The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 9 (2023) 860–868

www.elsevier.com/locate/icte

Intelligent Multi-Path TCP Congestion Control for video streaming in Internet of Deep Space Things communication

Taeyun Ha, Arooj Masood, Woongsoo Na, Sungrae Cho*

The School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea

Received 7 June 2022; received in revised form 24 December 2022; accepted 14 February 2023
Available online 17 February 2023

Abstract

The vision of space exploration includes missions to deep space that produce significant amounts of video data and require reliable video streaming back to the Earth. The Internet of Deep Space Things (IoDST) is envisioned to provide communication services for video data streaming for the mission spacecrafts. Ensuring reliable communications in IoDST requires Transmission Control Protocol (TCP) layer functionalities. However, current TCP Congestion Control (CC) protocols provide poor performance in IoDST communications primarily owing to the dependence on pre-defined rules to determine the transmission rate in a single path TCP flow. This paper proposes a Multi-Path TCP (MPTCP) CC design for data streaming transmission in IoDST. We utilized Scalable Video Coding (SVC)-based streaming to overcome the Head-of-Line (HoL) blocking and proposed an intelligent CC scheme based on Q-learning and Deep Q-Network (DQN) to solve the problems of challenging link conditions in IoDST. Our proposed CC scheme determines the optimal congestion window for data transmission in IoDST communications to maximize the TCP throughput performance and streaming data playback. Simulation results show that our proposed CC scheme achieves TCP throughput performance by up to approximately 257.14% and 73.08% compared to TCP CUBIC and TCP Westwood. In addition, video streaming playback by up to approximately 164.86%, 104.17%, 75%, 157.89% and 66% compared to TCP CUBIC, TCP BBR, TCP Westwood, DRL-TCP and QLE-DS, respectively. Finally the total streaming data transfer time by up to approximately 61%, 20%, 63%, 21% and 19% compared to TCP CUBIC, TCP BBR, TCP Westwood, DRL-TCP and QLE-DS, respectively.

Keywords: Internet of deep space things; Multi-path TCP; Scalable video coding; Head of line blocking

1. Introduction

Recent advances in space technologies have enabled the realization of deep space exploration missions. These missions produce a significant amount of audio and video data related to the outer space planets to be delivered to Earth. In addition, these missions require autonomous video data streaming at high data rates. Successful video data streaming over deep space links among planets, mission spacecrafts, and crewed vehicles requires advanced communication and networking technologies to be developed [1]. The Internet of Deep Space Things (IoDST) system is envisioned to provide reliable communication and networking services to future deep space missions. An IoDST system comprises of three networks: a planetary surface network, a planetary spacecraft network, and an interplanetary backhaul network [2]. Among the three aforementioned networks, the interplanetary backhaul network is characterized by extremely long and variable propagation delays (e.g., ranging between 8.5 min and 40 min with respect to the Mars-to-Earth communication network). It is also associated with high link error rates and link outages [3]. This network imposes the most challenging problems in achieving reliable video streaming [4].

Transmission Control Protocol (TCP) layer functionalities are crucial for reliable video streaming and timely delivery of real-time video data. However, the currently existing window-based and acknowledgment-triggered TCP protocols have been designed to operate efficiently over the traditional error-free wired links [5–10] and wireless links with very low propagation delays [11,12] and suffer from high propagation delays and random link errors. [13] have shown that legacy TCP protocols provide very poor performance on deep space links with extremely high propagation delays and link errors. Some TCP protocols have also been developed for

* Corresponding author.
E-mail addresses: tyha@uclab.re.kr (T. Ha), arooj@uclab.re.kr (A. Masood), wsna@kongju.ac.kr (W. Na), srcho@cau.ac.kr (S. Cho).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2023.02.006
2405-9595/

links with large bandwidth delay product [14–17]. However, these approaches have been shown to perform well in satellite network scenarios in which the RTTs are much smaller and the error rates are much lower than the interplanetary backhaul links in the IoDST. Some studies have also been proposed to address the challenges caused by the unique characteristics of the interplanetary backhaul link networks [4,18–20]. For instance, [4] proposed a rate-based additive-increase multiplicative-decrease TCP mechanism to achieve reliable data transmission on deep space links. [18,19] proposed an adaptive rate control algorithm to improve the utilization of planetary links as a rate-based version of classic TCP. [20] proposed a deep-space transport protocol (DS-TP) to deal with the packet losses and link outages.

However, these approaches present the following limitations: (1) The rule-based approaches use a fixed set of rules to handle every situation. (2) These approaches do not leverage past experience (e.g., run time statistics data) to control the transmission rate. (3) When long RTTs are experienced, the older information about the links conditions is received at the source and the CC decision based on such past information may not be an optimal decision. (4) Single-path TCP results in reduced link utilization on long RTTs. However, future deep space missions require video data streaming that has high requirements on both bandwidth and transmission latency.

To address the aforementioned problems of the existing TCP approaches, we propose a learning-based solution using Q-learning and Deep Q-network (DQN) for Multi-Path TCP (MPTCP) Congestion Control (CC) due to its capability to learn to take the best actions according to run time states. In addition, CC in MPTCP allows to split a single TCP flow into multiple sub-flows across multiple paths to improve end-to-end link bandwidth utilization. Since the characteristics of paths (e.g., bandwidth and delay) may differ, this can lead to Out-of-Order (OFO) delivery at the destination. As a result, packets with higher sequence numbers have to be stored in the destination buffer till the reception of packets with lower sequence numbers, which causes Head-of-Line (HoL) blocking [21,22]. To overcome this challenge, we utilize Scalable Video Coding (SVC) streaming in which each chunk is encoded into one base layer (BL) and multiple Enhancement Layers (ELs), where BL is transferred with high priority than ELs and each layer of every chunk is fetched from only one sub flow. The main contributions of this paper are summarized as follows:

- We utilize SVC-based data streaming to prevent HoL blocking in MPTCP CC and to improve end-to-end link bandwidth utilization for reliable video streaming over interplanetary backhaul links in an IoDST communication.
- We choose DQN as the basis for our MPTCP CC design because the DQN agent learns to generate an optimal CC action to maximize the received reward in a particular state, and adjust the policy adaptively to maximize network performance for different conditions in challenging IoDST communication.
- We analyze the performance of the proposed TCP CC design in comparison to TCP Cubic [23], TCP BBR [24], TCP Westwood [10], DRL-TCP [25]. The simulation results reveal that our proposed MPTCP CC scheme outperforms comparative schemes in terms of streaming data playback.

