TCP-Drinc: Smart Congestion Control Based on Deep Reinforcement Learning[2]

Kefan Xiao, Shiwen Mao, and Jitendra K. Tugnait
Dept. Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201 U.S.A.
E- KZX0002@tigermail.auburn.edu, smao@ieee.org, tugnajk@eng.auburn.edu

1. Introduction and Motivation

The unprecedented growth of network traffic, in particular, mobile network traffic, has greatly stressed today's Internet. Although the capacities of wired and wireless links have been continuously increased, the gap between user demand and what the Internet can offer is actually getting wider. Furthermore, many emerging applications not only require high throughput and reliability, but also low delay. Although the brute-force approach of deploying wired and wireless links with a higher capacity helps to mitigate the problem, a more viable approach is to revisit the higher layer protocol design, to make more efficient use of the increased physical layer link capacity.

Congestion control is the most important networking function of the transport layer, which ensures reliable delivery of application data. However, the design of a congestion control protocol is highly challenging. First, the transport network is an extremely complex and large-scale network of queues. The TCP end host itself consists of various interconnected queues in the kernel. When the TCP flow gets into the Internet, it traverses various queues at routers/switches along the end-to-end path, each shared by cross traffic (e.g., other TCP flows and UDP traffic) and served with some scheduling discipline. Significant efforts are still needed to gain good understanding of such a complex network to develop the queueing network theory that can guide the design of a congestion control protocol. Second, if following the end-to-end principle, agents at end hosts have to probe the network state and make independent decisions without coordination. The detected network state is usually error-prone and delayed, and the effect of an action is also delayed and depends on the actions of other competing hosts. Third, if to involve routers, the algorithm must be extremely simple (e.g., stateless) to ensure scalability, since the router may handle a huge amount of flows. Finally, as more wireless devices are connected, the lossy and capacity-varying wireless links also pose great challenges to congestion control design.

Many effective congestion control protocols have been developed in the past three decades since the pioneering work [1]. However, many existing schemes are based on some fundamental assumptions. For example, early generation of TCP variants assume that all losses are due to buffer overflow, and use loss as indicator of congestion. Since such assumption does not hold true in wireless networks, many heuristics have been proposed for TCP over wireless to distinguish the losses due to congestion from that incurred by link errors. Moreover, many existing schemes assume a single bottleneck link in the end-to-end path, and the wireless last hop (if there is one) is always the bottleneck. Given the high capacity wireless links and the complex network topology/traffic conditions we have today [2], such assumptions are less likely to be true. The bottleneck could be at either the wired or wireless segment, it could move around, and there could be more than one bottlenecks. Finally, when there is a wireless last hop, some existing work [3] assumes no competition among the flows at the base station (BS), which, as shown in [4], may not be true due to coupled wireless transmission scheduling at the BS.

2. TCP-Drinc Design and Contributions

In [15], we aim to develop a smart congestion control algorithm that does not rely on the above assumptions. Motivated by the recent success of applying machine learning to wireless networking problems [5], and based on our experience of applying deep learning (DR) and deep reinforcement learning (DRL) to 5G mmWave networks [6], edge computing and caching [7]-[9], and RF sensing and indoor localization [10]-[12], we propose to develop a model-free, smart congestion control algorithm based on DRL. The original methods that treat the network as a white box have been shown to have many limitations. To this end, machine learning, in particular, DRL, has a high potential in dealing with the complex network and traffic conditions by learning from past experience and extracting useful features. A DRL based approach also relieves the burden on training data, and has the unique advantage of being adaptive to varying network conditions.

---

[2] A short review for [15]. This work was supported in part by the NSF under Grant CNS-1702957, and by the Wireless Engineering Research and Education Center (WEREC), Auburn University, Auburn, AL, USA.

In particular, we present TCP-Drinc in [15], acronym for Deep reinforcement learning based congestion control. TCP-Drinc is a DRL based agent that is executed at the sender side. The TCP-Drinc architecture is presented in Fig. 1. The agent estimates features such as congestion window difference, round trip time (RTT), the minimum RTT over RTT ratio, the difference between RTT and the minimum RTT, and the inter-arrival time of ACKs, and stores historical data in an experience buffer. Then the agent uses a deep convolutional neural network (DCNN) concatenated with a long short term memory (LSTM) network to learn from historical data and select the next action to adjust the congestion window size (see Fig. 2).

