IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 24, 2019, accepted January 8, 2020, date of publication January 13, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966045

GCLR: GNN-Based Cross Layer Optimization for Multipath TCP by Routing

TING ZHU, XIAOHUI CHEN, LI CHEN, WEIDONG WANG, AND GUO WEI
Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China

Corresponding author: Xiaohui Chen (cxh@ustc.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFA0701603, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT Multipath TCP has attracted increasing attention as a promising technology for 5G networks. To fully utilize network interfaces on multi-homed terminals and the whole network resources, MPTCP is proposed as an extension of TCP to transfer packets concurrently over multiple paths. Cross layer optimization techniques have been applied for MPTCP such as routing and path management. However, existing multipath routing algorithms and network modeling techniques are facing the challenges of subflow asymmetry due to network heterogeneity, thus cannot handle routing optimization problems comprehensively. To address these problems, in this paper, firstly, a novel Graph Neural Network (GNN) based multipath routing model is proposed to explore the complications among links, paths, subflows and the MPTCP connection on various topologies. Leveraging the GNN model, expected throughput can be predicted with given network topology and multipath routes, which can further be the guidance for optimzing the multipath routing. Then, GCLR, a GNN based cross layer optimization system for MPTCP by routing, is proposed with the help of SDN (Software Defined Networking). According to simulation results, our off-line learned GNN model can predict the expected throughput of specific MPTCP connections with very low error. Besides, it's validated that the model has high generalization ability in terms of connection arbitrary and topology arbitrary, it can maintain MSE (mean squared error) at a low level when the situations are not seen during training, which is sufficient for throughput prediction in multipath routing decisions. Finally, the online routing optimization system is realized using SDN, experimental results show that our proposed routing optimization system can achieve significant throughput enhancement compared with traditional multipath routing algorithms.

INDEX TERMS Routing, multipath TCP, graph neural network, cross layer optimization, software defined networking.

I. INTRODUCTION

As technologies evolve, networks are on a trend towards multi-path. However, traditional TCP (transmission control protocol), in essence, is designed to be a single-path protocol and incapable to make use of multiple paths concurrently. Multipath TCP (MPTCP) [1], proposed by the Internet Engineering Task Force (IETF), aims to boost data rate and move congestion away by using multiple alternate paths. Since MPTCP can fully utilize network resources between multi-homed devices, it has attracted increasing interests as a promising technology in mobile communications [2]–[6] and in data centers [7]–[11].

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

Random-based approaches such as Equal-Cost Multi-Path (ECMP) [12] are commonly applied for network routing in MPTCP. However, subflows in an MPTCP connection are coupled, so MPTCP hopes that multiple paths between hosts are disjointed and matched, but random-based approaches are not designed specially for MPTCP and may end up with using the same paths for distinct subflows, which will introduce avoidable bottlenecks and waste a large number of available network resources. Moreover, since traditional routing algorithms have very limited knowledge of network states, the connections are generally forwarded through the same paths, even though there are multiple paths available in the network. Consequently, the connections cannot fully benefit from multiple subflows provided by MPTCP and the performance improvement remains limited.

To solve these problems, cross layer optimization techniques are applied, the network information can be provided by the network layer, e.g. software defined networking (SDN) [13]. SDN separates control planes and data planes in communication systems and offers a complete view of the network to applications and enables network programmability through flexible rules. These rules can be used to ensure that the flows follow the optimal paths in the network [14]. MPTCP implementations can be integrated with SDN to achieve higher throughput, load balancing, security, and reliable transmission that improve transmission quality. In [15], the author improved the video quality combing MPTCP and SDN, and in [16], the network fairness for MPTCP is ensured using SDN.

Despite owning the knowledge and control of the network by SDN, network modeling techniques are significant especially in terms of finding the optimal routes in complex real-world networks. However, it is difficult to find out the optimal routes through traditional algorithms such as optimization-based approaches or mathematical modeling, because routes and link features will cause nonlinear influence to the performance of MPTCP connections. Since machine learning methods can ascertain the potential nonlinear relationships among features, they have been widely applied nowadays. In [17], a novel Graph Neural Network (GNN) model is proposed to predict the delays and jitters in a network. In [18], a scheduler for MPTCP is proposed based on Deep Reinforcement Learning (DRL) in heterogeneous networks. In [19], LSTM (Long Short Term Memory) and DRL are implemented to improve the congestion control of MPTCP connections. Unfortunately, the above machine learning network modeling methods can only work in specific scenarios, and they have to learn and update their models constantly in a new environment.

Because of coupled congestion control algorithms and schedulers among subflows in MPTCP, existing network modeling techniques have limited performance, which will further affect routing decisions. To understand the complicated relationships among links, paths, subflows and the MPTCP connection on various topologies for routing optimization, in this paper, a novel GNN based cross layer routing optimization system for MPTCP using SDN is proposed. Our contributions are listed as follows:

- We analyze the routing problems when connections are extended from TCP to MPTCP. First, coupled congestion control algorithms and schedulers make the network behavior dissimilar with pure TCP situations, besides, the number of subflows, asymmetry among subflows, overlapped routes will cause nonlinear influence to MPTCP connections, thus making network modeling and routing decisions more complicated. So that existing network traffic models are not able to predict network performance correctly and make multipath routing fail to satisfy transmission requirements.
- We model the multipath routing problems into graph problems and apply a novel GNN architecture to explore the complicated relationships among links, paths, subflows, and the MPTCP connection on various topologies based on the interactions among them. Evaluation results prove that while maintaining high accuracy in throughput predictions, our model can generalize over arbitrary topologies and connections that are not seen during training, which is sufficient to be the guidance of multipath routing decisions.
- We propose GCLR, a cross layer multipath routing optimization system, by combining the GNN model and SDN. In GCLR, SDN can provide a global view of the whole network and the further control of the MPTCP connection. According to the prediction of the GNN model, optimal routes can be determined from the multipath route candidate set, and finally, the corresponding routing rules will be sent to switches. The GCLR is evaluated by network simulator Mininet and SDN controller Floodlight, simulation results show that it can achieve significant enhancement in connection throughput compared with traditional routing algorithms.

The rest of this article is structured as follows. In Section II, related works including the analysis of multipath routing, the state of art cross layer optimization methods for MPTCP, and GNN are introduced. In Section III, we present the network modeling methods for multipath routing using graph theory and propose the GNN model, in IV the design of our GNN based cross layer multipath routing optimization system for MPTCP is presented using SDN. In Section V, the performance of our proposed model and system is evaluated. Finally, Section VI concludes the paper.

II. RELATED WORKS

In this paper, we aim at improving the performance of MPTCP connections by optimizing multipath routing, so in this section, we will first list some problems in multipath routing, then, the state of art cross layer optimization techniques and the graph neural network will be presented.

