges in intent translation, fulfillment, and conflict resolution processes. Integration of AI approaches is also a major issue that would enable mobile networks to perform the self-

configuration, self-optimization, and self-healing. AI offers powerful algorithms which can be leveraged to manage these challenges. However, more research on AI-enabled IDN needs to be done to improve intelligence in autonomous network operations.

References

- [1] Bin Han et al. “An abstracted survey on 6g: Drivers, requirements, efforts, and enablers”. In: *arXiv preprint arXiv:2101.01062* (2021).
- [2] NTT Docomo. “5G Evolution and 6G”. In: *Whitepaper, January* (2020).
- [3] Stephen S. Mwanje et al. “Intent-Driven Network and Service Management: Definitions, Modeling and Implementation”. In: (Dec. 2021). DOI: [10.36227/techrxiv.17075450.v1](https://doi.org/10.36227/techrxiv.17075450.v1). URL: https://www.techrxiv.org/articles/preprint/Intent-Driven_Network_and_Service_Management_Definitions_Modeling_and_Implementation/17075450.
- [4] Panagiotis Demestichas et al. “5G on the horizon: Key challenges for the radio-access network”. In: *IEEE vehicular technology magazine* 8.3 (2013), pp. 47–53.
- [5] Yong Li and Min Chen. “Software-defined network function virtualization: A survey”. In: *IEEE Access* 3 (2015), pp. 2542–2553.
- [6] Mojdeh Amani et al. “Programmable policies for data offloading in LTE network”. In: *2014 IEEE International Conference on Communications (ICC)*. 2014, pp. 3154–3159. DOI: [10.1109/ICC.2014.6883806](https://doi.org/10.1109/ICC.2014.6883806).
- [7] Lei Pang et al. “A Survey on Intent-Driven Networks”. In: *IEEE Access* 8 (2020), pp. 22862–22873. DOI: [10.1109/ACCESS.2020.2969208](https://doi.org/10.1109/ACCESS.2020.2969208).
- [8] Yiming Wei, Mugen Peng, and Yaqiong Liu. “Intent-based networks for 6G: Insights and challenges”. In: *Digital Communications and Networks* 6.3 (2020), pp. 270–280. ISSN: 2352-8648. DOI: <https://doi.org/10.1016/j.dcan.2020.07.001>. URL: <https://www.sciencedirect.com/science/article/pii/S2352864820302418>.
- [9] Barun Saha et al. “Intent-based Networks: An Industrial Perspective”. In: Oct. 2018, pp. 35–40. DOI: [10.1145/3243318.3243324](https://doi.org/10.1145/3243318.3243324).
- [10] Kashif Mehmood, Katina Kravevska, and David Palma. *Intent-driven Autonomous Network and Service Management in Future Networks: A Structured Literature Review*. 2021. DOI: [10.48550/ARXIV.2108.04560](https://doi.org/10.48550/ARXIV.2108.04560). URL: <https://arxiv.org/abs/2108.04560>.
- [11] Arthur Selle Jacobs et al. “Refining Network Intents for Self-Driving Networks”. In: *Proceedings of the Afternoon Workshop on Self-Driving Networks*. SelfDN 2018. Budapest, Hungary: Association for Computing Machinery, 2018, pp. 15–21. ISBN: 9781450359146. DOI: [10.1145/3229584.3229590](https://doi.org/10.1145/3229584.3229590). URL: <https://doi.org/10.1145/3229584.3229590>.
- [12] *What is intent-based networking (IBN)?* 2022. URL: <https://www.vmware.com/topics/glossary/content/intent-based-networking.html> (visited on 05/11/2022).

- [13] P. Martinez-Julia, J. Nobre, and D. Lopez. “Intent Classification”. In: (2022).
- [14] Zhengquan Zhang et al. “6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies”. In: *IEEE Vehicular Technology Magazine* 14.3 (2019), pp. 28–41. DOI: [10.1109/MVT.2019.2921208](https://doi.org/10.1109/MVT.2019.2921208).
- [15] C. B. Anuradha and Puneet Sharma. “Empowering the future 5G networks: an AI based approach”. In: *Telecom Business Review* 10.1 (2017), p. 53.

Declaration of Academic Honesty

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority nor has it been published yet.

Augsburg, 21.06.2022

(Vladyslav Kolesnykov)