Machine Learning for End-to-end Congestion Control

Ticao Zhang and Shiwen Mao, *Fellow, IEEE*

***Abstract*—End-to-end congestion control has been extensively studied for over 30 years as one of the most important mechanisms to ensure efficient and fair sharing of network resources among users. As future networks are becoming more and more complex, conventional rule-based congestion control approaches tend to become inefficient and even ineffective. Inspired by the great success that machine learning (ML) has achieved in addressing large-scale and complex problems, researchers have begun to shift their attention from rule-based method to ML-based approach. This paper presents a selected review on the recent applications of ML to the field of end-to-end congestion control. In this survey, we start with a brief review of the relationship between congestion control and ML. We then review the recent works that apply ML to congestion control. These works either help the agent to make an intelligent congestion control decision or achieve an enhanced performance. Finally, we highlight a series of realistic challenges and shed insights on potential future research directions.**

I. Introduction

The rapid development of communication technologies has triggered the emergence of new network architectures, such as cognitive radio networks, data center networks, ultra dense heterogeneous networks, and millimeter-wave (mmWave) networks. Each network has its own features and performance requirements, which may change dynamically. The increasing capability of the network also enables a variety of new services and applications, e.g., augmented reality (AR), on-line gaming, edge computing, and autonomous driving, entailing more stringent requirements on the communication network.

The transport layer plays an important role in the management of end-to-end connections for upper layer services. The performance of emerging new applications depends heavily on the interactions between the underlying network and the transport layer. End-to-end congestion control, as a fundamental part of the transport layer protocol (TCP), ensures network stability and fairness in resource utilization. Today's TCP congestion control mechanism relies on the design that was created in the 1980s for wired networks. It uses a set of pre-defined rules, e.g., halving the congestion window (often referred to as CWND) when a packet loss is detected, and adjusting the CWND according to measured round-trip time (RTT). The same TCP design and its variants have been employed over the past three decades. Although these congestion control mechanisms achieve great success, they may not perform well in today's or future highly-dynamic and complex networks where the network performance is affected by a variety of factors. The congestion control problem can be modeled as an optimization problem, while conventional rule-based methods are mostly heuristics with no guarantee to solve the complex problem. They often lead to sub-optimal solutions which may suffer poor performance.

Recently, machine learning (ML) has made breakthroughs in a variety of application areas, such as speech recognition, computer vision, and robot control. ML can learn from the collected data or environment and build models. Also, with the recent development of computing infrastructures (e.g., GPU, TPU, and ML libraries) and distributed data processing frameworks, there is now an increasing trend in leveraging ML to deal with complex networking problems. For some tasks, such as regression, classification, and decision making, ML performs pretty well. Considering that these tasks play basic but vital role in networking problems, it is imperative to embrace ML techniques for potential breakthroughs in end-to-end congestion control.

In [1], the connection between artificial intelligence (AI) and network traffic control (NTC) is discussed. The author show that AI-NTC could be the next frontier of network research and deep reinforcement learning (DRL) would be a promising model. Reference [2] investigates how ML could benefit network design and optimization. A workflow for applying ML in the network domain is provided. Reference [3] provides an overview of the state-of-the-art in deep learning, a branch of ML, for intelligent network traffic control systems. The above works are focused on network traffic control with AI [1], networking with ML [2], and network traffic control with deep learning [3]. They do not provide a detailed review on the relationship between ML and the specific problem, i.e., end-to-end congestion control, which is the goal of this article.

In the rest of this article, we first survey the state-of-art congestion control algorithms and discuss the challenges. We then introduce the concept of ML and the motivations for ML-based congestion control. We next present the recent advances in ML-based congestion control, and conclude this article with a discussion of potential future directions.

