KID: Knowledge Graph-Enabled Intent-Driven Network with Digital Twin

Xiaotian Chang[1], Chungang Yang[1], Hao Wang[2], Ying Ouyang[1], Ru Dong[1], Junjie Guo[1], Zeyang Ji[1], Xianglin Liu[1]
[1]The State Key Laboratory on Integrated Services Networks, Xidian University, Xi'an, China.
(emails: henucxt@163.com; chgyang2010@163.com; yingouyang224@163.com; 18302999373@163.com;
peterguo97@163.com; jzy15835274090@163.com; lxl08080627@163.com)
[2]54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China.(email:632606089@qq.com)

***Abstract*—To meet novel services and networking requirements towards the next generation applications, intent-driven network is proposed as a promising networking paradigm. It is with capabilities of intent refinement, policy generation, and state awareness. And these distinctive capabilities contribute to its wide applications to the next generation networks. However, current researches lack a generalization model of intent refinement. Additionally, it is difficult to extract available knowledge from huge raw data of the network status, and guarantee the precise generation of network policies. To solve these challenges, we present a knowledge graph-enabled intent-driven network with the digital twin, which is termed as KID in this work. In the KID, knowledge graph is utilized to represent user intents, abstract network status, and express network policies. And the digital twin is applied to validate intents as well as abstract the physical network. The KID enhances the capabilities of intent-driven networks to refine intents, contributing to the continuous assurance of accurate intent fulfillment. Finally, we present a proof of concept implementation of the KID. Simulation results verify the feasibility and effectiveness of the presented KID framework.**

***Index Terms*—Digital twin, intent-driven network, knowledge graph**

I. Introduction

It is vital for the next generation networks to provide customers with richer and more distinctive services. Therefore, how to best fulfill user requirements and implement relevant network services is a worthwhile research direction. To overcome the issues that the next generation networks face in terms of novel services and networking requirements, a novel networking paradigm called intent-driven network (IDN) is proposed. The IDN, which is also named as intent-based network, is a promisingly programmable and customizable automation network that can extract user intents, perceive network status, and optimize network configuration in real-time. It allows users to declare intents through high-level abstraction policies. Therefore, users can declaratively express the desired network status, without concerning with the specifications of refining high-level abstract policies into a low-level programming language [1]. In the IDN, the desired network status can be achieved automatically through intent refinement, verification, configuration, and optimization according to the declarative user intents.

The National Key Research and Development Program of China (2020YFB1807700).

Intent translation, as a key component of intent refinement [2], primarily focuses on converting user intents into network parameters. There are two typical forms of intent input: the natural language input [3]–[5] and the network parameter input [6], [7]. Intent translation is beneficial to transform user intents into network policies that can be recognized by the computer. However, intents may come from different users, applications, or from the underlying network status. In the face of various intents, current intent translation methods are difficult to extract comprehensive user intent information and only work well in a static and pre-designed model. Besides, it is hard to merge the gap between the expected intents and the network configuration due to various intent forms. As a result, a generic intent translation method is urgently needed to address these shortcomings. Additionally, another challenge that the IDN is currently working on is how to handle the huge raw data of the network status. The presence of a large amount of data from the underlying network has a significant impact on the efficiency and accuracy of intent translation and policy generation. Meanwhile, it is difficult for current network systems to extract knowledge from the huge raw data, and perceive the network status. Furthermore, the process of policy generation lacks the capability to reasoning based on the user intents and the network status. This is another issue that should be well treated with for the IDN. In summary, the major novel technical challenges are as follows:

- **Intent:** Due to different intent sources and diverse kinds of intents, current intent refinement models are difficult to integrate and merge different intents from various application scenarios. A generalization model of intent refinement is required to deal with various intents.
- **State:** A large amount of raw data is generated in the underlying physical network during the information interaction. Current IDNs call for a method that can extract useful knowledge from huge network status information.
- **Policy:** It is necessary for policy generation to consider both user intents and constraints of the underlying resources. However, many IDN systems generate policies through simple mapping in the policy repository. Obviously, this cannot guarantee the completeness or accuracy

of policy generation, in particular, when the intent may be changed with the dynamic network status.

