AI-driven Packet Forwarding with Programmable Data Plane: A Survey

Wei Quan, Ziheng Xu, Mingyuan Liu, Nan Cheng, Gang Liu, Deyun Gao, Hongke Zhang, *Fellow, IEEE,* Xuemin (Sherman) Shen, *Fellow, IEEE,* and Weihua Zhuang, *Fellow, IEEE*

***Abstract*—The existing packet forwarding technology cannot meet the increasing needs of Internet development due to its rigid framework. Application of artificial intelligence (AI) for efficient packet forwarding is gaining more and more interest as a new direction. Recently, the explosive development of programmable data plane (PDP) has provided another direction of packet forwarding driven by AI. Therefore, this paper presents a survey on the recent research in packet forwarding driven by AI and PDP. First, we describe two frameworks of the packet forwarding, i.e., the traditional AI-driven packet forwarding framework and the new PDP-assisted AI-driven packet forwarding framework. Then, we focus on performance improvement of the packet forwarding under the two frameworks in four measures: delay, throughput, security, and reliability, correspondingly in four sections. In each section, we discuss the evolution from simple packet forwarding, to packet forwarding performance improvement with the assistance of AI, to the latest research on AI-driven packet forwarding supported by PDP. Through the review of packet forwarding in different evolution stages, we present the development rules and new open issues of packet forwarding at the end of each section. Finally, we summarize this survey, identify three directions in the development of future AI-driven packet forwarding, and highlight the challenges and issues in future research.**

***Index Terms*—Machine learning, packet forwarding, programmable data plane.**

I. Introduction

PACKET forwarding (PF) is an essential operation of communication in the Internet. Switching devices store and forward received packets through a series of preset processes to complete the delivery of data. However, with the rapid development of network technologies, the exponential growth of global Internet traffic stimulates an unprecedented demand in four aspects: low delay, high throughput, high security, and high reliability [1] [2] [3]. Examples include a line-rate packet forwarding capacitated with 6.50*Tbps* [4], a low-latency packet forwarding on the order of 0.32*ms* [5], a secure packet forwarding against high-volume attacks [6], and a reliable packet forwarding for highly dynamic vehicular networks [7]. The framework of traditional packet forwarding is rigid, using the same process in most scenarios. Many efficient algorithms for improving network performance cannot be deployed in the traditional framework, such as dynamic network resource allocation for customized network services and network attack behavior monitoring for a secure network.

Deploying artificial intelligence (AI) in networks is one potential way to satisfy the aforementioned demands [8] [9] [10]. The application of AI aims to flexibly allocate network resources according to different network service demands, enabling the deployment of customized networks. At the same time, the application of AI is real-time. When the network state changes, the resource allocation can be adjusted in time to adapt to network dynamics. For example, AI can effectively model dynamic features of multiple heterogeneous network paths and find the optimal scheme of resource allocation, to maximize the throughput of multi-path packet forwarding [11]. In addition, the application of AI can help improve network security, such as detecting a Distributed Denial of Service (DDoS) attack by identifying complicated packet behaviors [12]. While AI can be applied to effectively realize functions that packet forwarding cannot perform, deploying AI in networks has challenges. For example, deploying AI in a network controller can introduce unexpected delays, which becomes a stumbling block to high-rate packet forwarding [13] [14]. Further, the effectiveness of an AI model is affected by the granularity of network state information that the AI model can acquire.

Recently, the rise of programmable data plane (PDP) provides a solution to the challenges of AI-based packet forwarding, and attracts extensive attention for further performance improvement. The PDP can program the running logic of switching devices so that the software can flexibly choose an appropriate plane to deploy. For example, PDP provides a programmable plane for an AI model to complete the deployment and avoid the delay caused by deploying in a controller [15]. Meanwhile, PDP can provide more detailed network state information for the AI model to improve the operation. Due to the programmability and flexibility, PDP has a huge potential in AI-driven packet forwarding.

Looking into the evolution of the Internet, we observe that AI was introduced when traditional PF could not adapt to the development, and PDP was created when AI-driven packet forwarding needs further improvement. The evolution of the

Corresponding author: Ziheng Xu.

Wei Quan, Ziheng Xu, Mingyuan Liu, Gang Liu, and Deyun Gao are with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: weiquan@bjtu.edu.cn; zihengxu@bjtu.edu.cn; mingyuanliu@bjtu.edu.cn; gangliu93@foxmail.com; gaody@bjtu.edu.cn).

Nan Cheng is with Key State Lab. of ISN, and the School of Telecommunications Engineering, Xidian University, Xi'an, 710071, P.R. China (email: nancheng@xidian.edu.cn).

Hongke Zhang is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China, and also with PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518040, China (e-mail: hkzhang@bjtu.edu.cn).

Xuemin (Sherman) Shen and Weihua Zhuang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L3G1, Canada (e-mail: sshen@uwaterloo.ca; wzhuang@uwaterloo.ca).

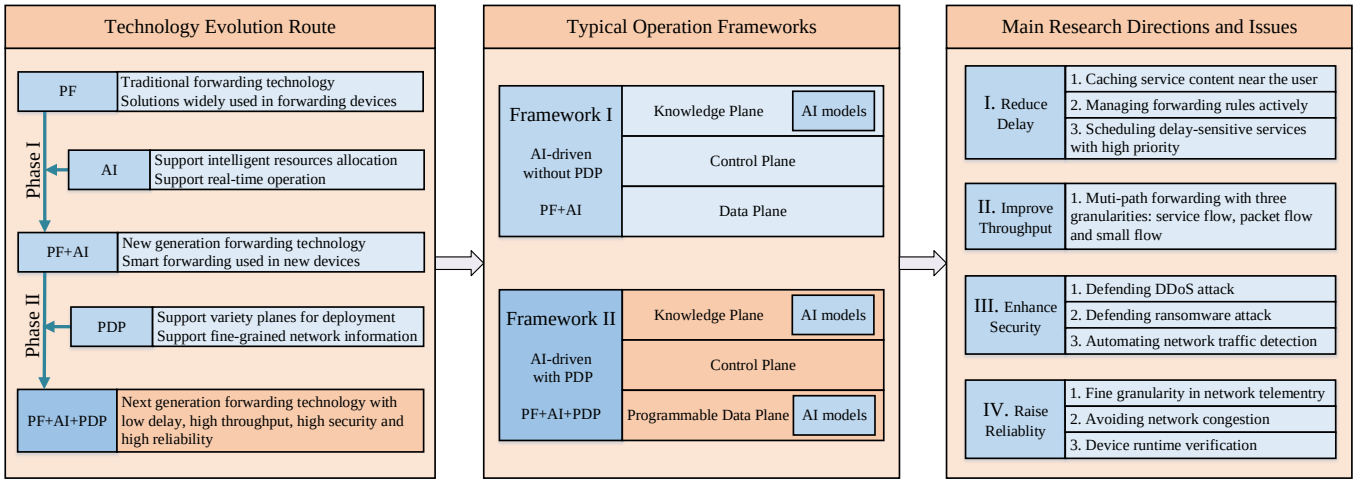


Fig. 1. A Road Map of This Paper

Internet is accelerated with the convergence of technologies. Globally, there have been extensive research activities on integrating PF, AI and PDP. A tutorial is in need to present a broad view of state-of-the-art packet forwarding that integrates the three technologies. Therefore, this paper surveys AI-driven packet forwarding mechanisms that can be deployed in PDP. A road map of this paper is shown in Fig. 1. First, we provide an overview of the traditional AI-driven packet forwarding framework and the new one supported by PDP. Based on that, we focus on research works that combine AI and PDP to improve the delay, throughput, security, and reliability of packet forwarding. A brief summary of future research directions and challenges is given at the end. The main purpose of this work is to provide a comprehensive survey on how PDP technologies can enhance AI-driven packet forwarding. Existing surveys have summarized research works on AI-driven and PDP-driven packet forwarding separately [16] [17] [18] [19] [20] [21] [22]. To our best knowledge, this survey is one of the first state-of-the-art overviews of the packet forwarding literature that combines AI and PDP.

The remainder of this paper is organized as follows. Section II describes the typical AI-driven packet forwarding framework and the new one supported by PDP. Section III and Section IV discuss research works on improving the delay and throughput performance respectively by combining AI and PDP. Then, Section V presents the security improvement and Section VI discusses the reliability improvement by the two technologies. Finally, Section VII identifies three potential directions and open issues in future AI-driven packet forwarding and Section VIII draws a conclusion.

II. Frameworks of AI-driven Packet Forwarding

In this section, we first introduce the framework of AI-driven packet forwarding, and analyze its advantages and limitations. Then, we give an overview of how PDP can enhance the framework (i.e., in overcoming the limitations) and discuss an effective approach that combines AI and PDP for packet forwarding.

Machine learning (ML), as one of the popular AI technologies, can make packet forwarding more secure, efficient and reliable [8] [23]. As shown in Fig. 2, the AI-driven packet forwarding framework without PDP includes data, control and knowledge planes from bottom to top. Information is transferred from one plane to another (shown by arrows in the figure), which forms a closed control loop. First, the data plane periodically reports network state information (e.g., interface throughputs) to the control plane during the process of forwarding packets. Next, the control plane collects and analyzes the information, builds a network state database, and reports the network state to the knowledge plane. Based on the network state, the knowledge plane selects an appropriate ML forwarding model according to the network performance demand. The ML forwarding models include global network secure forwarding model, global network reliable forwarding model, and so on. After selecting a model, the ML forwarding model generates a corresponding forwarding action message, and the knowledge plane distributes the action message to the data plane through the network strategy deployment module in the control plane. Finally, a closed control loop is completed to satisfy the network performance demand. As the closed control loop continues again and again, the whole network will eventually reach a steady and desired state.

In this closed control loop, there are two drawbacks: *(1) High interaction latency:* The interaction latency between any two planes is long [24], which is not suitable for delay sensitive applications. For example, the Ultra Reliable Low Latency Communication (URLLC) scenario requires 1*ms* latency [25] which is much shorter than the latency in the AI-driven framework; *(2) Coarse network state information granularity:* Due to the fixed data plane, network telemetry can obtain only coarse-grained network state information such as interface throughputs, but cannot obtain fine-grained information such as queue length of interfaces or specific flow rate [26]. Using the coarse-grained information will eventually lead to low accuracy of ML models, such as low accuracy of DDoS classification [27].

The programmable data plane provides possibility for overcoming the preceding drawbacks. The AI-driven packet forwarding framework supported by PDP is shown in Fig. 3. In

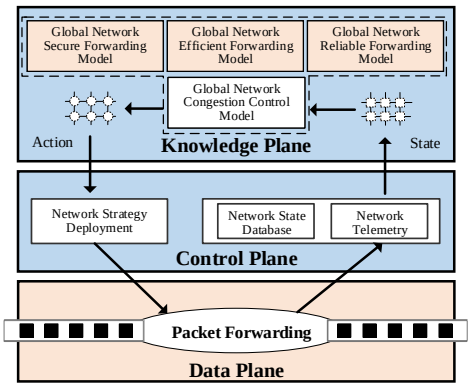


Fig. 2. AI-driven Packet Forwarding Framework without the PDP

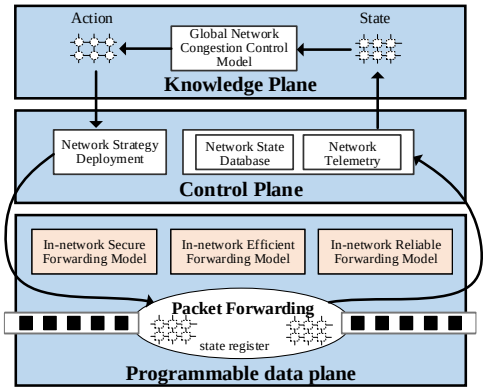


Fig. 3. AI-driven Packet Forwarding Framework with the PDP

comparison with that in Fig. 2, some models are transferred from the knowledge plane to the data plane, which reduces the interaction latency of the two planes and facilitates the ML model to acquire fine-grained network state information. The rest is still placed in the knowledge plane (e.g., global network congestion control model) to complete the overall regulation of the network.

The new framework has two advantages. On one hand, the flexible modification of the packet forwarding process provides a strong foundation for the optimization of the AI-driven framework. The PDP is adopted mainly for two reasons: (1) The PDP can provide line speed to reduce ML decision delay (the processing delay of a programmable switch is about several hundred nanoseconds [28], and the processing delay of a general ML platform is tens of microseconds [29]); (2) The match-action pipeline of PDP can dynamically adapt to network packet characteristics of the ML model, and execute its specific actions. The ML model deployed on the programmable data plane is called "in-network model". Depending on the requirements, the in-network model can have different functions. For example, Xiong *et al.* propose a mechanism to deploy the ML model on a programmable data plane to implement the in-network traffic classification [28]. In terms of network security, there are a large number of models, such as in-network abnormal flow monitoring model [30], in-network blackmail monitoring model [31], and in-network DDoS monitoring model [32] [33]. In terms of network performance, there are in-network cache model [34] [35], in-network forwarding rule management model [36], in-network interface queue management model [37], in-network packet scheduling model [38], in-network traffic flow scheduling model [39], in-network small flow scheduling model [40], and in-network Domain Name System (DNS) model [41]. In terms of network reliability, there are in-network devices running validation models [42].

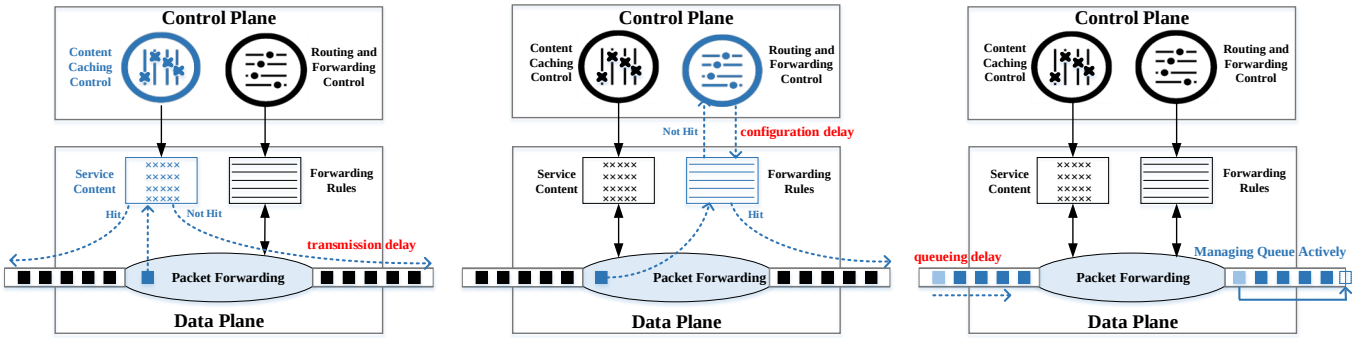
On the other hand, the PDP can collect the packet forwarding state information in fine granularity, which facilitates the accurate telemetry of network state to achieve high utilization of network resources. For example, Li *et al.* use PDP to collect network link state information, and use the information to determine whether the corresponding network link is congested [43]. They use an enhancement learning algorithm to minimize the maximum link utilization to avoid network congestion. The proposed global network congestion control model is deployed in the centralized knowledge plane, forming a closed control loop of "data plane uploading information - control plane analyzing information - knowledge plane establishing model and distributing action - control plane executing action - data plane updating forwarding rules".

III. Network Delay Performance Improvement by AI-driven Packet Forwarding with PDP

As a low delay is the basic requirement of most Internet services, the delay performance is usually one of the most important measures to be focused on. This section provides an overview of research works on reducing network delay by exploiting AI-driven packet forwarding with the PDP. The network delay that we discuss in this section is the time between a user sending a request packet and the user receiving a reply packet. During the packet transmission, the network delay can be divided into three main components: transmission delay, configuration delay and queueing delay. Corresponding to these three components, we can classify the methods of delay performance improvement into three approaches: (1) caching service content on switch near the user; (2) actively managing forwarding rules; (3) scheduling delay-sensitive services with high priority. Furthermore, with the help of AI and PDP, the delay performance can be enhanced by using the two frameworks discussed in Section II. In the following, we discuss delay performance maximization based on AI and PDP in each of the three approaches.

A. Caching service content near the user

Caching service content near the user is an idea of distributing service content data on switches in close proximity of the user in order to shorten the packet transmission distance and reduce the transmission delay. When a user requests some content, the corresponding switch will send the content data to the user in the shortest transmission distance. The specific working process of this mechanism is shown in Fig. 4(a). When the service request packet of the user arrives at a switch, the switch first matches the service content table. If hit (that is, the service content is stored on the current device), the stored service content will be fed back to the user directly; otherwise, the service request packet will be forwarded to



(a) transmission delay (b) configuration delay (c) queueing delay

Fig. 4. Three Network Delay Components and Corresponding Switch Operating Location (The Blue Part)

the neighbor switches and the neighbor switches will do the same mechanism. The service request packet is continuously forwarded until the service content is found. In this process, the transmission delay is the sum of the transmission time of each pair switches (one switch and its neighbor switch) in the transmission path.

