International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/39 Volume 11, Issue 5, September – October (2024)

RESEARCH ARTICLE

Reinforcement Learning based Adaptive Congestion Control for TCP over Wireless Networks

Hardik K. Molia
Computer Engineering Department, Government Engineering College, Rajkot, Gujarat, India.
hardik.molia@gmail.com

Received: 22 June 2024 / Revised: 15 September 2024 / Accepted: 04 October 2024 / Published: 30 October 2024

Abstract – Over the years, it has been observed that standard protocols designed for wired networks do not perform adequately when used for wireless networks. Researchers have proposed various protocols to enhance the functionalities of wireless network layers. TCP-Transmission Control Protocol is a transport layer protocol that experiences significant performance degradation in wireless networks. This is primarily because TCP considers any packet loss as a cause of network congestion, leading to an unnecessary reduction in transmission rate even when losses occur due to other reasons. This research work focuses in reviewing the existing approaches for improvement of TCP Congestion Control for wireless networks along with proposing TCP-RLACC (TCP with Reinforcement Learning based Adaptive Congestion Control). TCP-RLACC explores the network to select the most appropriate growth (linear, quadratic polynomial or exponential) of Cwnd – Congestion Window for adjusting transmission rate. TCP-RLACC is implemented in NS-3 simulator and evaluated with TCP Westwood+ for large number of wireless network scenarios. TCP-RLACC has shown significant improvements in terms of average throughput and end to end packet delivery ratio.

Index Terms – TCP, Congestion Control, Reinforcement Learning, Cwnd-Congestion Window, Throughput, Packet Delivery Ratio.

1. INTRODUCTION

Computer Networks were initially implemented as wired networks where the physical connections of cables carry the signals for the communications. With the introduction of wireless networks, the medium of communication is changed to the air. Wireless networks provide more flexible and convenient infrastructure as compared to the wired networks. While wired networks are preferred for large scale networks, wireless networks are preferred for networks such as home or office networks, sensor networks, ad hoc networks or IoT networks. Some wireless networks such as MANETs - Mobile Ad hoc Networks provide mobility without dedicated network infrastructure for setup which makes them suitable to form networks on temporary or emergency basis [1-3].

Computer networks, whether wired or wireless are operative with protocols at different layers of the service models to provide specific services in coordination with the available hardware resources. The protocols designed for wired networks may not provide satisfactory performances when used for wireless networks due to fundamental differences between wired and wireless communications. Wired networks are more reliable and stable due to physical connections whereas wireless networks are more vulnerable from transmission issues such as noise, attenuation, interference, physical obstacles etc. Wireless networks introduce high error rates and packet losses due to issues such as signal fading, multipath propagation, and collisions among wireless transmissions. One challenging issue in wireless networks is congestion control where it is necessary to identify whether packet losses are due to congestion or any other issues. Wired networks are configured with static topologies whereas wireless networks, specifically MANETs have dynamic topologies allowing nodes to move while being involved in communications. Routing is challenging in wireless networks due to mobility patterns and network partitions. As compared to wired networks, wireless networks often experience bandwidth and throughput constraints due to limited frequency bands. Wireless networks are inherently more vulnerable to unauthorized access and security issues compared to wired networks. As the wireless networks are more challenging to deploy with satisfactory performances, researchers have been actively participating towards proposing solutions to address specific issues [2-5].

TCP - Transmission Control Protocol is a transport layer protocol to provide process-to-process communication between two end devices. TCP provides congestion control, flow control and error control to ensure reliable communication. Congestion control and flow control set the transmission rate as per the capacities of network and receiver respectively. Error control supports acknowledgement based retransmissions. Over the years, numerous TCP variants are proposed to improve end to end performance for wireless networks. Many of these solutions are based on some heuristics or static rules, enabling them to improve performance for some specific type of wireless networks or scenarios. Proposing a TCP solution which is adaptive for different wireless networks and diverse set of scenarios has been an active area of research [6-8].