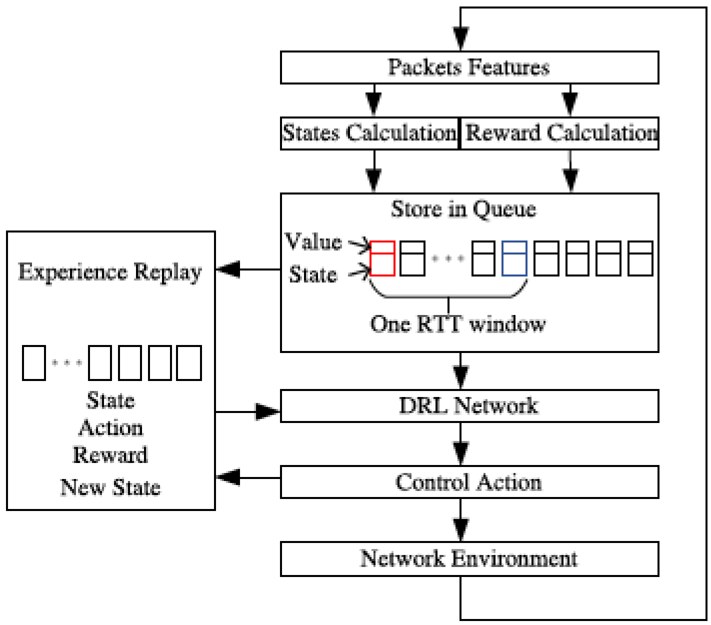

Figure 1 The proposed TCP-Drinc system architecture.

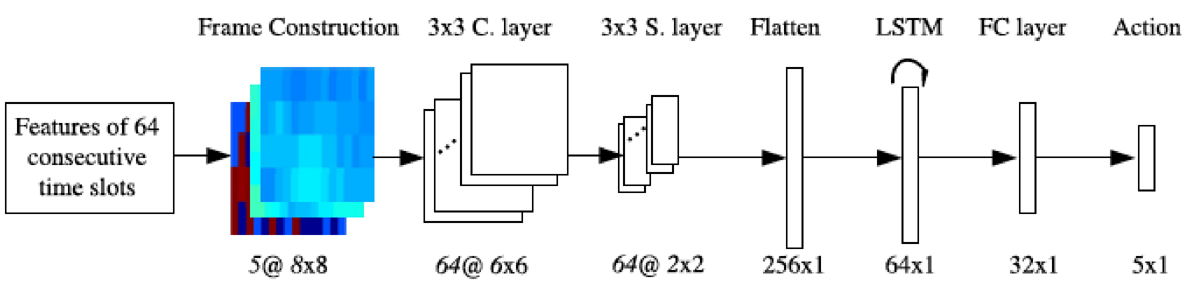

Figure 2 Design of the proposed DCNN (in the figure, "C." represents the convolutional layer, "S." represents the down sampling (pooling) layer, "FC" means fully connected).

The contributions of the work [15] are summarized as follows.

1) To the best of our knowledge, [15] is the first work that applies DRL to tackle the congestion control problem. Specifically, we propose a DRL based framework on (i) how to build an experience buffer to deal with the delayed environment, where an action will take effect after a delay and feedbacks are also delayed, (ii) how to handle the multi-agent competition problem, and (iii) how to design and compute the key components including states, action, and reward. We believe this framework could help to boost the future research on smart congestion control protocols.

2) The proposed TCP-Drinc framework also offers effective solutions to several long-existing problems in congestion control: delayed environment, partial observable information, and measurement variations. We apply DCNN as a filter to extract stable features from the rich but noisy measurements, instead of using exponential window moving average (EWMA) as a coarse filter as in previous works. Moreover, LSTM is utilized to handle the autocorrelation within the time-series introduced by delay and partial information that an agent senses.

3) We develop a realistic implementation of TCP-Drinc on the ns-3 [13] and TensorFlow [14] platforms. The DRL agent is developed with TensorFlow and the training and inference interfaces are built in ns-3 using TensorFlow C++. We conduct an extensive simulation study with TCP-Drinc and compare with five representative benchmark schemes, including both loss based and latency based TCP variants. TCP-Drinc achieves superior performance in throughput and RTT in all the simulations, and exhibits high adaptiveness and robustness under dynamic network environments.

3. Experimental Results

We examine the performance of TCP-Drinc and the baseline schemes under dynamic network settings. In particular, the simulation is executed 100 times, each lasting for 500s. The number of users is 5. The bottleneck capacity is varied at a frequency of 10 Hz; each capacity is randomly drawn from a uniform distribution in [5, 15] Mbps. The propagation delay is also varied at a 10 Hz frequency and each value is randomly drawn from a uniform distribution in [0:06, 0:16]s. In Fig. 10, we plot the combined RTT (*x*-axis) and throughput (*y*-axis) results in the form of of 95% confidence intervals. That is, we are 95% sure that the throughput and RTT combination of each scheme are located within the corresponding oval area. We find that TCP-Drinc achieves a comparable throughput performance with the loss based protocols, e.g., TCP-Cubic and TCP-NewReno. Furthermore, TCP-Drinc achieves a much lower RTT performance than the loss based protocols, e.g., at least 46% lower than TCP-NewReno and 65% lower than TCP-Cubic. Furthermore, TCP-Drinc achieves an over 100% throughput gain than TCP-Vegas at the cost of an only 15% higher RTT.