With the advancement of knowledge graph (KG) technology, several researchers have tried to explore and exploit KG into IDNs [8]. The KG is a graph-based data structure consisting of nodes and edges. A node represents an "entity", and an edge represents a "relationship" among the entities. It models and describes the data using more standardized conceptual models, such as property graphs [9] and resource description framework (RDF) graphs [10]. Additionally, reasoning is another characteristic of the KG and it helps discover the hidden or indirect associations among the data. Furthermore, the KG has a standardized and common construction process [11]. These advantages of the KG contribute to solving the problems that IDNs are currently facing to. Thus, researchers are presenting methods for intent representation based on the KG to address the issues associated with intent input. For example, in the INDIRA architecture [3] and the EVIAN system [12], researchers both map the intents to RDF graphs to represent user intents.

To promote the abstraction of the underlying network, we present a digital twin-based approach for dealing with the massive amount of data generated by the underlying network. The digital twin creates digital models of physical network entities and simulates their behavior in the real environment with the support of data. The stability and operational efficiency of the network system are enhanced through virtual-real interaction and data analysis [13]. In our work, a digital twin-like network scenario is presented by the KG technology. Using the KG construction technology, the knowledge can be extracted from the status information of the physical network to create a KG that we refer to as the network state knowledge graph (NSKG). The NSKG is capable of dynamically updating the attribute information of nodes and relations to reflect changes in network status. In this way, it realizes the abstraction of the physical network. And in IDNs, the digital twin layer, constructed on the basis of the KG, can be used to abstract the physical network and facilitate verification of intents.

In our work, we present a KG-enabled IDN with the digital twin, which is termed as KID. It constructs the intent knowledge graph (IKG) and NSKG. During the interaction of both KGs, the balance between user intents expressed by the IKG and network resource capacities expressed by the NSKG is achieved. Finally, network policies expressed by the KG are automatically generated through this process. They are then sent to the underlying network after the digital twin layer verifies their feasibility. Our novel contributions are summarized as follows:

- We present an IDN framework named the KID, aiming to merge the semantic gap between user intents, policy generation, and the underlying network.
- We develop an intent language specification and a generic intent refinement model. Additionally, an intelligent translation technique is designed to achieve intelligent intent translation which takes user intents into consideration as well as network status.
- Finally, we provide a use case of the presented KID framework to demonstrate the feasibility and effectiveness of the proposed scheme.

The remainder of this work is organized as follows. In Section II, we present an introduction to the KID framework. Following that, we present the key components of the KID in Section III. Then, the experimental use case and results are provided in Section IV. Finally, in Section V, we summarize our work.

II. KID: Knowledge Graph-Enabled Intent-Driven Network with Digital Twin

Current IDN systems are difficult to ensure accurate understanding of user intents, precise generation of network policies, and fine extraction of network status. To meet novel services and networking requirements towards the next generation applications, we present an IDN system based on the KG and the digital twin, as shown in Fig. 1. And it is divided into the application layer, intent layer, digital twin layer, and physical network layer. Upper layer intents and the underlying network status are connected by the KG. From the top-down, the KG completes the parsing and representation of intents. Meanwhile, from the bottom up, the KG abstracts the underlying network resources and understands the resource capabilities provided by the underlying network. And during the reasoning process, the intent requirements are matched to the capabilities of underlying resources. Finally, the KG-expressed network policies are formed. And they are validated via the digital twin layer to instruct the physical network in carrying out the intents.

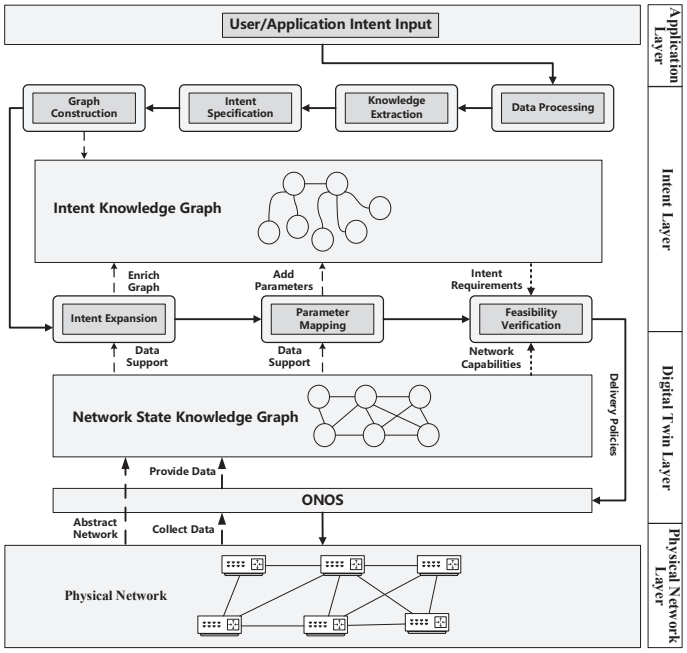


Fig. 1: The architecture of the KID.