A. TROUBLES IN MULTIPATH ROUTING

Although MPTCP can achieve throughput improvement under ideal network conditions, the realistic network environment is complicated and changeable which makes it difficult to reach the theoretical throughput. The structure of network topology and the heterogeneity among links and paths will introduce troubles in multipath routing, and the difficulties are listed above:

Firstly, routing should control the number of subflows in an MPTCP connection. Determining the optimal number of subflows is crucial to achieve the best performance of MPTCP, an optimal subflow number can bring throughput improvement without overwhelming network resources, and makes MPTCP more adaptive to maximize the utilization of network resources, while too many subflows will introduce extra transmission and control overhead on a network without significant throughput improvement, and also wastes network resources (like Ternary Content

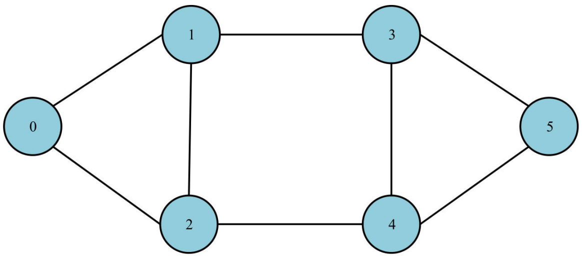


FIGURE 1. A 6-node network example for multipath routing.

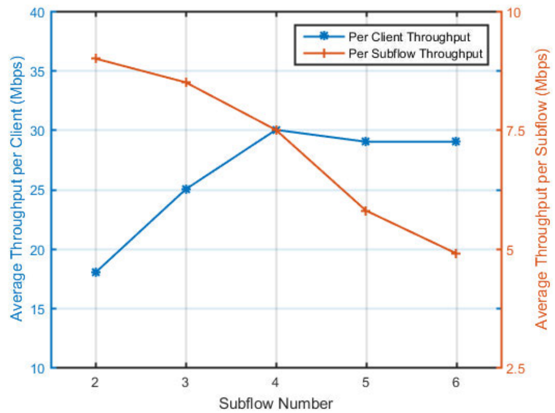


FIGURE 2. Per client and per subflow throughput with different subflow numbers.

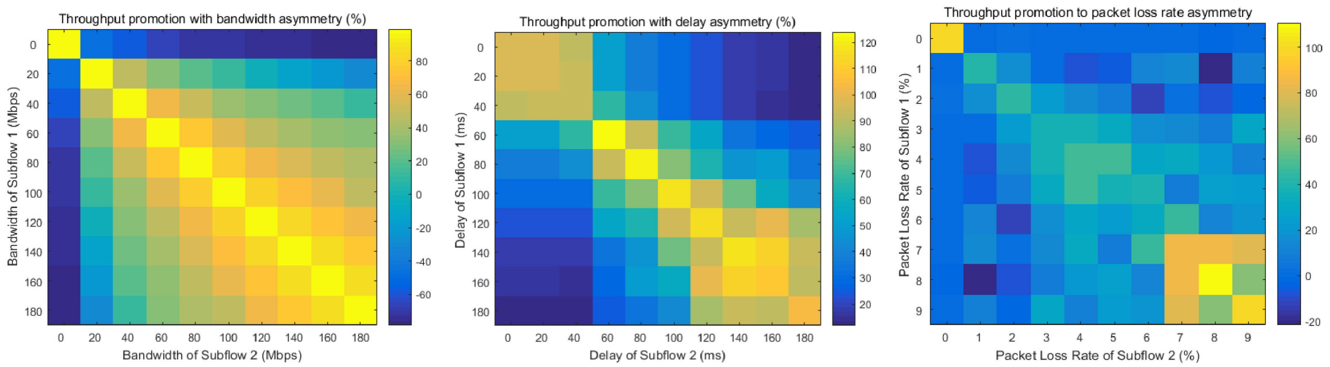
Addressable Memory, TCAM) by installing additional flow rules. On the other hand, the insufficient number of subflows cannot fully utilize network resources to reach the theoretical throughput. Take a 6-node network topology in fig.1 as an example, there are 6 hosts in the network, when the transmission request from node0 to node5 arrives, since MPTCP v0.90 multiple subflows can be created for each pair of IP addresses so that 8 subflows can be created, even before MPTCP v0.90, when only one subflow for each pair of IP addresses is permitted, there can be 4 pairs of IP addresses and thus 4 subflows can be established. But obviously, 2 subflows are enough to make full use of bandwidth in this network, which can be 0-1-3-5 and 0-2-4-5. Another example is shown in fig.2, it shows the per subflow throughput and per client throughput with increasing the number of subflows in another network topology. With the increase of subflow number, the traffic can be shared by each subflow, so that the per subflow throughput continues to decrease. However, the total throughput (per client throughput) reaches the highest when the number of subflows approaches the optimal value and starts to fall back with a further increase of the subflow number. It shows that in the topology of fig.2, 4 is the optimal subflow number to avoid the insufficiency or overflow of the transmission capacity.

Secondly, routing should avoid overlapping links traversed by subflows. MPTCP aggregates higher bandwidth by exploring network resources of different paths with multiple subflows. The overlapping link between subflows may very likely end up with becoming a bottleneck link that limits both subflows [20]. In a bottleneck link, bandwidth has to be shared by both subflows even though one subflow can already reach the same throughput, so that using only one of the two subflows is enough, the second subflow will introduce additional transmission and control overhead. Take the topology in fig.1 as an example, if the first subflow uses path 0-1-3-5, to avoid overlapping links, path 0-2-4-5 will be better than any other possible path because the bandwidth can be completely utilized while there is be no conflict between the two subflows. Otherwise, for example, if the second subflow chooses path 0-1-2-4-5 to route, there will form a bottleneck on link 0-1, the two subflows will compete the bandwidth on link 0-1 while link 0-2 keeps idle. The discovery and avoidance of shared bottleneck links is an important issue in MPTCP, the introduction of cross layer optimization based on SDN can effectively solve this problem with the help of the network layer information.

Thirdly, routing should consider the match of subflows. Dissimilar with regular TCP flows, coupled congestion control algorithms and schedulers in MPTCP make the performance of subflows interact with each other. If the difference among subflows is too large, the poorer subflow will drag down the performance of the better subflow, and ultimately affect the overall performance, so that the capability of subflows should be on similar levels. In fig.3, we studied the influence of subflow asymmetry in bandwidth, delay, and packet loss rate. For each experiment, a 2-subflow MPTCP connection with different subflow parameters will be created, and the throughput of the connection tp_m is measured, then 2 corresponding TCP connections are generated separately to get throughput tp_{s1} and tp_{s2}, finally, throughput promotion is defined as $tp_m/\max\{tp_{s1}, tp_{s2}\}$. According to the value of the throughput promotion, it is painted in different colors. The influence of bandwidth asymmetry is shown in fig.3(a). When the capacity of two subflows is symmetric, the throughput promotion reaches near 100% (colored in yellow). But when the gap between two subflows starts to increase, the throughput promotion has a drastic drop, more seriously, it even drops below 0 (colored in blue), which means that the additional subflow in MPTCP brings negative gain to the connection. This is mainly because MPTCP needs to reassemble the out-of-ordered data into a complete and orderly one at the receiver, but the subflow with larger bandwidth has to wait for the smaller one so that the total throughput is restricted. There exists the same situation in terms of delay asymmetry and packet loss rate asymmetry in fig.3(b) and fig.3(c), the performances of MPTCP connections with symmetric subflows are better than those with asymmetric subflows. The advantages of MPTCP decrease as subflow differences increase. So, routing should consider the match of subflows to avoid performance decrease in MPTCP due to subflow asymmetry.

