Self-learning Congestion Control of MPTCP in Satellites Communications

Tianle Mai
Beijing Advanced Innovation Center for Future Internet Technology Beijing University of Technology
Beijing, China
machealmai@gmail.com

Haipeng Yao
State Key Laboratory of Networking and Switching Technology Beijing University of Posts and Telecommunications
Beijing, China
yaohaipeng@bupt.edu.cn

Yaqing Jing
Beijing Advanced Innovation Center for Future Internet Technology Beijing University of Technology
Beijing, China
jyq@emails.bjut.edu.cn

Xiaobin Xu
Beijing Advanced Innovation Center for Future Internet Technology Beijing University of Technology
Beijing, China
xuxiaobin@bjut.edu.cn

Xiaolong Wang
Beijing Advanced Innovation Center for Future Internet Technology Beijing University of Technology
Beijing, China
wxlong124@163.com

Zhe Ji
State Key Laboratory of Networking and Switching Technology Beijing University of Posts and Telecommunications
Beijing, China
jiz18@bupt.edu.cn

***Abstract*—The past few years have witnessed a wide deployment of low earth orbit (LEO) satellites communications and networking. With the explosive growth of new businesses, satellite network is expected to provide global coverage and high bandwidth availability service. Toward this end, Multipath TCP(MPTCP) is a promising transport protocol to use in LEO satellites networks. MPTCP can not only achieve seamless handover, but also enhance throughput by using multiple paths transmission mechanism. However, following the improvement of the performance and scalability, it also brings unprecedented challenges for congestion control of multiple sub-flows. Especially, currently works on the congestion control largely relies on a manual process which presents a poor performance in the high-dynamic complexity network environment. Inspired by the recent success of applying machine learning in many challenging control decision domains, such as video game, self-driving, we employ deep deterministic policy gradient for learning the optimal congestion control strategies by interacting with the underlying network environment. Some simulation results demonstrated the effectiveness and feasibility of our architecture and algorithms.**

***Index Terms*—MPTCP, low earth orbit, congestion control, DDPG**

I. INTRODUCTION

Recently, satellite communications and networking has received a large amount of attention from both academics and industries. Owing to the unique advantages, such as global coverage, strong resilience, and high bandwidth availability, satellite communications have gradually infiltrated into every variety of practical fields ranging from disaster rescue to military mission [1]. Specifically, low earth orbit(LEO) satellites system plays a significant role in currently satellite communications. LEO satellites system is deployed at altitudes between 500km and 1500km above the earth's surface. Compared to Medium Earth Orbit(MEO) and Geostationary Earth Orbit(GEO), LEO satellites system provide higher throughput and lower propagation delay. In addition, it takes less energy to deploy the satellites into LEO compared to MEO and GEO [2]. Therefore, currently, most satellite communication system uses LEO satellites as infrastructure. However, because of the extremely high moving speeds of LEO satellites, the frequent satellite handover in LEO satellites system is a pending problem that needs to be solved urgently. For example, in the Iridium system, the visible LEO satellite for the ground terminal switches every 8 to 10 minutes. The intermittent handover may result in routing failures, channel quality changing, packet blocking and ultimately lead to service performance degradation.

With the increasing of multi-homed multiple interfaces devices, Multipath-TCP(MPTCP) become a promising transport protocol to match the requirements of today's networks [3]. Recently, there have been a series of works on satellite communications from the perspective of MPTCP. In [4], Paasch *et al*. discussed the efficiency of different MPTCP sub-flow mechanisms, congestion control algorithms and other parameters tuning. In [5], Du *et al*. designed an MPTCP-aware SDN controller which is able to identify MPTCP sub-flows, and accommodated the sub-flows with disjoint satellite paths. In [6], Giambene *et al*. presented a novel PBNC-MPTCP mechanism which combined MPTCP with network coding to protect TCP transmissions in a PEP-based satellite scenario.