Adaptive TCP observes the real time network conditions to adjust its mechanism for better decision making. As TCP is strongly associated with end to end congestion control, network conditions such as throughput, delay, error rate, ACK patterns can be monitored to set transmission rate accurately. Though many existing TCP variants are adaptive, they are often limited in their adaptability to handle dynamic network conditions effectively or may not be effective for wireless networks [6-8].

This research work reviews existing approaches for improvement of TCP Congestion Control for wireless networks along with proposing a reinforcement learning based solution for adaptive congestion control. The proposed solution is named TCP-RLACC (TCP with Reinforcement Learning based Adaptive Congestion Control). TCP-RLACC explores the network to select the most appropriate growth (linear, quadratic polynomial or exponential) of Cwnd – Congestion Window for adjusting transmission rate. This solution is expected to be adaptive to form its decision making policy for any network through Reinforcement Learning. It is implemented in NS-3 simulator and evaluated with large number of wireless network scenarios. Section-2 discusses review of existing approaches. Section-3 discusses the proposed solution and Section-4 discusses results with key observations. Section-5 is conclusion.

2. RELATED WORK

This section reviews some of the recent work related with improvements of TCP's congestion control. The review starts with discussing how ML-Machine Learning can be used for networking followed by discussions of recent TCP variants. The discussion covers TCP variants that do not utilize ML, followed by those that do incorporate ML techniques.

2.1. Machine Learning and Networking

A workflow for ML for networking is given [1] to discuss how RL-Reinforcement Learning is suitable for designing a congestion control algorithm that fits all network states. A discussion on feasibility is also given with reference of computation load. A state-of-the-art discussion on using ML in communication networks is given along with future directions [2]. A detail discussion on basis of TCP and types of TCP CC-Congestion Control algorithms is given [3]. The proposed algorithms are classified into loss-based algorithms, delay-based algorithms and hybrid algorithms (combination of loss-based and delay-based methods). This work also discussed how online learning based TCP solutions are better as compared with offline learning based TCP solutions due to their abilities to adapt to the network conditions on the fly instead of using fixed mappings and pre-defined actions. Application of ML in Wireless Networks along with open issues is given [4]. It is stated that ML in wireless communications is still at initial stage and needs further investigation. A review of ML for End to end Congestion Control is given [5]. It is discussed that the conventional rule-based algorithms are more susceptible to unpredictable factors, resulting in poor performance. ML based algorithms can learn from network environment to take decision accurately. A review of TCP performance enhancement in IoT and MANET is given highlighting approaches, tools and open issues [6]. A discussion on ML algorithms for wireless sensor networks and challenges of security is given [7]. Role of ML algorithms for WSN and IoT is discussed along with analysis of issues and future directions [8].

TCP Congestion Window is a TCP state variable to set transmission rate of sender. It is number of bytes TCP sender can transmit before waiting for an acknowledgement. The Congestion Control algorithm used by TCP sets the congestion window value dynamically by monitoring the network statistics. Throughout the paper, the term *Cwnd* will refer to the Congestion Window, *RTT* will refer to Round Trip Time, *RL* will refer to Reinforcement Learning.

2.2. TCP Variants – Traditional Approaches

NexGen D-TCP - Next Generation Dynamic TCP congestion control algorithm has adaptive increase adaptive decrease paradigm to adjust *Cwnd* by estimating accessible bandwidth [9]. This approach enables full utilization of bandwidth with minimization of packet loss due to congestion and wireless loss. A discussion on TCP for Low-Power and Lossy Networks is given [10]. The main objective of this work is to address challenges like fitting full scale TCP in limited memory of LLN platforms and integration of TCP from traditional OS to embedded OS. It is shown that modern low-power sensor platforms are capable of running full-scale TCP efficiently. TCP-NACK – TCP with Negative Acknowledgement is proposed to differentiate congestion losses from other losses due to issues related to wireless communications [11]. NACK notification informs the sender for reception of corrupted packets so they can be retransmitted immediately without reducing *Cwnd*. FAIR+ algorithm initiates congestion control based on queue level notification by relay node and sets value of *Cwnd* based on TCP flow's utilization level [12]. This approach is ECN - Explicit Congestion Notification based and implemented with DQDM, GBRC, and AWR algorithm. DQDM - Dual Queue Dual Marking algorithm analyzes congestion severity, GBRC - Growth Based Rate Control algorithm initiates *Cwnd* reduction, and the AWR algorithm implements faster recovery by increasing the *Cwnd*. BCCPS is BBR-based Congestion Control and Packet Scheduling scheme [13]. BBR - Bottleneck Bandwidth and Round-trip propagation time is used for dynamically adjusting the sending rates of each sub-flow in MPTCP - Multipath TCP according to real probing