II. Challenges and Opportunities

TCP is designed to provide reliable transmission of packets across an end-to-end connection, where congestion control is incorporated. A typical TCP congestion control works as follows. When started, the endpoint should quickly increase its sending rate to achieve high utilization of network resources. However, when congestion is detected, endpoints involved should reduce their sending rates; when congestion

T. Zhang and S. Mao are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA.
Digital Object Identifier: 10.1109/MCOM.001.1900509

is gone, endpoints should then increase their rates for high utilization of network bandwidth. Usually the rate keeps on increasing/decreasing, following the network congestion state. Detecting of network congestion is usually performed at network edge, without coordination and communication among the users, using loss or delay as indicators.

A summary of the major congestion control mechanisms and their pros and cons are provided in Table II in [4]. These schemes use some signals to detect congestion and then adjust the CWND or sending rate according to some predefined rules. Depending on the type of signals that are used as indicators of congestion, congestion control mechanisms can be categorized into several classes. For example, TCP variants Tahoe, Cubic, Reno, and New Reno are loss-based mechanisms. These mechanisms use packet loss to detect congestion. The additive increase multiplicative decrease (AIMD) algorithm is used to adjust the CWND. Loss may be a signal of congestion, but sometimes congestion happens before the bottleneck buffer is full. Early detection of congestion may help to avoid sharp decrease in throughput. Consequently, delay-based mechanisms such as TCP Vegas and Verus are developed, which use delay as an indicator of congestion. More recently, some hybrid mechanisms try to combine merits of the existing solutions. For example, Compound TCP, which is available on all Microsoft Windows machines, uses the sum of a delay-based window and a loss-based window as its congestion window. Veno, a hybrid between Vegas and Reno, adopts an explicit model of the bottleneck buffer occupancy. BBR, adopted by Google, estimates both bandwidth and RTT, aiming to keep CWND equal to the Bandwidth-Delay Product (BDP), which is the optimal operating point.

we have to point out that such rule-based mechanisms have several limitations:

- *Adapt to new networks*: Congestion control algorithms that are designed for one specific network may not apply to other types of networks. For example, TCP-NewReno is originally designed for wired links where packet loss is often interpreted as congestion. In wireless networks, packet loss may be caused either by link errors or congestion. The sender will always halve CWND even if the loss is caused by link errors. As a result, the link bandwidth will not be effectively utilized. Furthermore, the CWND update mechanism is not effective to adapt to various network topologies. For example, satellite network links where the RTT is large require a more aggressive CWND increase. Whereas, mobile ad hoc networks (MANET) where the BDP is low, may demand a more conservative CWND increase.
- *Learn from the past*: The rule-based approach uses a fixed set of rules to handle every situation. It does not leverage past experience and assumes no prior information such as link bandwidth, channel characteristics, and the number of flows on shared links. Suppose that NewReno can learn the link bandwidth information from past experience, it could speed up the sending rate more aggressively in the slow start phase to improve link utilization. Inability to learn from the past prevents endpoints to take actions proactively.

TABLE I
SUPERVISED VS. UNSUPERVISED LEARNING

	Supervised learning	Unsupervised learning
Training data	labeled	unlabeled
Discrete case	classification	clustering
Continuous case	regression	dimensionality reduction
Accuracy of results	high	less accurate
Number of classes	known	not known

- *Performance*: Today's network is becoming highly dynamic and complex. Ruled-based congestion control is designed based on human's understanding of the network. As human knowledge may not always accurately characterize the network features, the resulted congestion control mechanism may not be effective and can only achieve a sub-optimal performance.

III. WHY CONGESTION CONTROL WITH ML

ML is a subset of AI, where the machine (or, the agent) has the ability to accomplish a task when facing new data or new environment after a training process. Existing ML techniques generally fall into three categories: supervised learning, unsupervised learning, and reinforcement learning.