However, as a multiple paths competing transmission mechanisms, how to allocate the network resource efficiently and fairly among competing sub-flows is challenging [7]. In order to address these problems, a series of works have been made to improve the performance. In [8], Raiciu *et al*. proposed a coupled multipath congestion control algorithm RTT-Compensator

978-1-5386-7747-6/19/$31.00 ©2019 IEEE

for meeting three goals of improve throughput, do no harm, and balance congestion. In [9], Cao *et al*. proposed an approximate iterative algorithm with the "Congestion Equality Principle" to solve the multipath congestion control. In [10], Hassayoun *et al*. presented Dynamic Window Coupling(DWC) mechanism to maximize throughput and be fair to other flows. However, these algorithms largely relies on the manual process, which has poor scalability and robustness in complex system control. Therefore, there is a need for more powerful methods to deal with the challenges faced in networking.

Inspired by recent success of applying machine learning in other challenging domain, such as video game, autonomous vehicles. In this paper, we try to apply a deep reinforcement learning approach for optimizing the congestion control strategies to maximize throughput and guarantee fairness. In addition, some simulation results are presented to evaluate the correctness of our architecture and algorithm.

The rest of this paper is organized as followed. In Section II, we present a new architecture of combining MPTCP with satellite communications and formulate the problem of multipath congestion control in MPTCP. In Section III, we apply deep deterministic policy gradient algorithm for searching the optimal strategy of congestion control. In Section IV, we present a simulation result to demonstrate the performance of our architecture and algorithm, and summarize the work in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we firstly describe the system model of combination of MPTCP and satellite communications. Then, we formulate the problem of multipath congestion control in MPTCP.

A. System Model

Multipath-TCP, as a recently proposed TCP extension, splits a single data stream into multiple paths for the end-to-end transmission. As shown in Fig. 1, the transport layer of traditional TCP is divided into two parts: the MPTCP layer and the TCP layer (sub-flow layer). The MPTCP layer implements various functions for managing TCP sub-flows such as path optimization, packet scheduling, and congestion control. In addition, it is transparent and provides a standard TCP interface for the application layer to hide the complexity management of multiple paths.

As mention above, this multiple path transmission capabilities will improve bandwidth utilization, enhance the recovery capability of the connection, and adaptively transfer data from the congestion path to the non-congested path. Accordingly, deploying MPTCP in LEO satellite networks can not only enhance the transmission bandwidth, but also smoothly shift traffic on the disconnected sub-flow to other flow during LEO satellites handover.

As shown in Fig. 2, assuming a pair of ground stations S and D are communicating with each other through LEO satellite systems using MPTCP, where S is the source node and D is the destination node [11]. SAT_1, as the visible satellite of source node, plays the role of the entry satellite of ground station. At the same time, SAT_2 is an exit satellite.

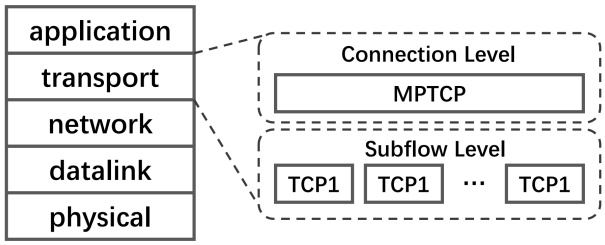

Fig. 1. MPTCP Protocol Stack.

During the transmission process, S possesses more than one IP addresses $S_i(1 \leq i \leq n)$, while D has $D_i(1 \leq i \leq n)$. The S established n path TCP sub-flows from source node to destination node. We denote them as $< S_1, D_1 >, < S_2, D_2 >$ .... $< S_n, D_n >$. The data packet will be routed with the IP address pair $< S_1, D_1 >, < S_2, D_2 >$ .... $< S_n, D_n >$ which can be calculated by source-address-based routing algorithm.

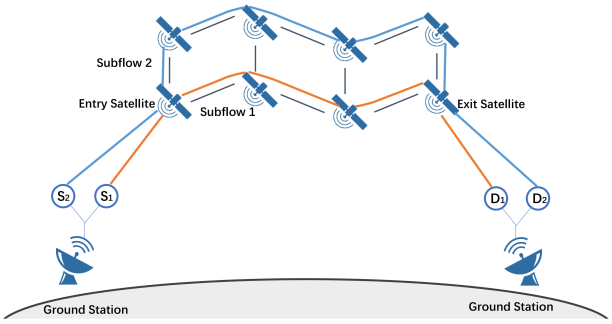

Fig. 2. Satellite Communications with MPTCP.