In order to reduce the transmission delay, researchers have explored many effective schemes for caching service content near the user. For example, Azath *et al.* design a cloud-based fog paradigm architecture [44]. By layering fog, temporary storage nodes are used in the fog layers to speed up the caching of service content. This architecture can improve the efficiency of caching service content and reduce content transmission delay. Majeed *et al.* propose an active service content dissemination mechanism [45]. The server pushes the critical service contents to its neighbor switches to achieve content caching. This active dissemination mechanism shortens the distance between the user and the switch which stores service content, and reduces the transmission delay. Patra *et al.* establish an optimization model for the selection of nodes for service content caching [46]. They try to select a relay node to realize a low average transmission delay from the node to every user who requests the content. Furthermore, they propose a fast algorithm with time complexity (O($n$log$n$)) to solve the optimization model.

ML for caching service content: Due to the dynamic, complex and heterogeneous characteristics of caching service content, it is challenging to select appropriate service contents stored on switches near corresponding users to ensure the best quality of service for each user. But with the assistance of ML, effective service content caching can be realized and network delay can be further reduced. Specifically, the service content that users may need can be predicted through machine learning. Then, we choose the content with a high probability and cache it on the switches, which can provide users with satisfactory service experience and reduce the transmission delay of service content.

Tsai *et al.* build a deep learning model for social media networks [47]. They obtain the user's conversation information and adopt the convolutional neural network (CNN) to analyze the context of the sentence, and then predict the service content that the user may request. According to the prediction, the corresponding service content is cached in the switch near the user in advance. It is shown that the model can reduce the transmission delay by 30% as compared with the traditional method. Cheng *et al.* propose a new mechanism of caching service content, which uses new Bayesian learning methods and reinforcement learning [48]. The new Bayesian learning methods is to predict user preferences and estimate individual content request probability (ICRP). Then a caching strategy is formed based on the results, and reinforcement learning is used to further optimize the caching strategy. Finally, the transmission delay for users to obtain the required services can be reduced. Saputra *et al.* design an active caching mechanism for service content caching [34]. First, they collect access information of users from the content server (CS) and use deep learning to predict users' service content requirements. To address the problem of privacy disclosure caused by sharing users' data during information collection, they improve the architecture by changing the machine learning model into distributed deep learning to protect users' privacy, and further reduce errors in content requirement prediction. The proposed mechanism reduces the network transmission delay as compared with other machine learning algorithm.

ML and PDP for caching service content: Because the PDP has flexible programming characteristics, it can provide multi-dimensional network state information for ML. Based on the multi-dimensional information, the accuracy of predicting users' content requirements can be further improved. Liu *et al.* propose a deep-learning-based content popularity prediction (DLCPP) mechanism [49]. The mechanism collects the spatial-temporal joint distribution data of network state and predict the content popularity based on distributed deep learning. In addition, they put forward a new caching scheme, which effectively adapts to the DLCPP, improves the caching performance, and reduces the service content transmission delay.

B. Managing forwarding rules

The application of actively managing forwarding rules configures packet forwarding rules in advance and adjusts the rules according to the network states, which can reduce the interaction frequency surge between switches and controllers in actual operation of the system and finally reduce network delay. As shown in Fig. 4(b), in a network under centralized control such as a software-defined network (SDN), forwarding

rules are in general passively sent to switches according to request packets from users. When a request packet is forwarded, a switch that receives a request packet will first check if there is any request content stored locally. If not, the request packet needs to be forwarded. The switch matches with the table of local forwarding rules. If hit, request packets will be forwarded according to the rules to the next switch to request content; otherwise, the packet information is uploaded to the controller to request a forwarding rule. The controller generates and distributes new forwarding rules according to the network state information so that the request packet can be forwarded correctly and the user can quickly acquire corresponding content. This passive distribution mode not only causes the frequency surge of interactions, but also limits the number of forwarding rules (i.e., only the rules corresponding to the request packet that has been forwarded will be distributed). In this process, configuration and distribution time of forwarding rules is a key factor that affects the network delay of packet forwarding from the user and the server.

Managing forwarding rules actively can solve the problems to some extent and, on this basis, researchers put forward various solutions. Luo *et al.* formulate an NP-hard problem of minimizing the end-to-end network delay [50]. They design a heuristic caching algorithm for forwarding rules to reduce the interaction delay between switches and the controller, and minimize the end-to-end transmission delay of data packets. Huang *et al.* study the compromise point between the cost of switch caching and the cost of network controller distributing forwarding rules, and build a minimum weighted flow provisioning model [51]. According to the network flow information acquisition, they design an off-line algorithm (i.e., after obtaining the network flow information) and two online algorithms (i.e., before getting the flow information).

ML for managing forwarding rules: Similar to caching service content, managing forwarding rules can benefit from ML to further reduce the configuration delay in various aspects. Bera *et al.* design a mobility-aware adaptive flow-rule placement scheme for mobile access networks [52]. In the presence of user mobility, the scheme predicts the next possible location of the user through a K-order Markov chain, and deploys the forwarding rules to the corresponding switches in advance through the active forwarding rule deployment method, so as to reduce the network configuration delay. In actual forwarding, the switch has limited cache space to store forwarding rules. How to select appropriate forwarding rules stored in the switch and the rest in the controller to achieve a balance is a research issue. Mu *et al.* design a mechanism based on deep reinforcement learning to analyze and optimize this balance, which aims at reducing the controller overhead while ensuring the switch caching [53]. Filali *et al.* present a load balancing mechanism for multiple SDN controllers to reduce the configuration and distribution time of forwarding rules [36]. First, the mechanism predicts the load of forwarding control devices (e.g., SDN controllers) through Auto Regressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM). Then, according to the predicted results, a migration algorithm for forwarding rules based on reinforcement learning is adopted to avoid a large service response delay in the high-load SDN controller, which can minimize the configuration and distribution delay of forwarding rules.

PDP for managing forwarding rules: Due to the programmable characteristic of PDP, we can reconstruct the caching architecture and accelerate the storage and matching speed of forwarding rules. Zhang *et al.* use PDP to design a behavioral-level caching mechanism [54]. By uniformly caching all entries, this mechanism speeds up the matching of forwarding rules and reduces network delay. Parizotto *et al.* propose the ShadowFS system and construct the data plane [55]. The system uses smaller caches to store and manage entries, which increases the speed of monitoring and telemetry on the data plane. Therefore, the switch can obtain high frequency information and feed this information back to the controller to adjust the forwarding rules and reduce network delay. Grigoryan *et al.* redesign the caching architecture based on PDP and FPGA, which includes two caching layers in the architecture [56]. The architecture can accelerate the cache speed and realize fast forwarding rule matching when packet is forwarded.

C. Scheduling delay-sensitive services with high priority

Scheduling delay-sensitive services with high priority reduces the packet queuing delay of different delay-sensitive services on switches by active queue management. As shown in Fig. 4(c), in the traditional packet forwarding model, packets of various service flows enter the queue according to their arrival time at the switch, and then leave the queue according to the first-in first-out (FIFO) principle after processing at the switch. Therefore, the queuing time is a key factor that affects the end-to-end network delay in packet forwarding.

In order to reduce the packet queuing time, a variety of active queue management (AQM) mechanisms has been proposed [57]. Olariu *et al.* design a queueing mechanism based on packet delay [58]. The mechanism divide Voice over Internet Protocol (VoIP) packets into five priorities according to the delay and put them into different push-in first-out (PIFO) queues. The design can avoid most congestion of data packets and reduce the end-to-end delay. Qiu *et al.* propose a backpressure queue scheduling algorithm in order to make the switches respond to emergency packets in a timely manner under an elephant flow [59]. The algorithm provides the shortest forwarding path for emergency packets and ensures that queues at the switches will not be congested on this path, so as to reduce the end-to-end network delay of emergency packets.

ML for scheduling delay-sensitive services: ML can help to scale services and provide different priority queues for different services. Alnoman *et al.* design a two-class priority queuing system based on supervised learning [37]. They train and test the system from simulated data sets and then identify delay-sensitive applications in an IoT network based on characteristics such as type and location. The system assigns high priority queuing for the delay-sensitive applications to reduce the end-to-end delay. Furthermore, different from this binary division (i.e., delay-sensitive and delay-insensitive ser-

vices), Zhu *et al.* propose SmartTrans, which provides multi-level priority queues for different flows [60]. They use deep learning to classify and predict the ranking of different flows, and provide corresponding priority queues according to the prediction results. In addition, they expand the buffer of the switch and improve throughput when the network flow surges.

PDP for scheduling delay-sensitive services: PDP can provide fine grained queueing information for ML, but there are some challenges in implementing AQM on PDP. Researchers try to deploy AQM in programmable switches. Papagianni *et al.* implement an AQM algorithm (PI2) that needs only the information on the data plane [61]. This algorithm uses PDP and the information of inlet and outlet pipeline of the switch. The queuing delay can be reduced in a short time once PI2 is deployed. Kundel *et al.* also implement an AQM algorithm (CoDel) based on PDP [62]. These researchers demonstrate that AQM can be supported for queue management in a network composed of programmable switches. Alcoz *et al.* propose a solution (strict-priority PIFO (SP-PIFO)) to the problem of deploying PIFO on hardware [63]. Specifically, they use the P4 programmable language to dynamically adjust the priority of packets based on network state and provide corresponding priority queues. The SP-PIFO can be fully implemented on Barefoot Tofino switches and has a low hardware overhead.

ML and PDP for scheduling delay-sensitive services: Using ML to realize AQM has a challenge. Some AQM algorithms are so complicated that it is difficult to deploy them in traditional switches. The PDP provide a solution to this issue. Shi *et al.* design a two level queueing management mechanism based on ML and PDP [64]. This mechanism uses OpenFlow switch architecture and extreme gradient boosting model (XGBoost) to accurately obtain queueing delay of switches and provide differentiated service priority for applications, which can improve the quality of service and reduce the end-to-end delay. The PDP not only can provide a platform for deploying complex ML and AQM algorithms, but also can provide refined network information for ML. Zhang *et al.* propose an application classification method based on a hybrid deep neural network [65]. This method automatically obtains a large amount of accurate network flow processing information through PDP, and then learns it through the hybrid deep neural network to classify applications at a high accuracy.

D. Remarks

We start our review from three traditional network delay reduction methods. Then, we discuss how ML and PDP help further reduce the network delay based on these three ways respectively. In the following, we provide an overview of the three approaches.

Caching service content near the user: The caching service content mainly has two stages: distinguishing service content popularity and choosing their storage location. In the first stage, we want to find out the popularity of each content, which helps choose its storage location. In the second stage, we try to select an appropriate location of each content to reduce the transmission delay.

Most of the researchers adopt machine learning to determine the caching of service contents in the first stage. They use a variety of machine learning models to reach a popularity ranking. When going further into the second stage, selecting an appropriate location should be based on the switch states and link states for data transmission. For example, if a switch which stores a large amount of high popularity contents is broken, its stored contents cannot be quickly transmitted to users and the service quality is affected. Alternatively, when the transmission link is congested, a switch closest to the user no longer has the minimal transmission delay. In short, the choice of caching location is not simply a ranking of content popularity, but also depends on the actual network environment. Therefore, further research on machine learning models is required to improve the caching service content performance.

Another research issue is how to better exploit the PDP for content caching. The PDP can help with caching service content in both two stages. In the first stage, due to its programmable capability, the PDP can observe the forwarding of service content and provide ML with accurate forwarding states to improve the ranking accuracy. In the second stage, the PDP can provide comprehensive network state information to help ML select an appropriate cache location. That is, the PDP has a potential to improve service content caching, which is a future research direction.

Actively managing forwarding rules: Manage forwarding rules includes three aspects: deploy forwarding rules in advance, optimize existing forwarding rules, and balance the cost of switches and the controller. The first aspect is to deploy forwarding rules before the arrival of packets to avoid frequent interactions between controllers and switches due to switches having no forwarding rules to match. The second aspect is based on the existing forwarding rules. When the network state changes, the forwarding rules need to be adjusted to adapt to the new environment. The third aspect is related to hardware. The switches often have a small cache space for forwarding rules, so it is inevitable to select limited appropriate rules and store them in switches and the rest will be processed by the controller.

Traditional forwarding rule management mainly focuses on the second and third aspects. It looks for better rules under existing forwarding rules or improve the hardware performance.

Machine learning has succeeded in the first aspect. Through machine learning algorithms, the obtained network state information can be converted into the prediction of future network state, so that forwarding rules can be deployed in advance and network delay can be reduced. However, there are limited studies on the adjustment of existing forwarding rules, which need to be researched in the future. The dynamic operation of machine learning is the key to further improve the forwarding rules in the second aspect.

The PDP is different from the ML. The PDP is mainly adopted in the third party. The programmable thought of PDP is very conducive to realize the balance of hardware cost and even decrease the cost, which improves the performance of the switch itself and the interaction with the controller.

In addition, the PDP can obtain more detailed network

state information, which can enhance the accuracy of ML algorithm. In the future, the integration of ML and PDP in the forwarding rules management will be a research direction. How to combine the advantages of the two techniques needs more investigation, for managing forwarding rules.

Scheduling delay-sensitive services with high priority: The essence of scheduling delay-sensitive services with high priority is how to reduce the queuing delay of delay-sensitive services. The AQM is a basic technique of all methods and is the key of scheduling delay-sensitive services with high priority.

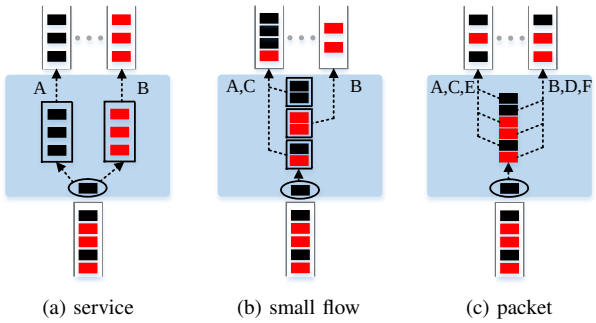
In a traditional framework, the AQM has plenty of mechanisms for different optimization objectives, such as achieving low delay or high throughput, but these mechanisms mainly adjust the queuing itself. ML can help AQM in another aspect, by classifying each service flow according to service states. The AQM schedules the service flow based on the classification. On the other hand, the programmability of PDP can flexibly deploy and realize various AQM algorithms. Also, the PDP can provide accurate network state information for traffic classification. Integrating ML and PDP for AQM is expected to provide a platform for deploying complicated algorithms and to enhance the service differentiation. However, related research is still in its infancy. There will be more studies on AQM based on both ML and PDP, as it deserves further research.

The three delay reduction methods are independent of each other in operation, and their effective integration will lead to a complete structure of comprehensive delay reduction mechanism. As the Internet requires lower and lower network delays, more and more studies on the topic will continue to appear. The studies discussed in this section based on ML and PDP for reducing network delay are summarized in Table I for easy reference.

IV. Network Throughput Improvement by AI-driven Packet Forwarding with PDP

The exponential growth of network traffic increases the throughput demand for various services. This section discusses three flow granularities in traditional PF networks for throughput improvement, and presents how to further improve throughput with ML only and with both ML and PDP, respectively.

With the popularity and development of the Internet, the amount of packet transmitted in the network increases exponentially, and the throughput performance of various services needs to be further improved. An advanced packet forwarding mechanism provides a solution to this requirement. In addition to reducing network delay, the packet forwarding mechanism can maximize throughput through parallel multi-channel scheduling of packets. Specifically, the packet forwarding mechanism distributes a certain number of packets to different network links according to a scheduling algorithm, so as to achieve link bandwidth aggregation and maximize network throughput. According to the number of packets to be scheduled (i.e., the granularity of flow), the packet forwarding algorithm can be classified into three categories: (1) multi-path scheduling with service granularity (all packets of a service flow); (2) multi-path scheduling with small flow granularity (a fraction of packets of a service flow); (3) multi-path scheduling with packet granularity (one packet of a service flow) [66].



(a) service (b) small flow (c) packet

Fig. 5. Three Flow Granularities of Multi-path Forwarding

In multiplex scheduling with service flow granularity, as shown in Fig. 5(a), the packet forwarding mechanism allocates service flows to different network links according to the scheduling algorithm, where all packets in a service flow are forwarded into the same network link. Many algorithms based on service flow granularity are studied. Zhou *et al.* propose a weighted cost multipath algorithm (WCMP) to control the traffic size of each path [67]. The algorithm sets weights for each path according to the network state information, and allocates service flows in different size into different path, to improve the network link utilization. Kheirkhah *et al.* design maximum multiPath TCP (MMPTCP) algorithm to allocate elephant flows (long duration flows) and mouse flows (short duration flows) [39]. The algorithm solves the problem of poor performance of mouse flows in traditional multiPath TCP (MPTCP) by randomly scattering packets in the network. Wang *et al.* and Liu *et al.* use software-defined networking to perceive network topology and link resource information, and adaptively allocate network links to different service flows according to network resource state [68] [69].

The preceding multi-path scheduling algorithms of service flow granularity improve network throughput via matching service flows with different network links. One idea is to allocate the service flows with their unique requirements, such as assigning delay-sensitive service flows to links with a good forwarding environment. Another idea is to allocate the flows based on their sizes. For example, we can distribute an elephant flow on a link with more available resources and distribute a mouse flow on a link with less resources. However, the packet forwarding mechanism based on service flow granularity leads to the different size of service flow in each link, making the utilization of links in an unbalanced state.