RESEARCH ARTICLE

rates, rather than relying solely on loss information. The packet scheduling scheme is used for managing the order of packet delivery in heterogeneous wireless networks. BBRp is a new BBR version that allows fine-tuning the congestion control pace to correctly aggregate packets at the wireless bottleneck and exploit the bottleneck bandwidth [14]. Dynamic adjustment of *Cwnd* for TCP Westwood is given by calculating threshold value using *RTT* – Round Trip Time value. Binomial algorithm is introduced for nonlinear growth of *Cwnd* [15]. TCP CERL+ is an advancement to TCP CERL - Congestion Control Enhancement for Random Loss that uses dynamic threshold for *RTT* which is calculated as average *RTT* and its minimum measurements made over the connection to estimate the queue length of the bottleneck link for congestion evaluation [16]. TCP-TACK presents a new acknowledgement called Tame ACK [17]. The conventional ACK scheme is received packet driven. TACK balances byte counting ACK and periodic ACK for controlled ACK frequency that improves performance by handling contention related issues effectively. This scheme reduces number of ACKs while carrying more information in ACKs. TCP-WBQ is a backlog queue based congestion control mechanism for heterogeneous wireless networks to notice real congestion and shield against random packet loss and oscillations of latency [18]. Backlog queue is maintained at sender and does not require ECN capability. This approach introduces multiplicative increase instead of additive increase of *Cwnd*. TCP-LoRaD is a Loss Recovery and Differentiation algorithm for TCP improvement over MANETs under noisy channels [19]. This solution uses loss differentiation algorithm to identify the packet loss more efficiently as compared to TCP-WELCOME. TCP-AW is a combination of TCP Westwood and Adaptive Reno. The lightweight ERE - Eligible Rate Estimation calculation resembles TCP-AW but does not require ACK history. An improved TCP optimization with flow control and acknowledgment aggregation is proposed for removing the bottleneck of packet losses through flow control and AMC -Adaptive Modulation Coding [20]. This solution is specifically designed to communicate over time varying channels.