B. Problem Formulation

The main reason for network congestion is because of uncoordinated between the network resources and network load. When the network load higher than the maximum capacity of network resources, it will cause packet loss, reduced data throughput, retransmissions, and performance degradation.

For MPTCP, the congestion control adopted congestion windows (CWND) to adjust the congestion condition, where CWND limits the amount of packet each TCP can send before receiving an ACK. Each sub-flow is correspondingly adjusted CWND according to the number of received acknowledgment packet (ACK), round-trip time (RTT), and retransmissions rate. Due to the requirement effectiveness and efficiency, the IETF MPTCP working group stated that the design of MPTCP congestion control algorithm must have three objectives: improve throughput, harmless, and balance congestion.

Improve throughput: The performance of a multi-path flows should at least better than a single flow on the best path [8].

Harmless:A multi-path flow cannot preempt too much network resource to guarantee it not unduly harm other single path TCP flow.

Balance congestion:A multi-path connection should remove traffic from the path with a high congestion condition as much as possible, thus satisfying the first two objectives.

III. DEEP REINFORCEMENT LEARNING

In this section, we firstly formulate a markov decision process to model the learning process of intelligent end point. Then, we apply the deep deterministic policy gradient for searching the optimal congestion control strategies.

A. Markov Decision Process

As described above, a joint congestion control problem with multi-objective among competing sub-flows are formulated. However, the current solution of congestion control is largely relied on the complex manual process(white-box). These white-box approaches present poor scalability and robustness under the high-dynamic network environment. It generally requires pre-defined abstraction of the underlying environment and meticulously designed heuristic algorithm. In this paper, inspired by many successes of applying machine learning in challenging control domains, we try to apply deep reinforcement learning for learning the optimal congestion control strategies.

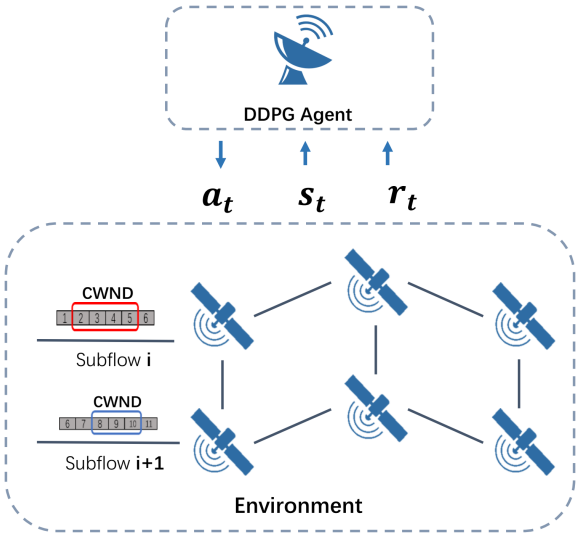

Fig. 3. The MDP process of strategy optimization.

Reinforcement learning is an important branch of machine learning where an agent learns the policy in an interactive environment by performing actions and receiving a reward. A Markov Decision Process (MDP) is formally described as the mathematical frameworks for modeling reinforcement learning. In our scenario, the AI agent and the underlying network environment interact continually for learning congestion control strategies.

As shown in Fig. 3, at each stage, the intelligent agent firstly observes the environment state s_t from the underlying networking and take a congestion control decision according to the current strategy $\pi(a|s)$. Following the action issued, the environment state will transfer to the s_{t+1} and the intelligent will receive an immediate reward R. Specifically, the congestion state can be represented by the congestion window size $cwnd_t$, round-trip time rtt_t, ACK number ack_t, and retransmissions rate rta_t of each sub-flow. The actions can be represented by congestion window adjustment $[cwnd_i]$ of each sub-flow, and the reward can be by the throughput.

Definition 1: The three components of the intelligent agent:

- State:
$$s^t = [(cwnd_t, rtt_t, ack_t, rta_t)_i]$$
where $cwnd_t$ represent the the number of packets sent by sub-flow in the unite time; rtt_t represent the time cost for a signal packet to travel from the source to the destination and back again; ack_t represent the the number of packets fail to receive the ACK packet; rta_t represent the cumulative rate number of retransmissions of the sub-flow i.
- Action:
$$a^t = [cwnd_1, cwnd_2, ..., cwnd_i]$$
- Immediate reward:
$$R = \sum_i (\alpha cwnd_t - \beta rtt_t - \epsilon rta_t - \kappa ack_t)$$
where α, β, ϵ, κ are the system parameter.