In order to avoid uneven utilization of network link resources in the multi-path scheduling with service flow granularity, multi-path scheduling with packet granularity has been studied to distribute packets in a service flow to different network links as shown in Fig. 5(c). All the red rectangles represent packets belonging to the same service flow and are assigned to different network links. Zhang *et al.* propose Hermes, a network balancer, to allocate link resources according

TABLE I
SUMMARY OF PUBLICATIONS ON IMPROVING NETWORK DELAY BASED ON ML AND PDP

Paper	Method of reducing network dealy[1]	Technique based on	Algorithms, Models, ML types	Where ML is used	Main ideas
[47]	1	ML	Convolutional Neural Network, Deep Learning	Social media network	Apply sentence analysis on the data, predict service requirements and cache content in advance.
[48]	1	ML	Bayesian Learning Method, Reinforcement Learning	The wireless network edge	Use Bayesian learning methods to predict user preferences and individual content request probability and cache content in advance.
[34]	1	ML	Distributed Deep Learning	Mobile edge network	Collect user information by content servers and predict users' content requirements while ensuring privacy.
[49]	1	ML&PDP	Deep Learning	Information-centric networking	Obtain the spatio-temporal joint distribution data of network state, predict the popularity of content and cache the service content with high popularity.
[52]	2	ML	K-order Markov Chain, Greedy Algorithm	Mobile access network	Predict the user's future location based on k-order Markov chain, and generate packet forwarding rules by the greedy algorithm.
[53]	2	ML	Deep Reinforcement Learning	Data center network	Formulate the optimization equations of switch cache cost and controller distribution cost, use deep reinforcement learning to obtain the approximate optimal solution.
[36]	2	ML	Auto Regressive Integrated Moving Average, Long Short-Term Memory (LSTM), Reinforcement Learning	Multi-access edge computing networks	Predict SDN load and migrate high load in advance to relieve local network pressure.
[54]	2	PDP	Behavioral-Level Caching Mechanism	-	Unify all cache entries in a switch.
[55]	2	PDP	ShadowFS	-	Use a small cache to manage entries and improve the ability to obtain accurate information with high change frequency.
[56]	2	PDP	-	-	Redesign cache architecture, replaces TCMA to FPGA and accelerate caching speed.
[37]	3	ML	Supervised Learning	Internet of Things	Distinguishe application priorities and creates two queues to provide priority queues.
[60]	3	ML	Deep Learning	Data center network	Distinguish the priorities of service flows and set up multiple priority queues, expand the switch caching space.
[61]	3	PDP	PI2	-	Use the PDP and inlet and outlet pipe information of switch to implement AQM on the data plane.
[62]	3	PDP	CoDel	-	Use PDP to implement an AQM algorithm (CoDel).
[63]	3	PDP	Push-in First-out Algorithm	-	Dynamically adjust the priority of packets based on network state and provide priority queuse.
[64]	3	ML&PDP	eXtreme Gradient Boosting Model	Internet of Things	Establish two-layer queue management and provide differentiated service guarantee mechanisms for applications.
[65]	3	ML&PDP	Hybrid Deep Neural Network	-	Obtain accurate network information, automatically extract characteristics and classify traffic.

[1] 1: caching service content near the user; 2: managing forwarding rules actively; 3: scheduling delay-sensitive services with high priority

to packet granularity [70]. The Hermes can ensure high utilization of network link resources by sensing the network state of available link resources and cautiously adjust the forwarding rules. Dan *et al.* utilize the centralized control characteristic of SDN to obtain the network link state information, and present a multi-channel scheduling mechanism with packet granularity to improve the network resource utilization [38]. Liu *et al.* propose a scheduling method of packet granularity on the basis of PDP, encapsulate different network-layer headers for different packets of one service flow, and forward them through different network links [71] [72]. The method not only improves the secure transmission capability of packets, but also achieves network link bandwidth aggregation. The preceding packet granularity multiplexing algorithms improve network throughput by fitting different network links with different packets. However, due to unique characteristics of different network links such as in terms of delay, the packet granularity multiplexing algorithms suffer from packets out of order [73].

In order to overcome the problems of uneven utilization of network link resources with the service flow granularity and packets out of order with packet granularity scheduling, a compromise is made with small-flow granularity scheduling which breaks up the service flows into a number of small streams and distributes the streams to different network links, as shown in Fig. 5(b). Vanini *et al.* design a small-flow switching multiplexing mechanism, dynamic packet scheduling and adjusting with feedback (DPSAF), which enables service flows to be flexibly "cut" and distributed to network links [74]. The mechanism ensures that packets arrive in order and improves the utilization of network links. Shi *et al.* develop a multi-path scheduling mechanism based on network state information, which allocates network link resources with small flow granularity and dynamically adjusts the number of packets in a small stream according to network link state [40]. Katta *et al.* use PDP for network probes to perceive resource utilization of network links [75]. They "slice" the service flows into streams according to the gap between adjacent packets arriving at a switch, and distribute the streams to each network link according to the perceived utilization.

ML for throughput: All the three flow granularities of packet forwarding face one problem: how to select an appropriate network link for the flow/packet/stream. ML can offer some advantages in this link resource allocation to service flows. First, we can use ML for service classification similar to that in scheduling delay-sensitive services with high priority in Section III, to provide different link resources for services with different priorities; second, we can apply ML to predict the future state of network links based on the existing network state information and to select an optimal packet transmission path to maximize throughput.

Pasca *et al.* apply machine learning techniques to SDN [76]. They establish a decision tree classifier through supervised learning, which can classify and assign priority to service flows according to their protocol and source/destination address. As the switch processes the packets, the switch allocates different flows to different paths based on the priority. This mechanism improves performance of high-priority services and maximizes network throughput. Ji *et al.* propose an MPTCP-based automatic learning selection path mechanism (ALPS-MPTCP) [77]. This mechanism automatically obtains the network link states and adaptively selects some high-quality paths to transmit data packets. Li *et al.* design a multipath packet forwarding congestion control mechanism based on ML to solve the low throughput problem caused by heterogeneous links in multipath forwarding [11]. They use reinforcement learning to analyze network congestion and dynamically adjust the congestion window to avoid congestion, so that aggregate throughput can be improved. It is worth noting that the training of a reinforcement learning model is an off-line process, does not influence the decision making process, does not introduce extra delay and overhead. Azzouni *et al.* propose NeuRoute, an ML algorithm based dynamic routing framework [78]. The framework uses deep learning to learn service flow characteristics and predict network flow changes. The new forwarding rules can be generated and distributed to each switch, which ultimately increase network throughput. Gilad *et al.* present a high-performance multipath congestion control architecture [79]. This structure uses on-line convex optimization to solve the balance problem of fairness and high performance. Kanagarathinam *et al.* propose intelligent multipath switch (SMS) for MPTCP in a wireless network [80]. The SMS can use machine learning to dynamically adjust and manage MPTCP substreams based on the state of the wireless network to improve network throughput. Mohammed *et al.* develope a deep reinforcement learning algorithm [81]. By analyzing the state of the wide area network (WAN), the algorithm adjusts the transmission path of the service flow, and reconfigures the traffic of each link, which improves the utilization and throughput of the network link.

ML and PDP for throughput: The flexible programmable characteristic of the PDP can provide more detailed network state information for ML in parallel multipath scheduling of packet forwarding [82]. With the help of PDP, the network throughput performance can be enhanced. Basat *et al.* design an in-band telemetry mechanism of low overhead based on PDP to monitor the hop latency, queue depth, and link utilization of all paths in parallel multi-forwarding of packets [83]. Based on the perceived refined network state information, they propose a precise parallel multiplexing mechanism which can improve the network throughput. Liu *et al.* utilize PDP to measure fine-grained network state information and adopt a deep Q-learning model to make decisions on packet forwarding path selection [84]. The model can reduce the out-of-order packets rate in parallel multi-channel transmission, which improves the network throughput. Hardegen *et al.* implement feature prediction for network flow based on ML and collect a variety of heterogeneous network link state information based on PDP [85]. They select each packet forwarding path according to the feature prediction and network link states to maximize the network throughput. Li *et al.* propose a mechanism to classify application flows with different QoS requirements based on C4.5 decision tree and adjust SDN switch packet queue depth [86]. The mechanism provides different transmission parameters for application flows with different priorities, which can reduce delay and increase throughput for the application flows

with the highest priority. Liu designs a deep reinforcement learning based routing (DRL-R) algorithm [87]. The algorithm transforms the performance requirements of service flow into the resource requirements. Then, according to the requirements, the algorithm classifies service flows and allocates corresponding resources. In addition, the algorithm pays attention to network states regularly, adjusts forwarding rules adaptively, and optimizes network resource allocation. Hu *et al.* propose EARS, an intelligence-driven experiential network architecture for automatic routing [88]. The EARS uses deep reinforcement learning to control switches and modify forward policies by interacting with the network environment to improve network throughput.

Summary and Remarks: In the traditional Internet multi-path forwarding, forwarding algorithms with three flow granularities have their own advantages and drawbacks. Algorithms based on service flow can effectively transmit the same service flow over a unique link, but can lead to uneven utilization of link bandwidth. Although algorithms based on packet granularity can improve link bandwidth utilization, they can lead to disordered packets. The forwarding algorithms based on small flow is between the two. It can achieve an appropriate utilization of bandwidth while decreasing out-of-order packets. It achieves satisfactory performance in all aspects, but not outstanding. On the other hand, forwarding based on small flow is a static balance. When the network fluctuates, the balance is broken, and the throughput performance deteriorates significantly.

The introduction of ML and PDP does not change the three granularities methods, but solves the problem of choosing an appropriate forwarding path. ML dynamically selects network links through various models to achieve reasonable resource allocation. In the small flow granularity, ML can change the static balance into dynamic balance, adapt to various network fluctuations within a certain range, and stably maintain a high performance of network throughput for a long time. The PDP is a technique to facilitate ML. It gathers the current network state more accurately and provides them to the ML model. The research on ML and PDP in path selection is at an early stage and most are based on flow granularity. In the future, studies on the other two granularities should be carried out, integrating ML and PDP closely with traditional solutions, to enhance throughput improvement.

The research works on improving network throughput based on ML and PDP discussed in this section are summarized in Table II for easy reference.

V. Security Improvement by AI-driven Packet Forwarding with PDP

Ensuring data security of users and forwarding devices is gaining increasing importance, which puts forward a higher requirement on the network security. This section summarizes recent works on how PDP can improve AI-driven packet forwarding in terms of data security. That is, AI-driven packet forwarding with PDP can defense three typical attacks (i.e., distributed denial of service, ransomware, and abnormal traffic) with high accuracy and low latency, and mitigate the attacks with high efficiency.

A. Distributed Denial of Service

Distributed Denial of Service (DDoS) attack is that an attacker controls multiple systems and constantly sends malicious traffic to the victim to suppress the server, host or application, which makes the computing or networking system overloaded. In the defense of traditional DDoS attack, a switch identifies the field of forwarding packet, and filters out the DDoS attack packets by extracting feature information. However, due to the limited performance of a traditional switch, the defense method cannot reach a high speed and high efficiency.

ML for DDoS: The dynamic feature extraction of ML can effectively solve the previous issues. ML can help deal with complex network environment, and accurately and automatically detect DDoS attacks. Doshi *et al.* propose a mechanism that uses IoT network specific behavior, such as a limited number of IoT nodes, regular forwarding time interval between adjacent packets, to conduct DDoS attack feature analysis [12]. In particular, they first capture network traffic, sorting packets based on information such as address and time. Then, they extract features from the packets and classify them into normal traffic and DDoS attack traffic. The mechanism can accurately detect DDoS attacks. Idhammad *et al.* design a DDoS detection mechanism based on semi-supervised learning to solve the problems of low accuracy and high false identification rate in typical DDoS detection [32]. The detection mechanism consists of two main parts: unsupervised and supervised. Unsupervised learning estimates the entropy of network traffic features based on a time sliding window algorithm, and then calculates the information ratio of gains. Network traffic with high information ratio of gains is considered abnormal. Supervised learning uses an ML classifier based on an extra-tree algorithm to accurately classify abnormal traffic and reduce the false positive rate of unsupervised learning. The detection failure rate of the mechanism is low. Yuan *et al.* propose a DDoS attack detection method, DeepDefense, based on deep learning [89]. It uses a bi-directional recurrent neural network and data sets to learn patterns from network traffic sequences and track network attack activities. The Deepdefense is better than shallow machine learning method in recognition error rate and generalization. Prez-Daz *et al.* design an SDN-based security architecture consisting of two independent systems, an intrusion prevention system (IPS) and an intrusion detection system (IDS) [90]. The IDS is deployed at the host, and is trained by machine learning for network traffic identification. If a flow is recognized as an attack, the flow will be processed by the IPS module in the controller to defend against the DDoS attackers. The proposed architecture is practical for identifying and mitigating DDoS attacks. Niyaz *et al.* propose a multi-vector DDoS detection system based on deep learning [91]. The system monitors network traffic, analyzes and evaluates traffic tracks in different scenarios, and determines whether DDoS attacks exist.

In addition, there are many ML algorithm models, and how to choose an appropriate model for DDoS attack defense has been studied. In order to solve the problem of new DDoS attacks in an SDN environment, Santos *et al.* realized three

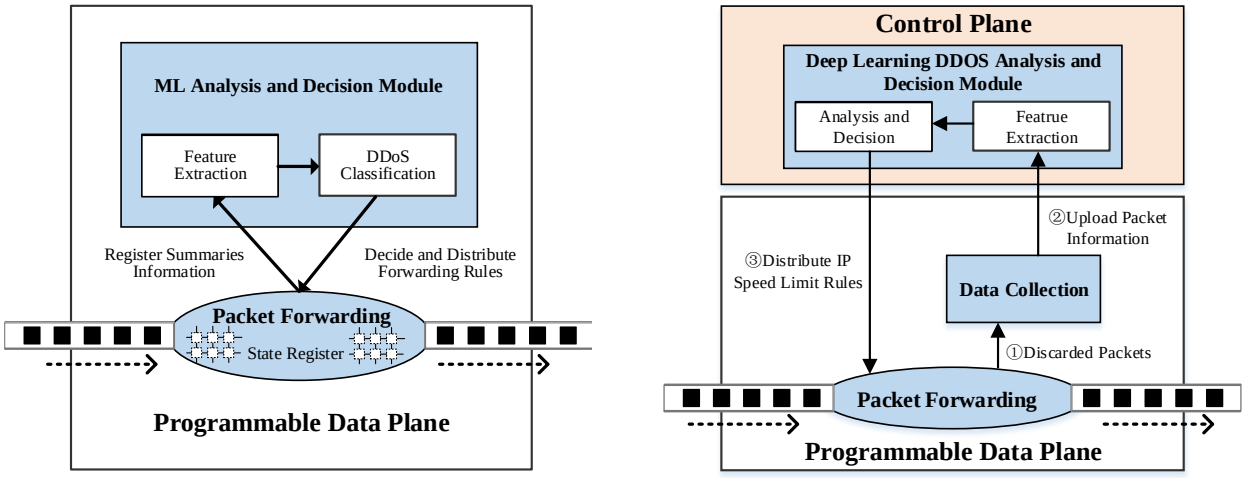
TABLE II
SUMMARY OF PUBLICATIONS ON IMPROVING NETWORK THROUGHPUT BASED ON ML AND PDP

Paper	Technique based on	Algorithms, Models, ML types	Where ML is used	Main ideas
[76]	ML	Supervised Learning	-	Classify and prioritize service flows, assign different flows to different paths based on priority.
[77]	ML	K-NN Algorithm, Random Forest, K-Means Algorithm, Reinforcement Learning	Internet of Things	Automatically obtain network link states and adaptively select high-quality paths to transmit packets.
[11]	ML	Reinforcement Learning, Hierarchical Tile Coding Algorithm	Heterogeneous network	Dynamically adjust the congestion window size to reduce congestion and improve aggregation throughput.
[78]	ML	Deep Learning	-	Predict network traffic changes and generate new forwarding rules to improve network throughput.
[79]	ML	Online Convex Optimization	-	Propose multipath fairness and switch performance optimization equations and solve by online convex optimization.
[80]	ML	Smart Multipath Switch	Wireless network	Dynamically adjust and manage MPTCP substreams according to the state of the wireless network.
[81]	ML	Deep Reinforcement Learning, Upper-Confidence Algorithm	Wide Area Network	Analyze the wide area network states, adjust the transmission path of service flows and reconfigure the traffic on each link.
[83]	ML&PDP	In-band Network Telemetry	-	Obtain information such as delay, queue depth, and link utilization to achieve fine multipathing control.
[84]	ML&PDP	Distributed Asynchrono Deep Reinforcement Learning	-	Analyze the network states and determine the forwarding path, reduce the packet out-of-order rate.
[85]	ML&PDP	Deep Neural Network	-	Predict network traffic and select an appropriate path to forward.
[86]	ML&PDP	C4.5 Decision tree, Low Latency Queueing	-	Classify applications and configure priorities, adjust queueing depth of SDN switches to forward packets of applications with different priorities.
[87]	ML&PDP	Deep Q-Network, Deep Deterministic Policy Gradient	-	Transform performance requirements of the service flow into the resource requirements, allocate network resources for flows and adaptively adjusts the forwarding rules.
[88]	ML&PDP	Deep Learning aAlgorithm	-	Interactive network environment periodically and modified forwarding rules dynamically.

different types of DDoS attack detection (flow table attack, bandwidth attack and controller attack) using four machine learning methods: support vector machines (SVM), multilayer perceptron (MLP), decision tree and random forest in an SDN simulation environment [92]. The results show that the decision tree approach has the shortest processing time, and the random forest approach has the highest absolute value accuracy.