- Supervised learning: supervised learning uses a set of labeled samples to learn a mapping between the input and output spaces. Depending on whether the output is continuous or not, supervised learning can be categorized as *regression* and *classification*. Deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), naive Bayesian (NB), decision tree (DT), and support vector regression are typical techniques in supervised learning.
- Unsupervised learning focuses on classifying unlabeled samples into different clusters. It is mainly used for data dimension reduction in the continuous cases or clustering in the discrete cases. Restricted Boltzmann machine (RBM), autoencoder (AE), Gaussian mixture model (GMM), principal component analysis (PCA), and k-means clustering are usually used in unsupervised learning.
- Reinforcement learning (RL): RL is an environment-based approach where the agent is trained to solve decision making problems through interactions with the environment. Typical RL techniques include Q-learning, the actor-critic algorithm, and the deep deterministic policy gradient (DDPG) algorithm.

A comparison between supervised and unsupervised learning is given in Table I. The latest breakthroughs, such as deep learning, generative adversarial networks (GAN), and transfer learning, also have great potentials to deal with many complex problems such as end-to-end congestion control.

Conventional congestion control only considers several measurements such as packet loss and/or RTT as indicator of congestion. The decision making process all relies on these measurements and the pre-defined rule based on human's understanding of the network. The rule-based mechanism is

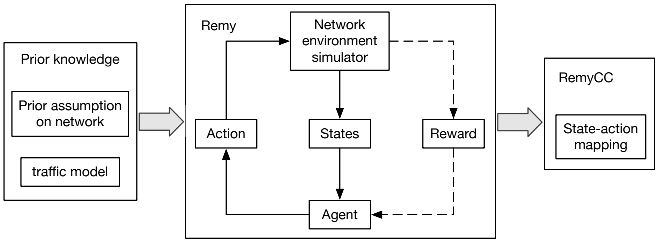

Fig. 1. An illustration of Remy congestion control (figure courtesey of [5]).

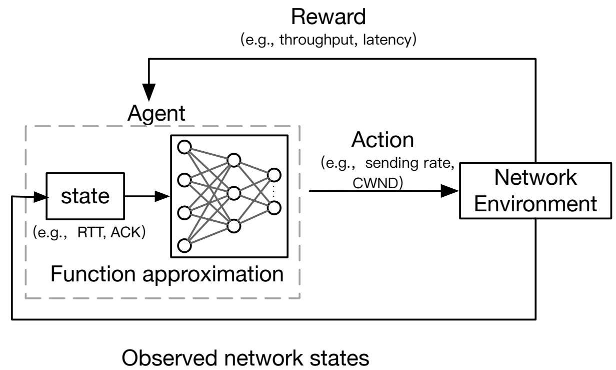

Fig. 2. Framework of DRL for congestion control.

more susceptible to many unpredictable factors, resulting in poor performance. ML, on the other hand, aims to construct algorithms or models that can learn to make decisions directly from the past experience or the network environment. It does not need accurate network models. Hence, it has the potential to outperform the rule-base mechanism.

Nowadays intensive computational resources are provided by both the central controller and edge servers. Cutting-edge technologies such as edge computing, network slicing, software-defined networking are changing the way how network traffic is managed. Moreover, the emergence of several dedicated libraries, such as TensorFlow, Caffe, and PyTorch, has greatly simplified the process of building an ML model. By leveraging these computational resources, ML-based congestion control is becoming feasible. In practice, a model can be trained with global information over a long time interval with the computation resources. The model parameters can be updated once upon a time. In the online deployment stage, the decision can be inferred using the trained ML model with a few computations.

IV. Recent Advances on ML-based Congestion Control

Over the past few years, there have been several works on ML-based congestion control. TCP Remy [5] is the first example, where the authors model congestion control as a decision-making problem under uncertainty. Each endpoint can make decisions on whether to send packets or not. As shown in Fig. 1, the network model is assumed to be Markovian. The network state consists of three variables: an exponentially-weighted moving average (EWMA) of the inter-arrival times of ACKs, an EWMA of the sending times of those same packets, and the ratio between the most recent RTT and the lowest measured RTT. The traffic is modeled as a stochastic process that switches unicast flows on or off between sender-receiver pairs. After observing the network state, the agent adjusts the CWND to achieve a balance between high throughput and low latency. This ML-based algorithm outperforms the human-designed end-to-end algorithms, including Cubic, Compound, and Vegas [5]. However, TCP Remy works well at the price of a stringent assumption on the network and traffic models.