B. Deep Deterministic Policy Gradient

As the problem formulated above, learning the congestion control policy in a real-world network environment requires to deal with the high-dimensional state space and continuous action spaces. However, traditional reinforcement learning exists computational complexity and memory complexity issue in complexity system control. Recently, with the development of deep learning algorithm, the deep reinforcement learning has made great success in much challenging control decision domain. The powerful function approximation and representation learning capabilities of deep learning take a step further for reinforcement learning in the complexity high-dimensional problems [12]. Therefore, in this paper, we apply a deep reinforcement learning approach deep deterministic policy gradient(DDPG) for searching the optimal policy.

As shown in Fig. 4, the DDPG consists of two eponymous components. One is the deterministic policy network (Actor) $\mu(s|\theta^\mu)$ and the other is the Q-network (Critic) $Q(s, a|\theta^Q)$. The DDPG implements a policy iteration mechanism alternating between a policy improvement (actor) and a policy evaluation (critic) [13]. The actor aims at improving the current policy $\mu(s|\theta^\mu)$ through policy gradient, and the critic is to evaluate how good is the policy for the current parameters

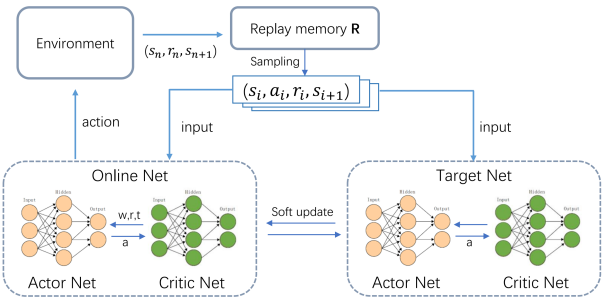

Fig. 4. The architecture of DDPG.

θ^{μ}. In addition, in order to improve the convergence and stability, the DDPG adopt two tricks named experience pool and the target network.

Experience Reply: The transition data $[s_t, a_t, r_t, s_{t+1}]$ during actual experience are stored in a replay memory R, and then sample a mini-batch transition data from the R for training. Based on such storage-sampling mechanism, the experience reply breaks the chronological relationship among data to improve the convergence of the learning process.

Target Network: The frequently changed target function brought by TD error calculation will cause training difficult issue. So the target network fixes parameters of target function to enhance the stability. And every few steps, it replaces them with the latest updated network.

Based on this, the actor and critic network and two copied target network can be formulated as follow:

$$policy\ network \begin{cases} online : \mu(s|\theta^{\mu}) : gradient\ update\ \theta^{\mu} \\ target : \mu(s|\theta^{\mu'}) : soft\ update\ \theta^{\mu'} \end{cases} \quad (1)$$

$$Q\ network \begin{cases} online : Q(s,a|\theta^{Q}) : gradient\ update\ \theta^{Q} \\ target : Q(s,a|\theta^{Q'}) : soft\ update\ \theta^{Q'} \end{cases} \quad (2)$$

For the actor network, the policy objective function is defined as the long-term discounted sum of reward:

$$J(\theta^{\mu}) = E_{\theta^{\mu}}[r_1 + \gamma r_2 + \gamma^2 r_3 + ...] \quad (3)$$

where the γ is the discount factor which measure how much weight it should give to future rewards in the long-term reward.

The actor network continually updates the parameters θ^{μ} in the direction of policy gradient. In [14], Silver *et al.* demonstrated that the policy gradient with respect to θ^{μ} is equivalent to the expected gradient of the Q-valued function. Therefore, the policy gradient of $J(\theta^{\mu})$ can be described as:

$$\frac{\partial J(\theta^{\mu})}{\partial \theta^{\mu}} = E_s[\frac{\partial Q(s,a|\theta^{Q})}{\partial a}\frac{\partial \mu(s|\theta^{\mu})}{\partial \theta^{\mu}}] \quad (4)$$

