IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 21, 2020, accepted July 3, 2020, date of publication July 13, 2020, date of current version July 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008721

Intelligent Routing Orchestration for Ultra-Low Latency Transport Networks

QINGMIN MENG[1,2], (Member, IEEE), JINGCHENG WEI[1], XIAOMING WANG[1,2], (Member, IEEE), AND HAIYAN GUO[1,3]

[1]Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[2]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
[3]Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210003, China

Corresponding author: Qingmin Meng (mengqm@njupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61801240, in part by the Open Research Fund of the National Mobile Communications Research Laboratory, Southeast University, under Grant 2019D16, and in part by the Open Research Foundation of the Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space (Nanjing University of Aeronautics and Astronautics), Ministry of Industry and Information Technology, under Grant KF20181910.

ABSTRACT Autonomous driving scenarios face the need for millisecond real-time response, which has led to the study of mobile networks with high speed and ultra-low latency. Software-defined networking (SDN) is recognized as a key technology for next-generation networks because it contains advanced functions such as centralized control, software-based traffic analysis, and forwarding rules for dynamic updates. In this paper, an SDN with flexible architecture is considered and a transport component is proposed. The component based on mesh topology is an example of joint route prediction and forwarding. First, different from existing transport protocols, the component can adopt a software-defined stream access control strategy that includes an extended forwarding mechanism (retransmission) to improve the short-term response performance. Second, we evaluate the impact of route prediction on transport network performance by using offline training and prediction. The key challenge here is that a suitable model needs to be trained from a limited training sample dataset, which will dynamically update the forwarding rules based on current and historical facts (network data). By introducing a parallel neural network classifier, an intelligent route arrangement is implemented in this work. Experimental results over different traffic patterns verify the advantages of the design. Not only does it enhance the flexibility of SDN, but it also significantly reduces the signaling overhead of the transport network without reducing the network throughput.

INDEX TERMS SDN, traffic control, routing, machine learning, prediction.

I. INTRODUCTION

Since 2018 autonomous driving technology has developed rapidly in China and the United States. Most of the leading companies adopt the intelligent way of single vehicle, that is, the vehicle's perception of the environment and decision-making on driving are completed by on-board sensors and calculation processing units. Single vehicle intelligence is difficult to complete the perception and real-time decision of complex road environment. The application and breakthrough of 5G in the field of autonomous driving and connected vehicles will break the limitation of the vehicle intelligence. 5G Ultra-Reliable and Low Latency Communications (URLLC) application scenarios such as autonomous driving, telemedicine, and industrial control have brought higher requirements for transmission delay, and this scenario is becoming one of the topics in the most popular frontier. At the same time, industry and academia have discussed the benefits of introducing software-defined networking (SDN) and network function virtualization (NFV) to 5G network architectures. SDN is a new type of network architectures and an implementation of network virtualization. The SDN processing function belongs to the 2-3 layers in the Open System Interconnection Reference Model (OSI), namely the data link layer and the network layer. This process involves switches and routers, and also lead to a 5G network architecture based on SDN/NFV. Therefore, this paper will discuss carrying-like networks (called transport

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

networks hereafter) and its intelligent routing in a new generation of mobile networks under the SDN concept. Subsequent sections A and B will describe the context design.

A. SOFTWARE DEFINED NETWORK

Compared with the existing mobile network, the next-generation mobile network that gradually introduces the SDN architecture needs to solve a variety of challenges in order to achieve flexible control of network traffic [1], [2]. The challenges faced can be summarized as the use of heterogeneous network environments (wired and wireless), the complexity of network management, increasing mobile traffic demand and diversified service demand, etc. Generally, network activities related SDN may be divided into several main categories, such as network operations and maintenance, configuration management, service orchestration, and policy management. Instead of letting intermediate network nodes manage traffic, SDN guides network traffic from the outside device in a software way. Building intelligent SDN should be a major task for next-generation mobile networks. This task facilitates the integration of artificial intelligence (AI) and machine learning into one SDN platform to replace manually completed network automation operations and significantly reduce the labor costs of network management [3].

In this work we try a combination of machine learning on mobile networks from the two aspects. First, we study a transport network based on a flexible SDN network architecture, which boosts the need for low time delay on the network side [4]. Secondly, we propose a routing orchestration method embedded in machine learning, which can enhance the agility of traffic engineering [5]. Route prediction (Here interchangeable with routing orchestration) is a basic task of traffic engineering and should be considered as a complex multivariate and multidimensional estimation problem. Machine learning has the ability to use algorithms to parse data, that is, to generate models through operations on the data. The input to the algorithm may even be limited historical data. Therefore, machine learning facilitates the SDN central controller to track seemingly random trends, and make effective decisions to complete intelligent configuration management and service orchestration in a network programmable process, which is the subject of this work.

B. MOTIVATIONS FOR THE TRANSPORT COMPONENT

The goal of this design is intelligent routing related to packet forwarding. By using the output from the predictors in the SDN central controller, a series of actions on various nodes within the transport component is periodically triggered. This series of actions ensures that the delay in the transport network is within a controllable range. Aiming for this goal, we summarize what we want to achieve into a comprehensive traffic control problem, which involves routing and flow rules in the control plane (CP). For example, multiple entities of SDN can successively collect various kinds of network data and complete path judgment under different time granularity. In addition, we need to consider flow rules in each node of the component, where multipath should stimulate the splitting of packets into the links of its next hop, to obtain a flow control with low congestion at times of high traffic. Path orchestration with small time granularity facilitates support for ultra-low latency service scenarios but requires a large overhead of controlled information and high computational complexity for online analysis [6]. Here we propose a wire-oriented, short-to-medium (such as 200 milliseconds) routing decision that uses an offline machine learning algorithm to learn the network state samples from a previous time window. The result of route prediction in the CP of SDN is then used as the sequence of path indexes on the next time window. In addition, the number of retransmissions within each trigger time slot defined by the software can ensure, as far as possible, a sub-millisecond transmission delay.

In summary, this paper makes the following contributions.

1) We propose the transport component with virtual control under a flexible network architecture. That is, the SDN central controller regularly arranges the forwarding paths within each component, instead of real-time path decision for edge equipment. The central controller uses a parallel deep neural network classifier to predict the path sequence in the next time window, to effectively achieve fast inference and reduce the control information cost.

2) For single constrained path selection, the component utilizes the enhanced stream access control strategy, including retransmission and splitting. The experimental traffic control can reduce transmission delay of the transport network.

3) We realize a complete simulation of computing, storage, and networking. The results of a large number of offline data analysis experiments show that the proposed schemes have a better performance than the centralized Open Shortest Path First (OSPF) based scheme.

The organization of the paper is as follows. Section II reviews relevant literature. Section III presents the software-defined network, the component, routing and forwarding as well as stream access control and data preprocessing. In Section IV, we introduce the multiple logistic regression (LR) classifier and the parallel deep neural network classifier. Section V presents the design in detail. Section VI presents simulation parameters, the network structure, performance evaluation, and discussion. Section VII is the conclusion of this paper.

II. RELATED LITERATURE

With the rise of autonomous driving technology, the demand for URLLC or event-defined URLLC is receiving attention, for applications such as self-driving cars, high-speed trains and drones. The speed of one mobile device can reach 500 km/h in the autonomous driving scenario [7]. This type of scenario places higher requirements on the end-to-end delay of the mobile network, and the delays of the user plane (UP) and the CP have become important performance indicators for 5G & Beyond and 6G wireless access networks.

Advanced network architectures that include wireless access networks and transport networks can be divided into centralized, flexible, and distributed types. Different cloud computing entities, such as cloud processing centers or Cloudlets, will be located in different locations of such network architectures. In these three architectures, the types of nodes and links of the fronthaul (FH) and the backhaul (BH) are very different. Usually, the part connecting the base station to the wireless access network is called FH, and the part connecting the access network to the core network is called BH. For the next generation of SDN-based mobile networks, converged FH and BH should significantly improve the performance and cost-efficiency of wireless access networks [8]. In LTE-like mobile communication systems, a base station of a small cell needs to be connected to a gateway through BH, and its delay is an important cause of the typical delay in LTE systems [9]. The literature in [10] suggests that the UP and the CP delays of SDN based core networks need to be less than 5 ms and 50 ms, respectively. For the SDN-based network, different combinations of FH and BH will vary end-to-end delays. By comparison, the typical end-to-end delay of existing cellular systems (LTE) is near 100 ms, and it is clearly far away from the transmission delay requirements of 5G URLLC application scenarios [11], [12]. Inspired by the convergence of FH and BH, this paper attempts to design a flexible network. The main improvement is to enhance wired BH by using a mesh topology, thereby shortening FH links and reducing the network side delay of the SDN-based network.