Unlike the off-line training approach adopted in TCP Remy, Performance-oriented Congestion Control (PCC) [6] uses on-line training and does not make similar assumptions on the network model. In each micro-experiment, PCC chooses a sending rate and observes the selective ACK (SACK), which is used to measure the utility of an action (delivery ACK, loss, and latency). PCC runs several micro-experiments continuously and an empirically-optimal rate control policy is learned in an online manner, such that the utility can be maximized. PCC achieves 10 times performance improvement, with better fairness and stability, over TCP CUBIC. However, its performance over wireless networks is affected by bufferbloat and it has not been tested in real-world network environments.

A. Congestion Control with DRL

More recently, advanced learning algorithms such as RL and DRL have been incorporated into congestion control design. Just as human learns a skill from past experience, with RL the endpoint can learn the optimal congestion control policy from its observations of the network environment and past experiences. These mechanisms do not rely on pre-defined rules and exhibit a stronger ability in intelligently adapting to the changing environment. DRL embraces the advantages of DNN to train the learning process and to achieve scalability (over RL), thereby the learning speed and the learning ability can both be improved.

In congestion control, each endpoint adjusts its sending rate or CWND based on its observations of measured RTT, ACK, and so on. This can be viewed as a decision-making problem. Under uncertain and stochastic environments, the decision-making problem is usually modeled as a Markov decision process (MDP). The goal is to find an optimal policy for the MDP so that the expected cumulative reward can be maximized. As shown in Fig. 2, a DRL agent interacts with the network environment by adjusting its actions (e.g., the sending rate or CWND) based on observed network state (e.g., RTT, CWND, and the inter-arrival time of ACKs). A DNN is trained to map state to action, so that the reward metric (e.g., throughput and/or latency) can be optimized.

The recent works on DRL based congestion control are summarized in Table II. QTCP is the first work that leverages RL to design congestion control algorithms [7]. It helps the senders to gradually learn the optimal congestion control policy in an on-line manner, without requiring prior knowledge of the network model. Reference [8] presents a DRL-based scheme termed Aurora, which uses a fully connected DNN to learn the state-action pairs from stored historical data. This method is surprisingly robust to environments outside the scope of training. TCP-RL, proposed in [9], uses RL to dynamically configure the congestion control parameters for both short and long TCP flows when environment changes. The performance evaluation is based on a real implementation.

References [10] and [11] investigate the multi-path TCP (MPTCP) congestion control problem. A single agent is used in [10] to dynamically and jointly optimize congestion control for all active flows. Apart from incorporating a long short-term memory (LSTM) model in the DRL framework to better capture the network dynamics, reference [10] for the first time integrates the actor-critic mechanism into DRL for continuous congestion control. In [11], the authors focus on the congestion control performance degradation problem of MPTCP caused by path diversity. For practical deployment, the authors in [11] propose an asynchronous RL-based framework to decouple model training and execution. The congestion control rules is generated offline and applied for real-time online decision making of window adjustment. Reference [12] considers the initial congestion window (IW) selection problem in a mmWave network. A DRL-based online decision making approach is proposed to adjust the IW such that the flow completion time (FCT) is optimized.

B. Performance Enhancements

The congestion control problem has its own characteristics and is impacted by many factors. ML techniques, especially supervised learning and unsupervised learning, have high potential to classify packet losses and predict congestion related parameters for an improved performance.

1) Loss Classification: As mentioned before, the inability to distinguish packet loss caused by congestion and degraded channel quality in wireless networks will result in poor throughput. For next generation cellular networks where wireless links are more easily blocked, accurately identifying the cause of packet loss will be of vital importance to improve the congestion control performance.