For the critic network, it receives the pairs $[a, s]$ and calculate the $Q(s, a)$ to evaluate the utility of action a. The agent update the parameters θ^{Q} according to the target value. The gradient information is:

$$\frac{\partial L(\theta^{Q})}{\partial \theta^{Q}} = E_{s,a,r,s'}[(TargetQ - Q(s,a|\theta^{Q}))\frac{\partial Q(s,a|\theta^{Q})}{\partial \theta^{Q}}] \quad (5)$$

where

$$TargetQ = r + \gamma Q'(s', \mu(s'|\theta^{\mu'})|\theta^{Q'}) \quad (6)$$

In DDPG, the parameters of target network is copied over from the online network by polyak averaging every some-fixed-number steps [15]:

$$soft\ update : \begin{cases} \theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \end{cases} \quad (7)$$

The congestion control policy of the mobile users with DDPG algorithm is shown in detail as Algorithm 1.

Algorithm 1 The DDPG algorithm for mobile users.

Set initialized online network weights θ^{Q}, θ^{μ} randomly
Set initialized target network weights $\theta^{Q'} \leftarrow \theta^{Q}, \theta^{\mu'} \leftarrow \theta^{\mu}$
Set initialized replay buffer R
for $t = 1, 2, 3$ **do**
  Set initialized random process $\mathcal{N}$ for action exploration
  Receive a initial state s_1
  for $t = 1, 2, 3$ **do**
    Actor takes an action according to the current policy and random noise:
    $a_t = \mu(s_t|\theta^{\mu}) + \mathcal{N}_t$
    Execute action a_t and receive a new state s_{t+1}
    Storage the transition $[s_t, a_t, r_t, s_{t+1}]$ in R
    Sample a mini-batch from the R randomly
    Calculate mean squared error of Q-network:
    $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^{Q}))^2$
    $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update online Q-network based on Adam optimizer
    Calculate policy gradient of policy network:
    Update online policy network based on Adam optimizer
    $\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^{Q})\nabla_{\theta^{\mu}}|_{s=s_i, a=\mu s_i}\mu(s|\theta^{\mu})|_{s_i}$
    Update the target networks:
    $\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$
    $\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$
  end for
end for

IV. SIMULATION

We present some simulation results to demonstrate the feasibility and correctness of our architecture and algorithms. In our experiment, we simulated a network, which has 6 nodes and 11 full-duplex links. For simplicity, we assume that the source node possesses two IP address and the destination node posses one IP address. We used a policy network with two connected hidden layers and a deep Q-valued network with three convolutional neural networks. In addition, we trained the DDPG agent for 100K steps, and we set the discount γ is 0.9, and τ is 0.4. Our experiment environment is developed

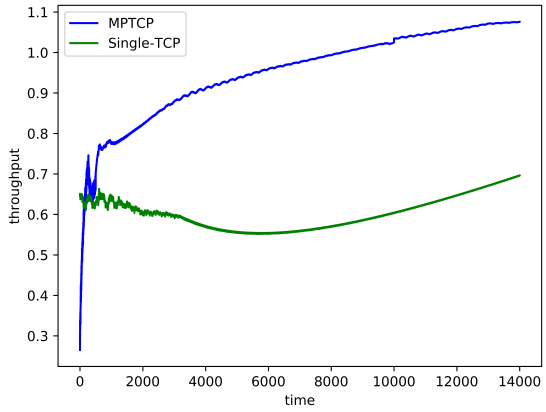

Fig. 5. The throughput performance(TCP vs. MPTCP).

on [16], [17], where we applied NS3 for network simulation and used TensorFlow for DDPG agent constructing.

We first compared the MPTCP algorithms with single-path TCP. As shown in Fig. 6, according to the multiple path transmission mechanisms, the MPTCP present a higher throughput than traditional single-path TCP transmission. This result demonstrates the superiority of applying MPTCP for improving throughput performance.

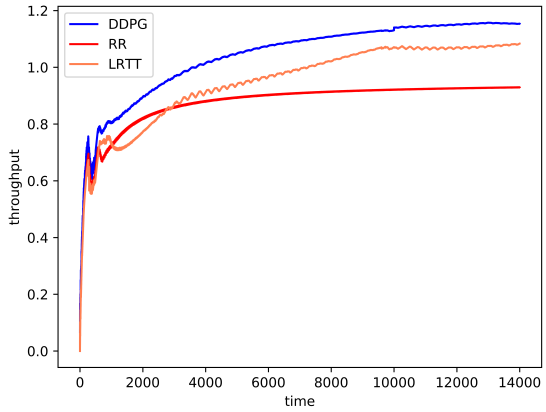

Fig. 6. The throughput performance.