B. CROSS LAYER OPTIMIZATION FOR MPTCP

The most common routing scheme of MPTCP is usually combined with randomized load balancing technologies such as ECMP [12], of which the limitation is that avoidable



(a) Subflow asymmetry in bandwidth (b) Subflow asymmetry in delay (c) Subflow asymmetry in packet loss rate

FIGURE 3. Throughput promotion with path asymmetry in terms of a) bandwidth, b) delay, c) packet loss rate.

bottlenecks may be created while massive network resource is idle.

Practical multipath routing algorithms are barely satisfactory because MPTCP is a transport layer protocol and the MPTCP stack has very limited knowledge of the underlying network to adopt efficient routing decisions. However, the advantages of MPTCP cannot be fully utilized by relying only on the traditional transport layer information, and useful network information on other layers need to be effectively used for an efficient multipath routing. For this reason, in recent years, related research has begun to combine MPTCP with the relevant information from other layers to effectively enhance its performance by cross layer optimization technologies.

In this paper, we focus on cross layer optimization at the network layer. For MPTCP, network layer information can be utilized mainly in subflow management and subflow routing, subflow management includes the evaluation and control of existing subflows, and subflow routing determines the route of each subflow, they all contribute to the performance enhancement of MPTCP. In order to get the connection with the network layer, the cross layer optimization for MPTCP is often combined with network layer techniques such as SDN [13], [14], [21], MPLS (multi-protocol label switching) [22], [23], LIPS (locator/identifier separation protocol) [24], [25], or source routing [26]. To fully utilize network resources for MPTCP, researches have been studied in different aspects. In [27], the author adopted a naive algorithm to avoid the overlapping among subflows. It first calculates the shortest path and deletes it from the topo, and then recalculates the shortest path for the second subflow. This idea is also applied in [26], but they only consider the overlapping between subflows and neglect the capacity of links, though there won't form a bottleneck on links with sufficient bandwidth. Further, algorithms starts to consider the traffic on subflows, the author of [28] prefers paths with slighter traffic and assigns them to the route of subflows. And in [14] it chooses subflows with high bandwidth and small differences for data transmission. However, as we mentioned before, coupled congestion control algorithms and schedulers make subflows interact with each other and make MPTCP dissimilar with traditional TCP, so a comprehensive network model is urgent for the guidance of multipath routing, and this is what we've done in this paper.

Cross layer optimization methods have also been applied at other layers. For example, at the physical layer and link layer, cross layer optimizations are considered from the perspective of the perception of link on-off status [29]–[31], the prediction of link parameters [32], and the coordination of network resources [33], [34]. And as for the application layer, the perception for deadline and importance [35]–[37], the perception of distortion rate [38]–[41], and file size [42] are usually considered.

C. GRAPH NEURAL NETWORKS AND MESSAGE PASSING NEURAL NETWORKS

As mentioned before, it is difficult to model the MPTCP network through traditional algorithms because there exists nonlinear relationships among links, paths, subflows, and the MPTCP connection. Existed machine learning techniques can only work in trained environments, and they have to learn and update their models constantly in a new environment. In this paper, the graph neural network is adopted in network modeling, GNN is good at solving problems related to graphs, and multipath routing can be abstracted as graph problems, so GNN is a best choice for solving multipath routing problems.

Firstly, a network consists of hosts and links and can be represented as a topology graph. Then, the relationships, between links and paths, and between the MPTCP connection and subflows, are coupled. Factors interact with each other and can be represented as a relationship graph. Therefore, multipath routing for MPTCP can be abstracted as optimization problems of graphs.

The graph structure has the ability of relational expressions, and the complex relationships among nodes in graphs pose a huge challenge to traditional machine learning algorithms. The convolutional neural network (CNN) has achieved great success in the Euclidean domain. But it can only operate on regular data such as arrays and matrices.

However, more and more application data belongs to the non-Euclidean domain and needs to be represented by graphs, which makes CNN stretched, so a neural network aiming at learning graphs is crucial for solving complicated graph problems.

Under the impetus of deep learning technologies, researchers propose the graph neural network [43], which combines the thoughts of convolutional networks, cyclic networks, and deep auto-encoders. Now GNN has been widely used in non-Euclidean domains such as applied physics, molecular chemistry, and knowledge maps. Compared with traditional solutions (e.g. CNN), the advantages of GNN in processing graph structure data are listed as follows [44]:

- **Input order**. Nodes in graphs are unordered, but CNN superimposes the features of nodes in a specific order and its inputs are sequential. So the inputs of nodes in different orders in CNN will result in different output results, which is contrary to the characteristics of graphs. Therefore, CNN has to traverse all the input orders of nodes, which leads to additinal overhead. As for GNN, it is designed for the structure of graphs, features can be propagated separately on each node, so that the input order of nodes can be ignored, and the inputs in different orders can come to the same output result.
- **Learning object**. In graph structures, an edge represents the dependency information between two nodes, for CNN, such dependency information is only regarded as the feature of two nodes, and only the characteristics of two nodes can be learned. However, for GNN, the information of edges can be propagated along with the graph structure, that is to say, the GNN model can learn the structural features of a graph, which is called the hidden state in GNN. Such structural feature is the key feature in graphs.
- **Generalization ability**. Generalization ability is an important topic of high-level artificial intelligence, the reasoning process of the human brain is based on the graph extracted from daily experience. CNN is able to generate synthetic images and documents by learning the distribution of data, but still cannot learn reasoning graphs. However, since GNN learns the structural features of graphs, it can generate new graphs from unstructured data such as scene images and story documents, which makes higher-level artificial intelligence neural network models possible. In this paper, the generalization ability of GNN in terms of connection arbitrary and topology arbitrary for multipath routing is utilized.

Message Passing Neural Network (MPNN) [45] is a GNN framework, MPNN updates node representations by stacking multiple graph convolutional layers using synthesis methods. MPNN consists of two phases, which are the message passing phase and the readout phase. During the message passing phase, T times of spatial graph convolutions are performed, and hidden states of nodes are iterated through message function $M_t()$ and update function $U_t()$:

$$h_v^t = U_t(h_v^{t-1}, \sum_{w \in N(v)} M_t(h_v^{t-1}, h_w^{t-1}, e_{vw})), \tag{1}$$

where e_{vw} represents the characteristics from node v to node w. New states are obtained by message function $M_t()$, and then, the real states are updated by update function $U_t()$. The readout phase is indeed a pooling operation that computes the feature vector of the entire graph, which is defined as

$$y = R(h_v^T | v \in G). \tag{2}$$

MPNN was originally applied to the field of quantum chemistry to learn the relationships between molecular structures and molecular properties to predict the properties of unknown molecules [45]. Later, MPNN was also applied to delay and jitter prediction in the network [46].