Various ML models have been used to infer the cause of packet loss at network edge. For example, Reference [13] uses expectation-maximization (EM) to identify the packet losses caused by contention and congestion in optical networks. The classification helps to improve the TCP throughput. Reference [14] develops a loss predictor based congestion control mechanism with supervised learning for wireline networks. It achieves a better tradeoff of throughput and delay compared to NewReno. Such classifiers usually achieve a higher classification accuracy than non-ML approaches. TCP variants built upon these ML-based classifiers have been shown to outperform the standard rule-based solutions.

2) Congestion Prediction: Congestion prediction plays an important role in dynamic routing, congestion control, congestion avoidance, and proactive network management. In practice, when congestion actually happens, it has already affected the throughput performance significantly, and it may be too late to adopt further actions. If congestion or congestion related parameters, such as TCP throughput and RTT, can be accurately predicted, the sender can proactively respond to congestion. Currently, there are mainly two directions of research on estimation of congestion related parameters, i.e., formula-based and history-based. Formula-based approaches integrate the sender's measurements, such as RTT, packet loss rate, and CWND into a formula to generate predictions. However, timely gathering of such information is not easy especially in today's highly dynamic and complex networks. Moreover, the ever-evolving feature of TCP makes it hard to maintain an accurate formula-based model. History-based approaches refer to some time series analysis techniques, e.g., the exponential weighted moving average (EWMA) algorithm used by TCP for estimating RTT. However, time series analysis may also be inaccurate in some cases.

To deal with the limitations of conventional mechanisms, ML-based approaches (mainly supervised and unsupervised learning) have been adopted for congestion prediction. Reference [15] applies support vector regression (SVR), a useful ML tool for multivariate regression, to predict TCP throughput. Recently, Reference [14] develops a loss predictor using random forest, a supervised learning technique, to predict the probability of packet loss caused by congestion. This method can predict and reduce packet loss events, lower the frequency that sending rate is reduced, and achieve a higher throughput. These works all predict congestion related parameters from passive measurements with ML approaches, which have great potential in parameter prediction.

C. ML-based Congestion Control Workflow

As real network data is generally hard to obtain and label, we have to point out that RL, which does not require network data, will be a dominant model to implement ML-based congestion control. It has the potential to find the best decision based on *trial and error* and help the endpoints to quickly react to environment changes. Supervised learning and unsupervised learning can leverage historical data for potential performance improvement. For example, Reference [12] combines the merit of both DRL and supervised learning. The system is implemented with DRL for online-decision making. A supervised learning approach is utilized to extract features from the collected data during online learning for performance improvement.

Fig. 3 illustrates the generic workflow for implementing congestion control with ML. The problem is firstly formulated as a decision-making problem. Depending on the measured signals and the goal, the corresponding state, action, and reward should be properly defined. In the model training process, various training methods can be used to help the model to learn the best control policy through interactions with the environment. The training data can be collected, based on which features can be extracted with supervised/unsupervised learning for potential performance improvement. Finally, the model is ready to be deployed in a real environment.

V. Challenges and Outlook

ML exhibits great potential for end-to-end congestion control. However, most ML algorithms so far are optimized for

TABLE II
A SUMMARY OF ML-BASED CONGESTION CONTROL SCHEMES

Algorithm	States	Actions	Rewards	Pros	Cons
QTCP [7] (2018)	Sending interval, ACK interval, RTT	Select CWND with RL	Large throughput, low latency	1. Generalization to different network topologies, 2. Higher throughput than NewReno	Limited performance evaluation
Aurora [8] (2019)	Latency gradient, latency ratio, sending ratio	Adjust sending rate with DRL	Large throughput, low latency, low packet loss rate	1. Robust to environment, 2. Outperforms BBR, PCC-Vivace and Remy	Limited test on network changes
TCP-RL [9] (2019)	Throughput, RTT, loss rate	Adapt the initial window of connection and the congestion control algorithm	Large throughput, low RTT	1. Dynamical parameter configuration, 2. Improved TCP transmission performance	Untested in highly dynamic network conditions
DRL-CC [10] (2019)	Sending rate, RTT, RTT deviation, goodput, CWND	Select CWND with single agent DRL in MPTCP	Large goodput, fairness	1. Performance improvement in terms of goodput, 2. Flexible and robust, 3. TCP friendly	Large state space increases the complexity
SmartCC [11] (2019)	Sending rate, ACK interval, (subflow)	Select congestion control rules in MPTCP	Large throughput, low RTT, low jitter fairness	1. Addresses path diversity in HetNet, 2. Asynchronous learning framework to reduce overhead, 3. Performance improvement in throughput, RTT, e.t.c.	1. Does not consider TCP friendliness, 2. Lack real deployment test
IW-DRL [12] (2019)	FCT, RTT, throughput	Select the initial CWND with DRL	Low FCT	1. Algorithms converges stable and fast, 2. Adaptive on-line decision making, 3. Software-defined networking (SDN) implementation, 4. Flow completion time (FCT) reduction	Does not optimize congestion directly