Then, we compared the DDPG algorithm with traditional data scheduling algorithms the round-robin(RR) and the lowest RTT first RR(LRTT). The RR schedule each sub-flow with no priority and adopt a round-robin mechanism. The LRTT schedule the data to each sub-flow based on the lowest RTT is beneficial. These algorithms adopt pre-defined deterministic strategies which is hard to meet the complex network environment [18]. As shown in Fig. **??**, based on the strong fitting ability of deep neural networks, our algorithms present a higher throughput than RR and LRTT algorithms.

V. CONCLUSIONS

Due to the low orbit and short range, the LEO satellites can be accessible by terrestrial users with high throughput and lower propagation delay. However, due to the high moving speeds of LEO satellites, the frequent handover is a challenging problem that needs to be solved urgently. In this article, we combined the MPTCP with satellite communications for improving the feasibility and performance of LEO satellites communications system. However, designing a congestion control mechanism in MPTCP for meeting complexity network state is a non-trivial task. In this paper, we applied a deep reinforcement learning for learning the optimal congestion control strategies for each sub-flow. Finally, simulation results were given to evaluate our proposed algorithm.

REFERENCES

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2018.

[2] E. Papapetrou, S. Karapantazis, and F. N. Pavlidou, "Distributed on-demand routing for leo satellite systems," *Computer Networks*, vol. 51, no. 15, pp. 4356–4376, 2007.

[3] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath tcp," in *Usenix Conference on Networked Systems Design Implementation*, 2012.

[4] C. Paasch, R. Khalili, and O. Bonaventure, "On the benefits of applying experimental design to improve multipath tcp," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pp. 393–398. ACM, 2013.

[5] P. Du, S. Nazari, J. Mena, R. Fan, M. Gerla, and R. Gupta, "Multipath tcp in sdn-enabled leo satellite networks," in *Military Communications Conference, MILCOM 2016-2016 IEEE*, pp. 354–359. IEEE, 2016.

[6] G. Giambene, D. K. Luong, M. Muhammad *et al.*, "Network coding and mptcp in satellite networks," in *Advanced Satellite Multimedia Systems Conference and the 14th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2016 8th*, pp. 1–8. IEEE, 2016.

[7] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp," in *Usenix Conference on Networked Systems Design Implementation*, 2011.

[8] C. Raiciu, D. Wischik, and M. Handley, "Practical congestion control for multipath transport protocols," *University College London, London/United Kingdom, Tech. Rep*, 2009.

[9] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath tcp," in *Network Protocols (ICNP), 2012 20th IEEE International Conference on*, pp. 1–10. IEEE, 2012.

[10] S. Hassayoun, J. Iyengar, and D. Ros, "Dynamic window coupling for multipath congestion control," in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, pp. 341–352. IEEE, 2011.

[11] P. Du, X. Li, Y. Lu, and M. Gerla, "Multipath tcp over leo satellite networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pp. 1–6. IEEE, 2015.

[12] H. Yao, T. Mai, X. Xu, P. Zhang, M. Li, and Y. Liu, "Networkai: An intelligent network architecture for self-learning control strategies in software defined networks," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1.

[13] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, pp. 1008–1014, 2000.

[14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *ICML*, 2014.

[15] M. Volodymyr, K. Koray, S. David, A. A. Rusu, V. Joel, M. G. Bellemare, G. Alex, R. Martin, A. K. Fidjeland, and O. Georg, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[16] https://github.com/JamesRaynor67/mptcp_with_machine_learning.git.

[17] https://github.com/kallen666/MPTCP-Deep-Reinforcement-Learning.git.

[18] C. Paasch, S. Ferlin, O. Bonaventure, and O. Bonaventure, "Experimental evaluation of multipath tcp schedulers," in *Acm Sigcomm Workshop on Capacity Sharing Workshop*, 2014.