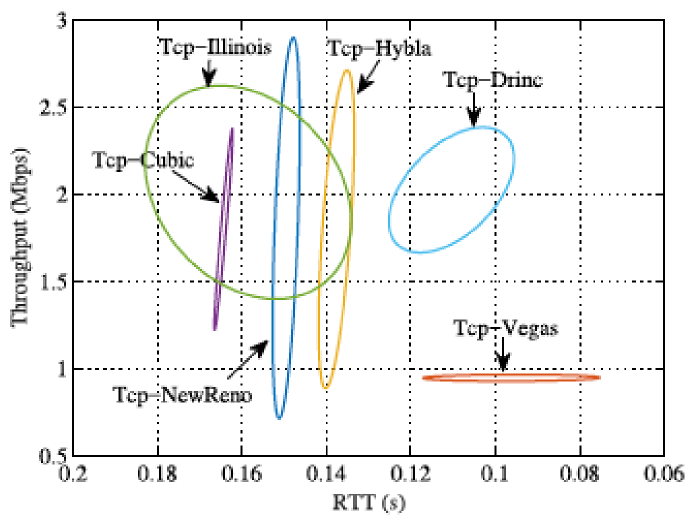

Figure 3 Throughput and RTT of the TCP variants under randomly varied network parameters. Each oval area represents the 95% confidence interval.

To study the fairness performance of the algorithms, we evaluate the Jain's index they achieve in the simulation. The average fairness index and the corresponding 95% confidence intervals are presented in Table 1. TCP-Vegas and TCP-Illinois achieve the best fairness performance among all the algorithms. TCP-Drinc can still achieve a considerably high fairness index (only 1.9% lower than the best). Note that the best fairness performance of TCP Vegas is achieved at the cost of a much poorer throughput performance. It is also worth noting that the 95% confidence interval of TCP-Drinc is the smallest among all the schemes, which is indicative of its robustness under varying network conditions.

Table I Jain's Fairness Index Achieved by the Congestion Control Schemes

	TCP-Cubic	TCP-Hybla	TCP-Illinois	TCP-NewReno	TCP-Vegas	TCP-Drinc
Average fairness index	0.7873	0.8025	0.8125	0.7562	0.8214	0.8058
95% Confidence Interval	[0.7005, 0.8741]	[0.7008, 0.9042]	[0.7114, 0.9136]	[0.6284, 0.8839]	[0.7115, 0.9313]	[0.7233, 0.8882]
Confidence Interval Span	0.1736	0.2034	0.2022	0.2555	0.2198	0.1649

4. Conclusions

In [15], we developed a framework for model-free, smart congestion control based on DRL. The proposed scheme does not require accurate models for network, scheduling, and network traffic flows; it also does not require training data, and is robust to varying network conditions. The detailed design of the proposed TCP-Drinc scheme was presented and the trade-offs were discussed. Extensive simulations with ns-3 were conducted to validate its superior performance over five benchmark algorithms.

References

[34] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Comput. Commun. Rev*., vol. 18, no. 4, pp. 314-329, Aug. 1988.

[35] Y. Zhao, B. Zhang, C. Li, and C. Chen, "ON/OFF traffic shaping in the Internet: Motivation, challenges, and solutions," *IEEE Netw*., vol. 31, no. 2, pp. 48-57, Mar./Apr. 2017.

[36] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in *Proc. USENIX NSDI*, Lombard, IL, USA, Apr. 2013, pp. 459-471.

[37] Y. Zaki, T. Potsch, J. Chen, L. Subramanian, and C. Gorg, "Adaptive congestion control for unpredictable cellular networks," *ACM SIGCOMM Comput. Commun. Rev*., vol. 45, no. 4, pp. 509-522, Oct. 2015.

[38] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao. (Sept. 2018). "Application of machine learning in wireless networks: Key techniques and open issues." [Online]. Available: https://arxiv.org/abs/1809.08707

[39] M. Feng and S. Mao, "Dealing with limited backhaul capacity in millimeter wave systems: A deep reinforcement learning approach," *IEEE Commun.*, vol.57, no.3, pp.50-55, Mar. 2019.

[40] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J*., to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8493155

[41] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning based mode selection and resource management for green fog radio access networks," *IEEE Internet Things J*., vol.6, no.2, pp.1960-1971, Apr. 2019.

[42] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 28-35, June 2018.

[43] X. Wang, X. Wang, and S. Mao, "RF sensing in the Internet of Things: A general deep learning framework," *IEEE Commun*., vol. 56, no. 9, pp. 62-69, Sept. 2018.

[44] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763-776, Jan. 2017.

[45] W. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng*., to be published. [Online]. Available: https://ieeexplore.ieee.org/document/8468057

[46] G. F. Riley and T. R. Henderson, ``The ns-3 network simulator," in *Modeling and Tools for Network Simulation*, K.Wehrle, M. Gunes, and J. Gross, Eds. Berlin, Germany: Springer, 2010, pp. 15-34.

[47] M. Abadi et al., ``Tensorflow: A system for large-scale machine learning," in *Proc. USENIX OSDI*, Savannah, GA, USA, Nov. 2016, pp. 265-283.

[48] K. Xiao, S. Mao, and J.K. Tugnait, "TCP-Drinc: Smart congestion control based on deep reinforcement learning," *IEEE Access Journal*, Special Section on Artificial Intelligence and Cognitive Computing for Communications and Networks, vol.7, no.1, pp.11892-11904, Jan. 2019. DOI: 10.1109/ACCESS.2019.2892046.