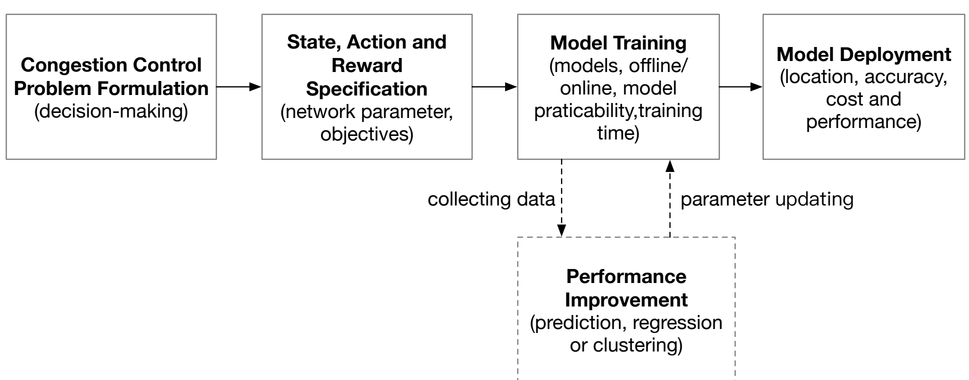

Fig. 3. Workflow of ML-based congestion control.

simple network topologies. There is no fully implemented and well-tested ML-based congestion control mechanisms in a real environment. We outline some potential future research directions here.

A. Real Data Collection

Collecting a large number of high quality data along with the network profiles is challenging. Most of the existing works rely on simulated dataset, which is generated based on a specific network type. Hence the resulted ML model may not be compatible with a practical system. For supervised learning, the collection of labeled data may be labor-intensive and costly. It could save a lot of repeated experiments if there are unified open-source datasets and a shared standard platform for researchers to compare their developed mechanisms. Moreover, due to the high test cost of large-scale networks and the difficulty in accessing these networks, simulators with high speed and high fidelity are in great demand.

B. Fairness, Robustness, and Generalization

ML-based congestion control might be unfair to existing solutions. If ML-based congestion control is trained in an environment where it competed with existing mechanisms, it might learn to occasionally cause packet loss to force other TCP protocols to back off, so that it can occupy more network resources. How to coexist properly with other existing protocols is challenging. Moreover, due to specific requirements of network systems, network protocols often require a worst performance guarantee. ML-based congestion control needs to be robust to the rapidly changing environment. Finally, the generalization ability of the ML algorithm is also required to ensure that the model can adapt to network dynamics without re-training the model every time the network condition changes.

C. Cross-layer Optimization

In ML, it is common to use the reward function as the optimization objective. Common reward functions can be maximizing throughput, minimizing latency, reducing loss, fairness, or a combination of these functions. Since the reward function can be chosen by the designer, some higher layer application metrics, such as users' quality of experience (QoE), can also be incorporated in the reward function design. For example, it is reported that by year 2020, more than 80% of the Internet traffic will be video related and much of it will be carried by TCP. Congestion control algorithms that aim to maximize users' QoE could be of great importance. Cross layer optimization is a promising research direction.