The remainder of this paper is organized as follows. In Section 3, we present the system model of this study. Then, the design of our proposed DQN-based MPTCP CC scheme is presented in Section 4. In Section 5, we analyze the performance of our proposed DQN-based MPTCP CC scheme and give the concluding remarks on this work in Section 6.

2. Related work

Multi-Path TCP (MPTCP) standardized by IETF in 2013 that enables simultaneous data transmission using multiple interfaces with different IP addresses has been proposed to compensate for link instability, intermittent connectivity, and high error rates. MPTCP is suitable for supplementing unstable connection in the deep space environment, and various studies exist accordingly. MPTCP uses an algorithm called [26] as a default congestion control algorithm, and it is an algorithm that dynamically controls the aggression of each flow by combining sub-flows within the increase function. Based on this, various algorithms were developed, and [27] could improve performance by balancing sub-flow, and [28] suggested an algorithm that could provide two problems at the same time by setting the problems of the existing LIA algorithm as optical resource pooling and response. In addition, [29] developed a delay-based algorithm that formalizes the congestion control problem of MPTCP and achieves subdivided load balancing using packet queue delay as a congestion signal. [30] proposes an approach that can evaluate the performance of MPTCP flows through joint WiFi and cellular network connections in consideration of various characteristics of cellular and Wi-Fi networks. [31] developed a new congestion control algorithm that shows fast responsiveness when millimeter wave link states are changed from visible to invisible to achieve high throughput and low end-to-end delays in 5G mobile networks. [32] considered the data center network situation and proposes an improved MPTCP protocol that adjusts the time granularity of congestion detection and control in a number of sub-flows. So far, studies aimed at improving overall throughput or reducing delays have been discussed, and studies related to energy efficiency also exist. [33,34] are papers that value energy efficiency, and [33] has a peculiarity in solving this problem using genetic algorithms.

As such, many fixed control policies with relatively low computational overhead have been developed mainly because even small overheads can be a factor in sensitive performance degradation. However, there are also future-oriented studies that try to solve the problem of congestion control using reinforcement learning or machine learning.

[25] proposed a congestion control framework based on in-depth reinforcement learning, which uses LSTM and reinforcement learning to set appropriate congestion control

windows, and [35] proposes distributed learning methods to adapt to highly volatile environments and realize efficient congestion control. In addition, [36] presents a practical approach to how much deep learning models can improve 5G network services. [37] proposed instrumentation tile coding algorithms and Q-learning function estimation methods using asynchronous reinforcement learning frameworks, as existing multipath congestion control mechanisms have several performance problems due to QoS characteristics of various links.

Even if such a dynamic congestion control method is used, there is a HoL blocking problem that is not solved. To solve this problem, the concept of a scheduler was introduced. [38] conducts research on modeling schedules for multi-path transmission and identifying sub-flows. In [39], asymmetric problem mitigation techniques using packet retransmission systems, load distribution, and bandwidth estimation-based mechanisms were introduced. [40] aims to reduce buffer blocking time by developing an analysis model for buffer blocking time to improve QoS in a wireless environment. [41] explains the trade-off relationship that the higher the throughput, the lower the QoS, and proposes a new scheduling policy that mitigates jitter by sending packets from different sub-flows. [42] proposes a new path failure detection method called feedback-based path failure detection to reduce transmission interruption time by determining a fast path failure, preventing duplicate transmission interruption events and unnecessary retransmissions. [43] conducts and discusses a comprehensive experimental analysis that reveals performance problems caused by packet scheduling. [44] proposes a new data scheduling algorithm using the Deep Q Network and improves MPTCP data scheduling performance in asymmetric paths.

The widespread application of multiple hosts in the network results in performance bottlenecks in FatTree topologies, collapsing the Incast communication pattern and severely affecting network energy, [45] proposes a multi-level cooperative MPTCP Incast performance evaluation model for queues based on the Markov characteristics of the queuing network and MPTCP scheduling process. [46] proposes a unique MPTCP scheduling scheme called Efficient Bandwidth Aggregation (EBA). EBA adaptively enables packet replication through parallel paths, effectively preventing HoL blocking. However, this shows a decrease in overall performance by losing some bandwidth. In [47], the model proposed in accurately computes and allocates the forward delay of each MPTCP path and allocates data to the multiple paths according to the forward delay difference calculated using a new multi-learning support forward delay estimator. l^2-MPTCP dynamically manages path usage and selects optimal path collection for bandwidth aggregation and multipath transmission using promising reinforcement learning power. In [48], to solve the problem of next-generation mobile networks, the proposed smart MPTCP path controller that processes MPTCP subflows while having various network communication interfaces. The proposed method controls MPTCP subflows using information exchanged at connection initiation to properly map clients to appropriate network server interfaces, and extends them by analyzing them over 5G networks and video traffic. [49] proposes a Q-Learning-based energy-aware data scheduling mechanism for MPTCP-based media services. It models multipath scheduling as a Q-learning process and discretizes high-dimensional continuous Q-tables using a novel quantum clustering approach. In [50], they introduce an MPTCP scheduler that can fully utilize the channel resources available on multiple interfaces of MPTCP, and [51] proposes an MPTCP model based on reinforcement learning using streaming power spectral density analysis.

Since the aforementioned constraints on HoL Blocking of MPTCP are similar to those of SVC, good synergy is expressed when the two are combined. SVC is an extension of the H.264/AVC standard that provides users with a variety of video quality. SVCs encode video by dividing it into base layers and reinforcement layers, and adapt to dynamic network conditions by flexibly adjusting the number of layers. [52,53] Such SVC technology has various studies considering the characteristics of video streaming transmission over a wireless network. [54] proposes a protocol for transmitting through multiple paths with the aim of increasing the transmission efficiency of hierarchical data, and [55] proposes a framework for optimizing the coding and speed of video files using cache. [56] proposes a Quality of Experience recognition priority algorithm for SVC-based video streaming, and [57] proposes a video encoding and caching technique that minimizes average delays. In [58], they aim to maximize the quality of experience by selecting the optimal number of video layers through optimal connectivity, and similarly [59] proposes a collaborative spectral technique to maximize the quality of experience and minimize the total power consumption.