III. NETWORK MODELING

The MPTCP connection can be modeled by graphs to ascertain the complicated relationships among links, paths, subflows, and the MPTCP connection, in this section, the method of network modeling and the proposed GNN model will be introduced in detail.

A. NETWORK MODEL

The network topology can be abstracted by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, which consists of a set of nodes $\mathcal{N}$ and links $\mathcal{L}$, where nodes $n_i \in \mathcal{N}$, represent for hosts, routers, and switches in the network, and links $l_i \in \mathcal{L}$, represent for physical transmission links. In this work, in order to build a comprehensive model for multipath routing, bandwidth bw_i, delay de_i, and packet loss rate lo_i are considered as link properties. Path p_i is a sequence of links that connects 2 hosts (the server and the client), which can be expressed as

$$p_i = (p_i^1, p_i^2 \ldots, p_i^{|p_i|}), \tag{3}$$

where p_i^j is the index of the jth link in path i, and $|p_i|$ is the number of links in path i. The bandwidth, delay and packet loss rate of path i can be expressed as bw_{p_i}, de_{p_i} and lo_{p_i}, separately.

The MPTCP connection c consists of several subflows $s_i \in \mathcal{S}$, traditional TCP is a special case of MPTCP and can be regarded as an MPTCP connection with only one subflow, so the model in this paper also works when routing for traditional TCP connections. Subflow s_i has a corresponding path p_i with it, and their subscripts are consistent. The throughput of connection c is notated as tp_c.

B. RELATIONSHIPS IN GRAPHS

In this paper, in order to build up a comprehensive model for multipath routing, three main features, bandwidth, delay, and packet loss rate of a link will be considered in a network topology. Take path $p_i = (p_i^1, p_i^2 \ldots, p_i^{|p_i|})$ as an example, since p_i is the route of subflow s_i, the delay of s_i is the sum

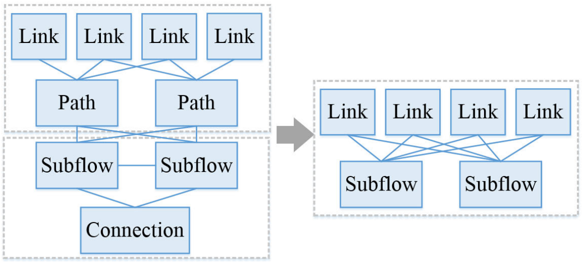


FIGURE 4. The mutually influential relationships in multipath routing.

of each link that s_i passes, which can be expressed as

$$de_{p_i} = \sum_{j=1}^{|p_i|} de_{p_i^j}. \tag{4}$$

And it's the same to bandwidth and packet loss rate, the bandwidth is determined by the minimum value of bw_i^j, which is

$$bw_{p_i} = \min(bw_{p_i^j}), \tag{5}$$

and the packet loss rate follows the multiplication rule, which is

$$lo_{p_i} = 1 - \prod_{j=1}^{|p_i|}(1 - lo_{p_i^j}). \tag{6}$$

As for the MPTCP connection, the throughput of the MPTCP connection is equal to the sum of throughput of all the subflows in this connection

$$tp_c = \sum_{s_i \in c} tp_{s_i}. \tag{7}$$

From all the above, we can conclude from the graph structure of network topology that the state of a path is determined by all the links that this path passes. In turn, on the other hand, the state of each link depends on all the paths passing through that link.

For the two graphs, due to the data sequence number mechanism shared by all MPTCP subflows and the coupled transmission control algorithms, we can acknowledge that the state of each subflow in an MPTCP connection is determined by the paths that each subflow passes, which is

$$tp_{s_i} = f(bw_{p_1}, \ldots, bw_{p_{|c|}}, de_{p_1}, \ldots, de_{p_{|c|}}, lo_{p_1}, \ldots, lo_{p_{|c|}}), \tag{8}$$

where $|c|$ is the number of subflows of connection c.

The mutually influential relationships in multipath routing can be represented by fig.4. When we hide paths and connection in the graph, we can get:

- The state of all subflows in an MPTCP connection is determined by the state of all links that each subflow passes.
- The state of all links in the network topology is determined by all the subflows in the MPTCP connection.

Define the hidden state of a link as h_{l_i}, and in correspondingly way the hidden state of a subflow is defined as h_{s_k}. We expect that link state vectors contain link information such as delay, packet loss rate, link utilization, etc,

Algorithm 1 MPTCP Network Model for Multipath Routing

Input: x_s, x_l
Output: h_s^T, h_l^T, o

# Initial phase
for s in S **do**
  $h_s^0 = [x_s, 0, \ldots, 0]$
end for
for l in L **do**
  $h_l^0 = [x_l, 0, \ldots, 0]$
end for
# Message passing phase
for $t = 1 : T$ **do**
  for s in S **do**
    for l in L **do**
      $h_s^t = RNN_t(h_s^t, h_l^t)$
      $\tilde{m}_{s,l}^{t+1} = h_s^t$
    end for
    $h_s^{t+1} = h_s^t$
  end for
  for l in L **do**
    $m_l^{t+1} = \sum \tilde{m}_{s,l}^{t+1}$
    $h_l^{t+1} = U_t(h_l^t, m_l^{t+1})$
  end for
end for
# Readout phase
$o = O(h_s^T, h_l^T)$

and path state is expected to contain the information of end to end connection parameters like RTT, throughput, packet loss rate, etc. Finally, the relationships can be formulated in a mathematical way as

$$\begin{aligned} h_{l_i} &= f(h_{s_i}, \ldots, h_{s_{|k|}}), \\ h_{s_k} &= g(h_{l_1}, \ldots, h_{l_i}), \quad l_i \in \mathcal{S}, \end{aligned} \tag{9}$$

where f and g are some functions that describe the underlying relationships among links and subflows.

C. THE ARCHITECTURE OF GNN MODEL

In the GNN model, information dissemination and output are the key steps to obtain the hidden states of nodes and links, so that different variants of GNN vary in aggregators and updaters. Among these GNN variants, the message passing neural network architecture is applied to solve the multipath routing problems. The step-by-step process of the graph neural network model is shown in algorithm.1.