A. Application Layer

The user or application intent input module is designed at the application layer. It is primarily aimed at providing interfaces to the users and applications. Meanwhile, this module enables

various sources and types of intent data to be entered into the system through front-end interface. It also allows users to declare their intents via text or voice. Then the intent data is sent to the data pre-processing module in the intent layer.

B. Intent Layer

The intent layer is responsible for completing the process of intent translation and policy generation. It is composed of the following modules: data processing, knowledge extraction, intent specification, graph construction, intent expansion, parameter mapping, and feasibility verification. At the intent layer, the workflow of the modules is approximately as follows: (1) To guarantee the data quality, the data processing module cleans and integrates the information of original intents. (2) The knowledge extraction module uses the algorithm model to perform entity recognition and relationship extraction on the pre-processed intent data. Then the triple data "entity-relation-entity" is generated. This intent triple data is referred to as the intent knowledge in this article. (3) The intent specification module standardizes the names of entities and relations in the constructed intent knowledge. (4) Using the normalized intent knowledge, the graph construction module creates a KG and saves it to the graph database. (5) To realize the intent expansion, the intent expansion module adds corresponding conditions and attribute information to the intent entities. (6) The parameter mapping module adds network parameter information to the intent entities, relations, and attributes in the IKG. (7) Feasibility verification module verifies the feasibility of intent execution according to the constraints of the underlying resources. It contributes to the ability of the IDN to adapt to the network autonomously.

C. Digital Twin Layer

The purpose of the digital twin layer is to abstract the physical network and to evaluate the feasibility of the intent execution. In our work, the NSKG and the open network operating system (ONOS) are used to construct a fundamental digital twin layer. The digital twin layer abstracts the physical network to ensure that the physical network can be accurately depicted by the KG. Additionally, it provides data support for the process of intent refinement in the intent layer. Furthermore, once the user intents are verified at the digital twin layer, the ONOS will instruct the physical network layer to execute the network policies which are from the intent layer.

D. Physical Network Layer

The physical network layer is a real-world network environment. It is composed primarily of the physical entities that comprise the end-to-end network, such as terminals, switches, routers, and other network element devices.

III. Detailed Scheme Design for the KID

In this section, we introduce the process of intent refinement by the KG first. Then, a digital twin scenario constructed based on the NSKG is presented to abstract the underlying network and verify the feasibility of the intent execution. Finally, we describe the process of network policy generation.

A. Intent

Considering the characteristics of the KG technology itself, we design a generic intent refinement model on the basis of the KG. Specifically, its working mechanism is as follows:

At first, the system pre-processes the intent data after receiving it. The program reads the data of intent input and loads it into memory. Following that, the regular matching rules are invoked to eliminate special symbols. Finally, the cleaned and filtered intent data is stored in the file.

Following data pre-processing, the intent data is sent into knowledge extraction model, which adopts the joint extraction method of subject model and object model based on Bi-directional Long Short-Term Memory. By fine-tuning the BERT [14] model, the subject model generates subject entities. Additionally, the object model predicts the corresponding object entities and relations based on the output subject entities. At the last layer of the subject and object models, the sigmoid activation function is used to predict multiple subject entities [15]. Finally, the knowledge extraction module generates intent triple data, which is referred to as intent knowledge. In training the knowledge extraction model, we borrowed the approach in [15]. Meanwhile, we reconstructed our own dataset according to the application scenario and trained the model through PyTorch. The basic form of the input data for the training is "xxx, from xxx to xxx, the time requirement is from xxx to xxx."