3. System model

3.1. Network model

In this study, we consider an interplanetary backhaul network in the IoDST communication system in which the source and destination end-points are basically the mission spacecraft on Mars and the ground station on the Earth, respectively, and planetary gateway spacecraft and relay spacecrafts orbitting around the outer space planets are the networking entities providing seamless communication services across the Mars-to-the Earth communication network, as shown in Fig. 1. The interplanetary bottleneck link has a capacity of 1 Mb/s and bit error rates on the order of 10^{-1} with a propagation delay of 8.5 to 40 min. The channel is characterized by a low signal-to-noise ratio and is modeled as Additive White Gaussian Noise (AWGN). The IoDST communications require reliable transmission of streaming data that has high requirements on bandwidth and network throughput maximization. In this study, we consider CC in MPTCP that establishes several subflows over multiple paths to improve end-to-end link bandwidth utilization for a single application, i.e., video streaming in IoDST communication. Since the characteristics of multiple paths in MPTCP may differ, we utilize SVC to prevent HoL

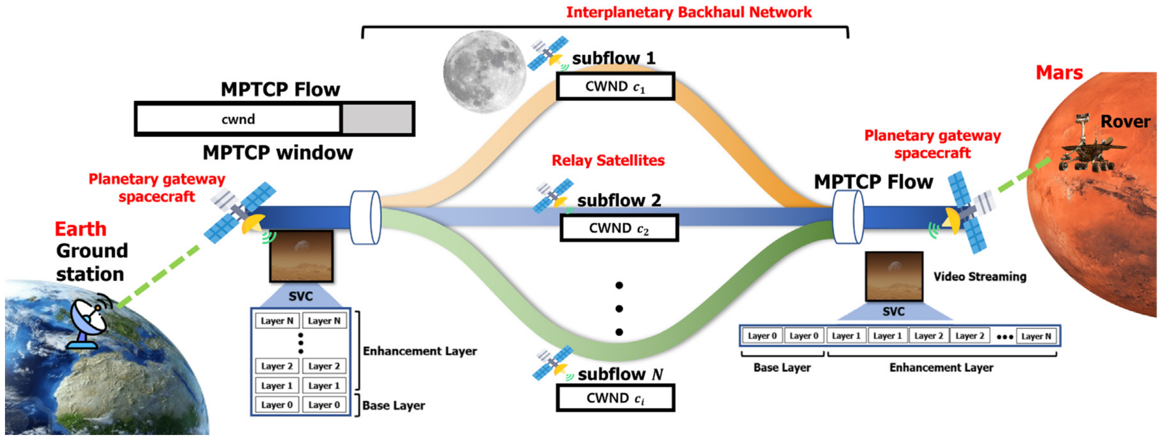


Fig. 1. Network model of an IoDST communication for SVC-based CC scheme for MPTCP.

blocking in video data transmission. SVC partitions video data into several chunks and encodes a chunk as one BL and several ELs, where the BL is transmitted with higher priority than ELs. Thus, a CC is performed on the sending host in MPTCP so that each layer of every chunk is transmitted from only one sub flow.

3.2. Scalable video coding (SVC) model

Each chunk is encoded into ordered layers (0 to M): one BL (layer 0) and multiple ELs (layer $M > 0$) that further improve the chunk quality based on layer $M-1$. For decoding a chunk up to enhancement layer M, a destination end-point must receive all layers from 0 to M. In this study, we assume all the layers (0 to M) have been inserted in an ascending order in the buffer [60]. Therefore, the weight increases with the decrease in the layer. We define the layer weight function as follows:

$$W(Ch_t^i) = (M - L(Ch_t^i)) \times W \tag{1}$$

where Ch_t^i is the ith chunk in the buffer at t-time slot. $L(\cdot)$ is a function that outputs the layer number of the chunk. W is the weight of the chunk and the default value is set as 100, and M is the number of layers, which is set as 3.

3.3. Overview of MPTCP CC

MPTCP extended the CC mechanism of TCP to multiple congestion windows Cw on the sub flows in a way coupled to the total window size. An MPTCP connection comprises N sub flows, where $N \geq 1$, and the congestion window Cw of an ith sub flow is updated based on the parameters of α and β. The update of the congestion window c of an ith sub flow of the t timeslot is defined as follows:

$$Cw_t^i = \alpha \times Cw_{t-1}^i + \beta \tag{2}$$

where Cw_{t-1}^i is the size of the congestion window of the ith sub flow in the $t-1$-time slot, and α and β are the parameters for window adjustment. To reduce the number of possible actions, we restrict α and β on some discrete values and the congestion window c is updated for N sub flows of an MPTCP connection every t-time slot.

3.4. SVC-based MPTCP CC as a learning task

In this section, we explore a brief introduction to DQN. We present the design of an SVC-based CC scheme for MPTCP as a Reinforcement Learning (RL) task with state and action spaces and with an immediate reward function using (1) and (2) to generate optimal CC for uncertain network conditions in an IoDST communication. DQN is a method that uses deep learning as a function approximator for Q-learning. With a function approximation using a neural network, the problem of reinforcement learning of the existing deep learning method was solved using Experience Replay Buffer and Fixed Q Targets.

It is an algorithm that learns in a model-free environment, which refers to learning a policy that allows the agent to take a specific action in a specific state. Maximize the expected value of the overall compensation for successive steps starting from the current state. This concept is also applicable in environments where transitions from one state to another occur stochastically or rewards are given stochastically. In this paper, DQN is used to consider the dynamic deep space environment and various states for various paths.

