Machine Learning for Internet Congestion Control: Techniques and Challenges

Lei Zhang, Yong Cui, Mowei Wang, Zhenjie Yang, and Yong Jiang
Tsinghua University

Internet congestion control (CC) remains a cornerstone issue in networking fields. It has attracted much research attention in academia, industry, and Internet standards organization. This article focuses on the machine learning (ML) technologies for Internet congestion control. Specifically, it summarizes the main reasons why network operators should apply ML in congestion control, surveys the latest advances of learning-based CC approaches, and explores challenges of standardizing CC with machine learning. This article provides two aspects challenges of learning-based CC that could motivate researchers to propose novel algorithms and develop standards of Internet CC with advanced ML techniques.

Internet CC is an important networking issue that the Internet Engineering Task Force (IETF) has been paying attention for more than 20 years.[1,2] As of now, research on CC can be divided into three phases. At the first stage, researchers proposed a CC scheme that all flows or users followed, and studied its effectiveness to deal with congestion. It may be called the "homogeneous CC paradigm." Subsequently, CC became the default deployment.[3,4] Many studies try to develop new schemes to improve CC and studied how these new schemes coexist with the default ones. It might be called the "competing CC paradigm." In the latest phase of CC study reviewed in this article, there is no assumption of what schemes are used by others; a flow is trying to learn how to survive well given other traffic. It might be called the "heterogeneous CC paradigm." For the first two phases, those schemes mostly deal with the complexity of network topology, the different number of flows, and their traffic demand/dynamics, which are already very complicated. For the third phase, it is considerably more complicated due to another dimension: how the other flows behave.

Digital Object Identifier 10.1109/MIC.2019.2948520
Date of current version 18 December 2019.

Recently, ML has emerged as one of the most prominent new approaches for realizing network control policies. Generally, ML techniques automatically learn policies from historical data and model the mapping from inputs to outputs without predefined rules. Among ML methods, offline learning is suitable for the scenarios where it can be assumed that the behavior of others has “converged” and assumed not to change much. Whereas online learning provides a game situation between the flows or users, these flows or users could play a cooperative environment, trying to achieve some common goals. There are interesting recent works on these lines by ML experts, e.g., the DeepMind people.[5] The CC problems can be cast with either of the above cases. Hence, some research works propose CC schemes with ML techniques and it is necessary to develop IETF standards of learning-based congestion control.

In this article, we present a survey on Internet CC from the ML perspective. First, we describe the reasons to apply ML techniques for congestion control. Then, we survey the state-of-the-art CC schemes and analyze their technical characteristics. We then discuss the challenges for developing the standards of learning-based CC in the real world. We hope that this study can encourage the researchers to design novel algorithms or develop Internet standards for congestion control.

WHY CC WITH MACHINE LEARNING?

ML is suitable and efficient for learning complex behaviors, where it is not easy to find the relationship between the input and the output. Specifically, ML can provide new possible ways to generate control policies by training a learning-based agent. The adoption of ML as a solution in network system is becoming a reality.[9] As of now, the network management research group has successfully proposed two learning-based Internet drafts[6,7] in IETF.

Traditional Internet CCs only consider several metrics as decision signals,[1–4] such as packet loss and round-trip time (RTT). The existing rule-based methods elaborately make use of the above signals but achieve poor throughput when running in links with high stochastic packet loss or network jitter. In fact, decision-making can be affected by many factors, including traffic pattern, link failure, dynamic latency, packet loss, and diverse application requirements. It is difficult to get optimal or near-optimal control policies from complex network behaviors following predefined rules. ML can provide possible ways to generate models via the training approaches. It also has the ability to model the inherent relationships between the inputs and the outputs of the network environments.[8] Among the state-of-the-art techniques in machine learning, deep reinforcement learning (DRL),[20] as one of the latest breakthroughs’ techniques, makes it easy to react to multidimensional feedbacks directly from network environment in variable network conditions.[13,14] In the following, we will introduce the representative efforts that conduct CC with ML methods.

NOVEL RESEARCH WORKS OF CONGESTION CONTROL

For Internet CC, the core problem is to make the decision about how and when to send data. Researchers develop flexible strategies using ML approaches to cope with varying network conditions. The most representative research works are shown in Table 1. Remy,[9] Indigo,[10] Aurora,[13] and Custard[14] perform optimization by learning the control rules offline, while PCC[11] and Vivace[12] work in an online learning manner. All of these schemes have different objective functions or utility functions as their optimization objectives. They choose different input signals, output, and ML methods, respectively. They also evaluate under different experimental environment. Next, we analyze the main techniques of these schemes.