PDP for DDoS: PDP has the characteristics of flexible programming, which can quickly collect attack flow information and update switch forwarding rules in real time. Such real-time updates can enhance defense capability. Bunia *et al.* establish an SDN-based framework called SoftThings, where SDN switches continuously monitor the traffic of IoT devices and provide the traffic information to the cluster SDN controller [93]. The controller detects abnormal behaviors by analyzing traffic, and dynamically sets traffic rules for switches to defend against DDoS attacks. In this framework, abnormal traffic can be quickly detected at the network edge. Shang *et al.* propose FloodDefender, a network defense framework, to address the problem that communication links between two planes in the SDN architecture are vulnerable to DDoS attacks [94]. The framework monitors network states in real time. When an attack occurs, the neighbor switch takes over the work of the victim switch and sends the received data stream to the interceptor of the controller to identify whether it is normal traffic. In addition, the controller updates the new attack stream characteristics into the interceptor in real time to improve the efficiency of interception.

ML and PDP for DDoS: The integration of PDP and ML is the development direction of DDoS attack defense, which is mainly studied in three aspects. Firstly, the PDP provides multiple planes to better deploy ML models, which can realize a DDoS monitoring algorithm on line speed and greatly reduce DDoS attack monitoring latency [95] [96]. Secondly, fine-grained network state information can be supplied by the PDP, including queue length, network delay and so on, to provide more multidimensional features for



(a) Detection Framework Based On Forwared Packets

(b) Detection Framework Based On Discarded Packets

Fig. 6. Two Mechanisms about Defending DDoS Attacks with ML and PDP

accurately monitoring DDoS attacks [97] [98]. Thirdly, the flexible and programmable characteristics of the PDP provide the foundation for customized DDoS attack monitoring [99]. The DDoS attacks monitoring algorithm can be dynamically selected according to requirements of security level. Therefore, aggregating ML and PDP for more accurate monitoring of DDoS attacks has become a research hotspot in the community.

Musumeci *et al.* use ML and PDP to detect DDoS attacks during packet forwarding in real time [100]. Their mechanism workflow is shown in Fig. 6(a). In the packet forwarding process with PDP, the numbers of IP, UDP, TCP and SYN packets are counted, and the local ML analysis and decision module can use these numbers to judge whether there is a DDoS attack. If there is, the decision forwarding engine distributes the forwarding rules containing the attacker's IP and other information in the form of table entry, and blocks the attacks. Mi *et al.* design an ML-Pushback mechanism with deep learning and PDP to accurately identify and mitigate DDoS attacks [101]. The workflow is shown in Fig. 6(b). The data collector is customized with PDP, and gathers real-time discarded packets in the network. When the number of discarded packets reaches a threshold, the data collector automatically extract packet information and uploads it to the control plane. When the deep learning module of the control plane receives the information, the feature extraction sub-module will extract the IP source/destination address, protocol type, interface number and other features, and then hand them to the decision tree of the analysis and decision module to judge whether there is a DDoS attack. If a DDoS attack exists, the features of DDoS attacker are further extracted. Then, the analysis and decision module distributes the forwarding rules to switches to limit the traffic from the attackers.

Hu *et al.* propose a prototype system to defend DDoS attack in real time [102]. During the process of packet forwarding, they adopt flow-based information collection methods in SDN switches to quickly gather network state information, and use the support vector machine (SVM) model for data analysis to detect and judge DDoS attacks. At the same time, they design a DDoS attack mitigation mechanism based on the white list, which can dynamic update the packet forwarding rules to block the service packets that are not in the white list and effectively reduce the damage of DDoS attacks. Chen *et al.* propose a distributed intrusion prevention system, CIPA, for programmable networks and SDNs, based on artificial neural network (ANN) [103]. The CIPA distributes computing power to some or all of the switches in the network, monitors the network and forms a global view of the network states, and performs high-precision intrusion detection and mitigation on discovered malicious online traffic. Phan *et al.* design a hybrid stream working mechanism based on SVM and self organizing map (SOM) [104]. The mechanism uses SDN to collect switch flow information and classify the flow by SVM. Then, SOM is used to make the final decision, and the switch transmission rules are changed to reduce network attacks. Chen *et al.* use extreme gradient boosting (XGBoost) as a detection method for cloud-based SDN networks [105]. The XGBoost classifier detects DDoS attacks on packets collected by TcpDump. The method has high detection accuracy, low false positive rate, fast detection speed and scalability.

B. Ransomware

Ransomware is a kind of virus software that extorts the files of victims. The attacker holds the file until the victim pays a lot of money to redeem it. Ransomware comes in many forms, and the traditional defense methods do not provide comprehensive protection against ransomware.

ML for ransomware: ML can help defend ransomware attacks by quickly detecting and filtering the attack packets at the interfaces of switches. Poudyal *et al.* analyze the raw data, library files, number of functional interface calls and other multi-level data of the user's computer, and use Bayesian Network, Random Forest and other supervised learning algorithms to detect ransomware [106]. Zhang *et al.* first analyze the opcodes of the user terminal and generate an N-gram sequence, then extract the features from the sequence with the TF-IDF data mining algorithm, and finally construct a ransomware detection model with the extracted features [107].

TABLE III
SUMMARY OF PUBLICATIONS ON DEFENDING DDOS ATTACK BASED ON ML AND PDP

Paper	Technique based on	Security type	Algorithms, Models, ML types	Main idea
[12]	ML	DDoS	Random Forests, SVM, K-nearest Neighbors, Decision Trees, Neural Networks	Classify packets, extract traffic characteristics, and classify them into normal traffic and attack traffic.
[32]	ML	DDoS	Semi-supervised Learning, Time Sliding Window Algorithm, Extra-trees Algorithm	Unsupervised learning extracts the characteristic entropy of data stream and calculates the information gain ratio. Supervised learning distinguishes abnormal flow and reduces the false positive rate of unsupervised learning.
[89]	ML	DDoS	Bi-directional Recurrent Neural Network	Use bi-directional recurrent neural network and data sets to learn DDoS attack patterns and track attack activity.
[90]	ML	DDoS	Intrusion Prevention System, Intrusion Detection System	Intrusion prevention system uses machine learning on the host to identify DDoS attacks. DDoS attacks are sent to the intrusion petection system of the controller for processing and interception.
[91]	ML	DDoS	Deep Learning	Monitor network traffic, analyze and evaluates traffic tracks in different scenarios, and determine whether DDoS attacks exist.
[92]	ML	DDoS	Support Vector Machines, Multiple Layer Perceptron, Decision Tree, Random Forest	Test the effectiveness of different machine learning algorithms against DDoS attacks.
[93]	PDP	DDoS	-	Collect traffic characteristics, provide them to the controller, analyze and distinguish data flows, and adjust forwarding rules dynamically.
[94]	PDP	DDoS	-	The neighbor switch performs the work of the victim switch. The controller identifies and analyzes the data flow, and updates the interception rules in real time.
[100]	ML&PDP	DDoS	Random Forest, Knearest Neighbors, Support Vector Machine	Collect the number of IP, UDP, TCP and SYN packets, analyze the data and identify DDoS attacks by local ML, update the flow rules to intercept DDoS packets.
[101]	ML&PDP	DDoS	Deep Learning, Decision Trees	Collector collects the packets discarded by the switch and sends the packets to the controller for feature extraction. The controller identifies DDoS attacks and updates the forwarding rules in the switch to limit the rate of DDoS attacks.
[102]	ML&PDP	DDoS	sFlow-based Method, Supervised Learning, Support Vector Machines	Use sflow-based information collection to quickly collect network state information, use support vector machines to analyze and identify DDoS attacks, update forwarding rules in time, and minimize looses caused by DDoS attacks.
[103]	ML&PDP	DDoS	Artificial Neural Network	Assign computing tasks to multiple switches, collect data from each switch, and form network state monitoring to detect DDoS attack flows with high accuracy.
[104]	ML&PDP	DDoS	Support Vector Machine, Self Organizing Map	SDN collects switch information. Support vector machine classifies data flows. Self organizing map determines whether data flows are attack flows and modifies switch forwarding rules to reduce attacks.
[105]	ML&PDP	DDoS	Extreme Gradient Boosting	Collect switch inbound packets, detect and identify DDoS attacks.

Lee *et al.* make comprehensive use of information entropy and ML methods to classify ransomware [31]. Their classification scheme accurately identifies ransomware. Omar *et al.* design NetConverse to identify ransomware [108]. Specifically, they analyze network flow, extract its characteristics, and compare them with the characteristics of the learned ransomware to determine whether it is a ransomware flow. Shaukat *et al.* propose RansomWall, a layered defense system for defending against crypto-ransomware [109]. RansomWall combines static and dynamic analysis to extract file features and send them to the feature collector. The feature collector sends suspicious file to the machine learning model to make the final judgment on whether it is ransomware.

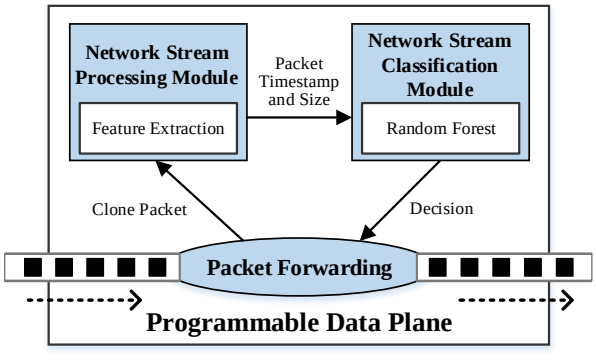


Fig. 7. Mechanism on Defending Ransomware Attack with ML and PDP

ML and PDP for ransomware: Different from the infor-

mation provided for DDoS attack defense, the PDP provides the real-time and accurate network endogenous state information to ML for a better ransomware detection. Cusack *et al.* propose a ransomware precise detection mechanism based on ML and PDP [110]. As shown in Fig. 7, the mechanism mainly includes two modules: network stream processing module and network stream classification module. First, the network stream processing module uses the PDP to quickly collect the information of each packet, and extracts the network stream features (packet 5-tuple information), including packet timestamp, packet size and byte number. Then, the network stream classification module gathers the feature information to identify the ransomware stream by the random forest algorithm. By adjusting the number (40) and depth (15) and the maximum amount of features in random forest decision trees, the accuracy of detection ransomware reaches 86%, and the failure rate of detection is only 11%. Combining the advantages of IDS and SDN, Boero *et al.* use the SVM algorithm as the core of the system to detect ransomware on the SDN controller side with the characteristics of byte rate, packet rate and average packet length [111]. Experimental results show that the proposed scheme has a high detection rate.

C. Automating Network Traffic Detection

Automating Network Traffic Detection (ANTD) mainly addresses the attacks caused by abnormal network traffic. The ANTD detects anomalies and security attacks by analyzing network traffic. Especially in the data center, a large amount of network traffic needs to be monitored for a safe network environment. However, the ANTD technique has low performance in a traditional network and brings huge hidden danger for network security.

ML for ANTD: ML provides a powerful tool for the ANTD. It can construct an abnormal network traffic model and accurately detect abnormal traffic, which achieves a high security level. Salman *et al.* establish an abnormal traffic detection framework for IoT network devices based on ML [30]. In this framework, network traffic information is collected from IoT network devices. Then, the information is extracted into 39 network traffic features for further classification. The ML algorithm keeps running to identify abnormal network traffic. The mechanism can detect abnormal traffic with a high accuracy. Ji *et al.* utilize ML to model a deep attack pattern of network abnormal traffic [112]. Specifically, they design a mechanism based on ML and SVM algorithm to analyze abnormal network traffic and predict the categories of network attacks. Niu *et al.* propose a network intrusion detection method based on Transfer Component Analysis (TCA) [113]. The method uses TCA to map the traffic features of the source domain and the traffic features of the target task to a shared subspace for domain adaptation, and then uses K-Nearest Neighbors (KNN), SVMs and Random Forests (RF) as base classifiers for training detection models to find abnormal traffic. Kong *et al.* propose an abnormal traffic identification system (ATIS) based on SVM [114]. The system determines abnormal traffic in four steps: data collection, traffic features extraction, data processing, and SVM classification. The system can classify and identify various attack traffic applications. Kong *et al.* design an anomaly encrypted traffic detection method based on machine learning and behavior characteristics [115]. This method extracts behavior characteristics of application programs and classifies abnormal traffic based on machine learning. This method can effectively improve the accuracy of abnormal encrypted traffic detection. Yong *et al.* design a parallel cross convolutional neural network (PCCN) for network traffic detection [116]. The PCCN fuses two branches of convolutional neural network, with excellent learning ability. It can complete the learning of traffic characteristics in the case of a small number of samples. In addition, it adopts an improved original flow feature extraction method and achieves fast convergence speed of network traffic classification and identification, which lead to a quick detection.

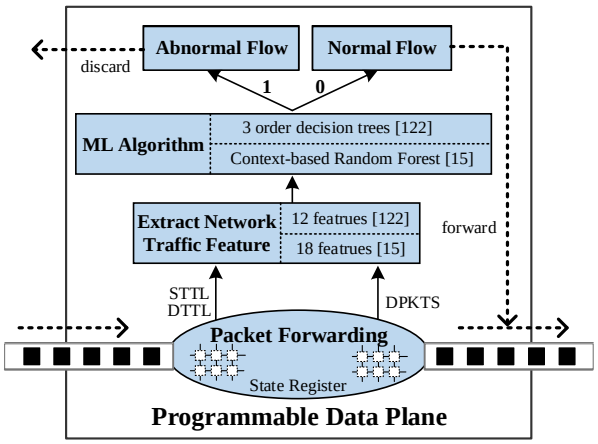


Fig. 8. Two Mechanisms about ANTD with ML and PDP

PDP and ML for ANTD: The high processing performance of the PDP facilitates the line-speed analysis and detection of network abnormal traffic. The PDP can be used to design a fine-grained traffic collection and analysis mechanism in the packet level to ensure the realtime and accuracy of abnormal traffic analysis [117] [118] [119] [120]. Using PDP to monitor network traffic can avoid sending too many network packets to the centralized control plane, and can reduce abnormal network traffic analysis decision delay [121]. Therefore, integrating ML and PDP to improve ANTD has become a hot topic in the security field.

Lee *et al.* propose an endogenous abnormal traffic analysis mechanism based on PDP, and deploy an ML random forest algorithm on switches to realize real-time online abnormal traffic detection in networks [122]. The specific workflow is shown in Fig. 8. First, 12 network traffic features are selected as the judgment basis, e.g., the TTL from source to destination terminal (STTL), the TTL from destination to source terminal (DTTL) and the number of packets from destination to source terminal (DPKTs). Then, the 3-order random forest decision tree algorithm is used to carry out abnormal analysis of network traffic. If abnormal network traffic is confirmed, the corresponding packet forwarding rules will be distributed to the switches and the abnormal traffic packets are discarded; otherwise, the packets will be forwarded following the default

TABLE IV
SUMMARY OF PUBLICATIONS ON DEFENDING RANSOMWARE BASED ON ML AND PDP

Paper	Technique based on	Security type	Algorithms, Models, ML types	Main idea
[106]	ML	Ransomware	Bayesian Network, Random Forest, Supervised Learning	Analyze the original data, library file, number of function interface call of user, and detect ransomware by supervised learning algorithm.
[107]	ML	Ransomware	Term Frequency-Inverse Document Frequency Data Mining Algorithm	Analyze the operation code of user and generat N-gram, extract sequence features by term frequency-inverse document frequency data mining algorithm, construct ransomware detection model.
[31]	ML	Ransomware	Information Entropy, Decision Tree, Deep Learning	Classify ransomware according to different file formats based on information entropy and ML methods.
[108]	ML	Ransomware	Decision Tree	Analyze network flow, extract features, compare with ransomware feature database, determine whether it is ransomware.
[109]	ML	Ransomware	Gradient Tree Boosting Algorithm	Extract file features by static and dynamic analysis, send features to collectors. Collector send suspicious files to machine learning engines to identify ransomware.
[110]	ML&PDP	Ransomware	Random Forest, Decision Trees	PDP quickly collects data flow information, extracts features. ML collects features and determines whether it is ransomware.
[111]	ML&PDP	Ransomware	Support Vector Machines	Collect byte rate, packet rate and average packet length, and detect ransomware by support vector machines.

flow table rules. Busse-Grawitz *et al.* deploy a supervised learning model on the top plane of PDP to detect the endogenous traffic in the network [15]. First, due to the memory and floating operation constraints in the PDP, they modify the feature selection process in the supervised learning to adapt the information from the PDP. Then, they design a real-time accurate network traffic analysis mechanism based on PDP and ML. In the model training stage of the mechanism, the labeled network traffic data is used to train the classification model of supervised learning, and then the classification model is deployed on the PDP to realize the line speed online classifier. The workflow is shown in Fig. 8. First, 18 network traffic features are extracted by counting the first three packets of a network traffic as the judgment basis. Then, the random forest algorithm with semantic association is used to conduct abnormal network traffic analysis. If abnormal network traffic exists, the corresponding traffic will be discarded according to the packet forwarding rules; otherwise, the packets will be forwarded following the default flow table rules.