Here, we summarize current transport networks and routing techniques. The literature in [13] provides a detailed overview of network layer routing and network layer multipath solutions that promise increased throughput (through the use of concurrent multipath) and improved reliability and fault tolerance. The work in the literature studies two important design problems, i.e., how the CP calculates and chooses routing, and how the UP splits the data flow on the calculated path. Wireless mesh networks and Ad hoc networks are two kinds of multi-hop networks that adopt a packet radio communication mode. They adopt OLSR (Optimized Link State Routing Protocol) and AODV (Ad hoc On-demand Distance Vector Routing) respectively [14], [15]. Earlier literature in [16], [17] and [18] all refer to link-state based protocols. In these OLSR scenarios, each router needs to know the entire topology. In the AODV scheme, each router only needs to know the network status information of its surrounding neighbors [19]. In addition, for a combined system containing a wired network and a wireless mesh network, the author of literature in [20] discusses OSPF and other routing protocols. The literature in [21] studies the solution for small and medium-sized cellular BH in 5G networks. This work indicates that the delay target for 5G requires advanced link scheduling and routing design for mesh networks. For mmWave connections, the study also considers routing algorithms for backup paths. In view of differential services in WDM broadband networks, the work in [22] studies priority-based routing and access control.

Next, we summarize the related network applications of AI and machine learning into transport networks and routing. These applications may be considered as techniques for introducing data analysis into the application layer. For the mesh topology studied, the redundant link needs to optimize the path decision, which provides an opportunity for the introduction of advanced data analysis and automation into the transport network. The literature in [23] outlines the application of deep learning in network traffic control. Its preliminary results show that compared with traditional shortest path algorithms, a routing scheme based on deep learning can save the cost of control information and improve the performance of network throughput. For a transport network based on a mesh topology of 9 nodes, the author in [24] proposes an intelligent traffic control. The experimental results show that if a traditional routing protocol is adopted to calculate the optimal path, it may lead to low network congestion performance. This is because, in the presence of a high traffic load, the traditional routing strategy repeats the optimal path with low efficiency. The active routing mechanism in the literature can be summarized as follows: deep learning can learn the experience related to network congestion from data experience, thus reducing delays and improving packet loss rates. For both the central router schemes and the distributed router schemes, the authors of literature [25] propose a load balancing routing scheme assisted by machine learning, which uses network state information (NSI) in the form of queue length to train the neural network and make route predictions. The results show that under the specific network state data, the scheme assisted by deep learning is inferior to the machine learning scheme in terms of delay performance.

The final part of this section provides an overview of how machine learning can improve SDN performance. Although telecommunication networks are faced with the demand for increasing traffic, the resource efficiency of the existing transport network is often not high because of the lack of flexibility in resource allocation. One of the technical features of SDN is the use of dynamic resource allocation to track or adapt to variable traffic in the network. With the steady development of core network technology in recent years, SDN may be applied to the last segment of the next generation of mobile networks. This is due to the heterogeneous nature and complexity of the optical devices that make it up [26]. Moreover, the software-defined methods bring new control network solutions to operators, such as medium or short-term traffic engineering [27]. This new type of traffic engineering further promotes the application of machine learning in transmission networks.

The aforementioned literature in [1] reviews how machine learning algorithms are applied to SDN from the perspectives of traffic classification, routing optimization, prediction of quality of service (QoS) /quality of experience (QoE), and resource management and security. For dynamic routing in mobile metropolitan core networks, the work in [28] proposes a traffic prediction method based on machine learning. By taking advantage of programmability and complete network visibility, the method can perform network reconfiguration based on historical and current traffic loads. The authors of the literature [29] propose a machine-learning framework for cloud computing assisted resource allocation. The central network architecture in the literature requires the BH with a large traffic volume to transfer measurement data or signaling from BS to the cloud. Reference [30] summarizes the global, central visualization, and control features of software-defined networks. The literature also evaluates the routing framework for path calculation in the SDN central controller. This improves network QoS performance, such as latency and network capacity. The study points out that although software-defined routing is still being explored, SDN and its functions should adapt to the number of service requests. Aiming at the application of software-defined frameworks and video streaming services, the literature in [31] studies the joint access control and wired routing design. This work points out that SDN can support the dynamic function of future networks and intelligent applications by using networked operating systems. In addition, compared with the Q learning-based scheme and the open shortest path priority scheme, the SDN based scheme studied in this literature has a better performance. Furthermore, the literature in [6] proposes a routing strategy based on unsupervised deep learning running in the SDN central controller. In this strategy, the controller can collect network traffic information and regularly train Convolutional Neural Networks (CNN) to adapt to changing traffic patterns.

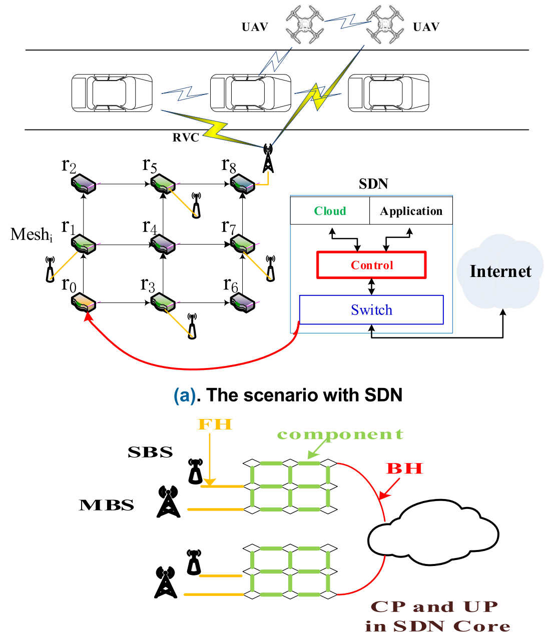
III. SOFTWARE DEFINED MOBILE NETWORKS AND COMPONENTS

This section describes the system studied and specifically discusses the components.

A. THE MOBILE NETWORKS

Fig. 1(a) shows a scenario that contains the studied component and conceptual SDN entities. The component can not only connect the core network to FHs, but also help to network multiple edge devices into a mesh topology, thereby improving the latency performance of the software-defined networks. Here N is the network size of the mesh topology. All links in Fig. 1(a) are bidirectional, and the number of nodes (routers) in the component can be configured according to the distribution of small base stations (SBSs) and a macro base station (MBS) in the coverage area. Each node in the component is configured with a buffer or Cloudlet to ensure that each node has an infinite queue length. Through FHs, SBSs are connected to nodes at the edge of the component and/or to adjacent SBSs or an MBS. By the flexible architecture of converged components and BHs, the SBS is also connected to the CP of the SDN. The CP can receive NSI from components through the uplink. In Fig. 1, UAV, and RVC are Unmanned Aerial Vehicle and Roadside-to-Vehicle, respectively.

Fig. 1(b) shows the studied flexible architecture or enhanced backhaul where a complete transport network consists of components as well as traditional BHs [8]. The connection between the component and the SDN Core is called enhanced backhaul. Hence the whole software-defined mobile network (SDMN) consists of SDN entities (switches, controllers, cloud processing centers, and applications), enhanced backhauls, FHs, SBSs, and MBS. Each component can be represented by a finite graph $G = (V, \bar{E})$. Where V represents the set of routers as $R = \{r_0, r_1, \ldots, r_{N-1}\}$, and $\bar{E}$ represents the set of edges (links). In addition, the geographically adjacent components form a cluster, and the components in a cluster are managed by the same SDN central controller. The design characteristics of the studied SDN are summarized as follows. 1) The central controller knows NSI of each component. 2) The controller can allocate transmission resources according to available resources and QoS metrics. 3) The controller can perform network configurations [4].



(a). The scenario with SDN

(b). The studied flexible architecture or enhanced backhaul

FIGURE 1. (a) The scenario with SDN. (b) The studied flexible architecture or enhanced backhaul.