Offline Learning

Remy[9] takes the target network assumption and the traffic model as prior knowledge and automatically generates a CC algorithm for the corresponding environment. In the offline phase, Remy uses an objective function to guide the rule generation process. The learned rule, i.e., RemyCC, maps the specially designed network states to the corresponding parameters about the congestion windows (cwnds). Whenever the sender receives an acknowledgment (ACK), RemyCC looks up its

Table 1. Novel CC with machine learning.

Congestion control	ML method		Objective function (Utility function)	Action	Experimental environment
Remy[9]	Offline learning	A tabular method	Throughput and delay	Cwnds and pacing	NS-2
Indigo[10]		Imitation learning	The ideal cwnds	Cwnds adjustment	Mahimahi
Custard[14]		Trust region policy optimization (TRPO)	Throughput, delay and loss rate	Sending rate	Emulab
Auraro[13]		Proximal policy optimization (PPO)	Throughput, delay and loss rate	Sending rate	Mininet
PCC[11]	Online learning	Rate probing	Throughput and loss rate	Sending rate	GENI Emulab Planetlab
Vivace[12]		Convex optimization	Sending rate, RTT gradient and loss rate	Sending rate	Emulab Mahimahi

mapping rule and changes the cwnds according to the current network state. Although Remy helps us to improve transmission efficiency, its performance could greatly degrade if the network assumptions are violated. Indigo[10] is another method of learning-based CC scheme with the data gathered from Pantheon,[10] a system for evaluating CC schemes. Indigo learns to "imitate" the oracle rule offline. The oracle is constructed with ideal cwnds given by the emulated bottleneck's bandwidth-delay product.

Aurora[13] and Custard[14] employ DRL to generate a policy that maps observed network statistics to choose the sending rate. DRL[20], as a novel ML algorithm, trains an agent which can sample the network state, learn the policy, and improve its behavior by constantly interacting with an environment, as shown in Figure 1. The input of the agent is the network state (e.g., bandwidth, RTT, loss rate, etc.) and the output is the action, i.e., sending rate or cwnds. The goal (termed "reward") of reinforcement learning is to maximize discounted cumulative reward from the environments. Reinforcement learning is suitable for the sequential decision-making problem that can make decisions not only in discrete space (e.g., cwnds), but also in continuous space (e.g., sending rate). Aurora and Custard use different input signals and learn to make the decision by exploration–exploitation behavior. Despite the fact that the offline learning schemes can converge quickly and obtain more information, the general applicability is limited to the network scenarios where they have not been trained for.

Online Learning

PCC[11] and Vivace[12] are based on online learning. They attempt to adopt a trial-and-error mechanism to decide the sending rate. PCC's default objective function involves the throughput and the loss rate, while Vivace adopts a more complex utility function that replaces the absolute value of RTT with the "RTT gradient," i.e., the RTT with respect to time. With the carefully engineered utility function, Vivace aims to guarantee some desirable properties (e.g., fair convergence). Due to the characteristics of online learning, PCC and Vivace provide no-regret guarantees even under complete uncertainty about the environment, i.e., without inferring anything about the relation between policies and the induced utility values.[12] Both PCC and Vivace focus on looking for the change in the sending rate that may lead to the best performance, without directly interpreting the environment or making use of previous experience. Although online learning can react to network conditions quickly, its performance may diminish in some cases as their greedy exploration could be trapped at a local optimum.[22] It should be noted that online learning usually has long convergence time.[19]

CHALLENGES FOR LEARNING-BASED CONGESTION CONTROL

Latest CC schemes are indeed capable of learning useful strategies to adapt to the network environments. However, there are several challenges for Internet standardization. In this section, we attempt to present several issues of

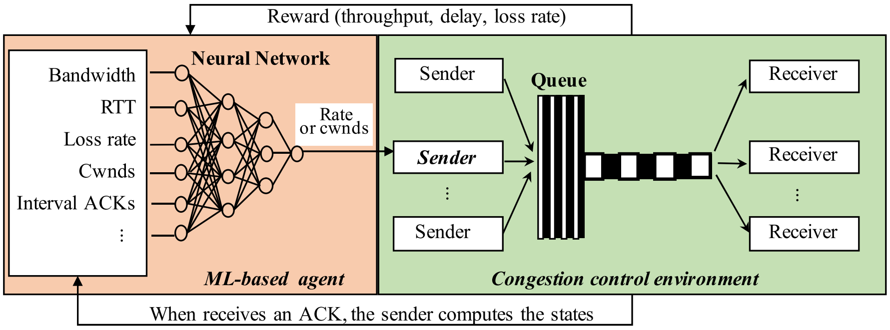

Figure 1. Architecture of CC with reinforcement learning.