3.4.1. State space

The RL agent observes the state of an MPTCP connection with N sub flows at each time slot t, and obtains a CC decision for SVC chunks transmission on each sub flow i. In an IoDST system, the network state of an MPTCP sub flow i at a t-time slot is defined as follows:

$$s_t^i = [Cw_t^i,\ Lr_t^i,\ Bw_t^i,\ Rtt_t^i], \tag{3}$$

where the state parameters $Cw_t^i,\ Lr_t^i,\ Bw_t^i,\ Rtt_t^i$ denote the congestion window, loss rate, available bandwidth, and delay, in t-time slot, respectively, to reflect the delay and bandwidth conditions of a sub route. The Lr_t^i, is defined as the ratio of the lost segments to the number of transmitted segments and the Bw_t^i, is defined as $Bw_t^i = \frac{Cw_t^i \times MSS}{Bw_{available}} \times 100$ where MSS is maximum segment size and $Bw_{available}^i$ is available bandwidth of ith subflow. Specifically, for an MPTCP connection with N sub flows, the state in t-time slot can be defined as:

$$S_t = [s_t^1, \ldots, s_t^i, \ldots, s_t^N]. \tag{4}$$

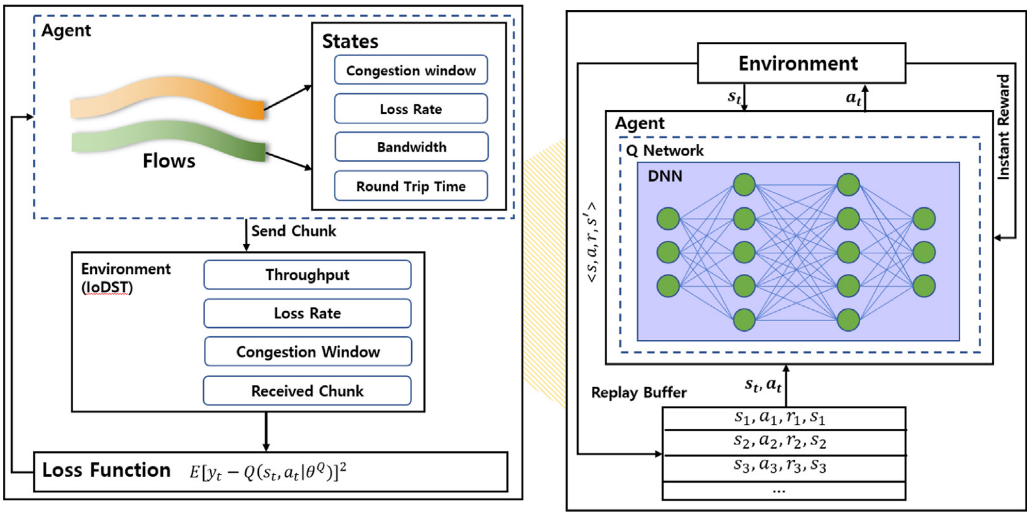


Fig. 2. RL for SVC-based CC scheme in MPTCP.

3.4.2. Action space

At each decision t-time slot, the agent observes the state s_t^i and performs an action a_t^i for MPTCP CC to determine the number of packets (chunks) transmitted at each sub flow i. To determine an action, a_t^i, for an ith sub flow, the values of α and β have been used in the Cw_t^i update. For an MPTCP connection with N sub flows, the action in t-time slot can be defined as:

$$A_t = [a_t^1, \dots, a_t^i, \dots, a_t^N]. \tag{5}$$

3.4.3. Reward function

Our objective is to design an optimal CC scheme that improves the link bandwidth utilization and throughput. Considering the network state S_t and action space A_t, the immediate reward function at t-time slot is formulated as follows:

$$r_t = log(\sum_{i=1}^{N} \sum_{j=1}^{C_t^i} W(Ch_t^j) \times LR_t^i) + log(\sum_{i=1}^{N} Th_t^i), \tag{6}$$

where $W(Ch_t^j)$ represents the layer weight function, Lr_t^i is the loss rate of the ith sub flow and Th_t^i is the achieved throughput.

4. DQN for SVC-based CC in MPTCP

Fig. 2 shows the operation of proposed scheme. The RL agent on the sender obtains information such as throughput, loss rate, current congestion window size, RTT, available bandwidth, and transmitted chunks in the environment. The agent selects an action that can maximize the return of the episode using state, which is the necessary information. According to this action, the size of the congestion control window is changed, and the result is substituted into the reward function to obtain a reward value for the corresponding action. The system proposed in this paper uses an experience replay buffer, which stores the values for state, action, and Q-value given at every step in the learning process of DQN and uses the stored experience to learn the network. With experience replay, the experience of learning remains in the replay buffer, which is used to update the network's weight. In this process, experience data that has undergone random-sampling as large as mini-batch size is extracted and weight is updated, which is less relevant than current data, so independence can be secured. In addition, in the weight update process, the target value is changed using the Q-learning method, and weight update is possible with new independent data for this changed value.

In Q-learning, the RL agent aims to find a policy $\pi(s_t)$ to map each state to a deterministic action, thereby maximizing its discounted cumulative reward

$$R_t = \sum_{t=1}^{T} \gamma^t r(s_t, a_t), \tag{7}$$

where $\gamma \in [0, 1]$ discounts future rewards. However, the Q-learning algorithm has a limitation in terms of handling high dimensional state space, which is called the curse of dimensionality. A deep version of Q-learning called DQN uses a Deep Neural Network (DNN) as a function approximator, which accepts state–action pair $(\mathbf{s_t}, \mathbf{a_t})$, as the input and outputs the corresponding Q-value, given by

$$Q(s_t, a_t) = \mathbb{E}[R_t | s_t, a_t], \tag{8}$$

where $\mathbb{E}[\cdot]$ is an expectation, and the action is derived by applying the following commonly used greedy policy:

$$\pi(s_t) = \underset{a_t}{\operatorname{argmax}}\, Q(s_t, a_t). \tag{9}$$

Based on Q-learning, the target value, y_t, for each state–action pair, $(\mathbf{s_t}, \mathbf{a_t})$, can be derived using the Bellman equation as follows:

$$y_t = r(s_t, \mathrm{a}_t) + \gamma Q(s_{t+1}, \pi(s_{t+1})|\theta^Q), \tag{10}$$

where θ^Q is the parameter of the DQN used to compute the target. Based on the target value, the DQN can be trained by minimizing the loss, $L(\theta^Q)$, where

$$L(\theta^Q) = \mathbb{E}\big[y_t - Q(s_t, \mathrm{a}_t|\theta^Q)\big]^2. \tag{11}$$

Algorithm 1 DQN for SVC-based CC algorithm in IoDST communication

1: Parameters: θ^Q
2: Randomly initialize Q value function $Q(\cdot)$ and policy $\pi(\cdot)$ with parameters θ^Q
3: Initialize replay buffer $\mathcal{B}$ with finite size of B
4: **for** episode n = 1, …, N **do**
5: Receive initial state s_t
6: Congestion window size
7: Loss rate for each subflow
8: Percentage of bandwidth utilization
9: Round Trip Time for each subflow
10: **for** t = 1,2, …, T **do**
11: Derive action $a_t \leftarrow \pi_E(a_t|s_t)$
12: Execute action a_t (change congestion window size)
13: Calculate reward r_t and next state s_{t+1}
14: Store experience in $\mathcal{B} \leftarrow \mathcal{B}$ (s_t, a_t, r_t, s_{t+1})
15: Sample random mini-batch of $\mathcal{B}'$ transitions
16: **for** each training step **do**
17: Perform a gradient descent according to Eq. (10)
18: **end for**
19: update network parameter θ^π
20: **end for**
21: **end for**
22: **end**