The input of the model includes link information x_l from network topology and subflow information matrix x_s from the MPTCP connection. First, the overlap of subflows should be considered in multipath routing, so we number all the links in the network topology uniquely and record their features in link information x_l:

$$x_l = \begin{vmatrix} bw_1 & bw_2 & \ldots & bw_{|\mathcal{L}|} \\ de_1 & de_2 & \ldots & de_{|\mathcal{L}|} \\ lo_1 & lo_2 & \ldots & lo_{|\mathcal{L}|} \end{vmatrix}, \tag{10}$$

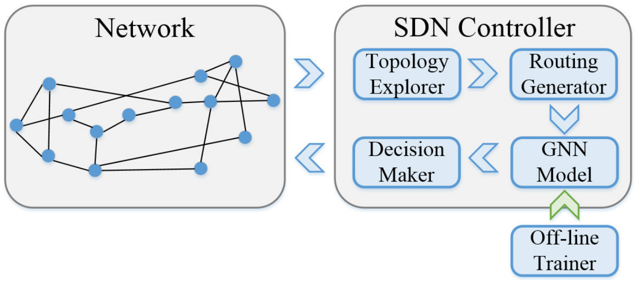


FIGURE 5. The SDN architecture of GCLR.

where $|\mathcal{L}|$ is the mode of $\mathcal{L}$, $\mathcal{L}$ is the number of links in the network topology. Then, as for the subflow information x_s, since the number of subflows in an MPTCP connection and the number of links that a subflow passes are variable, they are set as parameters.

In the MPTCP network model for multipath routing, line 2-7 is the initial process of the hidden state h_s and h_l, they are initialed with current value following with zeros. From line 9 to line 21 is the T times iteration of the message passing phase, since the training process is operated off-line, the number of training samples and iteration steps can be sufficient so that the model can be completely converged. Line 11-14 and line 17-20 are the message passing phase of links and subflows, separately. Information is aggregated through a recurrent neural network (RNN) for the iteration of subflows and through summation for links. For links, the order of subflows does not matter. But for subflows, sequential dependence between links in the same subflow caused by losses requires sophisticated message aggregation. So we use RNN here for it can record input sequence information. Since the state of subflows is determined by all links that these subflows pass, line 11-14 input the whole link set cyclically. Likewise, in line 18, all the subflows are summed. Line 13 and line 19 operate the update function from step t to $t + 1$, and line 23 is the readout phase of MPNN, $O()$ is a multi-layer perceptron with appropriate activations, it can finally calculate the feature vector of the graph based on the hidden state h_s^T and h_l^T.

IV. SYSTEM DESIGN

The architecture of the GNN based cross layer optimization for MPTCP by routing is shown in fig. 5. Based on the SDN architecture, the system is composed of the real network and the SDN controller. Several modules are designed in the SDN controller to realize multipath routing optimization with the help of the GNN model.

SDN is deployed among switches in data centers, it separates control planes and data planes to enable network programmability through flexible rules. Moreover, the availability of SDN has been extended to routers in WANs. In this paper, SDN is applied to provide a global view of the whole network and the further control of connections using the OpenFlow protocol [47], [48]. In the SDN controller, four modules are designed for multipath routing, they are the topology explorer module, routing generator module, GNN model, and decision maker. The role of these modules will be introduced in detail.

A. TOPOLOGY EXPLORER & DECISION MAKER

The topology explorer and decision maker are the bridges that connect the real network and the SDN controller. They are responsible for the information interaction between the network and the SDN controller.

Since network states change over time, the role of the topology explorer is to maintain the link information matrix x_l. Firstly, topology explorer is applied to collect and update various networking feedback from switches and routers in real-time, then x_l and the transmission requests will be sent to the routing generator for further predictions and decisions. The decision maker is for sending routing instructions back to the network. In the last step, based on the prediction of various routing strategies by the GNN model, decision maker chooses the optimal one for multipath routing and sends the configurations to the routers and switches in the network by flow table.

B. ROUTING GENERATOR

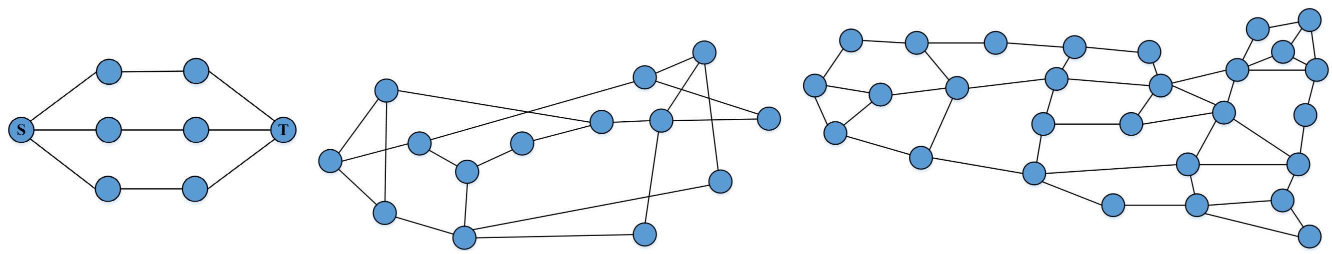
With the GNN model, the performance of specific routing results can be accurately predicted, but since the number of feasible routes increases exponentially as network size increases, it is unrealistic to iterate over all possible routes. So, it is significant for us to cover the optimal or near-optimal solution with a small routing candidate set. The routing generator is a key module in the SDN controller to generate the routing candidate set for multipath routing. The routing generator provides subflow information x_s for the model.

When there comes a transmission request from the network, according to the network state updated by the topology explorer, the routing generator generates a routing candidate set of the two hosts based on classic routing algorithms, greedy algorithms, and random algorithms. In the routing generator module, the number of subflows should be a parameter to test the optimal subflow number of the MPTCP connection. During routing generation, greedy algorithms are used to search for an optimal route while controlling the size of the routing candidate set, random algorithms are applied to maintain the diversity of the candidate set to avoid the algorithm getting trapped in a local optimum. The routes in the candidate set will then send to the GNN model to predict their corresponding expected performance. Benefit from GNN's fast single calculation time (around 1 ms), the number of routes in the candidate set can be large enough to cover potential optimal solutions.

C. GNN MODEL & OFF-LINE TRAINER

The GNN model is the core of GCLR. Since the GNN model has a strong generalization ability and can deal with most never-before-seen scenarios, it is divided into the off-line training stage and the on-line predicting stage.

The GNN model has the generalization ability in terms of connection arbitrary and topology arbitrary, so our off-line trained model can keep high performance when facing never-before-seen scenarios. For example, the prediction accuracy



(a) Directly Connected Network (b) National Science Foundation Network (c) US Backbone Network

FIGURE 6. Network topology for the evaluation of GNN model and multipath routing.

remains high when the characteristics of test samples (e.g. network topology, number of subflows) are not included in the training set. So, a sufficient number of training samples can satisfy most scenarios at the on-line predicting stage. At the off-line training stage, samples in different topology, by different routing algorithms, and with different subflow numbers are collected to cover most application scenarios. Then, these samples are divided into a training set and a test set, the test set does not participate in the training of the model, and is only used to calculate the performance trend of the model during the training process. In this work, the smoothed mean squared error (MSE) and the person correlation (ρ) is calculated. At the on-line predicting stage, routes in the candidate set will be sent to the GNN model to predict the corresponding expected throughput, then the results will be sent to the decision maker module.