Song *et al.* propose an abnormal traffic defense mechanism based on SDN and machine learning [123]. The mechanism first extracts the characteristics of network traffic, then evaluates the malicious degree using a random forest algorithm, and finally makes defense decisions to effectively prevent abnormal traffic transmission. Georgi *et al.* design a lightweight traffic detection and defense system [124]. The system periodically collects traffic statistic information from the SDN OpenFlow switch. Then, it extracts and aggregates a set of features to analyze traffic information, so as to achieve high performance abnormal flow detection. Tuan *et al.* use deep learning to detect abnormal traffic in an SDN environment [125]. They use the NSL-KDD data set to build a deep neural network anomaly flow detection model. Then the model is used to analyze network traffic and identify abnormal traffic. Experiments show that the model has great potential in abnormal flow detection.

D. Summary and Remarks

The three typical network attacks discussed in this section have their own unique characteristics and represent three aspects of network security. The DDoS defense deals with normal packets in normal stream, while ransomware studies the abnormal packets in normal stream and ANTD finds the abnormal stream. In addition, ransomware happens at the users' side and the ANTD is opposite. It takes place in the core network and exists in the traffic entering the forwarding devices. In this section, we make a brief summary of how to deal with the three network attacks.

DDoS: DDoS attack defense focuses on the features of attack packets. How to identify DDoS attack packets more efficiently and accurately is the research direction. ML can learn the features of known DDoS attack packets and then identify and filter other similar ones. While providing more accurate packet information, PDP can support an ML model to run on the data plane, avoiding the process of data upload and distribution, and improving the real-time interception. Existing solutions have the capability for successfully filtering DDoS attacks with high accuracy. However, identifying and filtering functions are in general deployed in the switch near the victim, which is passive and only triggered at the arrival of DDoS attack packets. Is it possible to intercept DDoS attacks at the edge of the network when the packets enter the network? If so, it will not only successfully defend against DDoS attacks, but also reduce a large number of junk packets in the network.

Ransomware: Ransomware is different from DDoS attacks. DDoS attacks use normal packets to bombard the victim.

TABLE V
SUMMARY OF PUBLICATIONS ON AUTOMATING NETWORK TRAFFIC DETECTION BASED ON ML AND PDP

Paper	Technique based on	Security type	Algorithms, Models, ML types	Main idea
[30]	ML	ANTD[1]	Convolutional Neural Network, Residual Neural Network, Recurrent Neural Network	Collect traffic information of the IoT device, extract characteristics, and use machine learning algorithms to classify and identify abnormal traffic.
[112]	ML	ANTD	Support Vector Machines	Use support vector machines algorithm to analyze network abnormal flow and predict the mechanism of network attack category.
[113]	ML	ANTD	Transfer Component Analysis, K-Nearest Neighbors, Support Vector Machine, Random Forests	Extract traffic characteristics of source and destination domain, carry out domain self-adaptation, classify and determine the abnormal traffic.
[114]	ML	ANTD	Support Vector Machine	Collect traffic information, extract traffic features, use support vector machine to determine abnormal traffic.
[115]	ML	ANTD	Intrusion Detection Systems, Decision Trees	Extract behavior characteristics of applications, use machine learning to classify and determine abnormal traffic, improve the accuracy of abnormal flow detection.
[116]	ML	ANTD	Convolutional Neural Network	Fuse two branch convolutional neural networks, adopt improved method for extracting original stream features, detect abnormal flow fast and efficiently.
[122]	ML&PDP	ANTD	Supervised Learning, Random Forest, Decision Tree	Collect 12 network traffic characteristics, use decision tree to analyze whether the network traffic is abnormal.
[15]	ML&PDP	ANTD	Supervised learning, Random Forest	PDP rapidly extracts 18 traffic features. Supervised learning realizes abnormal traffic identification.
[123]	ML&PDP	ANTD	Random Forest	Extract the characteristics of traffic, evaluate the malicious degree, and make decisions to prevent malicious flow transmission.
[124]	ML&PDP	ANTD	Bagged Trees	Periodically collect traffic statistics, extract and aggregate features to analyze traffic information, and implement high-performance abnormal traffic detection.
[125]	ML&PDP	ANTD	Deep Learning	Establish anomaly flow detection model based on NSL-KDD data set, detect the abnormal flow.

[1] 1. Automating Network Traffic Detection

Ransomware, however, sends "masquerading packets" which are packaged into ordinary packets for control and extortion. Therefore, the defense methods for them are different. DDoS mainly intercepts attack packets based on similarity, while ransomware focuses on identifying attack packets based on packet content. The ML for ransomware focuses more on obtaining information from users' terminal to help identify ransomware packets and intercept them. PDP, in terms of information precision, provides real-time and accurate packet information to assist ML operation. The blocking rate of each study is generally less than 90% which is not sufficiently high as desired. The reason is that ransomware can be packaged in various types of packets, which is very difficult to identify. In the future Internet, there can be more protocol packet format, which will lead to higher difficulties for ransomware identification. Finding them and extracting common features of ransomware packets correctly and quickly is the future research direction. How to integrate ML and PDP to maximize ransomware defense capabilities requires further studies.

ANTD: DDoS and ransomware defense is against the attack wrapped in a normal traffic, while ANTD is against abnormal one. ANTD requires a large number of packet characteristics as the basis for classification, and the ML model is more complex because of the diversity of the types of abnormal flow. Controllers deploying ANTD tend to require more resources to support the ML model. Therefore, how to optimize ML model and reduce the consumption of hardware resources while ensuring accuracy is a future research direction of ANTD.

The papers related to network security based on ML and PDP discussed in this section are summarized in Table III, Table IV and Table V respectively for easy reference.

VI. RELIABILITY IMPROVEMENT BY AI-DRIVEN PACKET FORWARDING WITH PDP

Reliability refers to the quality of data transmission and is an important issue of the Internet. This section offers an in-depth view of state-of-the-art reliable packet forwarding mechanisms that utilize AI and PDP technologies. First, research works on PDP-based network telemetry are discussed in detail. Then, congestion control and runtime verification are studied respectively.

A. Network Telemetry

Network telemetry refers to how information from various data sources is collected by using a set of automated communication processes and transmitted to the corresponding equipment for analysis tasks. The flexible programmable characteristic of the PDP and its customized header in packets can

help make the network telemetry accurate, convenient, and cost effective.

Basat *et al.* propose a network state telemetry mechanism based on PDP to collect information such as interface queue length, packet forwarding processing delay, packet forwarding interface [83]. The mechanism needs to insert only one bit telemetry information when packets are forwarded. The network state telemetry information can be partitioned and restored in multiple packets, which greatly reduces the overhead of telemetry. Zhou *et al.* design a real-time traffic monitor [126]. The monitor closely deploys multiple information collection modules in programmable switches and periodically collects switch information to obtain the states of the network. Holterbach *et al.* use the PDP to quickly detect network failures, and send rerouting signals to the control plane to ensure rapid recovery of services [127].

Although the PDP can obtain plenty of network state information, it cannot actively detect network state according to network needs because of its passive acquisition mode. The integration of ML and PDP can solve the problem, which realizes automatic and intelligent network telemetry.

Hyun *et al.* use an in-band telemetry mechanism of the PDP to obtain accurate network state information [128]. Based on the information, they use ML to build a knowledge-defined network whose system framework is shown in Fig. 9. The system is mainly divided into four planes, i.e., programmable data plane, management plane, control plane and knowledge plane. In the programmable data plane, the network state information is collected and summarized in the process of packet forwarding and uploaded to the management plane and the control plane. The management plane acquires the network state information and then carries on data statistics, analysis and processing, and finally constructs the network state database. The control plane gathers the short-term reports from the management plane and the information from the programmable data plane, and then show the network configuration and state for visualization. The knowledge plane acquires the long-term data of network state from the management plane, extracts the features of the network state and trains the ML model to form the knowledge of network traffic scheduling and abnormal detection. Then, the packet forwarding strategies are generated through the knowledge and deployed to the control plane. The control plane translates the strategies into packet forwarding rules and distributes the rules to the programmable data plane.

Lazaris *et al.* propose a DeepFlow framework to enable fine-grained measurements in programmable switches [129]. The framework can adaptively measure the activity of service flow and provide different fine-grained network state information for different activity levels. In addition, the framework uses historical measurements to train a cloud-based deep learning model to provide short-term traffic predictions when switch resources are limited. Pashamokhtari establishes a security monitoring mechanism for the Internet of Things [130]. The PDP is used to dynamically monitor the packet information of various devices in the Internet of Things. A set of machine learning models are used to classify and detect the information to find whether there is malicious behavior. Janakaraj *et al.*

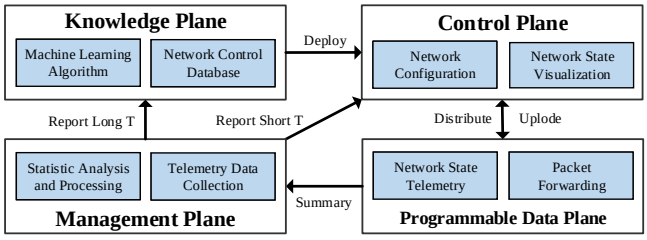


Fig. 9. A Framework of Network Telemetry with ML and PDP

design a distributed in-band network telemetry system (S-INT) and a wireless network operating system (WINOS) [131]. The S-INT reduces the overhead of monitoring network traffic by embedding a specialized INT header in the data stream. At the same time, the WINOS system can summarize distributed telemetry information and connect with an SDN network to realize fast machine learning algorithm and network control.

The research of network telemetry not only includes the integration of ML and PDP to improve network performance, but also includes the improvement of network telemetry technique, in-band network telemetry (INT). Hohemberger *et al.* formulate an INT's assignment plan model and enhance it with machine learning algorithms [132]. The enhanced model can improve the ability of INT to identify abnormal network states. Yang *et al.* propose a high-speed network telemetry mechanism called FAST-INT [133]. The Fast-INT monitors network change events through a reinforcement learning algorithm, and dynamically deploys and adjusts INT monitoring tasks to achieve efficient network monitoring in a short time. Vestin *et al.* develop a fast INT reporting collector based on the PDP [134]. The collector uses P4 language and is deployed on the stream processor of switches. It can quickly obtain the switch data plane information, while requiring a low network overhead and switch load. Mayer *et al.* propose a soft-failure localization framework based on ML [135]. The framework can be applied in the case of failure of telemetry equipment that cannot use INT to detect. In this framework, an artificial neural network is used to simulate network failures, and the approximate location of failures can be obtained quickly, which speeds up failure location determination.

B. Network Congestion Control and Management

Network congestion control and management is to ensure the network transmission performance remains at a high level for a long time. The PDP can help to find network congestion quickly and upload a congestion signal to the controller in real time, realizing the in-network congestion control and management. Feldmann *et al.* use PDP to identify elephant flow at line-speed level and assign an independent queue for each elephant flow through multi-queue management [136]. Further, they monitor queue state information in real time to detect network congestion. If congestion is about to occur, a congestion signal will be sent to inform the upstream node to change the packet forwarding rules, which can avoid network congestion and achieve accurate congestion control.

The PDP has advantages on network information acquisition in congestion control, while ML can predict occurrences of network congestion. Zhou *et al.* propose a supervised learning

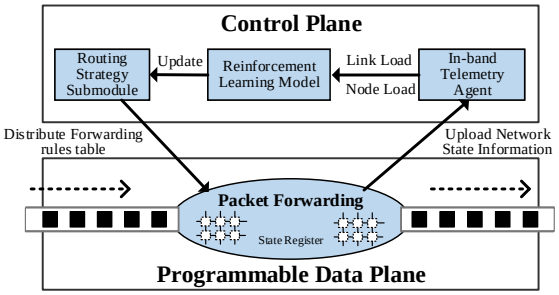


Fig. 10. Framework of Network Congestion Control with ML and PDP

regression model that can capture global routing behavior [137]. The model can obtain whole network routing information and predict network congestion.

In recent years, researchers have explored using ML and PDP to realize efficient and predictable in-network congestion management. Mai *et al.* propose a network congestion detection and management mechanism based on reinforcement learning for network congestion caused by instantaneous burst of elephant flows [138]. When an elephant flow fluctuates, the mechanism can adjust queueing assignment based on real-time feedback from reinforcement learning to avoid network congestion. Jain *et al.* obtain a large amount of data from practical telecommunication network and train the network traffic prediction model [139]. The model can predict the occurrence of network congestion and use big data to adjust the packet forwarding rules and improve the quality of service. Li *et al.* design a TCP-proximal policy congestion control (TCP-PPCC) algorithm [140]. The algorithm obtains network state information through the PDP, updates the forwarding policy offline, and further adjusts the new policy online. It can effectively prevent network congestion. Li *et al.* use the PDP to accurately collect network link state information, based on which, they determine whether the corresponding network link is about to be congested [43]. Further, they adopt an enhancement learning algorithm to minimize the maximum link utilization to avoid network congestion. The workflow of this mechanism is shown in Fig. 10. First, the network state information is collected through in-band telemetry when the packets are forwarded in the PDP. Then, the in-band telemetry agent in the control plane gathers the collected network state information and extracts the load of each network link and each node device as the state parameters for reinforcement learning. After that, the reinforcement leaning model generates routing adjustment strategies to avoid network congestion based on the state parameters. Finally, the control plane translates the strategies into packet forwarding rules and distributes them to the network devices on the programmable data plane.

C. Network Device Runtime Verification and Management

With the popularity of programmable switches, customized packet forwarding becomes the basis of more and more research works. Along with the trend, verification and management of the functional feasibility of programmable network devices and the validity of the runtime forwarding table

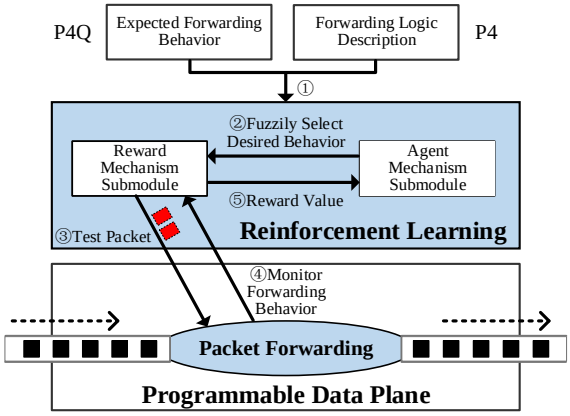


Fig. 11. Framework of Runtime Verification with ML and PDP

have become a research hotspot. The functional feasibility verification of network device refers to the analysis of whether the PDP packet forwarding processing has bugs [141] [142] [143] [144]. The validity of the runtime forwarding table refers to the analysis of whether there is an abnormal packet forwarding table when the PDP is running [144] [42] [145]. For example, malicious attackers can tamper a forwarding table to interrupt the user's services [144]. A verification mechanism is completed by Zhou *et al.* They design an anomalous runtime forwarding table mechanism [42]. The mechanism uses an intermediate representation based on a binary decision diagram to form the data plane probe of switch and to monitor the anomaly of packet forwarding. The mechanism can ensure that packets are transmitted efficiently and smoothly across the PDP.

In addition to manually collecting network packet forwarding state information to verify and manage the functional feasibility and the validity, researchers explore ML to realize automatic verification and improve the correctness and feasibility of packets forwarding management. Jagadeesan *et al.* formulate an approach to enhance automated verification with machine learning-based analytics and detect the faulty or malicious behaviors [146]. The approach takes the switch behavior as input and determines whether it is an faulty or malicious behavior through machine learning analysis. Furthermore, the approach has low operational requirements and can easily be deployed on the switch.

Shukla *et al.* propose a variation-coefficient and fuzzy-evaluation mechanism guided by reinforcement learning to verify the validity of packet forwarding logic and forwarding table during the operation of programmable network devices [147]. The workflow is shown in Fig. 11. First, the users describe expected packet forwarding behavior on the PDP through P4Q lightweight language, and provide packet forwarding logic described by P4 language. Second, the reward system of reinforcement learning generates two kinds of test packets according to information from the users. One is the ordinary packet, which is used for samples (seeds) of the initial environment state of reinforcement learning. The other is specific boundary packets (such as Ethernet address FF:FF:FF:FF:FF:FF), which is used to verify whether there are bugs in the packet forwarding logic. The reward system

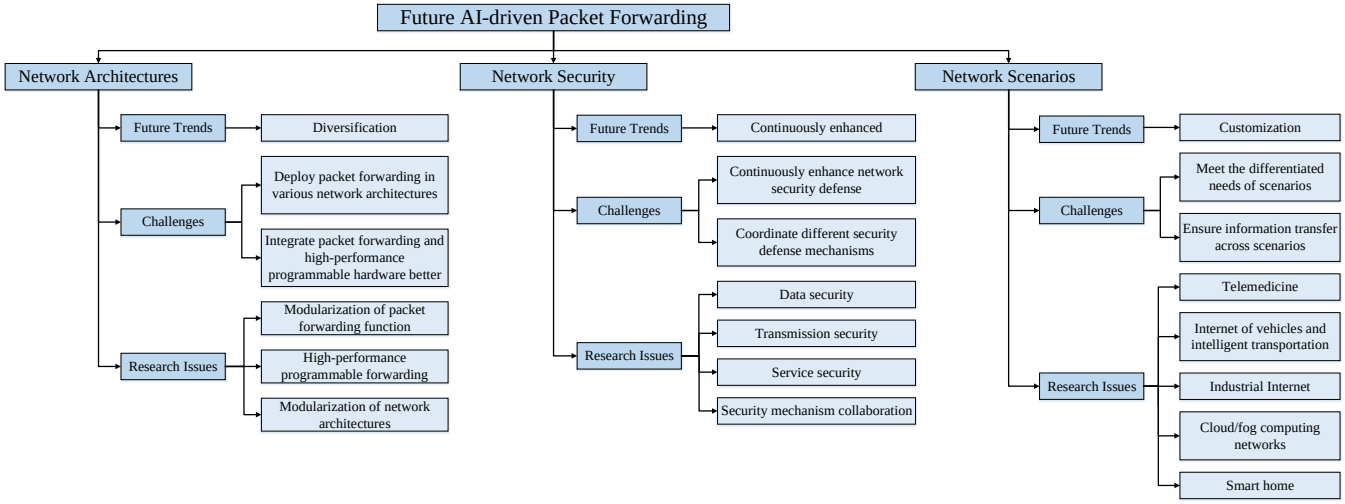


Fig. 12. Future Trends, Challenges and Open Issues of AI-driven Packet Forwarding

sends the ordinary test packets to complete the initialization of the reinforcement learning system. Then, the reward system sends out a specific boundary test packet, at which time the agent fuzzily selects an action expected to forward the packet. Meanwhile, the reward system monitors the forwarding process of the packet in real time. Finally, the forwarding of this packet is finished and the reward system compares the two actions and gives a reward value. The agent checks whether the reward value triggers the bug threshold. If it is, the agent will notify the user that there is a bug in the packet forwarding.