Thus, in this study, we propose a DQN-based MPTCP CC for SVC-based streaming to realize video data streaming in the IoDST communication. The specific proposed scheme is shown in the algorithm 1. First of all, the algorithm randomly initializes the value of the Q and the replay buffer (lines 1–3). The system collects the size of the congestion window, the loss rate of each subflow, the utilization of bandwidth, and the RTT of each subflow that is received from the environment (lines 5–9). Then, by selecting the action returned maximum reward, the system executes an action to change the size of the congestion window (lines 11–13). After executing an action, the system stores the state, action, reward, and next status in the replay buffer and performs random sampling (lines 14–15). Finally, gradient decent is performed for each learning step to update the network parameters (lines 16–19). According to the algorithm, the time complexity of our scheme is $O(N * T^2)$.

5. Performance evaluation

5.1. Simulation setting

To evaluate the performance of the proposed DQN-based MPTCP CC scheme, we assume two asymmetric links with different RTTs and loss rates. The detailed link settings is as shown in Table 1. Based on the link settings, link 1 is assumed to outperform link 2. For training the DQN, we used the experience reply memory and the reply buffer size and batch size is set to 512 and 10^{-6}, respectively. We used the ReLU activation function, and identity activation function for hidden layer as the activation function of all the hidden layers. In addition, the Reward discount factor γ is set to 0.95, and there are two hidden layers. The number of neurons for each hidden layer is 256. Also, we used Adam optimizer as the optimizer. We used channel model which is provided by National Aeronautics and Space Administration (NASA) to reflect more realistic and practical channels in the universe [61–63]. To evaluate the performance, three simulation experiments were conducted: the percentage of layers, TCP throughput, and HoL blocking percent and total completion time for transfer in the total streaming data.

Table 1
Simulation setting for link 1 and link 2.

	link 1	link 2
RTT	8.5 min–20 min	10 min–40 min
Loss rate	$10^{-2} - 10^{-4}$	$10^{-3} - 10^{-5}$
Channel rate	115,200 bit/s	240,000 bit/s

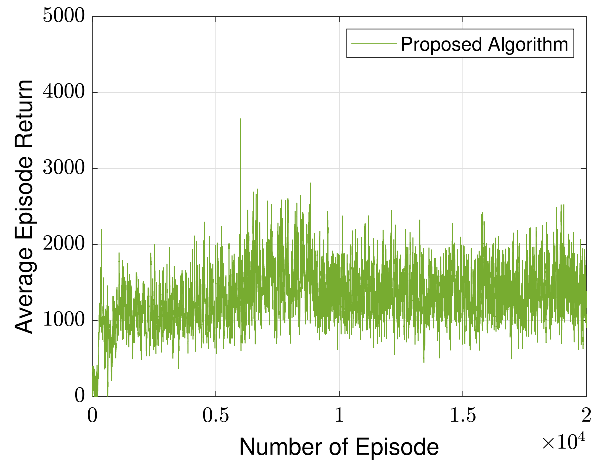


Fig. 3. Average Reward of Proposed Algorithm.

5.2. Simulation

In the simulations, we conduct an experiment to determine the successful transmission of chunk (BL only, up to EL, nothing) that has been transmitted from the streaming data.

Fig. 3 shows the convergence performance of the proposed scheme. As shown in the figure, it can be seen that each episode converges to a return value of between 1000 and 2000. This graph shows that this system has established a policy that guarantees convergence and performs optimal behavior in IoDST.

Fig. 4 shows the percentage of the layer that is successfully received and played when the streaming data is sent over the interplanetary backhaul link in IoDST communication. The proposed algorithm exhibited the best performance. However, in the comparison schemes, BL is often transmitted through slow paths and is not played normally. TCP Westwood exhibits a slightly higher EL playback percentage as opposed to TCP Cubic. The proposed algorithm reduces this to 2% in an environment where the probability of streaming data not being played was up to 68%. TCP BBR and DRL-TCP [25] transmit as much as the maximum congestion window size each subflow can spend without considering the characteristics (latency, loss rate) of each subflow, so of course throughput can prevail, but it causes HoL Blocking and does not meet

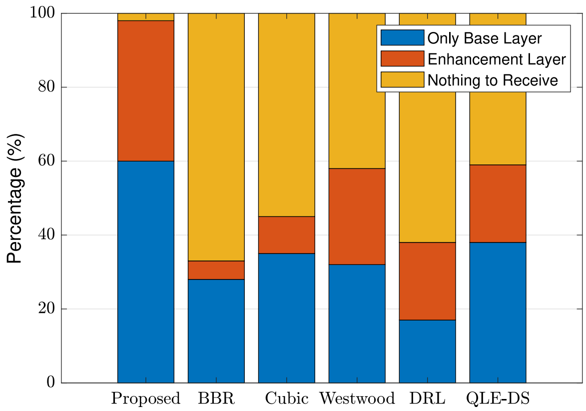


Fig. 4. Percentage of Layers Received.

the conditions of SVC, which increases the probability of not playing. For QLE-DS, the model shows average performance, which is about 66% lower than the proposed algorithm. This is due to the lack of constraints or considerations for high error rates, high RTT, and severe asymmetry in the IoDST environment.

Performance evaluation according to the asymmetry of RTTs of each link is also conducted. The asymmetry of the links is defined based on the relative error given by,

$$A_{1,2} = \frac{RTT_1 - RTT_2}{RTT_1} \times 100, \tag{12}$$

where RTT_1 is the RTT of link 1, and RTT_2 is the RTT of link 2.