V. PERFORMANCE EVALUATION

The evaluation of our works can be divided into GNN model evaluation and multipath routing evaluation. Firstly, we will introduce the simulation platform and data collection. Then, the GNN model is evaluated in terms of the accuracy of throughput prediction and the generalization in topology arbitrary and connection arbitrary. Finally, the cross layer multipath routing algorithm based on SDN is evaluated compared with traditional routing algorithms.

A. SIMULATION PLATFORM

To get MPTCP connection data and test the GNN model under different conditions, our data collection, model training, and performance validation are performed on a simulation platform. In this part, the simulation platform and the data collection process will be introduced.

The simulation environment is configured in a Linux server with Intel i7-4790k with 8 CPU cores, 16GB memory, and 300GB storage. In order to support the MPTCP protocol, the operating system is Ubuntu 14.04 LTS and installed with MPTCP kernel v0.89 as the most widely used MPTCP kernel version. The experimental network topologies are simulated by Mininet 2.2.2, and Floodlight v1.2 is used as the SDN controller in the cross layer optimization. The GNN model is implemented with Tensorflow in Linux userspace. The MPTCP kernel is modified so that it can communicate MPTCP-level information with the Linux userspace through kernel logs.

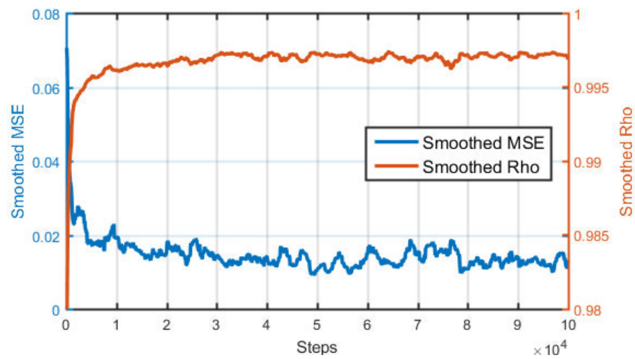


FIGURE 7. The training process of throughput prediction of the GNN model.

To ensure topological diversity, the directly connected network, the National Science Foundation Network (NSFNet), and the US Backbone Network in fig.6 are simulated by Mininet. Then, topology parameters x_l including bandwidth, delay, and packet loss rate are randomly set. Finally, we generate MPTCP connections by Iperf to collect training samples. In the MPTCP connections, communication peers, subflow numbers, and the route of each subflow x_s are randomly configured to simulate all possible scenarios. Both x_l and x_s are recorded as training samples.

B. EVALUATION OF GNN MODEL

GNN model is the most important component of GCLR, only if the model's prediction accuracy is high, the cross layer multipath routing optimization can achieve better performance. The GNN model is evaluated in terms of the accuracy of throughput prediction and the generalization ability in topology arbitrary and connection arbitrary.

For the evaluation of prediction accuracy, MPTCP samples are randomly divided into either the training set or the test set. The test set does not participate in the training of the model and is only used for performance evaluation at the training state. The MPTCP model has been trained for 100k steps and the convergence processes of MSE and ρ with the increase of training steps are shown in fig. 7. In fig.7, the smoothed MSE drops rapidly in the first 3k steps and turns into a steady state, while the ρ value following the opposite trend. After 100k steps, the smoothed MSE arrives at 0.016 and ρ reaches 0.994, so we can conclude that the GNN model can provide accurate predictions for different MPTCP connections and be the guidance for multipath routing optimization.

TABLE 1. The MSE of the GNN model in terms of subflow numbers and topologies.

	2 Subflows	3 Subflows
Directly Connected	0.003	0.004
NSFNet	0.014	0.016

TABLE 2. The validation of generalization ability (in MSE).

	Topology arbitrary	Connection arbitrary
Control group	0.016	0.016
Experimental group	0.022	0.034

Then, we analyzed the performance of the GNN model when training with samples with different subflow numbers (MPTCP connections with 2 and 3 subflows) and in different topologies (directed connected network and NSFNet), since MSE and ρ have similar but opposite trends, we only use MSE to show the performance of the model and the results are shown in table1. In table1, in the same topology, the performance of the GNN model slowly decreases as the subflow number increases. This is probably because that the effect of the subflow aggregation on MPTCP throughput follows similar rules, so that the subflow number has a small influence on the model. In terms of differences in topologies, the model has a very low MSE in the directly connected network, this is because it is a simple topology and has no overlapped subflows, so throughput predictions are much easier. As for complex topologies such as the NSFNet, there are more links in subflows, and various overlapped links exist, so the MSE is a little bit higher but still acceptable for multipath routing optimization.

To validate the generalization ability for topology arbitrary, samples in the NSFNet (fig.6(b)) are sent to the GNN model at the training stage and samples in the US Backbone Network (fig.6(c)) will only appear as the test set in the experimental group. The two network topologies are composed of similar network structures but they are not exactly the same. Therefore, if GNN has the ability to learn the structural characteristics of graphs, there will not be severe performance degradation in never-before-seen topologies (US Backbone Network). As for the generalization ability for connection arbitrary, the number of subflows in the MPTCP connection is considered. Samples of 4-subflow connections are set as the experimental group and will not be used while training, and then the throughput of them will be predicted to validate the generalization ability for connection arbitrary. The results are shown in table2, where samples in experimental groups are from US Backbone Network and 4-subflow connections, respectively. Although the results in experimental groups are not as good as that in control groups, they can still maintain MSE at a low level, which is sufficient for throughput prediction in multipath routing decisions. The result for connection arbitrary is not as good as that of topology arbitrary, this is because the increase of subflow number brings the increase in the input dimension, but there is no corresponding treatment for that in the GNN model.

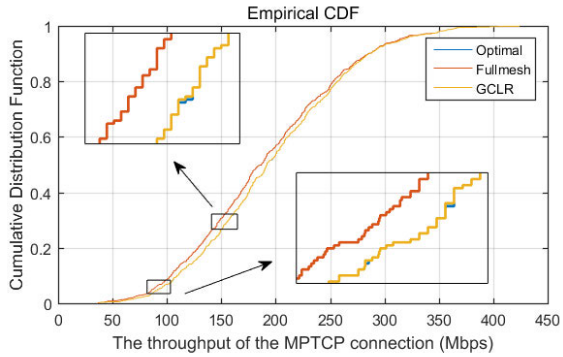


FIGURE 8. The throughput cumulative distribution function of GCLR, fullmesh, and the optimal value in the directly connected network.

C. EVALUATION OF MULTIPATH ROUTING

The evaluation of multipath routing (GCLR) also includes two parts.