VII. Further Trends, Challenges and Open Issues

Packet forwarding is the foundation and the core of the Internet and still has a long way to go in the future. AI-driven packet forwarding technology is the basic future direction, and intelligence is one of the most important features of future networks [148]. This section discusses the future AI-driven packet forwarding technology from three development trends, and highligets challenges and research issues respectively. The structure of this section is shown in Fig. 12.

A. AI-driven Packet Forwarding for Diverse Network Architectures

Future Internet architecture is one of the most discussed research hotspots in recent years. Traditional IPv4 networks can no longer meet the needs of the current diversified and intelligent network requirements. Thus various new network architectures have emerged, such as open programmable network (ForCES, SDN), new service-oriented network system (SOI, NetServ, COMBO, SONA), content center network (NDN, DONA, PSIRP, NetInf), new mobility-oriented network system (MobilityFirst, HIP, LIN6, Six/One), intent-driven networks (IDN), and smart identifier networks (SINET). The explosion of proposed new network architectures has brought about today's Internet, where the traditional IPv4 network is deployed as the core and diversified new network architectures are accessed as private networks.

1) Challenges: Packet forwarding technology is an indispensable part in both traditional and new network architectures. However, different network architectures have different deployment planes and functional requirements for packet forwarding, which leads to the first challenge of furture packet forwarding technology – *How to properly deploy packet forwarding in various network architectures?* In addition, to satisfy differentiated network architecture requirements, high-performance programmable hardware emergies and can be utilized for efficient and custom packet forwarding. *How can we better integrate packet forwarding and high-performance programmable hardware?* The integration becomes a new challenge. On one hand, diverse network architectures have different requirements, and the compatibility of high-performance programmable hardware in the context becomes a problem. On the other hand, the enrichment of forwarding functions requires high-performance hardware to provide more programming interfaces, which undoubtedly puts forward new challenges for hardware design.

Furthermore, a careful look at today's Internet results in a question: whether today's Internet with IPv4 as the core network and new architectures as the private networks will continue to have a development momentum? The new architecture is supposed to break the bottleneck of the traditional Internet and replace IPv4 network. But the actual situation is that the widespread IPv4 popularing provides poor compatibility between new network architectures. New network architectures can exist only in the form of private networks. *Can we develop a super network?* We call this super network as "unified network". It should have excellent forwarding compatibility, which can replace IPv4 network completely at low cost. At the same time, it should have extensive backward compatibility, which can easily access various new networks and realize the true integration of the Internet.

2) Reasearch Issues: In response to the challenges, we identify three further research directions.

- **Modularization of packet forwarding functions.** Functional modularity is not a new concept in programmable network. In network architecture ONOS, various network

functions exist in the form of APP, and the ONOS can arbitrarily take some functions to meet user needs. In essence, packet forwarding in the Internet is a network function, and its basic elements are the packet storage and packet forwarding. Hence, it is possible to represent it as separate modules for multiple API interfaces and for embedding in different network architectures. Even if the network architecture changes, packet forwarding can be updated and implemented.

- **High-performance programmable forwarding.** The enhancement of high-performance programmable forwarding can be carried out from two aspects: general hardware chip and interface enrichment. The high-performance programmable forwarding must be deployed on specific network devices. For example, DPDK is deployed on supported nics and FPGA is deployed on programmable network cards. All these require unique interface and hardware of forwarding devices. General hardware chips may be an answer. The packet forwarding can be integrated into a chip placed in the general network hardware. Devices in various network architectures can install this network hardware to implement the packet forwarding. Interface enrichment addresses the integration of programmable forwarding capabilities with high-performance hardware. The richer the forwarding functions, the more interfaces the hardware needs to provide. How to design high-performance programmable hardware supporting multiple forwarding functions deserves further studying.
- **Modularization of Internet architectures.** The modularization of Internet architectures is an important approach towards the super network. Compared to the modularization of forwarding functions, network architecture modularization looks at the network from a higher perspective. The core of the super network can be similar to that of an Android system. The system itself does not realize any network functions, but establishes a framework for the operation of all network functions and coordinates mutual communications of various networks. Such approach allows for freedom to design a unique network architecture and install it in the system as an APP. The network architecture can operate internally as an independent entity while communicating with other network architectures as a subnet.

B. AI-driven Packet Forwarding for Enhanced Network Security

We discuss network security separately because network security is receiving more and more attentions in the global community. With the rapid advances of information technology, the digitalization of information dominates the industry, and network security has become one of the most important technical issues.

1) Challenges: Network security defense and security attack are an opposing and unified topic. For any security attack, given a sufficient time, each security defense mechanism can be developed. In turn, each security defense mechanism faces vulnerabilities to new security attacks. Therefore, security attack and security defense are two sides of the coin. There is no security attack that cannot be solved and no security defense that cannot be broken. *How to continuously enhance network security defense* is a challenge to the development of network security solutions. In addition, most existing network security defense mechanisms defend against a specific type of security attacks. However, paying too much attention to each specific type of security attack defense leads to lack of the coordination between security defense mechanisms and causes new vulnerabilities. *How to defend more types of security attacks? How to coordinate different security defense mechanisms?* These challenging question requirs more research efforts.

2) Reasearch Issues: In view of the challenges, we focus on typical network security defense mechanisms related to packet forwarding technology and divide them into four categories: data security, transmission security, service security and security defense mechanism collaboration.

- **Data securtiy.** Data security includes data encryption and data recovery. For data encryption, data sources or switching devices encrypt important data before sending them into the network. In this way, hackers cannot decrypt the packets to obtain the original information. AI can help the switch select an appropriate encryption algorithm and generate random encryption keys to further improve data security. Blockchain establishes an information chain with multiple nodes responsible for security. The information in blockchain is difficult to be modified. Future research may focus on the integration of blockchain and AI-driven packet forwarding technology to realize decentralized data exchange and intelligent forwarding. In addition, blockgraph brings more possibilities for blockchain [149]. Blockgraph transforms two-dimensional lines into three-dimensional graphs, further expanding the secure responsibility of nodes and enhancing mutual supervision between nodes.

  In packet transmission, hackers can destroy a packet to make data incomplete when reaching the destination. Data recovery can be a solution to it. Data recovery can use network coding to increase redundancy in a packet. When the packets arrive at the destination terminal, the terminal only needs partial data to restore the complete original information. The network coding and AI-driven forwarding technology can be integrated. A network coding algorithm should be flexibly selected by intelligent forwarding technology, so as to ensure more secure data.
- **Transmission security.** While data security means putting the data in a safe, transmission security means hiding the delivery route. One popular approach in transmission security is multipath transmission technology, which separates data packets and sends them from different paths to hide the transmission routes. When a hacker intercepts packets on only one path, the original data cannot be completely obtained. AI can help switches select more secure routes to forward packets, further improving the transmission security. Another research hotspot is quantum communication, which uses quan-

tum entanglement technology to completely eliminate the possibility of path interception. Although quantum communication remains a theory, it may have a great potential to improve transmission security.

- **Service securtiy.** Service security does not protect packets in the network, but protects critical network nodes such as DNS servers and cloud servers. There are different servers that need to be defended, but the attack types are similar, including DDoS attacks and abnormal flow attacks. The typical defense mechanism for service security attacks is packet detection and filtering, which is discussed in Section V. In fact, the interception and filtering of abnormal packets also block the forwarding of normal packets, affecting the quality of service. Therefore, improving the accuracy of recognition by AI is an issue worth studying.
- **Security mechanism collaboration.** Collaborative security is no longer a single security defense, but a security defense for the entire network. Collaborative security establishes a intelligent collaborative system, which can flexibly schedule various security defense mechanisms (e.g., firewall, intrusion detection system, security audit system, and log analysis system) by AI algorithm to implement comprehensive network security defense. However, with more types of security attacks and more diversified security defense mechanisms, rapid scheduling among these mechanisms becomes increasingly difficult. Future security coordination systems with high efficiency are worth studying.

C. AI-driven Packet Forwarding for Customized Network Scenarios

"Internet Plus" has become a new model for Internet development. The formation of intelligent information platform providing services for vertical industries is another trend of Internet development. Vertical industries are diverse and have different performance requirements for the Internet. According to characteristics of the industries, the Internet should focus on providing efficient performance in a certain aspect, such as ultra-low delay for telemedicine, network collaboration for industrial Internet, and network intelligence for smart home. The Internet and various vertical industries have formed various kinds of private networks, which plays an important role in society.

1) Challenges: The traditional Internet has relatively single functions and simple forwarding technology, which cannot provide customized network requirements for various industries. *How to meet the differentiated needs of each scenario?* This is a major challenge in the trend of network scenario customization. It is an effective solution to select appropriate Internet function modules according to the requirements of each scenario, provide corresponding service performance, and form demand-centered forwarding scheduling. In addition, it is difficult for private networks to communicate with each other. Each network scenario uses independent network protocols and packet forwarding technologies to achieve efficient internal communication. However, if a device in one scenario wants to access a device in another scenario, problems may occur. The compatibility of network protocol recognition and packet forwarding technology becomes the largest obstacle. *How to ensure information transfer across scenarios* poses technical challenges.

2) Research Issues: There are many kinds of network scenarios. We select some popular scenarios, discuss the AI-driven packet forwarding in them, and point out their research issues.

- **Telemedicine.** Telemedicine is a popular network scenario in recent years. With the COVID-19 outbreak, the need for telemedicine has become even more obvious. Doctors can remotely control surgical equipment and operate on patients through a dedicated network, forming a new operation mode with no contact and zero infection. Telemedicine has a high requirement for network delay, and it is necessary to ensure that doctors at different physical locations carry out surgical operations synchronously. AI-driven packet forwarding can meet these needs. On one hand, AI can select a dedicated packet transmission path to ensure the stability of surgical operations. On the other hand, AI can adjust the different network delay of doctors at different physical locations to ensure the synchronization of operations. Therefore, how to further improve ultra-low delay and keep the operation synchronized are the research issues of telemedicine.
- **Internet of vehicles and intelligent transportation.** Internet of vehicles is another scenario with high requirements on network delay, which is an important part of intelligent transportation. Internet of vehicles ensures instant information exchanges between vehicles, establishes effective vehicle sensile driving models, and prevents traffic accidents through intelligent traffic control algorithms. Internet of vehicles and intelligent transportation also have high requirements on network computing and network reliability. Vehicle network information changes frequently and rapidly. Timely information processing and strategy generation are important prereguisites to ensure intelligent transportation. AI can adjust the load of data computing nodes in the vehicle network to improve computing speed. AI-driven packet forwarding can ensure high-speed and stable data transmission. However, the deployment of AI in vehicles puts high demands on the hardware, which hinders the further vehicle network development. Therefore, optimizing AI algorithms and proposing the AI deployment strategies are the research issues for Internet of vehicles and intelligent transportation.
- **Industrial Internet.** Industrial Internet is a platform for the effective integration of information and communication technology and industrial economy, which can promote the development of industry digitization, networking and intelligence. Industrial Internet focuses on network coordination ability. It connects the four systems of network, platform, data and security to provide a perfect application model for industrial production and services. Information collaboration and integration between systems is the key to an industrial Internet.

AI-driven packet forwarding technology can coordinate the work of the four systems and improve the network collaboration capability on the basis of ensuring the transmission performance. Therefore, delopying AI-driven packet forwarding for high network collaboration ability is the research issue of industrial Internet.

- **Cloud/fog computing networks.** Cloud/fog computing networks generally are an essential component of network scenarios. A cloud/fog computing network has high information computing and processing capacity, which can provide efficient information processing for many network scenarios. The deployment and allocation of resources are important issues for the development of cloud/fog computing. AI-driven packet forwarding can reduce the data transmission delay among computing nodes, balance the computing load of each node, and improve resource allocation. Utilizing AI-driven packet forwarding better for resource deployment while maintaining a high computing performance is the research point of cloud/fog computing in the future.
- **Smart home.** Smart home focuses on the ability to generate and respond to network control information. Smart home can obtain the needs of the user, generate the equipment control instructions and realize intelligent control of the homely electrical equipment. The response time of smart home program is closely related to the user experiences. The realization of fast and efficient equipment control through AI-driven packet forwarding is the research issue of smart home.

In addition to the three research directions discussed in this section, there are other development directions for AI-driven packet forwarding, such as network resource allocation, network devices management and network fault detection. These directions are also worth studying.

VIII. Conclusion

This paper presents a survey on AI-driven packet forwarding with PDP. We first discuss the typical framework of AI-driven packet forwarding and show the existing problems of this framework. Then, we introduce the new framework of PDP-assisted AI-driven packet forwarding to show that PDP can improve AI-driven packet forwarding. After that, we discuss the delay, throughput, security and reliability based on the evolution of packet forwarding: packet forwarding, AI-driven packet forwarding, and AI-driven packet forwarding with PDP, and show the existing studies on them. Finally, we elaborate our own views on the AI-driven packet forwarding evolution and propose three research directions.

AI with PDP can effectively improve the performance of packet forwarding in many aspects, which play an important role in the development of the Internet. However, there are still many challenges in this field. This paper attempts to study the current development of packet forwarding and discusses the future research direction. We hope that our research and discussion can provide more information for other scholars to study packet forwarding and make contributions to the development of advanced networking technology.

References

[1] G. Zhu and W. B. Kang, *Application and Analysis of Three Common High-Performance Network Data Processing Frameworks*. Big Data Analytics for Cyber-Physical System in Smart City, 2021.

[2] J. Nam, S. Lee, H. Seo, P. Porras, V. Yegneswaran, and S. Shin, "BASTION: A security enforcement network stack for container networks," in *2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020*, A. Gavrilovska and E. Zadok, Eds. USENIX Association, 2020, pp. 81–95. [Online]. Available: https://www.usenix.org/conference/atc20/presentation/nam

[3] B. Rauf, H. Abbas, A. M. Sheri, W. Iqbal, and A. W. Khan, "Enterprise integration patterns in sdn: A reliable, fault-tolerant communication framework," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6359–6371, 2021.

[4] T. Jepsen, A. Fattaholmanan, M. Moshref, N. Foster, A. Carzaniga, and R. Soulé, "Forwarding and routing with packet subscriptions," in *CoNEXT '20: The 16th International Conference on emerging Networking EXperiments and Technologies, Barcelona, Spain, December, 2020*, D. Han and A. Feldmann, Eds. ACM, 2020, pp. 282–294. [Online]. Available: https://doi.org/10.1145/3386367.3431315

[5] Z. Xiang, F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein, and F. H. P. Fitzek, "Reducing latency in virtual machines: Enabling tactile internet for human-machine co-working," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1098–1116, 2019.

[6] Y. Afek, A. Bremler-Barr, and S. L. Feibish, "Zero-day signature extraction for high-volume attacks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 691–706, 2019.

[7] H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, "Towards fast and reliable multihop routing in vanets," *IEEE Transactions on Mobile Computing*, vol. 19, no. 10, pp. 2461–2474, 2020.

[8] 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. Surv. Tutorials*, vol. 21, no. 1, pp. 393–430, 2019. [Online]. Available: https://doi.org/10.1109/COMST.2018.2866942

[9] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, no. 1, pp. 45–66, Jan. 2020.

[10] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, early access, 2021, DOI: 10.1109/COMST.2021.3135829.

[11] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, "Smartcc: A reinforcement learning approach for multipath tcp congestion control in heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2621–2633, 2019.

[12] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 29–35.

[13] Y. Fan, L. Wang, and X. Yuan, "Controller placements for latency minimization of both primary and backup paths in sdns," *Comput. Commun.*, vol. 163, pp. 35–50, 2020. [Online]. Available: https://doi.org/10.1016/j.comcom.2020.09.001

[14] R. Chai, Q. Yuan, L. Zhu, and Q. Chen, "Control plane delay minimization-based capacitated controller placement algorithm for SDN," *EURASIP J. Wirel. Commun. Netw.*, vol. 2019, p. 282, 2019. [Online]. Available: https://doi.org/10.1186/s13638-019-1607-x

[15] C. Busse-Grawitz, R. Meier, A. Dietmüller, T. Bühler, and L. Vanbever, "pforest: In-network inference with random forests," *CoRR*, vol. abs/1909.05680, 2019. [Online]. Available: http://arxiv.org/abs/1909.05680

[16] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2019.