Due to the possibility that the extracted intent knowledge may contain non-standard and ambiguous words, it is necessary to normalize the intent knowledge. Therefore, we design intent grammar rules based on the extended backus-naur form [16]. Intent grammar rules normalize the names of entities and relations in the intent knowledge. For example, we standardize "time" and "endpoint" by designing the following grammar rules: "<Start-Time>::= *start*<Qualifierl><Point-in-Time> <End-Time>::= *end*<Qualifier2><Point-in-Time>", "<Endpoint>::= <At_Where>|<Route-Where> <At_Where>::= *at*<Location> <Source>::= "from"<Location> <Destination>::= "to"<Location>". The grammar design for other entities and relations is similar. This standard paradigm of intent language specification improves the generality and effectiveness of the paradigm.

Then, Py2neo constructs the IKG using the intent knowledge. And the graph database Neo4j is used to store the IKG. Py2neo is a client library and toolkit for working with Neo4j from within Python applications and from the command line [17].

To improve the accuracy in representing intents, the initial IKG must be extended. As a result of intent expansion, the corresponding conditions and attributes are added to the intent entities. For instance, if the user types "establish a reassurance-level voice service from A to B", the phrase "reassurance-level voice service" will be parsed. The necessary requirements for "reassurance-level voice service", such as delay, bandwidth, and other parameters, are added. It's worth noting that the intent expansion module will expand the IKG in accordance with the information from the digital twin layer.

After realizing the expansion of the IKG, the system will

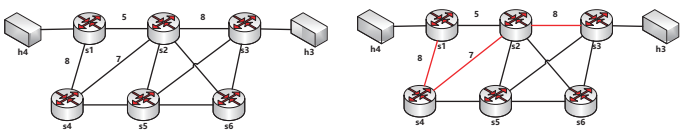
perform parameter mapping. Since the network parameter information is stored in the NSKG at the digital twin layer, the system will add the parameters to the intent knowledge based on the information in the NSKG. Firstly, the system parses the entities and relations in the IKG. Then, the system queries whether the names of these entities and relations are contained in the NSKG. When a consistent entity or relation name is matched in the NSKG, the system updates the IKG with the network parameter information that corresponds to these entity or relation names in the NSKG. Eventually, the network parameter data is added to the corresponding entities and relations in the IKG in the form of attributes.

So far, we have obtained the IKG containing detailed network parameter information. Following that, the feasibility verification module verifies the feasibility of intent execution. And during the process of feasibility verification, the system modifies the IKG according to the resource capabilities provided by the underlying network. Finally, network policies expressed by the KG can be formed and recorded in the IKG.

B. State

In combination with the ONOS, we construct the digital twin layer to abstract the physical network and verify the feasibility of intent execution.

At first, the system collects data from the physical network layer, such as network element entity data, resource data, log, protocol, routing, and status data, through real-time or non-real-time methods. Meanwhile, the ONOS can also obtain the physical network status information through relevant measurement means. After that, the collected data is processed and stored in the database. The system reads data from the database, and constructs the NSKG using the KG construction technique. Then, the system dynamically updates the attributes of nodes and relations in the NSKG based on the indicator data collected in real-time. This method reflects the changes of network status and achieves the purpose of abstracting the physical network.



(a) The network topology. (b) The diagram of route planning.

Fig. 2: The network topology.

Additionally, the digital twin layer supports the feasibility verification of intents during the intent refinement process. For example, according to the network topology shown in Fig. 2(a), the intent is identified and parsed to determine that the required minimum bandwidth resource between source node h4 and destination node h3 is 7. By default, the system executes the intent via the route s1 -> s2 -> s3. According to the NSKG, the default route's available minimum bandwidth is 5, which cannot meet the intent requirements. Therefore, it is necessary to find suitable paths for intent that can guarantee its execution. The planned route is s1 -> s4 -> s2 -> s3, as shown in Fig. 2(b). If there is no suitable path can be found based on the NSKG, this intent is deemed unsuitable for execution by the underlying network, and the result is fed back to the user.

C. Policy

We implement policy generation with the support of the KG during the interaction process between the IKG and the NSKG. In this way, we merge the semantic gap between user intents, policy management and network status.