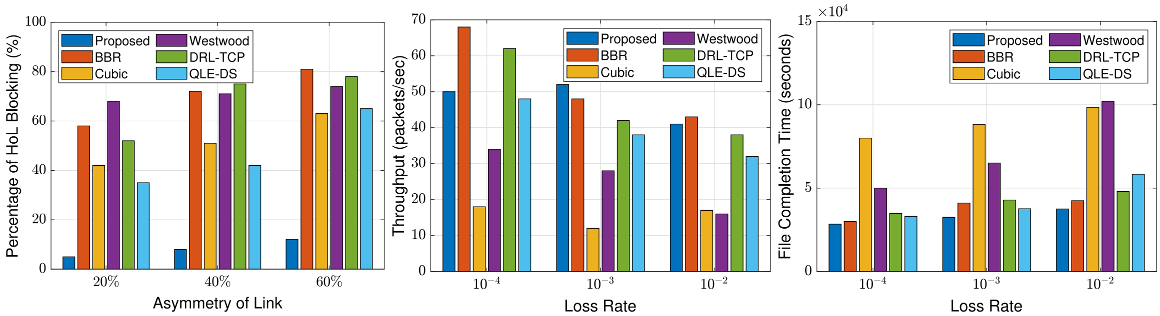
As shown in Fig. 5(a), it is evident that HoL Blocking increases with an increase in the asymmetry defined above. The proposed algorithm shows good performance at approximately 12% when the maximum asymmetry is 60%. In contrast, the other algorithms start at 50%, and if the HoL blocking is not considered in the design of MPTCP CC, it may result in significant OFO packet delivery. In addition, comparisons were made with the DRL-TCP system, which is the same reinforcement learning-based system, resulting in generally higher HoL Blocking than the variant TCP of other fixed control policies. This results in very low performance for streaming playback purposes in this paper because DRL-TCP maximizes throughput in communication over the probability of HoL Blocking, even though high throughput can be achieved. Similar to DRL-TCP, QLE-DS also adopts and uses reinforcement learning, and as a result, it shows that it is not possible to flexibly respond to HoL Blocking. This is presumed to be due to the development of the model focusing on DASH techniques, although the goal of video transmission is the same as in this paper.

Fig. 5(b) shows the TCP throughput performance. TCP BBR, which is currently widely used, has the highest performance in terms of throughput. This is followed by the throughput performance improvements achieved by the proposed algorithm because it does not currently send the maximum amount that can be sent in order, rather it sends only the appropriate amount. However, it appears to exhibit much better performance than algorithms that use static control policies such as TCP Cubic and TCP Westwood. This is because the transmission amount was adaptively determined to fulfill the purpose of this study by observing the link statistics and state space in RL. Moreover, for DRL-TCP, we use reinforcement learning-based algorithms similar to the algorithm proposed in this system, but achieve maximum throughput in certain situations. Similarly, it can be seen that QLE-DS also performs slightly higher than the proposed algorithm in environments based on low error rates similar to global communication situations, but degrades with communication situations similar to the IoDST environment. Like TCP BBR, this performance can be achieved because it aims only at the maximum throughput in a specific communication. The proposed DQN-based MPTCP CC scheme performed slightly lower than the current most widely used TCP BBR. However, in terms of playback data streaming which is the main focus in this study, the proposed CC scheme performed up to 60% better than all variants of TCP. The probability of video playback and throughput is trade-off from the perspective of the problem defined in this paper. Comparing Fig. 4 and Fig. 5(a), it can be seen that this paper proposes a suitable compromise of two conditions over all other schemes.

Fig. 5(c) measured the time it took to transmit one full streaming file. Obviously, for TCP Westwood and TCP Cubic, the proposed scheme achieves up to 259% higher performance. This is from delay due to buffer loss or insufficient SVC conditions because the two comparison models did not achieve adequate throughput in the IoDST environment or transfer to suitable subflows to prevent HoL blocking. In terms of throughput, TCP BBR, DRL-TCP and QLE-DS which have higher performance in certain situation than the proposed scheme, are also lower than TCP Cubic and TCP Westwood. However, it causes higher delay than the proposed algorithm. Because these three models achieve higher throughput than other comparative models, they rather cause more HoL Blocking, resulting in packet loss and the dimensions of the state set of reinforcement learning are relatively larger than the proposed model, the complexity increases and the corresponding factors appear to be included in the delay. The proposed scheme shows up to 20%, 21% and 19% better performance than TCP BBR, DRL-TCP and QLE-DS, respectively.

6. Conclusion

In this paper we realize a MPTCP CC scheme for reliable video data streaming in IoDST communication. We propose a learning based solution based on DRL to determine an optimal transmission rate against the challenging link conditions in IoDST by considering SVC, where each layer of every chunk can be fetched from only one sub flow of MPTCP. The performance of the proposed MPTCP CC scheme was compared with existing fixed control policy schemes where the proposed scheme exhibited excellent performance in terms of streaming data playback. However, TCP BBR showed slightly better throughput performance when compared to the proposed MPTCP CC scheme. In addition, the corresponding



(a) HoL Blocking Percentage (b) Throughput (c) File Completion Time

Fig. 5. HoL Blocking Percentage, Throughput, File Completion Time performance of the various TCP protocol.

communication environment was realized through computer simulations, instead of real experiments or emulated tests. Thus, this may be another relevant research topic of our future studies.

CRediT authorship contribution statement

Taeyun Ha: Conceptualization, Writing – original draft, Software. **Arooj Masood:** Methodology, Investigation. **Woongsoo Na:** Writing – review & editing, Validation. **Sungrae Cho:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A4A5034130). This research was supported by the Chung-Ang University Research Scholarship Grants in 2023.