[17] P. K. Donta, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Inf. Fusion*, vol. 49, pp. 1–25, 2019. [Online]. Available: https://doi.org/10.1016/j.inffus.2018.09.013

[18] Y. Cheng, J. Geng, Y. Wang, J. Li, D. Li, and J. Wu, "Bridging machine learning and computer network research: a survey," *CCF Trans. Netw.*, vol. 1, no. 1-4, pp. 1–15, 2019. [Online]. Available: https://doi.org/10.1007/s42045-018-0009-7

[19] R. Bifulco and G. Rtvri, "A survey on the programmable data plane: Abstractions, architectures, and open problems," in *2018 IEEE 19th*

International Conference on High Performance Switching and Routing (HPSR), 2018, pp. 1–7.

[20] E. Kaljic, A. Maric, P. Njemcevic, and M. Hadzialic, "A survey on data plane flexibility and programmability in software-defined networking," *IEEE Access*, vol. 7, pp. 47 804–47 840, 2019.

[21] S. Han, S. Jang, H. Choi, H. Lee, and S. Pack, "Virtualization in programmable data plane: A survey and open challenges," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 527–534, 2020.

[22] E. F. Kfoury, J. Crichigno, and E. Bou-Harb, "An exhaustive survey on P4 programmable data plane switches: Taxonomy, applications, challenges, and future trends," *IEEE Access*, vol. 9, pp. 87 094–87 155, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3086704

[23] H. Yao, T. Mai, X. Xu, P. Zhang, M. Li, and Y. Liu, "Networkai: An intelligent network architecture for self-learning control strategies in software defined networks," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4319–4327, 2018.

[24] N. Foster, N. McKeown, J. Rexford, G. M. Parulkar, L. L. Peterson, and O. Sunay, "Using deep programmability to put network owners in control," *Comput. Commun. Rev.*, vol. 50, no. 4, pp. 82–88, 2020. [Online]. Available: https://doi.org/10.1145/3431832.3431842

[25] A. S. Yogapratama and M. Suryanegara, "Dealing with the latency problem to support 5g-urllc: A strategic view in the case of an indonesian operator," in *2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP)*, 2020, pp. 96–100.

[26] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead, fine-grained end-to-end monitoring of wireless networks using in-band telemetry," in *2019 15th International Conference on Network and Service Management (CNSM)*, 2019, pp. 1–5.

[27] B. Zhang, T. Zhang, and Z. Yu, "Ddos detection and prevention based on artificial intelligence techniques," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 1276–1280.

[28] Z. Xiong and N. Zilberman, "Do switches dream of machine learning?: Toward in-network classification," in *Proceedings of the 18th ACM Workshop on Hot Topics in Networks, HotNets 2019, Princeton, NJ, USA, November 13-15, 2019*. ACM, 2019, pp. 25–33. [Online]. Available: https://doi.org/10.1145/3365609.3365864

[29] K. A. Simpson, R. Cziva, and D. P. Pezaros, "Seir: Dataplane assisted flow classification using ml," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[30] O. Salman, I. H. Elhajj, A. Chehab, and A. Kayssi, "A machine learning based framework for iot device identification and abnormal traffic detection," *Transactions on Emerging Telecommunications Technologies*, 2019.

[31] K. Lee, S.-Y. Lee, and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," *IEEE Access*, vol. 7, pp. 110 205–110 215, 2019.

[32] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for ddos detection," *Appl. Intell.*, vol. 48, no. 10, pp. 3193–3208, 2018. [Online]. Available: https://doi.org/10.1007/s10489-018-1141-2

[33] W. He, Y. Liu, H. Yao, T. Mai, N. Zhang, and F. R. Yu, "Distributed variational bayes-based in-network security for the internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6293–6304, 2021.

[34] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, D. Niyato, and D. I. Kim, "Distributed deep learning at the edge: A novel proactive and cooperative caching framework for mobile edge networks," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1220–1223, 2019.

[35] J. Vestin, A. Kassler, and J. kerberg, "Fastreact: In-network control and caching for industrial control networks using programmable data planes," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2018, pp. 219–226.

[36] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Preemptive sdn load balancing with machine learning for delay sensitive applications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 947–15 963, 2020.

[37] A. Alnoman, "Supporting delay-sensitive iot applications: A machine learning approach," in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2020, pp. 1–4.

[38] LI, Dan, LIN, Du, JIANG, Changlin, Wang, and Lingqiang, "Sopa: Source routing based packet-level multi-path routing in data center networks," *ZTE Communications*, vol. v.16;No.62, no. 02, pp. 46–58, 2018.

[39] M. Kheirkhah, I. Wakeman, and G. Parisis, "Mmptcp: A multipath transport protocol for data centers," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[40] J. Shi, W. Quan, D. Gao, M. Liu, G. Liu, C. Yu, and W. Su, "Flowlet-based stateful multipath forwarding in heterogeneous internet of things," *IEEE Access*, vol. 8, pp. 74 875–74 886, 2020.

[41] J. Woodruff, M. Ramanujam, and N. Zilberman, "P4dns: In-network dns," in *2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2019, pp. 1–6.

[42] Y. Zhou, J. Bi, Y. Lin, Y. Wang, D. Zhang, Z. Xi, J. Cao, and C. Sun, "P4tester: Efficient runtime rule fault detection for programmable data planes," in *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, 2019, pp. 1–10.

[43] Q. Li, J. Zhang, T. Pan, T. Huang, and Y. Liu, "Data-driven routing optimization based on programmable data plane," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, 2020, pp. 1–9.

[44] A. Mubarakali, A. D. Durai, M. Alshehri, O. Alfarraj, and D. Mavaluru, "Fog-based delay-sensitive data transmission algorithm for data forwarding and storage in cloud environment for multimedia applications," *Big Data*, 2020.

[45] M. F. Majeed, S. H. Ahmed, and M. N. Dailey, "Enabling push-based critical data forwarding in vehicular named data networks," *IEEE Communications Letters*, vol. 21, no. 4, pp. 873–876, 2017.

[46] T. K. Patra and A. Sunny, "Forwarding in heterogeneous mobile opportunistic networks," *IEEE Communications Letters*, vol. 22, no. 3, pp. 626–629, 2018.

[47] K. C. Tsai, L. Wang, and Z. Han, "Mobile social media networks caching with convolutional neural network," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2018, pp. 83–88.

[48] P. Cheng, C. Ma, M. Ding, Y. Hu, Z. Lin, Y. Li, and B. Vucetic, "Localized small cell caching: A machine learning approach based on rating data," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1663–1676, 2019.

[49] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for icn: A deep learning approach with sdn," *IEEE Access*, vol. 6, pp. 5075–5089, 2018.

[50] L. Luo, R. Chai, Q. Yuan, J. Li, and C. Mei, "End-to-end delay minimization-based joint rule caching and flow forwarding algorithm for sdn," *IEEE Access*, vol. 8, pp. 145 227–145 241, 2020.

[51] H. Huang, S. Guo, P. Li, W. Liang, and A. Y. Zomaya, "Cost minimization for rule caching in software defined networking," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1007–1016, 2016.

[52] S. Bera, S. Misra, and M. S. Obaidat, "Mobi-flow: Mobility-aware adaptive flow-rule placement in software-defined access network," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1831–1842, 2019.

[53] T. Mu, A. I. Al-Fuqaha, K. Shuaib, F. Sallabi, and J. Qadir, "SDN flow entry management using reinforcement learning," *ACM Trans. Auton. Adapt. Syst.*, vol. 13, no. 2, pp. 11:1–11:23, 2018. [Online]. Available: https://doi.org/10.1145/3281032

[54] C. Zhang, J. Bi, Y. Zhou, K. Zhang, and Z. Ma, "B-cache: A behavior-level caching framework for the programmable data plane," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00 084–00 090.

[55] R. Parizotto, L. Castanheira, R. H. Ribeiro, L. Zembruzki, A. S. Jacobs, L. Z. Granville, and A. Schaeffer-Filho, "Shadowfs: Speeding-up data plane monitoring and telemetry using p4," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[56] G. Grigoryan, Y. Liu, and M. Kwon, "Pfca: A programmable fib caching architecture," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1872–1884, 2020.

[57] S. Jung, J. Kim, and J.-H. Kim, "Intelligent active queue management for stabilized qos guarantees in 5g mobile networks," *IEEE Systems Journal*, pp. 1–10, 2020.

[58] C. Olariu, M. Zuber, and C. Thorpe, "Delay-based priority queueing for voip over software defined networks," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 652–655.

[59] T. Qiu, R. Qiao, and D. O. Wu, "Eabs: An event-aware backpressure scheduling scheme for emergency internet of things," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 72–84, 2018.

[60] K. Zhu, G. Shen, Y. Jiang, J. Lv, Q. Li, and M. Xu, "Differentiated transmission based on traffic classification with deep learning in

datacenter," in *2020 IFIP Networking Conference (Networking)*, 2020, pp. 599–603.

[61] C. Papagianni and K. D. Schepper, "PI2 for P4: an active queue management scheme for programmable data planes," in *Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies, CoNEXT 2019, Companion Volume, Orlando, FL, USA, December 9-12, 2019*. ACM, 2019, pp. 84–86. [Online]. Available: https://doi.org/10.1145/3360468.3368189

[62] R. Kundel, J. Blendin, T. Viernickel, B. Koldehofe, and R. Steinmetz, "P4-codel: Active queue management in programmable data planes," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2018, pp. 1–4.

[63] A. G. Alcoz, A. Dietmüller, and L. Vanbever, "SP-PIFO: approximating push-in first-out behaviors using strict-priority queues," in *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, Santa Clara, CA, USA, February 25-27, 2020*, R. Bhagwan and G. Porter, Eds. USENIX Association, 2020, pp. 59–76. [Online]. Available: https://www.usenix.org/conference/nsdi20/presentation/alcoz

[64] Y. Shi, Y. Zhang, H. Jacobsen, L. Tang, G. Elliott, G. Zhang, X. Chen, and J. Chen, "Using machine learning to provide reliable differentiated services for iot in sdn-like publish/subscribe middleware," *Sensors*, vol. 19, no. 6, p. 1449, 2019. [Online]. Available: https://doi.org/10.3390/s19061449

[65] C. Zhang, X. Wang, F. Li, Q. He, and M. Huang, "Deep learning-based network application classification for SDN," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 5, 2018. [Online]. Available: https://doi.org/10.1002/ett.3302

[66] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "On load distribution over multipath networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 3, pp. 662–680, 2012.

[67] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat, "WCMP: weighted cost multipathing for improved fairness in data centers," in *Ninth Eurosys Conference 2014, EuroSys 2014, Amsterdam, The Netherlands, April 13-16, 2014*, D. C. A. Bulterman, H. Bos, A. I. T. Rowstron, and P. Druschel, Eds. ACM, 2014, pp. 5:1–5:14. [Online]. Available: https://doi.org/10.1145/2592798.2592803

[68] Q. Wang, G. Shou, Y. Liu, Y. Hu, Z. Guo, and W. Chang, "Implementation of multipath network virtualization with sdn and nfv," *IEEE Access*, vol. 6, pp. 32 460–32 470, 2018.

[69] Y. Liu, X. Qin, T. Zhu, X. Chen, and G. Wei, "Improve MPTCP with SDN: from the perspective of resource pooling," *J. Netw. Comput. Appl.*, vol. 141, pp. 73–85, 2019. [Online]. Available: https://doi.org/10.1016/j.jnca.2019.05.015

[70] H. Zhang, J. Zhang, W. Bai, K. Chen, and M. Chowdhury, "Resilient datacenter load balancing in the wild," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2017, Los Angeles, CA, USA, August 21-25, 2017*. ACM, 2017, pp. 253–266. [Online]. Available: https://doi.org/10.1145/3098822.3098841

[71] L. Jin, W. Quan, G. Liu, D. Gao, C. H. Foh, and Q. Wang, "Dps: A delay-programmable scheduler for the packet out-of-order mitigation in heterogeneous networks," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[72] G. Liu, W. Quan, N. Cheng, N. Lu, H. Zhang, and X. Shen, "P4nis: Improving network immunity against eavesdropping with programmable data planes," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 91–96.

[73] E. Vanini, R. Pan, M. Alizadeh, P. Taheri, and T. Edsall, "Let it flow: Resilient asymmetric load balancing with flowlet switching," in *14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017*, A. Akella and J. Howell, Eds. USENIX Association, 2017, pp. 407–420. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/vanini

[74] K. Xue, J. Han, D. Ni, W. Wei, Y. Cai, Q. Xu, and P. Hong, "Dpsaf: Forward prediction based dynamic packet scheduling and adjusting with feedback for multipath tcp in lossy heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1521–1534, 2018.

[75] N. P. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, "HULA: scalable load balancing using programmable data planes," in *Proceedings of the Symposium on SDN Research, SOSR 2016, Santa Clara, CA, USA, March 14 - 15, 2016*, B. Godfrey and M. Casado, Eds. ACM, 2016, p. 10. [Online]. Available: https://doi.org/10.1145/2890955.2890968

[76] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, "Amps: Application aware multipath flow routing using machine learning in sdn," in *2017 Twenty-third National Conference on Communications (NCC)*, 2017, pp. 1–6.

[77] R. Ji, Y. Cao, X. Fan, Y. Jiang, G. Lei, and Y. Ma, "Multipath tcp-based iot communication evaluation: From the perspective of multipath management with machine learning," *Sensors*, vol. 20, no. 22, p. 6573, 2020. [Online]. Available: https://doi.org/10.3390/s20226573

[78] A. Azzouni, R. Boutaba, and G. Pujolle, "Neuroute: Predictive dynamic routing for software-defined networks," in *2017 13th International Conference on Network and Service Management (CNSM)*, 2017, pp. 1–6.

[79] T. Gilad, N. R. Schiff, P. B. Godfrey, C. Raiciu, and M. Schapira, "MPCC: online learning multipath transport," in *CoNEXT '20: The 16th International Conference on emerging Networking EXperiments and Technologies, Barcelona, Spain, December, 2020*, D. Han and A. Feldmann, Eds. ACM, 2020, pp. 121–135. [Online]. Available: https://doi.org/10.1145/3386367.3433030

[80] M. R. Kanagarathinam, H. Natarajan, K. Arunachalam, I. Sandeep, and V. Sunil, "Sms: Smart multipath switch for improving the throughput of multipath tcp for smartphones," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[81] B. Mohammed, M. Kiran, and N. Krishnaswamy, "Deeproute on chameleon: Experimenting with large-scale reinforcement learning and sdn on chameleon testbed," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, 2019, pp. 1–2.

[82] J. Lim, S. Nam, J. H. Yoo, and W. K. Hong, "Best nexthop load balancing algorithm with inband network telemetry," in *2020 16th International Conference on Network and Service Management (CNSM)*, 2020.

[83] R. B. Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, "PINT: probabilistic in-band network telemetry," in *SIGCOMM '20: Proceedings of the 2020 Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, Virtual Event, USA, August 10-14, 2020*, H. Schulzrinne and V. Misra, Eds. ACM, 2020, pp. 662–680. [Online]. Available: https://doi.org/10.1145/3387514.3405894

[84] K. Liu, "Distributed asynchronous learning for multipath data transmission based on p-ddqn," in *the 3th conference on advanced computing and endogenous security*, 2020.

[85] C. Hardegen and S. Rieger, "Prediction-based flow routing in programmable networks with p4," in *2020 16th International Conference on Network and Service Management (CNSM)*, 2020, pp. 1–5.

[86] F. Li, J. Cao, X. Wang, and Y. Sun, "A qos guaranteed technique for cloud applications based on software defined networking," *IEEE Access*, vol. 5, pp. 21 229–21 241, 2017.

[87] W.-x. Liu, "Intelligent routing based on deep reinforcement learning in software-defined data-center networks," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.

[88] Y. Hu, Z. Li, J. Lan, J. Wu, and L. Yao, "Ears: Intelligence-driven experiential network architecture for automatic routing in software-defined networking," *China Communications*, vol. 17, no. 2, pp. 149–162, 2020.

[89] X. Yuan, C. Li, and X. Li, "Deepdefense: Identifying ddos attack via deep learning," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017, pp. 1–8.

[90] J. A. Pérez-Díaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, "A flexible sdn-based architecture for identifying and mitigating low-rate ddos attacks using machine learning," *IEEE Access*, vol. 8, pp. 155 859–155 872, 2020.

[91] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based ddos detection system in software-defined networking (SDN)," *EAI Endorsed Trans. Security Safety*, vol. 4, no. 12, p. e2, 2017. [Online]. Available: https://doi.org/10.4108/eai.28-12-2017.153515

[92] R. Santos, D. S. Silva, W. E. Santo, A. R. M. Ribeiro, and E. D. Moreno, "Machine learning algorithms to detect ddos attacks in SDN," *Concurr. Comput. Pract. Exp.*, vol. 32, no. 16, 2020. [Online]. Available: https://doi.org/10.1002/cpe.5402

[93] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in iot using sdn," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, 2017, pp. 1–6.

[94] G. Shang, P. Zhe, X. Bin, H. Aiqun, and R. Kui, "Flooddefender: Protecting data and control plane resources under sdn-aimed dos attacks," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[95] . C. Lapolli, J. Adilson Marques, and L. P. Gaspary, "Offloading real-time ddos attack detection to programmable data planes," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 19–27.