2.3. TCP Variants – Machine Learning based Approaches

QTCP is a reinforcement learning based congestion control solution to set *Cwnd* for high throughput and low delay using Q-Learning [21]. The state space has variables such as average interval between sending two packets, average interval between receiving two ACKs and average *RTT*. The action space has three options to increase, decrease and no change in *Cwnd*. Reward function is formed using throughput and *RTT*. This solution is evaluated with TCP NewReno. TCP-Drinc -Deep ReInforcement learNing-based Congestion control adjusts *Cwnd* efficiently [22]. The state space has variables such as *Cwnd* difference, *RTT*, minimum *RTT* over *RTT* ratio, inter-arrival time of ACKs etc. The action space has five options to set increase / decrease *Cwnd* linearly / exponentially and no change. The reward function is formed using *RTT* and goodput. This solution is evaluated with TCP NewReno, TCP Cubic, TCP Hybla, TCP Vegas and TCP Illinois. Eagle is deep reinforcement learning based solution [23]. The state space has variables related with *RTT*, loss rate, delay, delivery rate etc. The action space has seven options to increase / decrease *Cwnd* by a factor of 2.89, 1.5, 1.05 and no change. The reward function is formed using goodness that is calculated as the ratio between the current *Cwnd* and the *Cwnd* that gave us the best utility. This solution is evaluated with TCP BBR and some other variants too. ML techniques (PCA - Principal Component Analysis, LR - Linear Regression and RF - Random Forest) are used to study the effect of link speed, received signal strength, *RTT*, and number of available access points on TCP throughput in WiFi [24]. Cell vs WiFi dataset is used for all three types of analyses. The first principal component from PCA is observed to be highly correlated with *RTT*. It is observed that LR model is unable to find relationship between throughput and other variables. It is also observed that RF model predicts throughput more accurately with *RTT* as compared to other variables. TCP - PPO2 -TCP with Proximal Policy Optimization algorithm is deep reinforcement learning based solution [25]. The state space has variables such as connection time, current size of *Cwnd*, number of unacknowledged bytes, *RTT*, throughput, number of packets losses etc. The action space has various levels of increase or decrease of *Cwnd* based on speed of network. Reward function is formed using observed throughput and latency. This solution is evaluated with TCP Cubic. PBQ-Enhanced QUIC is QUIC with Deep Reinforcement Learning Congestion Control Mechanism [26]. PBQ - Proximal Bandwidth-delay Quick optimization combines traditional bottleneck bandwidth and round-trip propagation time of BBR with PPO - Proximal Policy Optimization. PPO agent improves itself according to network state to set value of *Cwnd*. BBR specifies the pacing rate of the client. The state space has variables such as current *Cwnd*, average *RTT*, average packet loss, average throughput etc. The action space is continuous with *CwndRatio* is used to set value of *Cwnd*. Reward function is formed using observed throughput and packet loss rate. This solution is evaluated with QUIC variants with (TCP-BBR, TCP-BIC, TCP-Cubic, LEDBAT, TCP-NewReno, Remy and TCP-Vegas). Deep Reinforcement Learning Based TCP Congestion Control in UAV Assisted Wireless Networks is proposed for improving average throughput and reducing average latency by controlling the number of packets passing through the bottleneck link between UAV base stations [27]. DRLFcc is proposed for facilitating real-time adaptation of the congestion window for dynamic changes in network conditions while incorporating fast recovery mechanisms, thereby effectively enhancing network throughput and improving data transmission capacity recovery in high-loss

RESEARCH ARTICLE

wireless networks [28]. A distributed congestion control protocol is proposed using CatBoost algorithm that uses gradient boosting on decision trees [29]. Decision trees are used to predict whether the packets to be transmitted over the network will reach their destination on time or not. Network load state is represented with parameters related with utilization of transmission channels and occupancy of buffers. A dissemination protocol is also designed to make parameters available to all nodes in the network. A real time packet loss prediction is designed with ECN - Explicit Congestion Notification to inform a sender to reduce transmission rate before packet loss would have happened [30]. XGBoost model is used with training data arranged by running emulated iPerf TCP tests using mininet network emulation. 18 features related with values of *Cwnd*, *RTT*, *SSThresh*, *RTO* etc. are used. This solution is evaluated with TCP Reno and TCP Cubic. Table 1 shows outcome of literature review in a concise manner.

Table 1 Machine Learning based TCP

No.	Type of TCP solutions	Advantages	Limitations
1	Traditional TCP Variants	Suitable for wired networks. They can handle congestion effectively.	Do not perform satisfactorily when used for wireless networks. Not able to handle channel losses, random errors effectively.
2	TCP Variants using Supervised Learning	Suitable to include decision making capability to set parameters based on network conditions more accurately.	Do not perform satisfactorily when used for unknown network scenarios. Training dataset arrangement is a critical task. So, these variants become useful for limited types of network scenarios for which they are trained.
3	TCP Variants using Reinforcement Learning	Suitable to include decision making capability to set parameters based on network conditions more accurately. Suitable to find solutions which are adaptive and generalize for any network scenario.	These variants can also be used for unknown network scenarios as they will form their decision making policies by exploring the environments. There is no need of prior training so dataset is not required. But these algorithms take time to form accurate policy.