References

[1] I.F. Akyildiz, Ö.B. Akan, C. Chen, J. Fang, W. Su, InterPlaNetary internet: state-of-the-art and research challenges, Comput. Netw. 43 (2) (2003) 75–112.
[2] K. Bhasin, J.L. Hayden, Space internet architectures and technologies for NASA enterprises, Int. J. Satell. Commun. 20 (5) (2002) 311–332.
[3] S.-R. Cho, Adaptive forward error correction scheme for real-time communication in satellite IP networks, KSII Trans. Internet Inform. Syst. (TIIS) 4 (6) (2010) 1116–1132.
[4] O.B. Akan, J. Fang, I.F. Akyildiz, TP-Planet: A reliable transport protocol for interplanetary internet, IEEE J. Sel. Areas Commun. 22 (2) (2004) 348–361.
[5] V. Jacobson, Congestion avoidance and control, ACM SIGCOMM Comput. Commun. Rev. 18 (4) (1988) 314–329.
[6] S. Floyd, T. Henderson, A. Gurtov, The NewReno Modification to TCP's Fast Recovery Algorithm, Tech. rep., 2004.
[7] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP Selective Acknowledgment Options, Tech. rep., 1996, RFC 2018, IETF Network Working Group.
[8] L.S. Brakmo, S.W. O'Malley, L.L. Peterson, TCP Vegas: New techniques for congestion detection and avoidance, in: Proceedings of the Conference on Communications Architectures, Protocols and Applications, 1994, pp. 24–35.
[9] M. Mathis, J. Mahdavi, Forward acknowledgement: Refining TCP congestion control, ACM SIGCOMM Comput. Commun. Rev. 26 (4) (1996) 281–291.
[10] S. Mascolo, C. Casetti, M. Gerla, M.Y. Sanadidi, R. Wang, TCP Westwood: Bandwidth estimation for enhanced transport over wireless links, in: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, 2001, pp. 287–297.
[11] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, R.H. Katz, A comparison of mechanisms for improving TCP performance over wireless links, IEEE/ACM Trans. Netw. 5 (6) (1997) 756–769.
[12] G. Xylomenos, G.C. Polyzos, P. Mahonen, M. Saaranen, TCP performance issues over wireless links, IEEE Commun. Mag. 39 (4) (2001) 52–58.
[13] O.B. Akan, H. Fang, I.F. Akyildiz, Performance of TCP protocols in deep space communication networks, IEEE Commun. Lett. 6 (11) (2002) 478–480.
[14] C. Jin, D.X. Wei, S.H. Low, FAST TCP: motivation, architecture, algorithms, performance, in: IEEE INFOCOM 2004, Vol. 4, IEEE, 2004, pp. 2490–2501.
[15] I.F. Akyildiz, G. Morabito, S. Palazzo, TCP-Peach: a new congestion control scheme for satellite IP networks, IEEE/ACM Trans. Netw. 9 (3) (2001) 307–321.
[16] I.F. Akyildiz, X. Zhang, J. Fang, TCP-Peach+: Enhancement of TCP-Peach for satellite IP networks, IEEE Commun. Lett. 6 (7) (2002) 303–305.
[17] J. Wang, C. Jiang, H. Zhang, Y. Ren, V.C. Leung, Aggressive congestion control mechanism for space systems, IEEE Aerosp. Electr. Syst. Mag. 31 (3) (2016) 28–33.
[18] L. Grieco, S. Mascolo, A congestion control algorithm for the planetary internet, IFAC Proc. Vol. 38 (1) (2005) 119–124.
[19] L.A. Grieco, S. Mascolo, A congestion control algorithm for the deep space internet, Space Commun. 20 (3–4) (2006) 155–160.
[20] I. Psaras, G. Papastergiou, V. Tsaoussidis, N. Peccia, DS-TP: Deep-space transport protocol, in: 2008 IEEE Aerospace Conference, IEEE, 2008, pp. 1–13.
[21] Y.-s. Lim, E.M. Nahum, D. Towsley, R.J. Gibbens, ECF: An MPTCP path scheduler to manage heterogeneous paths, in: Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies, 2017, pp. 147–159.
[22] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, J. Liu, R. Li, A low-latency MPTCP scheduler for live video streaming in mobile networks, IEEE Trans. Wireless Commun. 20 (11) (2021) 7230–7242.
[23] S. Ha, I. Rhee, L. Xu, CUBIC: a new TCP-friendly high-speed TCP variant, Oper. Syst. Rev. 42 (5) (2008) 64–74.
[24] N. Cardwell, Y. Cheng, C.S. Gunn, S.H. Yeganeh, V. Jacobson, BBR: Congestion-based congestion control, Commun. ACM 60 (2) (2017) 58–66.
[25] Z. Xu, J. Tang, C. Yin, Y. Wang, G. Xue, Experience-driven congestion control: When multi-path TCP meets deep reinforcement learning, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1325–1336.
[26] C. Raiciu, M. Handley, D. Wischik, Coupled Congestion Control for Multipath Transport Protocols, Tech. Rep., 2011.