Firstly, we studied the advantages of GCLR in subflow selection over the traditional MPTCP fullmesh algorithm. The fullmesh algorithm will exhaust all feasible subflows without considering the impact of subflow asymmetry on throughput. Experiments are performed on the directly connected network (fig.6(a)), link parameters are set randomly. We transmit data for 10 seconds by MPTCP and calculate the average transmission rate. The fullmesh algorithm selects all subflows for data transmission, but GCLR only establishes appropriate subflows according to the guidance of the GNN model. Finally, we compare the throughput difference among GCLR, fullmesh, and the optimal solution obtained by traversal. The results are shown in fig.8, we can see that due to the ability of GCLR to predict the throughput of different subflow combinations, the CDF curves of GCLR and the optimal value almost completely coincide, which exceed the traditional fullmesh algorithm a lot. It can be seen from the enlarged graphs in fig.8 that the performance of GCLR cannot reach the optimal value only at a few individual points, this is due to the prediction error of the GNN model. However, since the average error of the GNN model is small (0.004), even though the optimal combination of subflows is not selected, the throughput of GCLR will not differ much from the optimal value.

Then, the performance of the GCLR in more complex real-world topologies is evaluated. In complex topologies, the feasible multipath routes increase exponentially with the topology size, it is unrealistic to obtain the optimal solution for multipath routing. Therefore, we only compare the performance improvement of the GCLR with the traditional ECMP algorithm. Here, we choose the more representative NSFNet as the experimental topology. First, link parameters are set randomly. Then, transmission requests are created randomly. Finally, then we establish the MPTCP connections by GCLR and ECMP separately to compare their difference in performance. The results are shown in fig.9, GCLR has significantly improved performance compared with ECMP, with an average throughput increase of 14.57%.

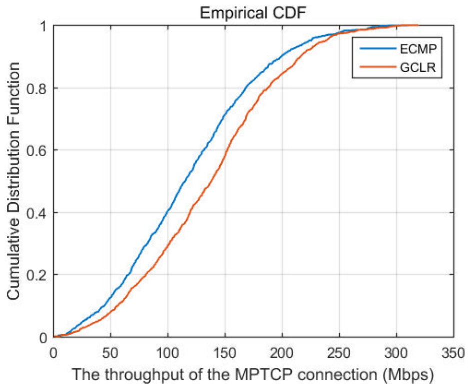


FIGURE 9. The throughput cumulative distribution function of GCLR and ECMP in the NSFNet.

Compared with ECMP, GCLR not only has the ability to control the number of subflows, but also can coordinate the routes among subflows. So GCLR can avoid network conflicts, make full use of network resources, and increase MPTCP throughput. The above experimental results have shown that this GNN based cross layer optimization system for MPTCP by routing can provide significant performance improvements to MPTCP connections.

VI. CONCLUSION

Leveraging the advantages of the GNN model that can learn the characteristics of graph structures, in this paper, we model routing problems as graph problems and propose a novel GNN based multipath routing model to explore the complications among links, paths, subflows and the MPTCP connection on various topologies. Then, based on the GNN model, the GNN based cross layer optimization system for multipath routing called GCLR is proposed. Evaluation results have shown that GCLR can achieve significant throughput enhancement in multipath routing optimization. At the same time, it has the generalization ability in terms of connection arbitrary and topology arbitrary.

REFERENCES

[1] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *TCP Extensions for Multipath Operation With Multiple Addresses*, document RFC 6824, Internet Engineering Task Force, 2013.

[2] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath TCP," in *Proc. ACM SIGCOMM Workshop Cellular Netw., Oper., Challenges, Future Design*, 2012, pp. 31–36.

[3] Y.-S. Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, and R. J. Gibbens, "Improving energy efficiency of MPTCP for mobile devices," 2014, *arXiv:1406.4463*. [Online]. Available: https://arxiv.org/abs/1406.4463

[4] G. Hampel, A. Rana, and T. Klein, "Seamless TCP mobility using lightweight MPTCP proxy," in *Proc. 11th ACM Int. Symp. Mobility Manage. Wireless Access*, 2013, pp. 139–146.

[5] C. Lee, S. Song, H. Cho, G. Lim, and J.-M. Chung, "Optimal multipath TCP offloading Over 5G NR and LTE networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 293–296, Feb. 2019.

[6] P. Dong, J. Wang, J. Huang, H. Wang, and G. Min, "Performance enhancement of multipath TCP for wireless communications with multiple radio interfaces," *IEEE Trans. Commun.*, vol. 64, no. 8, pp. 3456–3466, Aug. 2016.

[7] L. Ming, A. Lukyanenko, S. Tarkoma, and A. Ylä-Jääski, "MPTCP incast in data center networks," *China Commun.*, vol. 11, no. 4, pp. 25–37, Apr. 2014.

[8] C. Raiciu, C. Pluntke, S. Barre, A. Greenhalgh, D. Wischik, and M. Handley, "Data center networking with multipath tcp," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, 2010, p. 10.

[9] L. Wang, F. Zhang, J. A. Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "GreenDCN: A general framework for achieving energy efficiency in data center networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 1, pp. 4–15, Jan. 2014.

[10] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath tcp," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, 2011, pp. 266–277.

[11] R. Van Der Pol, S. Boele, F. Dijkstra, A. Barczyk, G. van Malenstein, J. H. Chen, and J. Mambretti, "Multipathing with MPTCP and OpenFlow," in *Proc. SC Companion, High Perform. Comput., Netw. Storage Anal.*, 2012, pp. 1617–1624.

[12] C. Hopps, *Analysis of an Equal-Cost Multi-Path Algorithm*, document RFC 2992, 2000.

[13] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[14] H. Nam, D. Calin, and H. Schulzrinne, "Towards dynamic MPTCP path control using SDN," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Jun. 2016, pp. 286–294.

[15] A. A. Barakabitze, I.-H. Mkwawa, L. Sun, and E. Ifeachor, "QualitySDN: Improving video quality using MPTCP and segment routing in SDN/NFV," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 182–186.

[16] 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, Sep. 2019.

[17] K. Rusek, J. Suárez-Varela, A. Mestres, pp. Barlet–Ros, and A. Cabellos-Aparicio, "Unveiling the potential of graph neural networks for network modeling and optimization in SDN," in *Proc. ACM Symp. SDN Res.*, 2019, pp. 140–151.

[18] H. Zhang, W. Li, S. Gao, X. Wang, and B. Ye, "Reles: A neural adaptive multipath scheduler based on deep reinforcement learning," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1648–1656.

[19] Z. Xu, J. Tang, C. Yin, Y. Wang, and G. Xue, "Experience-driven congestion control: When multi-path TCP meets deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1325–1336, Jun. 2019.

[20] L. Zongor, Z. Heszberger, A. Pašić, and J. Tapolcai, "The performance of multi-path TCP with overlapping paths," in *Proc. ACM SIGCOMM Conf. Posters Demos*, 2019, pp. 116–118.

[21] J. Pang, G. Xu, and X. Fu, "SDN-based data center networking with collaboration of multipath TCP and segment routing," *IEEE Access*, vol. 5, pp. 9764–9773, 2017.

[22] G. Armitage, "MPLS: The magic behind the myths [multiprotocol label switching]," *IEEE Commun. Mag.*, vol. 38, no. 1, pp. 124–131, Jan. 2000.