To improve the flexibility of the studied architecture, an IP-based communication mode can be adopted between different component clusters. This allows the CP of the SDN to manage multiple clusters. The resulting larger network-size mobile network can provide greater geographic coverage. SDMNs deployed alongside the highways will offer a new generation of networking facilities, which will undertake a large number of real-time communications operations involving unmanned vehicles and drones. Compared to traditional transport networks, components will be a community of computing, storage, and networking elements.

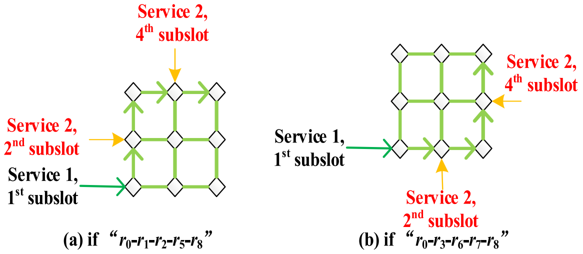
Here, we introduce the design idea of the CP. The cloud processing center in Fig. 1(a) is responsible for database processing and partial data analysis, and the application entity is responsible for the logic functions of customized network applications. The cloud processing center needs to receive NSI from the SDN center switch and perform routing calculations. The general functions of the central controller are data collection, decision making, and control. Its decision and control functions include routing sequence generation and distribution, and retransmission parameter distribution. Through the center switch, the central controller sends packets containing control information to the appropriate component. The router in the component parses the routing information in the packet to realize fast routing. The primary function of data analysis is data processing and route sequence prediction. It can extract historical network data from the database in the Cloudlet to form the feature vector. The vectors in a specific time window are stacked as a sample array. With the array as the input of the classifier, the machine learning algorithm will output the model. Using this model and the new network state information, the system can predict the next path index sequence. Thus, the central controller can use software to define future network events for both components and their internal nodes.

Now, we introduce the service of components. The service of the node within the component is divided into two categories, service 1 and service 2. The data flow of service 1 is from the center switch, and the data flow of service 2 is from the FH, which is connected with an SBS. Without losing generality, Fig. 1 considers how the CP can carry out route prediction and forwarding for high-priority service 1, that is, the path selection problem of a single constraint. The source node and destination node are r_0 and r_8, respectively. The base station connected to r_8 can use the LTE band or the new band for 5G to communicate with autonomous vehicles or drones [32].

B. ROUTING AND EXTENDED FORWARDING

First, the studied assembly is introduced where dynamic routing, the stream access control strategy, and extended forwarding are implemented. This paper proposes a type of path decision outside component, and store and forward inside component. In this scheme, each network node does not directly participate in the routing prediction, and each node will dynamically select the next hop node according to the instruction issued by the SDN central controller. A unique feature of the design is the extended forwarding or retransmission. Meanwhile, the flexible stream access control strategy can be adopted within the component. Here retransmission is seen as an extension of the stream access control strategy to deal with high traffic network conditions. Detailed stream access control can be found in subsection C of this section.

Second, we describe the extended forwarding mechanism in detail. Each node can execute data retransmitting except for the destination node. Link layer retransmission needs a simple calculation within the component. Here, node r_0 is assumed to have the role, that is 1) Collect the queue length information of each node at the end of the basic subslots (see the definition in subsection C) to determine whether each node retransmits. 2) Pre-process state data of each node at the end of each time window such as W_1. Retransmission does not depend on the protocol of the existing network layer, and the maximum retransmission number can be defined according to the parameters from the SDN application entity. The retransmission target will be aimed at millisecond-level return delay requirements. At the end of the basic subslots, if the queue of an intermediate node is not empty, the node immediately enters retransmission stage.



(a) if "r_0-r_1-r_2-r_5-r_8" (b) if "r_0-r_3-r_6-r_7-r_8"

FIGURE 2. Input control of data flow.

C. STREAM ACCESS CONTROL

A trigger time slot is defined as the data flow forwarding duration of a complete service throughout one component. Each trigger time slot of length T_0 contains a variable number of subslots of length T_{ss} where the N_f subslot is the basic subslots to forward a data packet to the next hops. The remaining N_b subslot of one trigger time slot is used for retransmission, which is used to transmit the remaining data packets in the queue of each node. Therefore, the minimum and maximum lengths of a trigger time slot are $T_{0b} = N_f T_{ss}$ and $T_{0m} = (N_f + N_b)T_{ss}$, respectively. In addition to the above retransmission, the access control strategy is considered where both input control at certain edge nodes and splitting of data packets at certain edge and/or intermediate nodes are implemented within the component, which boosts support for different priority services.

Assume that in Fig. 1, the input service type for node r_0 is high-priority service 1. The set of nodes containing service 2 is $\psi_2 = \{r_1, r_3, r_5, r_7\}$, the traffic volume of a single service 2 is assumed to be less than that of service 1, and the data flow destination of service 2 is r_8. At the beginning of the time slot, the packet of service 1 enters the queue of node r_0. As shown in Fig. 2, at the beginning of the subsequent subslot, the service 2 data flows are sequentially queued by the buffer to the nodes belonging to ψ_2. In Fig. 2(a), the routing decision selects the routing path "r_0-r_1-r_2-r_5-r_8". At the initial moment of the second subslot of this trigger time slot, node r_1 will be added to the service 2 data units. At the beginning of subslot 4, node r_5 will have an input of service 2. During the retransmission stage, all nodes have no external service input. Fig. 2(b) shows the input control or dynamics of another type of service 2, the path being "r_0-r_3-r_6-r_7-r_8". At the output end of the node r_3 in Fig. 2(b), the data packets stored in the node are sent to the next hop of the two links in accordance with even packets splitting.

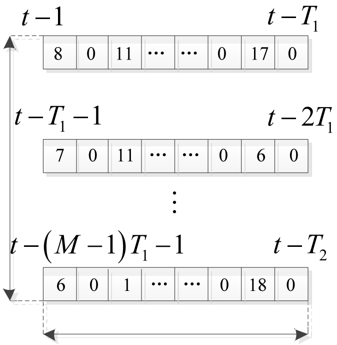


FIGURE 3. Historical data of queue length at node r_k.

D. DATA PREPROCESSING

NSI involves various types of network parameters, such as queue length of nodes, request rate of service packets, number of retransmission subslots, and queue delay of nodes. Only three types of network data are considered here, the data request rate f_1 for service 1, the data request rate f_2 for service 2, and the queue length f_3 for the node. This data can directly reflect the congestion of the transport network and indirectly reflect the influence of the routing decision.

In order to clearly define the design, three time windows W$_1$, W$_2$ and W$_3$ are defined, and the corresponding durations are T_1, T_2, and T_3, respectively. W$_1$ contains multiple variable-length trigger time slots, and T_1 is a multiple of T_{0m}, and $T_2 = T_1 *$ M (here T_1 and M are positive integers). In the beginning of each W$_1$, the component updates a path index, that is, the data flow of the services will be transmitted according to the same route. Every W$_2$, the proposed scheme carries out a route sequence prediction in test phase. It is important to note that the predictor does not predict a single path index, but rather a sequence of path indexes within the next W$_2$. W$_3$ represents the change cycle of Poisson distribution parameters λ of traffic flow. Within the same W$_3$, the parameter remains unchanged.

In order to effectively use the recent data of the network to predict the sequence of path indexes in the next W$_2$, the proposed scheme needs to collect and preprocess the historical data immediately adjacent to the previous W$_2$. The historical data of the destination node is not considered because it is assumed that the queue length of the destination node is always zero in each trigger time slot. Fig. 3 shows the NSI array at node r_k, (k = 0, *1*,..., *N*-2) where each row of the array is defined as "record," and each element in each row is defined as the processed NSI at the end of the trigger time slot (such as the queue length of the node). Now to define the reference time moment t. When a certain type of network data in the previous W$_2$ is collected at this moment, the CP of the SDN will output a historical data array with dimension $M \times T_1$. The scheme studied will pre-process M records for feature vectorization. Consider the simple averaging method, where each record is averaged to obtain a feature.

This study uses the number of data units arriving in each trigger time slot to characterize the service request rate in NSI.