D. Deployment in Real Environments

Most of the existing DRL-based congestion control algorithms are tested in a simulated environment. They are not tested in a real-world environment. In the future, the rule table needs to be embedded to the operating system kernel and the ML model needs to be implemented as a system service. Then, a prototype can be developed, based on which, the adaptation ability of the congestion control algorithms can be tested under dynamic network conditions.

VI. Conclusions

Today's network is becoming more and more complex, and it is imperative to embrace ML techniques to design effective congestion control algorithms. Despite the advances made in recent works, there is still considerable room for improving the network performance by redesigning smart congestion control protocols with ML. There is still a long way to put ML-based congestion control into practice for some practical issues. In this paper, we provided a selected review of the recent advances on ML-based end-to-end congestion control. We also discussed open problems that need to be further investigated from both networking and ML perspectives.

Acknowledgments

This work is supported in part by the NSF under Grant ECCS-1923717.

References

[1] J. Xu and K. Wu, "Living with artificial intelligence: A paradigm shift toward future network traffic control," *IEEE Netw.*, vol. 32, no. 6, pp. 92–99, Nov. 2018.

[2] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," *IEEE Netw.*, vol. 32, no. 2, pp. 92–99, Mar. 2018.

[3] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrows intelligent network traffic control systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, Fourthquarter 2017.

[4] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3584–3608, Fourthquarter 2019.

[5] K. Winstein and H. Balakrishnan, "TCP Ex Machina: Computer-generated congestion control," in *Proc. ACM SIGCOMM'13*, Hong Kong, China, Aug. 2013, pp. 123–134.

[6] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting congestion control for consistent high performance," in *Proc. USENIX NSDI'15*, Oakland, CA, May. 2015, pp. 395–408.

[7] W. Li, F. Zhou, K. R. Chowdhury, and W. Meleis, "QTCP: Adaptive congestion control with reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 445–458, Jul. 2019.

[8] N. Jay, N. Rotman, B. Godfrey, M. Schapira, and A. Tamar, "A deep reinforcement learning perspective on internet congestion control," in *Proc. 2019 Int. Conf. Machine Learning*, Long Beach, CA, June 2019, pp. 3050–3059.

[9] X. Nie, Y. Zhao, Z. Li, G. Chen, K. Sui, J. Zhang, Z. Ye, and D. Pei, "Dynamic TCP initial windows and congestion control schemes through reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1231–1247, Jun. 2019.

[10] 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.

[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 J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2621–2633, Nov. 2019.

[12] R. Xie, X. Jia, and K. Wu, "Adaptive online decision method for initial congestion window in 5G mobile edge computing using deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2019.

[13] A. Jayaraj, T. Venkatesh, and C. S. R. Murthy, "Loss classification in optical burst switching networks using machine learning techniques: Improving the performance of TCP," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 45–54, Aug. 2008.

[14] Y. Kong, H. Zang, and X. Ma, "Improving TCP congestion control with machine intelligence," in *Proc. 2018 Workshop on Network Meets AI & ML*, New York, NY, USA, Aug. 2018, pp. 60–66.

[15] M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A machine learning approach to TCP throughput prediction," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1026–1039, Aug. 2010.

Ticao Zhang received the B.E. degree in 2014 and the M.S. degree in 2017 from School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China. He is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at Auburn university, Auburn, AL. His research interests include video coding and communications, and optimization and design of wireless multimedia networks.

Shiwen Mao [S'99-M'04-SM'09-F'19] received a Ph.D. in electrical and computer engineering from Polytechnic University, Brooklyn, N.Y. in 2004. He is the Samuel Ginn Professor and Director of Wireless Engineering Research and Education Center at Auburn University, Auburn, AL. His research interests include wireless networks, multimedia communications, and smart grid. He is a co-recipient of several best paper awards and the best demo award from IEEE conferences. He received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, NSF CAREER Award in 2010, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a Fellow of the IEEE.