TCP Westwood+ differentiates congestion losses from random losses that are common in wireless networks by AIAD – Adaptive Increase / Adaptive Decrease congestion control. Westwood+ estimates network's bandwidth to adjust value of Cwnd. TCP Westwood performs bandwidth sampling every ACK while TCP Westwood+ performs bandwidth sampling every *RTT*, leading to more accurate estimations. TCP Westwood+ is selected as base variant to implement our proposed solution.

3. PORPOSED SOLUTION

RL-Reinforcement Learning is a branch of ML-Machine Learning to enable an agent to interact with its environment to learn through trials. RL algorithms do not need prior training phase as needed in supervised learning algorithms. An RL agent explores its environment to form a decision making policy to select next action based on current state. To form decision making policy, RL agent needs to receive feedback from the environment that is in the form of rewards or penalties. RL algorithm forms a decision making policy that maximizes cumulative reward on long run. The component of an RL based system include the agent, the environment, state space, action space and a reward signal. The general flow of RL based system is shown in Figure 1 and Figure 2 [31].

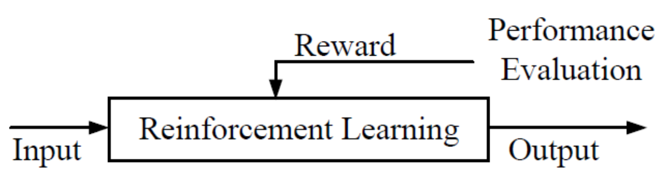

Figure 1 Reinforcement Learning [31]

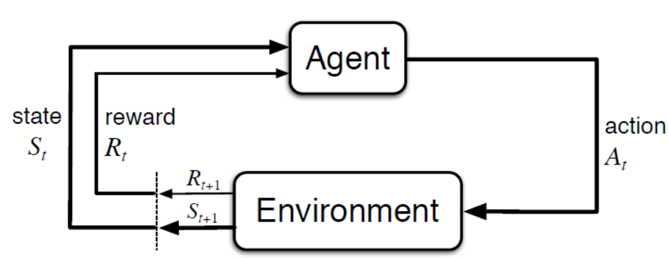

Figure 2 Agent Environment Interaction [31]

RESEARCH ARTICLE

The proposed work is TCP-RLACC - TCP with Reinforcement Learning based Adaptive Congestion Control. The components of TCP-RLACC along with its algorithm are discussed in this section.

3.1. TCP-RLACC Environment & Agent

TCP-RLACC learns from the network over which it facilitates end to end communication. This solution is specifically designed to improve end to end performance in wireless networks, so the environment is any wireless network under which TCP based communication is initiated. TCP based congestion control is primarily implemented with TCP Sender to set transmission rate by setting value of *Cwnd*. The agent of TCP-RLACC is congestion control algorithm at TCP sender.

3.2. TCP-RLACC State Space

As TCP is end to end protocol for which the statistics at the intermediate nodes and specific to the lower layers are not accessible. Though researchers have proposed cross-layer solutions for TCP, a wide range of TCP variants are still layered solutions. This makes the environment partially observable for TCP.

TCP-RLACC forms state based on analyzing values of *Cwnd* – Congestion Window, *RTT*-Round Trip Time and *RTO* – Retransmission Time Out status. The state is formed as a binary number of three bits – *B2 B1 B0*. So the state space has 8 possible states.

B2 represents change in EWMA - Exponentially Weighted Moving Average of *RTT* value. At any particular moment, current value of *EWMA of RTT* is calculated as shown in equation (1).

$$EWMA_{RTT_{New}} = (1-\alpha) * EWMA_{RTT_{Old}} + (\alpha) * RTT_{New} \quad (1)$$

B2 is set to 1 if $EWMA_{RTT_{New}} > EWMA_{RTT_{Old}}$, otherwise 0.

B1 represents change in *Cwnd*. It is set 1 if *Cwnd* is decreased, otherwise 0.

B0 is set to 1 if *RTO* is occurred, otherwise 0.