Defining the service input node in Fig. 2 by $\varphi = \{r_0, r_1, r_3, r_5, r_7\}$, then the request rate at node n, $(n \in \varphi)$ is represented as $\mathbf{P}(n) = [\mathbf{p}_1(n), \mathbf{p}_2(n), \ldots, \mathbf{p}_M(n)]^T$. Each element of the array represents a record $\mathbf{p}_i(n) = \left[p_{i,1}(n), p_{i,2}(n), \ldots, p_{i,T_1}(n)\right]^T$ and here $p_{i,j}(n)$ represents the number of data units arriving in the j-th, $(1 \leq j \leq T_1)$ trigger time slot in the i-th, $(1 \leq i \leq M)$ record. After averaging all recorded data, the corresponding statistics and features are defined as

$$\bar{p}_i(n) = \frac{1}{T_1}\sum_{j=1}^{T_1} p_{i,j}(n) \tag{1}$$

$$\mathbf{Sp}(n) = [\bar{p}_1(n), \bar{p}_2(n), \ldots, \bar{p}_M(n)] \tag{2}$$

The set of nodes containing the queue is $\psi = \{r_0, r_1, \ldots, r_{N-2}\}$. For the node queue length, the recent queue length array node m,($m \in \psi$) can be described as $\mathbf{B}(m) = [\mathbf{b}_1(m), \mathbf{b}_2(m), \ldots, \mathbf{b}_M(m)]^T$. Each element of the array represents a record $\mathbf{b}_i(m) = \left[b_{i,1}(m), b_{i,2}(m), \ldots, b_{i,T_1}(m)\right]^T$ and $b_{i,j}(m)$ represents the queue length of node m at the starting point in the i-th, $(1 \leq i \leq M)$ record, the j-th, $(1 \leq j \leq T_1)$ trigger time slot. After averaging the data of each record, the corresponding statistics and features of the queue are defined as

$$\bar{b}_i(m) = \frac{1}{T_1}\sum_{j=1}^{T_1} b_{i,j}(m) \tag{3}$$

$$\mathbf{Sb}(m) = \left[\bar{b}_1(m), \bar{b}_2(m), \ldots, \bar{b}_M(m)\right] \tag{4}$$

Using the request rate of service 1, the request rate of service 2, and the queue length of a network node, the sample vector is represented as $\mathbf{S} = \{\mathbf{Sp}(n), \mathbf{Sb}(m) \,|n \in \varphi, m \in \psi\}$. The number of features of the sample vector is defined as U, and the maximum value of each element of the sample vector is F, which is determined by the maximum request rate.

Assuming that there are z candidate paths in the component under study, the set of path indexes can be expressed as $o_i \in \{V_1, V_2, \ldots, V_z\}$. Since the predicted target is a sequence of path indexes of length M within the next W$_2$, the predicted sequence can be characterized as $O = [o_1, o_2, \ldots, o_M]$.

IV. CLASSIFIER COMPLEXITY AND MODELS

The computational complexity is important to predict the achievable performance of the scheme. The routing classification problem studied uses two widely used supervised learning algorithms, LR and Deep Neural Networks (DNN) [33]. In this paper, multi-class logistic regression is considered as a baseline solution (Baseline). LR may have a moderate performance with a small number of training samples. During the testing process, the sample vector is mapped into a sequence of path indexes.

Positive integer parameters are defined by U, F, z, M, respectively. Therefore, the complexity of searching the entire sample space is $O\left(F^U\right)$, and the complexity of the search path sequence space is expressed as $O\left(z^M\right)$ [22]. When M takes a relatively small value, the computational complexity of the path search is moderate. Considering the above reasons, the number of rows M and the number of features U in Fig. 3 may be set to a positive integer less than 50 and 400, respectively.

After discussing the computational complexity, we will compare LR with the BP-based DNN algorithm. The prediction results of Baseline are discrete classification values. As a typical binary classifier, LR can be used to deal with two types of classification problems. Of course, it can also be used to deal with multiple types of issues, but it needs to be converted to One-vs-All. The DNN can learn and store a large number of input-output mode mapping relationships. Its learning rule is to use the steepest descent method to continuously adjust the weights and thresholds of the network through back propagation to minimize the sum of squared errors of the network. The topological structure of the DNN model includes an input layer, an indefinite number of hidden layers, and an output layer. Here the DNN corresponds to a representation learning method, which includes shallow learning and deep learning. Shallow learning is a model with only one hidden layer, and deep learning may be a model with multiple layers and multiple learning mechanisms (such as back propagation). The DNN usually needs to use a large number of samples to train the model to obtain excellent performance and complete very complicated nonlinear classification.

We then briefly describe the classifier with Baseline where the One-vs-All multi-class logistic regression model is utilized [34],

$$y = \arg\max_{i} h_{\theta}^{(i)}(\mathbf{x}) = \arg\max_{i} g\left(\boldsymbol{\theta}_i^T \mathbf{x} + b_i\right) \tag{5}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_k]^T$ is the sample, y is the label number of the sample, $\boldsymbol{\theta}_i = [\theta_1, \theta_2, \ldots, \theta_k]^T$ is the weight coefficient, b_i is the offset, and $g()$ is the sigmoid function.

In order to improve mapping efficiency, we propose a parallel DNN model. By training M parallel DNN sub-networks $A = [a_1, a_2, \ldots, a_M]$, the proposed scheme implements a classifier with a relatively low network scale. The network structure is shown in Fig. 4. M DNN sub-networks or branches receive the same samples, but each DNN sub-network a_i,$(1 \leq i \leq M)$ only targets one index o_i,$(1 \leq i \leq M)$ at the corresponding position on the path index sequence. Each branch is an independent classifier. Therefore, the parallel structure reduces the number of classifications of each DNN sub-network to z classes.

V. SIMULATION OF TRANSPORT NETWORKS

In order to effectively evaluate the design of a transport network, we need to build our own network simulator and use it to generate NSI. The parameter initialization of the network simulator involves various parameter settings, such as the request rate of the services, the node's service rate (processing capacity), and the maximum number of retransmission subslots.

A flowchart including four phases of the dynamic routing is shown in Fig. 5. It is divided into Initialization, Training, Testing, and Transporting. Initialization includes service traffic generation, NSI collection, routing decisions, and data packet transmission & retransmission. After this phase the NSI datasets described in Section III are generated. The routing decision uses the Dijkstra-based OSPF algorithm [18], and the path sequence output by the algorithm is used as the label of the training process. As a typical single-source shortest path algorithm, the central Dijkstra algorithm finds the shortest path from a starting node to all other nodes by calculating the link weight matrix of a finite graph. To favor the description of OSPF, Fig. 5 also includes the OSPF functional module. Moreover, in order to describe the number of samples in the datasets, the total training window, the total learning length, and the total test length are defined as W$_4$, T_4, and T_4, respectively.

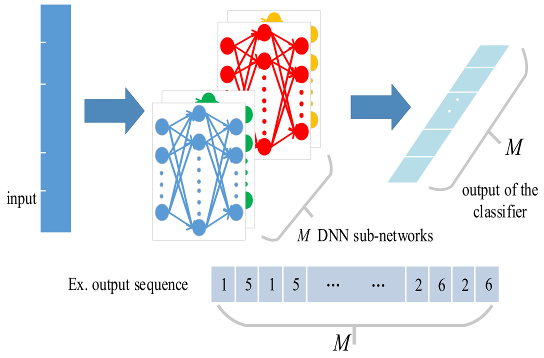


FIGURE 4. DNN network structure with M parallel branches.

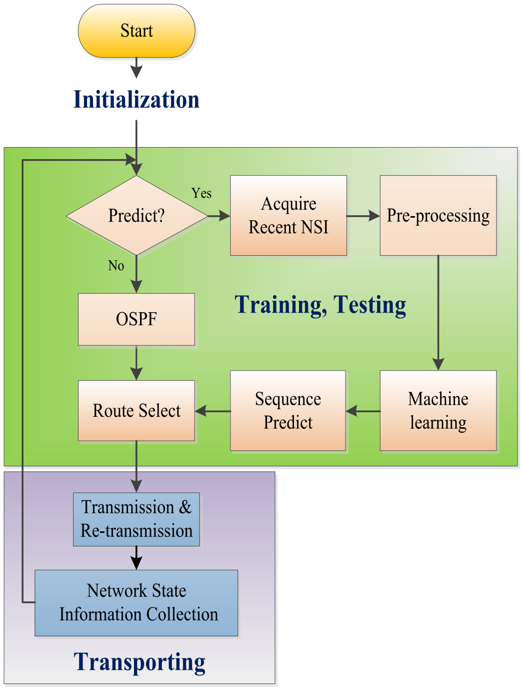


FIGURE 5. Four phases of the dynamic routing simulation.