[27] Q. Peng, A. Walid, J. Hwang, S.H. Low, Multipath TCP: Analysis, design, and implementation, IEEE/ACM Trans. Netw. 24 (1) (2014) 596–609.
[28] R. Khalili, N. Gast, M. Popovic, J.-Y. Le Boudec, MPTCP is not Pareto-optimal: Performance issues and a possible solution, IEEE/ACM Trans. Netw. 21 (5) (2013) 1651–1665.
[29] Y. Cao, M. Xu, X. Fu, Delay-based congestion control for multipath TCP, in: 2012 20th IEEE International Conference on Network Protocols, ICNP, IEEE, 2012, pp. 1–10.
[30] S.R. Pokhrel, M. Mandjes, Improving multipath TCP performance over WiFi and cellular networks: An analytical approach, IEEE Trans. Mob. Comput. 18 (11) (2018) 2562–2576.
[31] C. Lee, J. Jung, J.-M. Chung, DEFT: Multipath TCP for high speed low latency communications in 5G networks, IEEE Trans. Mob. Comput. 20 (12) (2020) 3311–3323.
[32] J. Ye, L. Feng, Z. Xie, J. Huang, X. Li, Fine-grained congestion control for multipath TCP in data center networks, IEEE Access 7 (2019) 31782–31790.
[33] W. Wang, X. Wang, D. Wang, Energy efficient congestion control for multipath TCP in heterogeneous networks, IEEE Access 6 (2017) 2889–2898.
[34] T.A. Le, C.S. Hong, M.A. Razzaque, S. Lee, H. Jung, ecMTCP: An energy-aware congestion control algorithm for multipath TCP, IEEE Commun. Lett. 16 (2) (2011) 275–277.
[35] B. He, J. Wang, Q. Qi, H. Sun, J. Liao, C. Du, X. Yang, Z. Han, DeepCC: Multi-agent deep reinforcement learning congestion control for multi-path TCP based on self-attention, IEEE Trans. Netw. Serv. Manag. 18 (4) (2021) 4770–4788.
[36] M.M. Roselló, Multi-path scheduling with deep reinforcement learning, in: 2019 European Conference on Networks and Communications, EuCNC, IEEE, 2019, pp. 400–405.
[37] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, S. Lu, SmartCC: A reinforcement learning approach for multipath TCP congestion control in heterogeneous networks, IEEE J. Sel. Areas Commun. 37 (11) (2019) 2621–2633.
[38] A. Singh, C. Goerg, A. Timm-Giel, M. Scharf, T.-R. Banniza, Performance comparison of scheduling algorithms for multipath transfer, in: 2012 IEEE Global Communications Conference, GLOBECOM, IEEE, 2012, pp. 2653–2658.
[39] G. Sarwar, R. Boreli, E. Lochin, A. Mifdaoui, G. Smith, Mitigating receiver's buffer blocking by delay aware packet scheduling in multipath data transfer, in: 2013 27th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2013, pp. 1119–1124.
[40] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, R. Boreli, DAPS: Intelligent delay-aware packet scheduling for multipath transport, in: 2014 IEEE International Conference on Communications, ICC, IEEE, 2014, pp. 1222–1227.
[41] F. Yang, Q. Wang, P.D. Amer, Out-of-order transmission for in-order arrival scheduling for multipath TCP, in: 2014 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2014, pp. 749–752.
[42] B.-H. Oh, J. Lee, Feedback-based path failure detection and buffer blocking protection for MPTCP, IEEE/ACM Trans. Netw. 24 (6) (2016) 3450–3461.
[43] B.Y. Kimura, D.C. Lima, A.A. Loureiro, Packet scheduling in multipath TCP: Fundamentals, lessons, and opportunities, IEEE Syst. J. 15 (1) (2020) 1445–1457.
[44] J. Luo, X. Su, B. Liu, A reinforcement learning approach for multipath TCP data scheduling, in: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC, IEEE, 2019, pp. 0276–0280.
[45] J. Yao, S. Pang, J.J.P.C. Rodrigues, Z. Lv, S. Wang, Performance evaluation of MPTCP incast based on queuing network, IEEE Trans. Green Commun. Netw. 6 (2) (2022) 695–703, http://dx.doi.org/10.1109/TGCN.2021.3125860.
[46] I. Khan, K. Chen, EBA: Efficient bandwidth aggregation for connected vehicles with MPTCP, IEEE Internet Things J. 9 (8) (2022) 5812–5823, http://dx.doi.org/10.1109/JIOT.2021.3065911.
[47] Y. Cao, R. Ji, L. Ji, G. Lei, H. Wang, X. Shao, l^2-MPTCP: A learning-driven latency-aware multipath transport scheme for industrial internet applications, IEEE Trans. Ind. Inform. 18 (12) (2022) 8456–8466, http://dx.doi.org/10.1109/TII.2022.3151093.
[48] S. Singh, M.K. Maheshwari, M.R. Kanagarathinam, A.R. Masud, I. Qasim, NexGen S-MPTCP: Next generation smart multipath TCP controller, IEEE Access 10 (2022) 96079–96090, http://dx.doi.org/10.1109/ACCESS.2022.3205418.
[49] L. Zhong, X. Ji, Z. Wang, J. Qin, G.-M. Muntean, A Q-learning driven energy-aware multipath transmission solution for 5G media services, IEEE Trans. Broadcast. 68 (2) (2022) 559–571, http://dx.doi.org/10.1109/TBC.2022.3147098.
[50] S. Aggarwal, S.K. Saha, I. Khan, R. Pathak, D. Koutsonikolas, J. Widmer, MuSher: An agile multipath-TCP scheduler for dual-band 802.11ad/ac wireless LANs, IEEE/ACM Trans. Netw. 30 (4) (2022) 1879–1894, http://dx.doi.org/10.1109/TNET.2022.3158678.
[51] C. Xu, J. Qin, P. Zhang, K. Gao, L.A. Grieco, Reinforcement learning-based mobile AR/VR multipath transmission with streaming power spectrum density analysis, IEEE Trans. Mob. Comput. 21 (12) (2022) 4529–4540, http://dx.doi.org/10.1109/TMC.2021.3082912.
[52] Y. Huo, C. Hellge, T. Wiegand, L. Hanzo, A tutorial and review on inter-layer FEC coded layered video streaming, IEEE Commun. Surv. Tutor. 17 (2) (2015) 1166–1207.
[53] P. Rahimzadeh, C. Joe-Wong, K. Shin, Y. Im, J. Lee, S. Ha, SVC-TChain: Incentivizing good behavior in layered P2P video streaming, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.
[54] C. Xu, Z. Li, J. Li, H. Zhang, G.-M. Muntean, Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks, IEEE Trans. Circuits Syst. Video Technol. 25 (7) (2014) 1175–1189.
[55] C. Zhan, K. Gao, Video delivery in heterogeneous wireless networks with network coding, IEEE Wirel. Commun. Lett. 5 (5) (2016) 472–475.
[56] B. Fu, D. Staehle, G. Kunzmann, E. Steinbach, W. Kellerer, QoE-aware priority marking and traffic management for H. 264/SVC-based mobile video delivery, in: Proceedings of the 8th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, 2013, pp. 173–180.
[57] K. Poularakis, G. Iosifidis, A. Argyriou, L. Tassiulas, Video delivery over heterogeneous cellular networks: Optimizing cost and performance, in: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, 2014, pp. 1078–1086.
[58] K. Zhu, L. Li, Y. Xu, T. Zhang, L. Zhou, Multi-connection based scalable video streaming in UDNs: A multi-agent multi-armed bandit approach, IEEE Trans. Wireless Commun. 21 (2) (2022) 1156–1169, http://dx.doi.org/10.1109/TWC.2021.3102626.
[59] S. Chatterjee, S. De, QoE-aware cross-layer adaptation for D2D video communication in cooperative cognitive radio networks, IEEE Syst. J. 16 (2) (2022) 2078–2089, http://dx.doi.org/10.1109/JSYST.2021.3123463.
[60] A. Elgabli, M. Felemban, V. Aggarwal, GiantClient: Video HotSpot for multi-user streaming, IEEE Trans. Circuits Syst. Video Technol. 29 (9) (2019) 2833–2843.
[61] F.A. Miranda, Advanced communications technologies in support of NASA mission, in: The European Conference on Antennas and Propagation, EuCAP 2018, (GRC-E-DAA-TN53779) 2018.
[62] J. Hamkins, L. Deutsch, D. Divsalar, S. Dolinar, D. Lee, F. Stocklin, J. Wesdock, C. Patel, Formulation of forward error correction coding recommendations for future NASA space communications, in: 2008 IEEE Aerospace Conference, IEEE, 2008, pp. 1–18.
[63] H. Hemmati, Deep Space Optical Communications, John Wiley & Sons, 2006.