[23] I. Lopez, M. Aguado, D. Ugarte, A. Mendiola, and M. Higuero, "Exploiting redundancy and path diversity for railway signalling resiliency," in *Proc. IEEE Int. Conf. Intell. Rail Transp. (ICIRT)*, Aug. 2016, pp. 432–439.

[24] D. Saucez, L. Iannone, O. Bonaventure, and D. Farinacci, "Designing a deployable Internet: The locator/identifier separation protocol," *IEEE Internet Comput.*, vol. 16, no. 6, pp. 14–21, Nov. 2012.

[25] Y. Benchaïb, S. Secci, and C.-D. Phung, "Transparent cloud access performance augmentation via an MPTCP-lisp connection proxy," in *Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, May 2015, pp. 201–202.

[26] P. Du, X. Li, Y. Lu, and M. Gerla, "Multipath TCP over leo satellite networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 1–6.

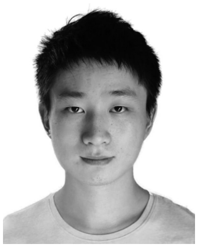
[27] M. Sandri, A. Silva, L. A. Rocha, and F. L. Verdi, "On the benefits of using multipath TCP and openflow in shared bottlenecks," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2015, pp. 9–16.

[28] J. Duan, Z. Wang, and C. Wu, "Responsive multipath TCP in SDN-based datacenters," in *Proc. IEEE Int. Conf. Commun.(ICC)*, Jun. 2015, pp. 5296–5301.

[29] Y. Lim, Y. C. Chen, E. M. Nahum, D. Towsley, and K. W. Lee, "Cross-layer path management in multi-path transport protocol for mobile devices," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1815–1823.

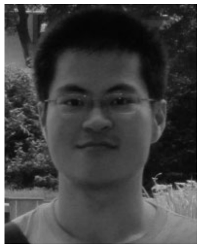
[30] H. Sinky, B. Hamdaoui, and M. Guizani, "Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4754–4764, Jul. 2016.

[31] J. Chung, D. Han, J. Kim, and C.-K. Kim, "Machine learning based path management for mobile devices over MPTCP," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 206–209.
[32] Y. Cao, C. Xu, J. Guan, Q. Wei, and H. Zhang, "Ant colony optimization based cross-layer bandwidth aggregation scheme for efficient data delivery in multi-homed wireless networks," in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Sep. 2013.
[33] S. R. Pokhrel, M. Panda, and H. L. Vu, "Analytical modeling of multipath TCP over last-mile wireless," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1876–1891, Jun. 2017.
[34] C.-L. Tsao, S. Sanadhya, and R. Sivakumar, "A super-aggregation strategy for multi-homed mobile hosts with heterogeneous wireless interfaces," *Wireless Netw.*, vol. 21, no. 2, pp. 639–658, Feb. 2015.
[35] C. M. Huang and M. S. Lin, "Partially reliable-concurrent multipath transfer (PR-CMT) for multihomed networks," in *Proc. Global Commun. Conf. (GLOBECOM)*, New Orleans, LA, USA, Nov./Dec. 2008.
[36] Y. Cao, Q. Liu, G. Luo, Y. Yi, and M. Huang, "PR-MPTCP+: Context-aware QoE-oriented multipath TCP partial reliability extension for real-time multimedia applications," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2016, pp. 1–4.
[37] X. Corbillon, R. Aparicio-Pardo, N. Kuhn, G. Texier, and G. Simon, "Cross-layer scheduler for video streaming over MPTCP," in *Proc. 7th Int. Conf.*, 2016.
[38] N. M. Freris, C.-H. Hsu, J. P. Singh, and X. Zhu, "Distortion-aware scalable video streaming to multinetwork clients," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 469–481, Apr. 2013.
[39] J. Wu, B. Cheng, C. Yuen, Y. Shang, and J. Chen, "Distortion-aware concurrent multipath transfer for mobile video streaming in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 688–701, Apr. 2015.
[40] J. Wu, C. Yuen, B. Cheng, Y. Yang, M. Wang, and J. Chen, "Bandwidth-efficient multipath transport protocol for quality-guaranteed real-time video over heterogeneous wireless networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2477–2493, Jun. 2016.
[41] J. Wu, B. Cheng, and M. Wang, "Improving multipath video transmission with raptor codes in heterogeneous wireless networks," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 457–472, Feb. 2018.
[42] M. Kheirkhah, I. Wakeman, and G. Parisis, "MMPTCP: A multipath transport protocol for data centers," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2016.
[43] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[44] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," 2019, *arXiv:1901.00596*. [Online]. Available: https://arxiv.org/abs/1901.00596
[45] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn. (JMLR)*, vol. 70, 2017, pp. 1263–1272.
[46] K. Rusek, J. Suárez-Varela, A. Mestres, P. Barlet-Ros, and A. Cabellos-Aparicio, "Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN," in *Proc. ACM Symp. SDN Res.*, 2019.
[47] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.
[48] W. Li, W. Meng, and L. F. Kwok, "A survey on OpenFlow-based software defined networks: Security challenges and countermeasures," *J. Netw. Comput. Appl.*, vol. 68, pp. 126–139, 2016.



TING ZHU received the B.S. degree in electronic information from the Huazhong University of Science and Technology, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree in electronic engineering and information science with the University of Science and Technology of China. His current interests include multipath communication, network protocols, and wireless network QoS.

XIAOHUI CHEN received the B.S. and M.S. degrees in communication and information engineering from the University of Science and Technology of China, Hefei, China, in 1998 and 2004, respectively. He is currently an Associate Professor with the Department of Electronic Engineering and Information System, University of Science and Technology of China. His current research interests include wireless network QoS, mobile computing, MAC protocol, and traffic model.



LI CHEN received the B.E. degree in electrical and information engineering from the Harbin Institute of Technology, Harbin, China, in 2009, and the Ph.D. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2014. He is currently a Faculty Member with the Department of Electronic Engineering and Information Science, University of Science and Technology of China. His research interests include energy efficiency in wireless communications and wireless optical communications. He received the Chinese Academy of Sciences President Award, in 2014.

WEIDONG WANG received the B.S. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1989, and the M.S. degree from the University of Science and Technology of China, Hefei, China, in 1993. He is currently a Full Professor with the Department of Electronic Engineering and Information Systems, University of Science and Technology of China. His research interests include wireless communication, microwave and millimeter, and radar technology. He is a member of the Committee of Optoelectronic Technology and the Chinese Society of Astronautics.

GUO WEI received the B.S. degree in EE from the University of Science and Technology of China (USTC), in 1983, and the M.S. and Ph.D. degrees in EE from the Chinese Academy of Sciences, in 1986 and 1991, respectively. He is currently a Full Professor with the Department of Electronic Engineering and Information System, USTC. His current research interests are wireless and mobile communications, wireless multimedia communications, and mmwave communication systems.