During Training, the CP can use the two classifiers, namely Baseline and parallel DNN models. The NSI dataset is utilized for offline supervised learnings. Testing is located after Training, and the CP uses the trained model and new samples to predict the corresponding path sequence. In Testing, the CP can uses one of the prediction schemes related to the two classifiers. Every W_2, the predictor outputs a sequence of path indexes. Compared to T_2 (interval of the inference phase), T_4 is about four orders of magnitude of T_2. In the end, Transporting includes packet transmission & retransmission, and the collection of network data.

The steps of Testing are outlined below.

Step 1) At the beginning of each W_2, the cloud processing center processes the data in the immediately preceding W_2 and obtains the feature vector. Then, it uses the trained classifier to obtain the sequence of path indexes;

Step 2) Every W_1, the component queries the sequence of the path index until the elements of the sequence of the path index are completely traversed;

Step 3) The above steps are performed iteratively until the total test length.

During Transporting, the component will forward the data units according to the predicted path index. The processing capacity of a node is defined as C_r units each subslot (“unit” is a traffic unit), and the link weight is inversely proportional to processing capacity. Assume that in the u-th subslot data units are stored in the node k, ($k \epsilon R$). In the $(u+1)$-th subslot, traffic needs to be forwarded from node k to node h, ($h \epsilon R$), and the link between nodes k and h is represented as l_{kh}. At the beginning of the u-th subslot, the queue length of node k and the queue length of node h are $E_k(u)$ and $E_h(u)$ respectively, and the traffic of adding service 2 on node k is $Q_k(u)$. If node k has no traffic input of service 2, then $Q_k(u) = 0$. The weight of the link is defined as

$$W_{kh} = (E_k(u) + Q_k(u)) / C_h \tag{6}$$

Equation (6) reflects the congestion level of the link l_{kh} owing to the size of the traffic flow, that is, the state or weight of the link. Updating the length of nodes k and h at the beginning of the $(u+1)$-th subslot,

$$E_k(u+1) = \max(0, E_k(u) + Q_k(u) - C_h) \tag{7}$$

$$E_h(u+1) = E_h(u) + \min(E_k(u) + Q_k(u), C_h) \tag{8}$$

The steps of Transporting are as follows.

(1) The source node receives the data packet from the center switch and parses out the routing control information;

(2) The first N_f subslots forward the data of the services, and calculate the weight of the link, and update the queue of the node, respectively.

(3) At the end of the first N_f subslots, if the node’s queue still has data units, the retransmission function needs to be enabled. The node will select the appropriate N_b according to the queue length, and update the link weight and the node queue length at the same time.

TABLE 1. Simulation parameter description.

Symbol	Quantity	Value
N	Number of nodes in the mesh network	9
T_{ss}	Subslot length	1/6 ms
N_f	Number of the basic subslots	4
N_b	Number of retransmission subslots	2
T_0	Maximum length of the trigger time slot	1 ms
T_1	W_1 duration	10 ms
T_2	W_2 duration	200 ms
T_3	W_3 duration	1000 ms
T_4	W_4 duration	1200 s
C_r	Node r service rate	12 units
z	Number of candidate routes	6

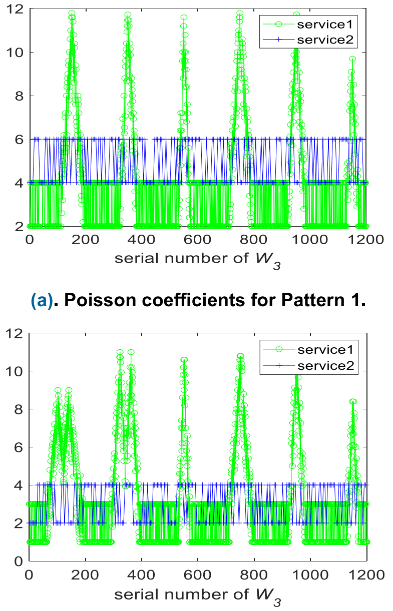
TABLE 2. DNN topology parameters.

Parameters	Value
Number of network layers	5
Number of neurons in the input layer	260
Number of hidden neurons in the 1st layer	400
Number of hidden neurons in the 2nd layer	150
Number of hidden neurons in the 3rd layer	100
Number of neurons in the output layer	6
Hidden layer activation function	ReLU

VI. PERFORMANCE EVALUATION

A. SYSTEM PARAMETERS AND DNN NETWORK TOPOLOGY

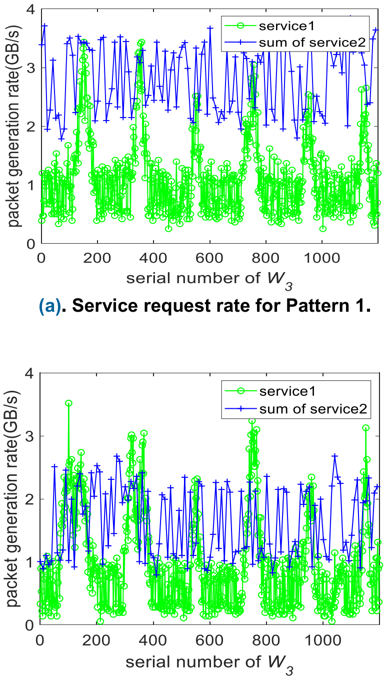
In the following network simulation, the configuration of components and services is represented as a single component. There is a data flow of a single service 1 and data flows of multiple services 2, respectively. At each trigger time slot, the above two types of data flows are added to the component, as defined in Fig. 2. Each unit of data flow contains 100 data packets with size of 1500 bytes. The system simulation parameters are shown in Table 1. In the configuration of T_{ss}, the maximum length of the trigger time slot representing short-term granularity is one millisecond, which also represents the expected time for the data unit to pass through the component. The maximum service rate at the component is $C_r * 100 * 1500/T_{ss}/1e9 = 10.8$ GBps (Gigabytes per second). In addition, T_2 represents short-to-medium time granularity. The uplink signaling overhead is defined as the amount of signaling from the component to the CP. Therefore, referring to Table 1, the proposed scheme reduces the signaling overhead by a factor of $M = T_2/\ T_1 = 20$. The parallel DNN algorithm in Training was developed using Python under the Tensorflow framework [35]. Matlab 2017a was used for the remaining network simulation. The simulation used a core I7 processor with 16GB RAM. We present the DNN network structure with M parallel branches or DNNs. Table 2 shows DNN topology parameters of the branch with three sub-layers each of which is a fully connected neural network layer.



(a). Poisson coefficients for Pattern 1.

(b). Poisson coefficients for Pattern 2.

FIGURE 6. (a) Poisson coefficients for Pattern 1. (b) Poisson coefficients for Pattern 2.



(a). Service request rate for Pattern 1.

(b). Service request rate for Pattern 2.

FIGURE 7. (a) Service request rate for Pattern 1. (b) Service request rate for Pattern 2.

B. EXPERIMENTS

1) TRAFFIC OF THE SERVICES

In order to simulate the actual arrival of data packets (units), it is assumed that the number of arrivals of data packets follows a Poisson distribution within a certain time, and the Poisson coefficient changes every T_3 triggering time slots. Fig. 6 shows the Poisson coefficient distribution where Fig. 6(a) and Fig. 6(b) show Pattern 1 and Pattern 2, respectively. Owing to the burst nature of service 1, the coefficient of this service in Pattern 1 for part of the time is presented as the shape of a single-peak triangle wave, and that in Pattern 2 as a single-peak and/or double-peak triangle wave. Service 2 has a non-burst nature, so the coefficient fluctuates randomly over every W_3.