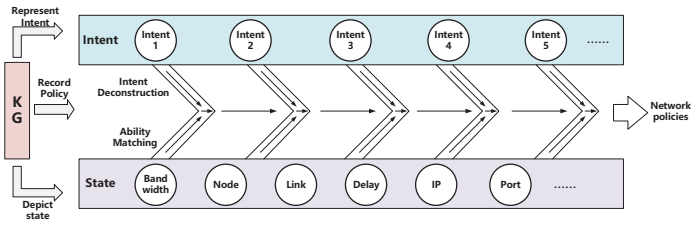


Fig. 3: The diagram of the interaction between the IKG and the NSKG.

The diagram of the interaction between the IKG and the NSKG is shown in Fig. 3. The system gradually implements intent expansion, parameter mapping, and intent verification through interaction between both KGs. It is in these processes that network policies are gradually formed. After implementing the parameter mapping, the system parses the IKG to understand the requirements of user intents. These intent requirements are expressed in terms of parameter information such as bandwidth, delay, and packet loss rate. Additionally, the NSKG is parsed to understand the resource capabilities provided by the underlying network. The network topology information in the NSKG is traversed by the system. Thus, the link connectivity of the network topology and link parameter information can be obtained. Then, by calculation, the links that can meet the intent requirements for bandwidth, delay, and packet loss rate are determined. Furthermore, all feasible paths are stored in the IKG. Throughout the process, the system completes the adaptation between the intent requirements and the capacity of the underlying network resources, and the feasibility of the intents is verified. Once all intents have been verified, network policies expressed by the KG are also formed. So far, the KID system completes the whole process of intent representation, policy recording, and network depiction, and realizes the knowledge-driven IDN through the KG.

IV. Experimental Implementation and Results

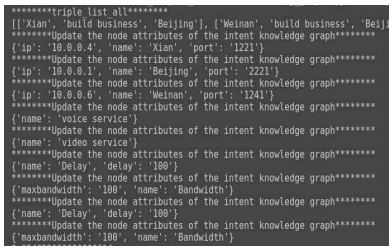
In this section, we implement a use case to demonstrate the feasibility and effectiveness of the proposed solution. Through the use case, we illustrate how the KID works. At last, the performance of the KID is evaluated by simulation experiments.

A. Experimental Implementation

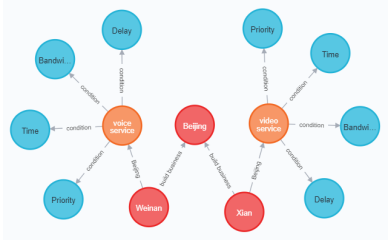
We input the intent as "Transmit an important video service from Xian user A to Beijing user B, the time requirement is from 10:30 on June 5, 2020, to 12:30 on June 5, 2020".

Meanwhile, multiple intents can be entered concurrently. The intent text is sent as an HTTP request to the system backend for data processing. Then, entity recognition and relation extraction are performed on the preprocessed data using the knowledge extraction model. The knowledge extraction model outputs triple data, which is saved in a file in JSON format. Based on the intent grammar rules, the triple data is normalized. Following that, Py2neo imports the triple data into Neo4j to create an IKG. During the data import process, the system detects the presence of intent entities that can be expanded based on the information contained in the NSKG.

In this use case, the entity "video service" has two conditions in the NSKG: "bandwidth" and "delay", with default attribute values. Therefore, when the intent knowledge is imported into Neo4j, the system appends corresponding conditions to the "video service". Meanwhile, the attribute values of both conditions are updated into the IKG. When the system detects that there is no intent entity can be expanded according to the NSKG, the intent expansion is completed.



(a) The process of updating parameters from the NSKG to the IKG.



(b) The final IKG displayed in the Neo4j.

Fig. 4: The process of forming the IKG.

Following the intent expansion, the parameter mapping is performed. When the system reads the intent entity "Beijing user B" and queries the node "Beijing user B" in the NSKG, it updates the attribute values of the node, such as IP address 10.0.0.4 and port number 1257, to the node "Beijing user B" in the IKG. Furthermore, similar steps are repeated until corresponding network parameters is added for all intent entities and relations. The process of updating parameter information from the NSKG to the IKG is illustrated in Fig. 4(a). And a view of the final IKG in Neo4j is shown in Fig. 4(b).

After successfully verifying the feasibility of intents, the system performs calculations using the NSKG to determine the forwarding paths of executable intents. As part of the network policies, all feasible paths are saved in Neo4j.