CC with ML for standards as follows. First, we discuss the challenges of ML for congestion control. Second, we explore the challenges of CC with machine learning.

Challenges of ML for Congestion Control

Input and Output Space The space of input and output determine the primary operation of learning-based algorithms. The input space of the existing CC schemes varies greatly. For example, Remy takes the interval of ACKs, the interval of packets sent and RTT as states, whereas PCC takes the sending rate. This provides different information for the learning. A unified interface of state should be provided for standards of learning-based congestion, and its design is very challenging. Meanwhile, the output space also affects the efficiency of learning. Traditional congestion controls (e.g., Cubic[3]) or standards (e.g., IETF RFC 2581[1]) usually make the decision on cwnds. As the problem of bufferbloat becomes more and more serious, recent researchers have proposed to use rate-based transmission.[11] For learning-based congestion control, a large output space is a big challenge that makes it difficult to learn a model with ML methods. Some research works propose to decrease the decision space by reducing the dimension. For example, Indigo[10] adopts the adjustment of previous cwnds (addition, subtraction, multiplication, and division) instead of the value of cwnds.

Experimental Environment A large amount of data and the experimental environment are important for ML algorithms, especially for offline learning methods. For the CC problem, there is no unified dataset for learning-based algorithms that is open sourced currently. In addition, simulators and emulators in learning-based CC as shown in Table 1, e.g., NS-3[15], Mininet[16], Mahimahi[17], and Emulab[21] can only provide the reproducible and rapid experimentation, but they fail to capture the dynamics of the real world. Although Stanford researchers have proposed the Pantheon[10] platform as the training ground which includes real network paths, it is not easy for researchers to tune their algorithms and use it as a performance benchmark or a performance comparison platform.

Universality Directly deploying the offline learning-based agent from simulator or emulator can reduce the performance. The general applicability of the trained model is one of the key challenges faced by offline methods. Most offline learning methods assume that the data follow the same distribution which is not the case for real-world traffic flows. Approaches like Remy,[9] Indigo,[10] and Aurora[13] train the model on specific network conditions and perform well, to some degree, across a range of specific test conditions. However, they could not guarantee the performance of the learning-based CC methods when testing outside of the training fields. To develop standards, the offline learning-based CC model should have high general applicability that can adapt to high variance and dynamic traffic environments.

Challenges of CC With Machine Learning

Fairness Fairness is a crucial consideration for the design of TCP CC schemes.[18] On the Internet, different CC schemes may exist at the same time and interact with each other. However, the existing CC schemes with ML techniques cannot guarantee fairness with legacy TCP. The congestion controls with ML are trained in the environment with their own objective functions. When competing with other protocols, CC with ML cannot dynamically modify their objective functions so that the CC makes decision based on the predefined optimization objective. Further, even if CC approaches are trained in an environment where it competes with other protocols, they might learn to occasionally drop packets to free up network capacity.[13] To develop standards, fairness is the critical factor to be considered in the design of CC with ML methods.

Efficiency and Effectiveness Today, a large number of applications have high efficiency requirements.[19] The CC algorithms must be robust in the transport layer as mentioned in IETF RFC,[2] and meet the efficiency requirements of real-time transmission. Learning-based CC models (offline generated) have limitations on computational overhead, energy consumption, and response time. However, the current learning-based schemes in Table 1 do not detail the overhead comprehensively. Additionally, offline-based models work under their optimization functions and are required to consider the corresponding fault tolerance control methods in order to expel bad policies. As standards for learning-based approaches, congestion controls with ML in practice require the learning-based model to take the real-time network state and immediately output the near-optimal policy online. The tradeoff between efficiency and effectiveness is important for the performance in practical network scenarios.

Multiple Objectives As mentioned above, the optimization objective is another core role of learning-based congestion control. The existing congestion controls in Table 1 often use a combination of throughput, latency, and loss rate as the objective function or the utility function. However, once the tradeoffs between the throughput, latency, and packet loss are determined by the designers, the optimization objective of the learning-based CC is fixed. With increasingly complicated and diverse applications, there are distinct and diverse network performance requirements. The learning-based CC algorithms for Internet standards should satisfy the diverse transmission requirements of applications. Therefore, diverse optimization objectives are also considered for standards to handle the different tradeoffs between the performance factors which the applications or users need.