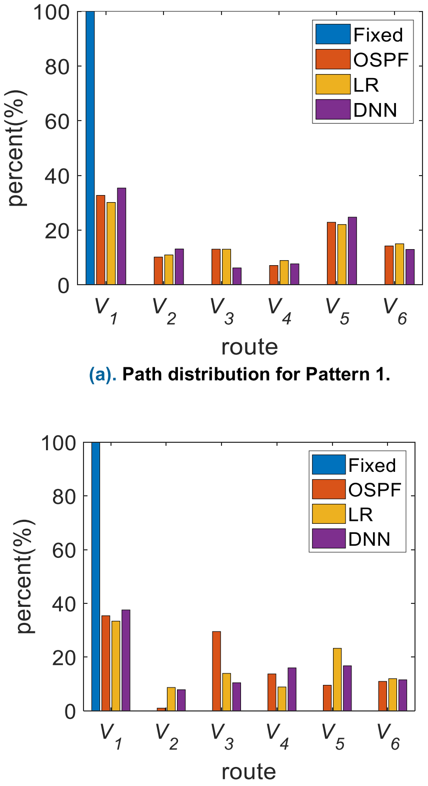
Fig. 7(a) and Fig. 7(b) illustrate the request rates for service 1 of Pattern 1 and Pattern 2, respectively. These figures also show the sum of the request rates for service 2 over W_3. According to Fig. 7, both in the short-term (T_1) and medium-long-term (T_3) time granularity, the data flow under study has excellent dynamics. In order to show the throughput of approximately 32 Gbps or 4 GBps, the requested rate is converted into GBps. Fig. 7(a) and Fig. 7(b), respectively, represents traffic scenarios with a high service request rate and a low service request rate.

2) PERFORMANCE IN RETRANSMISSION MODE

After the training of the classifier is completed during Training, the transport network uses four different routing schemes for Pattern 1 and Pattern 2, i.e., fixed routing (the path of all services remains the same), OSPF and two prediction schemes (Baseline and parallel DNNs). This work utilizes the following performance evaluation methods of delay, throughput, and control information (signaling) overhead. Fixed routing requires minimum signaling overhead, and OSPF requires maximum network information feedback and maximum signaling overhead. Both prediction schemes can reduce the signaling overhead by M times. Both the training samples and the first test samples are from the data of Pattern 1. In addition, the model trained on the training samples is used in the second test for the data of Pattern 2 to further evaluate the accuracy of the model under different service patterns.

In the $N = 9$ configuration, there are $z = 6$ optional paths from the source node r_0 to the destination node r_8, that is $\{V_1, V_2, \cdots, V_z\}$. Fig. 8 (a) and Fig. 8 (b) illustrate the path distribution in Pattern 1 and Pattern 2, respectively. Parallel DNNs have the legend "DNN". Pattern 2 has a lower rate of service requests and less congestion. Since fixed routing (legend "Fixed") is the selected path V_1, i.e., "r_0-r_1-r_2-r_5-r_8", so the probability of V_1 in Fig. 8 is 100%. Since training of the classifier requires the output from OSPF, the path distributions of OSPF, Baseline, and parallel DNNs are similar in Fig. 8 (a). In Fig. 8 (b), there is a certain deviation between OSPF and the two prediction schemes. This can be explained as the accuracy of both kinds of predictor is not high enough and they have limited generalization ability. Furthermore, in Pattern 2, OSPF has a low probability of selecting programs to V_2, but both predictors may have a higher probability of doing so. Therefore, the combined path scheduling and flow control help transform a non-ergodic process to an ergodic process, i.e., through all candidate paths.

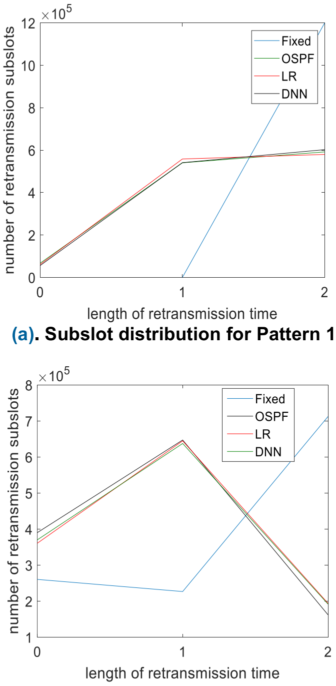
Fig. 9 (a) and 9 (b) show the number of retransmission subslots of Pattern 1 and Pattern 2, respectively, which reflect



(a). Path distribution for Pattern 1.

(b). Path distribution for Pattern 2.

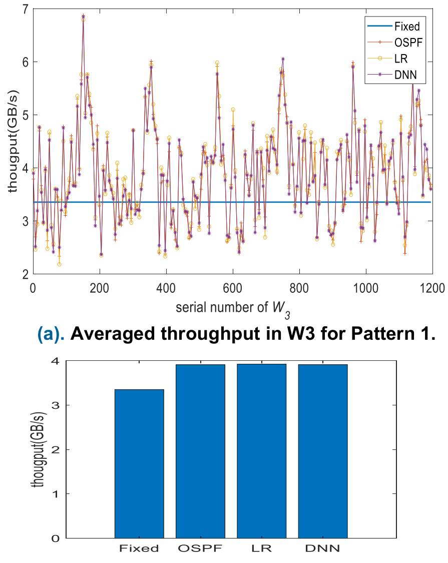
FIGURE 8. (a) Path distribution for Pattern 1. (b) Path distribution for Pattern 2.



(a). Subslot distribution for Pattern 1.

(b). Subslot distribution for Pattern 2.

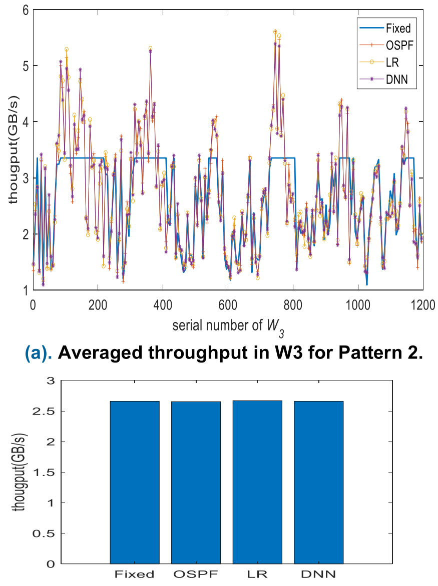
FIGURE 9. (a) Subslot distribution for Pattern 1. (b) Subslot distribution for Pattern 2.



(a). Averaged throughput in W3 for Pattern 1.

(b). Averaged throughput in W4 for Pattern 1.

FIGURE 10. (a) Averaged throughput in W3 for Pattern 1. (b) Averaged throughput in W4 for Pattern 1.



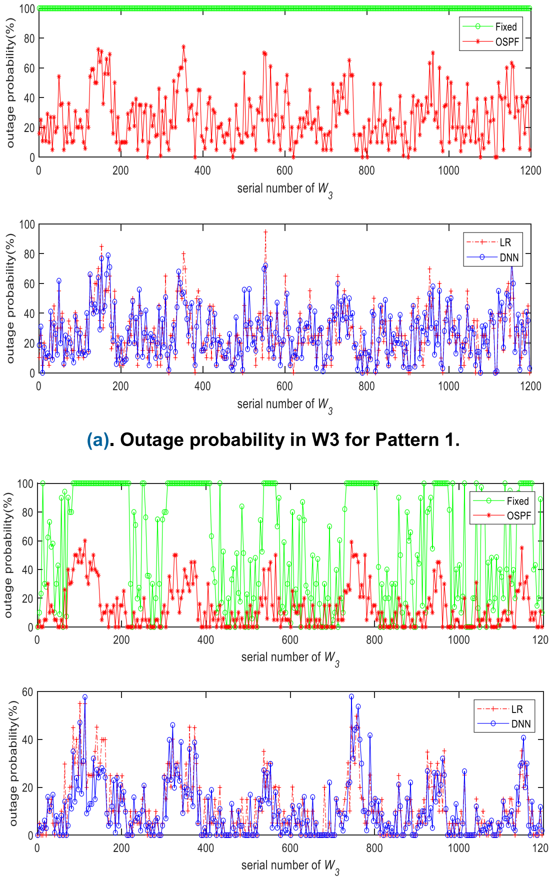
(a). Averaged throughput in W3 for Pattern 2.

(b). Averaged throughput in W4 for Pattern 2.