Additionally, we develop an ONOS-based routing algorithm application. When the ONOS receives the network policies, this routing algorithm application is able to calculate the intent satisfaction values of all feasible paths using the remaining bandwidth, delay, and packet loss rate collected by the system. And the optimal path is selected from feasible paths according to intent satisfaction values.

Finally, network policies based on the expression of the KG will be sent to the ONOS in the JSON format. Then, the ONOS starts the designed route planning application. It instructs the physical network to execute intents in accordance with the network policies that it has received.

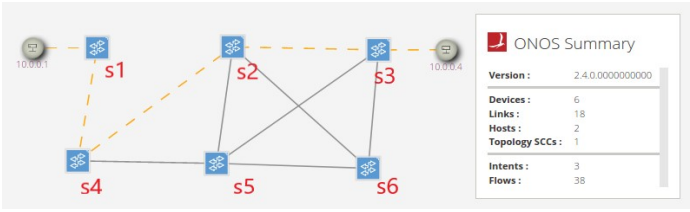


Fig. 5: Intent assurance for end-to-end routing.

B. Analysis of Results

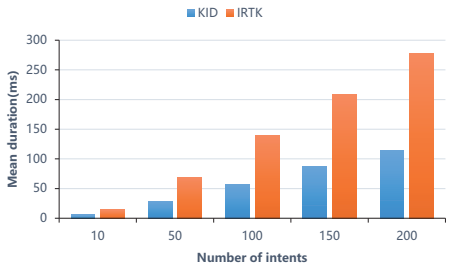
We create two virtual machines based on Ubuntu 16.04 system. One is used to run the KG-based intent translation system. And the other is used to build the network simulation environment and construct the NSKG.

At first, the assurance of the KID for end-to-end communication intent is tested. In our work, we use Mininet to create a network simulation environment with the same topology as that shown in Fig. 2. When the ONOS detects a link failure via the link subsystem, the system automatically determines a new path and quickly repairs the connection to maintain the end-to-end communication intent.

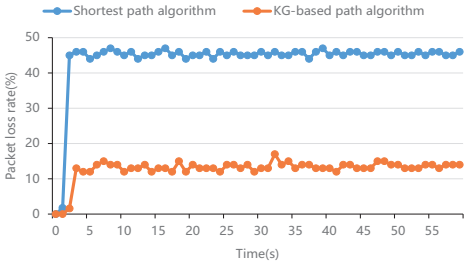
Fig. 5 shows the network topology diagram displayed in the ONOS control interface. It illustrates how the KID ensures end-to-end intent. Initially, the path between h1(10.0.0.1) and h4(10.0.0.4) is s1->s2->s3. When the link between s1 and s2 in the topology fails, the system can quickly reelect a new communication path and recover from the failure.

Then, the duration of the whole intent refinement process is tested. The measurement is performed over 200 iterations. In each iteration, a random intent is selected from a specific intent group. Through knowledge extraction, parameter mapping, and other processes, the duration required to complete intent refinement is determined. Finally, statistical mean values are calculated and bar charts are plotted. Meanwhile, we use the same method to test the time consumed by intent refinement toolkit (IRTK) [18] during the intent refinement. Eventually, a graph is obtained that compares the duration of intent refinement using the KID and the IRTK.

Fig. 6(a) shows the average duration of intent refinement. In the front-end interface of the system, 10, 50, 100, 150 and 200 user intents are input respectively. When the number of processed intents is 10, the KID takes about 5.68 ms. It



(a) The average duration of intent refinement.



(b) Link packet loss rate over time.

Fig. 6: The performance of the presented KID.

clearly shows that the KID takes significantly less time than the IRTK during intent refinement for the same number of intents. Furthermore, as the number of intents increases, the total duration of refinement increases roughly linearly.

Finally, the physical performance of the KID is evaluated in terms of ensuring video service. We use the video lan client to simulate end-to-end video transmission in Mininet. And the network topology is identical to that shown in Fig. 2. Additionally, the link's maximum bandwidth is set to 20M, and background traffic is pre-injected into the simulation network. Then, as the intent is issued in the simulation network, the system plans the path for intent execution. The routers follow the control path issued by the ONOS when forwarding packets for end-to-end video streams. Then, in the process of ensuring video service, iPerf is used to test the size of the link packet loss rate. Eventually, we compare the performance of the KG-based path algorithm against the shortest path algorithm in terms of reducing link packet loss rate in Fig. 6(b).