CONCLUSION

Although ML shows great potential in solving CC problems, there is still a long way to go for the industry to use CC with ML directly in practice due to some practical issues of ML for networking. In this article, we first analyze the advantages of using ML in Internet CC. Next, we summarize the latest CC schemes deriving from different learning techniques. However, some issues still remain to be addressed and we discuss the challenges for Internet standardization from the CC and ML perspectives.

ACKNOWLEDGMENT

The authors would like to thank Dah Ming Chiu for his helpful suggestions and comments. This work was supported in part by NSFC Project under Grant 61872211 and in part by the National Key R&D Program of China under Grant 2018YFB1800303.

REFERENCES

1. TCP Congestion Control, IETF RFC 2581. 1999. [Online]. Available: https://tools.ietf.org/html/rfc2581
2. Open Research Issues in Internet Congestion Control, IETF RFC 6077. 2011. [Online]. Available: https://tools.ietf.org/html/rfc6077
3. S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM Sigops Oper. Syst. Rev.*, vol. 23, no. 50, pp. 64–74, 2008.
4. S. W. O'Malley, L. S. Brakmo, and L. L. Peterson, "TCP vegas: New techniques for congestion detection and avoidance," in *Proc. ACM SIGCOMM Conf.*, 1994, pp. 24–35.
5. 2019. [Online]. Available: https://deepmind.com/blog/article/capture-the-flag-science.
6. Intelligent Reinforcement-learning-based Network Management, draft-kim-nmrg-rl-05. 2019. [Online]. Available: https://tools.ietf.org/html/draft-kim-nmrg-rl-05
7. Intelligent Reasoning on External Events for Network Management, draft-pedro-nmrg-intelligent-reasoning-00." 2019. [Online]. Available: https://tools.ietf.org/html/draft-pedro-nmrg-intelligent-reasoning-00

8. 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./Apr. 2018.
9. K. Winstein and H. Balakrishnan, "Tcp ex machina: Computer-generated congestion control," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 123–134, 2013.
10. F. Y. Yan *et al.*, "Pantheon: The training ground for internet congestion control research," in *Proc. USENIX Annu. Tech. Conf.*, 2018, pp. 731–743.
11. M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting congestion control for consistent high performance," in *Proc. 12th USENIX Conf. Networked Syst. Des. Implementation*, 2015.
12. M. Dong *et al.*, "PCC vivace: Online-learning congestion control," in *Proc. 15th USENIX Conf. Networked Syst. Des. Implementation*, 2018, pp. 343–356.
13. N. Jay, N. Rotman, B. Godfrey, M. Schapira, and A. Tamar, "A deep reinforcement learning perspective on Internet congestion control," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 3050–3059.
14. N. Jay, N. H. Rotman, P. Brighten Godfrey, M. Schapira, and A. Tamar, "Internet congestion control via deep reinforcement learning," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2018, pp. 1–10.
15. The network simulator—NS-3. 2001. [Online]. Available: https://www.nsnam.org/
16. N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. Mckeown, "Reproducible network experiments using container-based emulation," in *Proc. 8th Int. Conf. Emerg. Netw. EXp. Technol.*, 2012, pp. 253–264.
17. R. Netravali *et al.*, "Mahimahi: Accurate record-and-replay for HTTP," in *Proc. USENIX Annu. Tech. Conf.*, 2015, pp. 417–429.
18. Guidelines for Internet Congestion Control at Endpoints, IETF draft-fairhurst-tsvwg-cc-01, 2019. [Online]. Available: https://tools.ietf.org/html/draft-fairhurst-tsvwg-cc-01
19. B. Raouf *et al.*, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *J. Internet Services Appl.*, vol. 9, no. 16, pp. 1–99, 2018.
20. V. Mnih *et al.*, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
21. B. White *et al.*, "An integrated experimental environment for distributed systems and networks," *ACM SIGOPS Oper. Syst. Rev.*, vol. 36(SI), pp. 255–270, 2002.
22. M. Schapira and K. Winstein, "Congestion-control throwdown," in *Proc. 16th ACM Workshop Hot Topics Netw.*, 2017, pp. 122–128.

Lei Zhang is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University. Contact her at leizhang16@mails.tsinghua.edu.cn.

Yong Cui is currently a full professor at Department of Computer Science and Technology, Tsinghua University. Contact him at cuiyong@tsinghua.edu.cn.

Mowei Wang is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University. Contact him at wang.mowei@outlook.com.

Zhenjie Yang is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University. Contact him at yangzj15@mails.tsinghua.edu.cn.

Yong Jiang is a full professor at the Graduate School at Shenzhen, Tsinghua University. Contact him at jiangy@sz.tsinghua.edu.cn.