FIGURE 11. (a) Averaged throughput in W3 for Pattern 2. (b) Averaged throughput in W4 for Pattern 2.

the distribution of the number of retransmission subslots in each trigger slot. In one hand, OSPF and the two prediction schemes have a similar distribution. On the other hand, the number of retransmission subslots of these three schemes is less than that of fixed routes. This shows that dynamic routing can reduce network congestion and improve latency performance to a certain extent.

Fig. 10 and Fig. 11 show the average network throughput in both patterns. Fig. 10 (a) shows the throughput in W_3 (W3TH) under Pattern 1. The network throughput of the fixed route remains basically constant and is lower than that of the other three schemes. Fig. 10 (b) shows the performance within W_4(W4TH). It can be seen from Fig. 10 (b) that under the condition of high service request rate, the other three



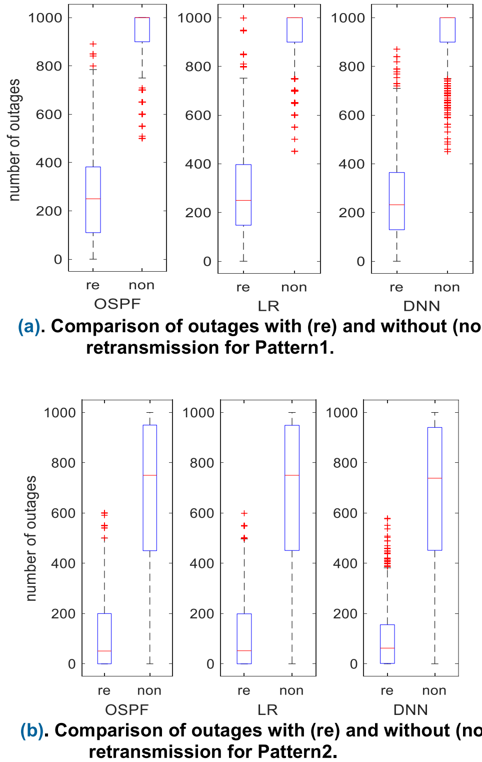
(a). Outage probability in W3 for Pattern 1.

(b). Outage probability in W3 for Pattern 2.

FIGURE 12. **(a) Outage probability in W3 for Pattern 1. (b) Outage probability in W3 for Pattern 2.**

schemes have better W3TH. Fig. 11 (a) shows W3TH within W_3 under Pattern 2. On the one hand, the curve of fixed routing is dynamic. On the other hand, its performance is lower than the other three solutions most of the time. It can be seen from Fig. 11 (b) that the W4TH of the four schemes under Pattern 2 are basically the same. This can be explained as the service traffic of Pattern 2 is relatively low. It can also be observed from Fig. 10 (a) and Fig. 11 (a) that in the medium-term statistical performance, the other three schemes are better than fixed routing.

Now, we define the outage probability as the proportion of time slots whose node queue length is greater than 0 at the end of the trigger time slot. The outage probability reflects the network congestion at a short time granularity. It can be seen from Fig. 12 (a) that the outage probability of fixed routing is always 1 in W_3. The congestion conditions of OSPF and the two prediction schemes change with the size of the service request rate. It can also be observed from Fig. 12 (b) that although the outage probability of fixed routing changes with the change of the service request rate, its value is significantly greater than these of the other three schemes. Furthermore, Fig. 12(a) and Fig. 12(b) show that the performance of Baseline and parallel DNNs is basically the same.



(a). Comparison of outages with (re) and without (non) retransmission for Pattern1.

(b). Comparison of outages with (re) and without (non) retransmission for Pattern2.

FIGURE 13. **(a) Comparison of outages with (re) and without (non) retransmission for Pattern1. (b) Comparison of outages with (re) and without (non) retransmission for Pattern2.**

3) COMPARISON OF THE NUMBER OF OUTAGES WITH AND WITHOUT RETRANSMISSION

In order to further compare the impact of the retransmission on outage performance, Fig. 13 shows a box plot of the number of outages in W_3 with and without retransmission (OSPF, Baseline, parallel DNNs). Fig. 13 (a) is a simulation for Pattern 1. In most cases, the number of outages of the three routing schemes with the retransmission (labeled "re") is only $15\%-40\%$ of the corresponding non-retransmission (labeled "non"). Fig. 13 (b) is a simulation for Pattern 2, which shows that the routing scheme with the retransmission can reduce the number of outages and that the performance of parallel DNNs is somewhat better than for OSPF and Baseline.

4) DISCUSSION

This paper proposes a method to enhance the flexibility of the network architecture. Here, the SDN controller periodically manages the components, instead of controlling each node at each time window (such as W_1). The reasons for the enhancement can be summarized as follows. By providing multiple redundant links or pipes in the network structure, the mesh topology helps to resist large traffic volumes and reduces time delays and congestion. In addition, this method also has the advantages of supporting the extension of coverage and relevant network configuration. According to the reference [36], the coverage distance of each small base station supporting a millimeter-wave connection is about 300 meters or less.

In order to achieve larger scale coverage (such as 900 meters), deploying components with the configuration of a 9-node mesh topology may be a promising backhaul solution.

The routing decision goal for a component is to choose a path from the candidate paths of the source-destination pair. The window length will affect the performance of this shortest path problem. The short window length helps real-time path selection. Its disadvantages are the significant increase in the amount of feedback and the need for shorter computation time for the CP of the SDN. A large window length will cause the network response time to be too long, which is not conducive to the removal of congestion in the transport network. In addition, the traditional routing method has a ''non-intelligent'' problem; that is, it is difficult to record the routing selection under the same situation in the past [24]. In order to solve this problem, we do not directly use the current state data for calculation, but use a small sample of historical NSI to train a classifier to achieve fast routing decisions under similar network conditions.

Finally, we further discuss the application of machine learning in routing. In machine learning-based route prediction, the mapping from feature vectors to candidate path series is a classification process of multi-class labels. The historical data forming the feature vector is only a subset of a very large sample space. This classification or sequence prediction often has multiple descriptions and presentations. On the one hand, the parallel DNN classifier proposed in this paper is convenient to output the sequence of paths, thereby reducing the signaling overhead and improving the inference efficiency. On the other hand, it may need to be improved in the following two aspects. Since the offline supervised learning algorithm cannot output path indexes in real time, it is necessary to introduce online machine learning algorithms (for example, reinforcement learning) to adaptively adjust the model according to any major events. Secondly, we may also need to apply machine learning methods to dynamically build resource models under different workloads, that is, upgrade the mesh components studied to Istio-like service grid [37]. Furthermore, the throughput performance of the transport network is constrained by the request rate of the service, the stream access control strategy, the mesh network topology, and the machine learning models. The influence of these parameters and models on the path distribution of the studied routing scheme needs further study.

VII. CONCLUSION

In order to realize the next generation software-defined network with large flow and ultra-low latency, this paper proposes a transport component with a mesh topology. This small network scale component can be used to enhance the flexibility of network architectures or backhauls in software-defined networks. For the path selection of a single constraint, the component internally uses a link layer retransmission mechanism as well as tentative input access control, which causes data packets to pass through the transport network at sub-millisecond rates. In addition, in order to reduce the network overhead caused by dynamic routing within a short time granularity, we propose the parallel DNN classifier and the path selection strategy. The intelligent and knowledge-based system in short-to-medium time window of length 200 ms can quickly predict the path index sequence in the time window in the future, thereby achieving a regular path orchestration. A large number of computer simulation results show that the proposed prediction scheme has better performance than the central OSPF in terms of the overhead of network control information. At the same time, in terms of throughput and outage probability, the proposed prediction schemes have basically the same performance as OSPF.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, ''A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019, doi: 10.1109/COMST.2018.2866942.

[2] W. Jiang, M. Strufe, and H. Schotten, ''Autonomic network management for software-defined and virtualized 5G systems,'' in *Proc. Eur. Wireless*, Dresden, Germany, 2017, pp. 1–6.

[3] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, ''A survey of networking applications applying the software defined networking concept based on machine learning,'' *IEEE Access*, vol. 7, pp. 95397–95417, 2019.

[4] P. Podili and K. Kataoka, ''Effective resource provisioning for QoS-aware virtual networks in SDN,'' in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–9.