Fig. 6(b) illustrates the effect of the KG-based path algorithm and the shortest path algorithm on the packet loss rate links. As seen in Fig. 6(b), the packet loss rate of links is roughly at a stable level after the end-to-end video service is established. When the system forwards packets using the shortest path algorithm, the link packet loss rate is approximately 45%. The link packet loss rate is approximately 13% when the system forwards packets along the paths planned by the KG-based path algorithm. Clearly, the KID has good physical performance when it comes to ensuring intent execution.

V. Conclusion

To deal with novel services and networking requirements towards the next generation networks more effectively, we presented a novel IDN framework named the KID. We developed an intent translation technique based on the KG that took into account both user intents and the constraints of network status. Furthermore, the digital twin layer was used to verify intents and deliver policies. Finally, we verified the feasibility and effectiveness of the presented scheme by a use case. Our research explored and exploited the IDN to solve complex problems to continuously guarantee intents. And it contributes to merging the semantic gap between user intents, network status, and policy generation for knowledge-driven IDNs.

References

[1] K. Parker, "Cisco Systems-Intent-Based Networking: Building the bridge between business and IT." [Online]. Available: https://securenetworkers.com/2018/08/31/intent-based-networking/. [Accessed:2022].

[2] Y. Ouyang, C. Yang, Y. Song, X. Mi, and M. Guizani, "A brief survey and implementation on refinement for intent-driven networking," *IEEE Network*, vol. 35, no. 6, pp. 75–83, 2021.

[3] 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.

[4] A. S. Jacobs, R. J. Pfitscher, R. H. Ribeiro, R. A. Ferreira, L. Z. Granville, W. Willinger, and S. G. Rao, "Hey, lumi! using natural language for {Intent-Based} network management," in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pp. 625–639, 2021.

[5] A. Chaudhari, A. Asthana, A. Kaluskar, D. Gedia, L. Karani, L. Perigo, R. Gandotra, and S. Gangwar, "Vivonet: Visually-represented, intent-based, voice-assisted networking," *arXiv preprint arXiv:1904.03228*, 2019.

[6] C. Prakash, J. Lee, Y. Turner, J.-M. Kang, A. Akella, S. Banerjee, C. Clark, Y. Ma, P. Sharma, and Y. Zhang, "Pga: Using graphs to express and automatically reconcile network policies," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 29–42, 2015.

[7] A. Abhashkumar, J.-M. Kang, S. Banerjee, A. Akella, Y. Zhang, and W. Wu, "Supporting diverse dynamic intent-based policies using janus," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pp. 296–309, 2017.

[8] C. Gutiérrez and J. F. Sequeda, "Knowledge graphs," *Communications of the ACM*, vol. 64, no. 3, pp. 96–104, 2021.

[9] R. Angles, H. Thakkar, and D. Tomaszuk, "Rdf and property graphs interoperability: Status and issues.," *AMW*, vol. 2369, 2019.

[10] "Rdf - Semantic Web Standards." [Online]. Available: https://www.w3.org/RDF/. [Accessed:2022].

[11] F. Li, W. Xie, X. Wang, and Z. Fan, "Research on optimization of knowledge graph construction flow chart," in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 9, pp. 1386–1390, 2020.

[12] H. Mahtout, M. Kiran, A. Mercian, and B. Mohammed, "Using machine learning for intent-based provisioning in high-speed science networks," in *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*, pp. 27–30, 2020.

[13] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13789–13804, 2021.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[15] J. Su, "Hybrid structure of pointer and tagging for relation extraction: A baseline." [Online]. Available: https://github.com/bojone/kg-2019-baseline. [Accessed:2021].

[16] Strumenta, "Ebnf: How to describe the grammar of a language." [Online]. Available: https://tomassetti.me/ebnf/. [Accessed:2020].

[17] Py2neo, "The py2neo v4 handbook." [Online]. Available: https://py2neo.org/v4/#the-py2neo-v4-handbook. [Accessed:2021].

[18] P. Widmer and B. Stiller, *Design and Implementation of an Intent-Based Blockchain Selection Framework*. PhD thesis, University of Zurich Zurich, Switzerland, 2020.