[96] M. Dimolianis, A. Pavlidis, and V. Maglaris, "A multi-feature ddos detection schema on p4 network hardware," in *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2020, pp. 1–6.

[97] F. Paolucci, F. Civerchia, A. Sgambelluri, A. Giorgetti, F. Cugini, and P. Castoldi, "P4 edge node enabling stateful traffic engineering and cyber security," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 1, pp. A84–A95, 2019.

[98] J. Vestin, A. Kassler, S. Laki, and G. Pongrcz, "Toward in-network event detection and filtering for publish/subscribe communication using programmable data planes," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 415–428, 2021.

[99] N. Narayanan, G. C. Sankaran, and K. M. Sivalingam, "Mitigation of security attacks in the sdn data plane using p4-enabled switches," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2019, pp. 1–6.

[100] F. Musumeci, V. Ionata, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-learning-assisted ddos attack detection with p4 language," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[101] Y. Mi and A. Wang, "Ml-pushback: Machine learning based pushback defense against ddos," in *Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies, CoNEXT 2019, Companion Volume, Orlando, FL, USA, December 9-12, 2019*. ACM, 2019, pp. 80–81. [Online]. Available: https://doi.org/10.1145/3360468.3368188

[102] D. Hu, P. Hong, and Y. Chen, "Fadm: Ddos flooding attack detection and mitigation system in software-defined networking," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

[103] X. Chen and S. Yu, "CIPA: A collaborative intrusion prevention architecture for programmable network and SDN," *Comput. Secur.*, vol. 58, pp. 1–19, 2016. [Online]. Available: https://doi.org/10.1016/j.cose.2015.11.008

[104] T. V. Phan, N. K. Bao, and M. Park, "A novel hybrid flow-based handler with ddos attacks in software-defined networking," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, 2016, pp. 350–357.

[105] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "Xgboost classifier for ddos attack detection and analysis in sdn-based cloud," in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 251–256.

[106] S. Poudyal, K. P. Subedi, and D. Dasgupta, "A framework for analyzing ransomware using machine learning," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 1692–1699.

[107] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on n-gram of opcodes," *Future Gener. Comput. Syst.*, vol. 90, pp. 211–221, 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.07.052

[108] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha, "Leveraging machine learning techniques for windows ransomware network traffic detection," *CoRR*, vol. abs/1807.10440, 2018. [Online]. Available: http://arxiv.org/abs/1807.10440

[109] S. K. Shaukat and V. J. Ribeiro, "Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning," in *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, 2018, pp. 356–363.

[110] G. Cusack, O. Michel, and E. Keller, "Machine learning-based detection of ransomware using SDN," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, SDN-NFVSec@CODASPY 2018, Tempe, AZ, USA, March 19-21, 2018*, G. Ahn, G. Gu, H. Hu, and S. Shin, Eds. ACM, 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3180465.3180467

[111] L. Boero, M. Marchese, and S. Zappatore, "Support vector machine meets software defined networking in ids domain," in *2017 29th International Teletraffic Congress (ITC 29)*, vol. 3, 2017, pp. 25–30.

[112] S. Ji, B. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *J. Netw. Comput. Appl.*, vol. 62, pp. 9–17, 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2015.12.004

[113] J. Niu, Y. Zhang, D. Liu, D. Guo, and Y. Teng, "Abnormal network traffic detection based on transfer component analysis," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.

[114] L. Kong, G. Huang, and K. Wu, "Identification of abnormal network traffic using support vector machine," in *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2017, pp. 288–292.

[115] B. Kong, Z. Liu, G. Zhou, and X. Yu, "A method of detecting the abnormal encrypted traffic based on machine learning and behavior characteristics," in *ICCNS 2019: The 9th International Conference on Communication and Network Security, Chongqing, China, November 15-17, 2019*. ACM, 2019, pp. 47–50. [Online]. Available: https://doi.org/10.1145/3371676.3371705

[116] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, and X. Wang, "Pccn: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, pp. 119 904–119 916, 2019.

[117] H. Kim and A. Gupta, "ONTAS: flexible and scalable online network traffic anonymization system," in *Proceedings of the 2019 Workshop on Network Meets AI & ML, NetAI@SIGCOMM 2019, Beijing, China, August 23, 2019*. ACM, 2019, pp. 15–21. [Online]. Available: https://doi.org/10.1145/3341216.3342208

[118] J. Hypolite, J. Sonchack, S. Hershkop, N. Dautenhahn, A. DeHon, and J. M. Smith, "Deepmatch: practical deep packet inspection in the data plane using network processors," in *CoNEXT '20: The 16th International Conference on emerging Networking EXperiments and Technologies, Barcelona, Spain, December, 2020*, D. Han and A. Feldmann, Eds. ACM, 2020, pp. 336–350. [Online]. Available: https://doi.org/10.1145/3386367.3431290

[119] D. Scholz, S. Gallenmüller, H. Stubbe, B. Jaber, M. Rouhi, and G. Carle, "Me love (syn-)cookies: SYN flood mitigation in programmable data planes," *CoRR*, vol. abs/2003.03221, 2020. [Online]. Available: https://arxiv.org/abs/2003.03221

[120] R. Negi, A. Dutta, A. Handa, U. Ayyangar, and S. K. Shukla, "Intrusion detection amp; prevention in programmable logic controllers: A model-driven approach," in *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1, 2020, pp. 215–222.

[121] L. Castanheira, R. Parizotto, and A. E. Schaeffer-Filho, "Flowstalker: Comprehensive traffic flow monitoring on the data plane using p4," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[122] J. H. Lee and K. Singh, "Switchtree: In-network computing and traffic analyses with random forests," *Neural Computing and Applications*, 2020.

[123] C. Song, Y. Park, K. Golani, Y. Kim, K. Bhatt, and K. Goswami, "Machine-learning based threat-aware system in software defined networks," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–9.

[124] G. A. Ajaeiya, N. Adalian, I. H. Elhajj, A. Kayssi, and A. Chehab, "Flow-based intrusion detection system for sdn," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 2017, pp. 787–793.

[125] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016, pp. 258–263.

[126] Y. Zhou, D. Zhang, K. Gao, C. Sun, J. Cao, Y. Wang, M. Xu, and J. Wu, "Newton: intent-driven network traffic monitoring," in *CoNEXT '20: The 16th International Conference on emerging Networking EXperiments and Technologies, Barcelona, Spain, December, 2020*, D. Han and A. Feldmann, Eds. ACM, 2020, pp. 295–308. [Online]. Available: https://doi.org/10.1145/3386367.3431298

[127] T. Holterbach, E. C. Molero, M. Apostolaki, A. Dainotti, S. Vissicchio, and L. Vanbever, "Blink: Fast connectivity recovery entirely in the data plane," in *16th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2019, Boston, MA, February 26-28, 2019*, J. R. Lorch and M. Yu, Eds. USENIX Association, 2019, pp. 161–176. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/holterbach

[128] J. Hyun, N. Van Tu, and J. W.-K. Hong, "Towards knowledge-defined networking using in-band network telemetry," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–7.

[129] A. Lazaris and V. K. Prasanna, "Deepflow: a deep learning framework for software-defined measurement," in *Proceedings of the 2nd Workshop on Cloud-Assisted Networking, CAN@CoNEXT 2017, Incheon, Republic of Korea, December 12, 2017*, P. Sharma and J. Hwang, Eds. ACM, 2017, pp. 43–48. [Online]. Available: https://doi.org/10.1145/3155921.3155922

[130] A. Pashamokhtari, "Phd forum abstract: Dynamic inference on iot network traffic using programmable telemetry and machine learning," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2020, pp. 371–372.

[131] P. Janakaraj, P. Pinyoanuntapong, P. Wang, and M. Lee, "Towards in-band telemetry for self driving wireless networks," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 766–773.

[132] R. Hohemberger, A. G. Castro, F. G. Vogt, R. B. Mansilha, A. F. Lorenzon, F. D. Rossi, and M. C. Luizelli, "Orchestrating in-band data plane telemetry with machine learning," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2247–2251, 2019.

[133] F. Yang, W. Quan, N. Cheng, Z. Xu, X. Zhang, and D. Gao, "Fast-int: Light-weight and efficient in-band network telemetry in programmable data plane," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1–5.

[134] J. Vestin, A. Kassler, D. Bhamare, K.-J. Grinnemo, J.-O. Andersson, and G. Pongracz, "Programmable event detection for in-band network telemetry," in *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*, 2019, pp. 1–6.

[135] K. S. Mayer, J. A. Soares, R. P. Pinto, C. E. Rothenberg, D. S. Arantes, and D. A. A. Mello, "Machine-learning-based soft-failure localization with partial software-defined networking telemetry," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 13, no. 10, pp. E122–E131, 2021.

[136] A. Feldmann, B. Chandrasekaran, S. Fathalli, and E. N. Weyulu, "P4-enabled network-assisted congestion feedback: A case for nacks," in *BS '19: 2019 Workshop on Buffer Sizing, Stanford University, Palo Alto, CA, USA, December 2-3, 2019*. ACM, 2019, pp. 3:1–3:7. [Online]. Available: https://doi.org/10.1145/3375235.3375238

[137] Z. Zhou, S. Chahal, T.-Y. Ho, and A. Ivanov, "Supervised-learning congestion predictor for routability-driven global routing," in *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2019, pp. 1–4.

[138] T. Mai, H. Yao, X. Zhang, Z. Xiong, and D. Niyato, "A distributed reinforcement learning approach to in-network congestion control," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, 2020, pp. 817–822.

[139] S. Jain, M. Khandelwal, A. Katkar, and J. Nygate, "Applying big data technologies to manage qos in an sdn," in *2016 12th International Conference on Network and Service Management (CNSM)*, 2016, pp. 302–306.

[140] J. Li, Y. Guan, P. Ding, and S. Wang, "Tcp-ppcc: Online-learning proximal policy for congestion control," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2020, pp. 243–246.

[141] J. Liu, W. T. Hallahan, C. Schlesinger, M. Sharif, J. Lee, R. Soulé, H. Wang, C. Cascaval, N. McKeown, and N. Foster, "p4v: practical verification for programmable data planes," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2018, Budapest, Hungary, August 20-25, 2018*, S. Gorinsky and J. Tapolcai, Eds. ACM, 2018, pp. 490–503. [Online]. Available: https://doi.org/10.1145/3230543.3230582

[142] R. Stoenescu, D. Dumitrescu, M. Popovici, L. Negreanu, and C. Raiciu, "Debugging P4 programs with vera," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2018, Budapest, Hungary, August 20-25, 2018*, S. Gorinsky and J. Tapolcai, Eds. ACM, 2018, pp. 518–532. [Online]. Available: https://doi.org/10.1145/3230543.3230548

[143] L. Freire, M. C. Neves, L. Leal, K. Levchenko, A. E. S. Filho, and M. P. Barcellos, "Uncovering bugs in P4 programs with assertion-based verification," in *Proceedings of the Symposium on SDN Research, SOSR 2018, Los Angeles, CA, USA, March 28-29, 2018*. ACM, 2018, pp. 4:1–4:7. [Online]. Available: https://doi.org/10.1145/3185467.3185499

[144] P. Zhang, "Towards rule enforcement verification for software defined networks," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[145] C. Zhang, J. Bi, Y. Zhou, J. Wu, B. Liu, Z. Li, A. B. Dogar, and Y. Wang, "P4db: On-the-fly debugging of the programmable data plane," in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, 2017, pp. 1–10.

[146] L. J. Jagadeesan and V. Mendiratta, "Analytics-enhanced automated code verification for dependability of software-defined networks," in *IEEE International Symposium on Software Reliability Engineering Workshops*, 2017.

[147] A. Shukla, K. N. Hudemann, A. Hecker, and S. Schmid, "Runtime verification of P4 switches with reinforcement learning," in *Proceedings of the 2019 Workshop on Network Meets AI & ML, NetAI@SIGCOMM 2019, Beijing, China, August 23, 2019*. ACM, 2019, pp. 1–7. [Online]. Available: https://doi.org/10.1145/3341216.3342206

[148] H. Zhang, W. Quan, H.-c. Chao, and C. Qiao, "Smart identifier network: A collaborative architecture for the future internet," *IEEE Network*, vol. 30, no. 3, pp. 46–51, 2016.

[149] D. C. Morales, P. B. Velloso, A. Guerre, T. M. T. Nguyen, G. Pujolle, K. A. Agha, and G. Dua, "Blockgraph proof-of-concept," in *SIGCOMM '21: ACM SIGCOMM 2021 Conference, Virtual Event, August 23-27, 2021, Poster and Demo Sessions*, M. Chiesa, D. R. Choffnes, A. Markopoulou, and M. P. Barcellos, Eds. ACM, 2021, pp. 82–84. [Online]. Available: https://doi.org/10.1145/3472716.3472866

Wei Quan (Member, IEEE) received the Ph.D. degree in communication and information system from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014.

He is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing. He has published more than 20 papers in prestigious international journals and conferences including IEEE Communications Magazine, IEEE WIRELESS COMMUNICATIONS, IEEE NETWORK, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS LETTERS, IFIP Networking, IEEE ICC, and IEEE GLOBECOM. His research interests include key technologies for network analytics, future Internet, 5G networks, and vehicular networks.

Dr. Quan is a TPC Member of IEEE ICC in 2017 and 2018, IEEE INFOCOM (NewIP Workshop) in 2020, ACM MOBIMEDIA in 2015, 2016, and 2017, and IEEE CCIS in 2015 and 2016. He serves as an Associate Editor for the Journal of Internet Technology, Peer-to-Peer Networking and Applications, and IEEE Access, and as a technical reviewer for many important international journals. He is also a Member of ACM and a Senior Member of the Chinese Association of Artificial Intelligence.

Ziheng Xu received the B.E. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, National Engineering Lab for Next Generation Internet Technologies.

His research interests include software-defined networking, concurrent multipath transfer, machine learning, high-speed railway networks, and programmable data planes.

Mingyuan Liu received the bachelor's degree in communication engineering, in 2018. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Beijing Jiaotong University.

His current research interests include future networks, software defined networking (SDN), and cybersecurity.

Nan Cheng (Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

He is currently a Joint Professor with the School of Telecommunication, Xidian University, Xian, China. He is also a Joint Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, and with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include performance analysis, MAC, opportunistic communication for vehicular networks, unmanned aerial vehicles, and application of artificial intelligence for wireless networks.

Gang Liu (Student Member, IEEE) received the B.E. degree from Tiangong University, Tianjin, China, in 2016 and the Ph.D. degree with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, in 2021.

His current research interests include information-centric networking, software-defined networking, network function virtualization, programmable network language, and stateful forwarding.

Deyun Gao (Senior Member, IEEE) received the B.E. and M.E. degrees in electrical engineering and the Ph.D. degree in computer science from Tianjin University, China, in 1994, 1999, and 2002, respectively.

He spent one year as a Research Associate with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Hong Kong. He then spent three years as a Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. In 2007, he joined the faculty of Beijing Jiaotong University, Beijing, China, as an Associate Professor with the School of Electronics and Information Engineering and was promoted to a Full Professor, in 2012. In 2014, he was a Visiting Scholar with the University of California at Berkeley, Berkeley, CA, USA. His research interests include the Internet of Things, vehicular networks, and the next-generation Internet.

Hongke Zhang (Fellow IEEE) received the M.S. and Ph.D. degrees in electrical and communication systems from the University of Electronic Science and Technology of China, Chengdu, China, in 1988 and 1992, respectively.

From 1992 to 1994, he was a Postdoctoral Fellow with Beijing Jiaotong University, Beijing, China, where he is currently a Professor with the School of Electronic and Information Engineering and the Director of a National Engineering Lab on Next Generation Internet Technologies. He has authored more than ten books and the holder of more than 70 patents. His research has resulted in many papers, books, patents, systems and equipment in the areas of communications and computer networks.

Prof. Zhang is the Chief Scientist of a National Basic Research Program of China (973 Program) and has also served on the editorial board of several international journals.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks.

Dr. Shen received the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society, and Technical Recognition Award from Wireless Communications Technical Committee in 2019, and the AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premiers Research Excellence Award in 2003 from the Province of Ontario. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16 and IEEE Infocom'14, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He was the elected IEEE Communications Society Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and the Member of IEEE ComSoc Fellow Selection Committee. He was the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE Network, and IET Communications. He is a Registered Professional Engineer of Ontario, Canada, a Fellow of Engineering Institute of Canada, Canadian Academy of Engineering, and Royal Society of Canada, a Foreign Member of Chinese Academy of Engineering, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.

Weihua Zhuang (Fellow, IEEE) has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, since 1993, where she is currently a Professor and a Tier I Canada Research Chair in Wireless Communication Networks. Dr. Zhuang is a Fellow of the Royal Society of Canada, the Canadian Academy of Engineering, and the Engineering Institute of Canada. She is also an elected member of the Board of Governors and VP Publications of the IEEE Vehicular Technology Society. She was a recipient of the 2017 Technical Recognition Award from the IEEE Communications Society Ad Hoc & Sensor Networks Technical Committee and a co-recipient of several best paper awards from IEEE conferences. She was the Technical Program Chair/Co-Chair of IEEE Vehicular Technology Conference (VTC) Fall 2017 and Fall 2016 and the Technical Program Symposia Chair of the IEEE Global Communications Conference (Globecom) 2011. She was the Editor-in-Chief of IEEE Transactions on Vehicular Technology from 2007 to 2013.