[5] R. Guerzoni, R. Trivisonno, I. Vaishnavi, Z. Despotovic, A. Hecker, S. Beker, and D. Soldani, ''A novel approach to virtual networks embedding for SDN management and orchestration,'' in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, Kraków, Poland, May 2014, pp. 1–7.

[6] B. Mao, F. Tang, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, ''A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks,'' *IEEE Wireless Commun.*, vol. 25, no. 4, pp. 74–81, Aug. 2018, doi: 10.1109/MWC.2018.1700417.

[7] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, ''The roadmap to 6G: AI empowered wireless networks,'' *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019, doi: 10.1109/MCOM.2019.1900271.

[8] K. M. S. Huq and J. Rodriguez, *Backhauling/Fronthauling for Future Wireless Systems*. Hoboken, NJ, USA: Wiley, 2016.

[9] G. Zhang, T. Q. S. Quek, M. Kountouris, A. Huang, and H. Shan, ''Fundamentals of heterogeneous backhaul design-analysis and optimization,'' *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 876–889, Feb. 2016, doi: 10.1109/TCOMM.2016.2515596.

[10] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer, ''Towards a cost optimal design for a 5G mobile core network based on SDN and NFV,'' *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 1061–1075, Dec. 2017, doi: 10.1109/TNSM.2017.2732505.

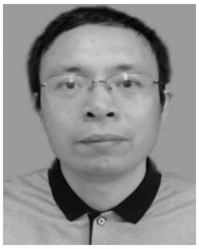
[11] O. N. C. Yilmaz, Y.-P.-E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, and J. Sachs, ''Analysis of ultra-reliable and low-latency 5G communication for a factory automation use case,'' in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., Jun. 2015, pp. 1190–1195.

[12] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, ''5G-enabled tactile Internet,'' *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016, doi: 10.1109/JSAC.2016.2525398.

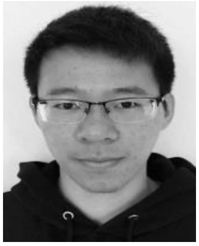
[13] J. Qadir, A. Ali, K.-L. A. Yau, A. Sathiaseelan, and J. Crowcroft, ''Exploiting the power of multiplicity: A holistic survey of network-layer multipath,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2176–2213, Fou. 2015, doi: 10.1109/COMST.2015.2453941.

[14] B. Fortz and M. Thorup, ''Internet traffic engineering by optimizing OSPF weights,'' in *Proc. IEEE Conf. Comput. Commun., 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Tel Aviv, Israel, Mar. 2000, pp. 519–528.

[15] L. Hanzo and R. Tafazolli, "A survey of QoS routing solutions for mobile ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 2, pp. 50–70, Feb. 2007, doi: 10.1109/COMST.2007.382407.
[16] S. Vutukury and J. J. Garcia-Luna-Aceves, "MDVA: A distance-vector multipath routing protocol," in *Proc. IEEE Conf. Comput. Commun., 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Anchorage, AK, USA, Apr. 2001, pp. 557–564.
[17] P. A. Humblet, "Another adaptive distributed shortest path algorithm," *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 995–1003, Jun. 1991, doi: 10.1109/26.87189.
[18] E. W. Djikstra, "A note on two problems in connection with graphs," *Numer. Math.*, vol. 1, pp. 269–271, Dec. 1959, doi: 10.1007/BF01386390.
[19] L. Zou, M. Lu, and Z. Xiong, "A distributed algorithm for the dead end problem of location based routing in sensor networks," *IEEE Trans. Veh. Technol.*, vol. 54, no. 4, pp. 1509–1522, Jul. 2005, doi: 10.1109/tvt.2005.851327.
[20] J. A. C. Fuertes, M. Philipp, and E. Baccelli, "Routing across wired and wireless mesh networks: Experimental compound Internet working with OSPF," in *Proc. 8th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Limassol, Cyprus, Aug. 2012, pp. 45–739.
[21] P. Wainio and K. Seppänen, "Self-optimizing last-mile backhaul network for 5G small cells," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 232–239.
[22] C.-L. Chang and S.-J. Kang, "Using reinforcement learning to the priority-based routing and call admission control in WDM networks," in *Proc. 5th Int. Multi-Conf. Comput. Global Inf. Technol.*, Valencia, Spain, Sep. 2010, pp. 126–130.
[23] N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue, and K. Mizutani, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 146–153, Jun. 2017, doi: 10.1109/MWC.2016.1600317WC.
[24] F. Tang, B. Mao, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 154–160, Feb. 2018, doi: 10.1109/MWC.2017.1700244.
[25] H. Yao, X. Yuan, P. Zhang, J. Wang, C. Jiang, and M. Guizani, "Machine learning aided load balance routing scheme considering queue utilization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7987–7999, Aug. 2019, doi: 10.1109/TVT.2019.2921792.
[26] R. Alvizu, G. Maier, N. Kukreja, A. Pattavina, R. Morro, A. Capello, and C. Cavazzoni, "Comprehensive survey on T-SDN: Software-defined networking for transport networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2232–2283, Jun. 2017, doi: 10.1109/COMST.2017.2715220.
[27] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of software-defined networking to traffic engineering," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 918–953, Sec. 2017, doi: 10.1109/COMST.2016.2633579.
[28] R. Alvizu, S. Troia, G. Maier, and A. Pattavina, "Matheuristic with machine-learning-based prediction for software-defined mobile metro-core networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 9, pp. D19–D30, Sep. 2017, doi: 10.1364/JOCN.9.000D19.
[29] J. Wang, J. Wang, Y. Wu, J. Wang, H. Zhu, M. Lin, and J. Wang, "A machine learning framework for resource allocation assisted by cloud computing," *IEEE Netw.*, vol. 32, no. 2, pp. 144–151, Apr. 2018, doi: 10.1109/MNET.2018.1700293.
[30] A. X. Porxas, S.-C. Lin, and M. Luo, "QoS-aware virtualization-enabled routing in software-defined networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 5771–5776.
[31] J. Yang, K. Zhu, Y. Ran, W. Cai, and E. Yang, "Joint admission control and routing via approximate dynamic programming for streaming video over software-defined networking," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 619–631, Mar. 2017, doi: 10.1109/TMM.2016.2629280.
[32] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018, doi: 10.1109/MNET.2018.1700105.
[33] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*. New York, NY, USA: Academic, 2105.
[34] A. Ng. *Multiclass Classification: One-vs-All*. Accessed: 2017. [Online]. Available: https://www.coursera.org/lecture/machine-learning/multiclass classification-one-vs-all-68Pol
[35] *Google TensorFlow*. Accessed: 2019. [Online]. Available: https://www.tensorflow.org
[36] H. Wang, P. Zhang, J. Li, and X. You, "Radio propagation and wireless coverage of LSAA-based 5G millimeter-wave mobile communication systems," *China Commun.*, vol. 16, no. 5, pp. 1–18, May 2019.
[37] G. Yu, P. Chen, and Z. Zheng, "Microscaler: Automatic scaling for microservices with an online learning approach," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Milan, Italy, Jul. 2019, pp. 68–75.



QINGMIN MENG (Member, IEEE) received the Ph.D. degree in communication and information systems from the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China, in 2007. He is currently an Associate Professor with the Nanjing University of Posts and Telecommunications (NUPT), Nanjing. His research interests include radio resource management and artificial intelligence for communications and networks.



JINGCHENG WEI received the B.S. degree in communication engineering from Hangzhou Dianzi University, Hangzhou, China, in 2017, and the M.S. degree in communication and information systems from the Nanjing University of Posts and Telecommunications. His research interests include radio resource allocation and artificial intelligence for software-defined networks.

XIAOMING WANG (Member, IEEE) received the Ph.D. degree in information and communication engineering from the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China, in 2016. He is currently a Lecturer with the Nanjing University of Posts and Telecommunications, Nanjing. His research interests include radio resource management, green communications, and machine learning in communications.

HAIYAN GUO received the B.Eng. and Ph.D. degrees in signal and information processing from the Nanjing University of Posts and Telecommunications (NUPT), Nanjing, China, in 2005 and 2011, respectively. From 2013 to 2014, she was a Postdoctoral Research Fellow with Southeast University. She is currently an Assistant Professor with NUPT. Her research interests include physical-layer security, cooperative transmission, energy harvesting, 5G networks, and speech signal processing.