The state is formed as binary number from bits *B2 B1 B0*. TCP-RLACC keeps maintaining state as and when new or updated values of *RTT*, *Cwnd* and *RTO* status are available. So it is a continuous process of deriving state. The reason of transforming continuous values in to a single discrete number representation is to simplify the state space by enabling finite number of possibilities.

Moreover, TCP-RLACC is RL implementation within a network protocol, specifically for wireless networks, so it should be as light as possible due to resource constraints and for faster execution.

3.3. TCP-RLACC Action Space

TCP-RLACC follows congestion control mechanism of underlying algorithm till any of these three conditions happen. 1. Retransmission Time Out, 2. Arrival of three DUPACKs, 3. *Cwnd* reaches *SSThresh*. TCP variants which are based on AIMD- Additive Increase Multiplicative Decrease has two types of increases (linearly and exponentially) for *Cwnd*. TCP-RLACC introduces increasing *Cwnd* with Quadratic Polynomial growth, where *Cwnd* is increased faster than linear growth and slower than exponential growth.

Linear Growth of *Cwnd* is considered as shown in equation (2).

$$Cwnd_{New} = Cwnd_{Old} + 1 \quad (2)$$

Quadratic Polynomial Growth of *Cwnd* is considered as shown in equation (3).

$$Cwnd_{New} = Cwnd_{Old} * 0.1 * (No.of\ RTTs\ since\ last\ update)^2 \quad (3)$$

Exponential Growth of *Cwnd* is considered as shown in equation (4).

$$Cwnd_{New} = Cwnd_{Old} * 2 \quad (4)$$

TCP-RLACC action space has five possibilities. 1. Increase *Cwnd* with linear growth, 2. Increase *Cwnd* with quadratic polynomial growth, 3. Increase *Cwnd* with exponential growth. 4. Decrease *Cwnd* by dividing it by 2 and 5. Decrease *Cwnd* by setting it to its initial value.

3.4. TCP-RLACC Utility and Reward

The utility is calculated for the duration of communication between two consecutive actions taken by TCP-RLACC. It is formed with average values of throughput, *RTT* and loss rate. Utility is calculated as shown in equation (5) with constant values show the importance factors for respective values.

$$Utility_{New} = 0.5 * Avg_{Throughput} + 0.3 * Avg_{RTT} + 0.2 * Avg_{Lossrate} \quad (5)$$

Reward is 1 if $Utility_{New} > Utility_{Old}$, otherwise 0.

3.5. TCP-RLACC Algorithm

TCP-RLACC is inspired from Q-TCP [21] which uses Q-Learning algorithm for learning purpose. The Q-matrix is a tabular representation of state-action spaces. The state space of Q-TCP is formed using avg_send: average interval between sending two packets, avg_ack: average interval between receiving two consecutive ACKs and avg_rtt: the average RTT. The action space of Q-TCP is formed to increase Cwnd by 10, to decrease Cwnd by -1 or no change in Cwnd. Utility function is formed using throughput and RTT values. This work has been main source of interest to propose TCP-RLACC.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/39 Volume 11, Issue 5, September – October (2024)

RESEARCH ARTICLE

TCP-RLACC's Q-matrix has 8 states and 5 actions leading to a matrix of 8 rows X 5 columns. Initially this matrix is initialized with 0. Subsequently, TCP-RLACC keeps updating this matrix (each of the 40 values) as it progresses with learning via interacting with the environment. Initial Q-Matrix is given in Table 2.

Table 2 TCP-RLACC – Initial Q Matrix

Q-Matrix	Action as per Section 3.3				
State as per section 3.2	1	2	3	4	5
000	0	0	0	0	0
001	0	0	0	0	0
…..	…..	…..	…..	…..	…..
111	0	0	0	0	0

The flow of TCP-RLACC is shown in Figure 3. TCP-RLACC is shown in algorithm 1. It is based on Q-Learning is shown below. α is learning rate which is most commonly set as 0.9. γ is discount factor which is most commonly set as 0.5.

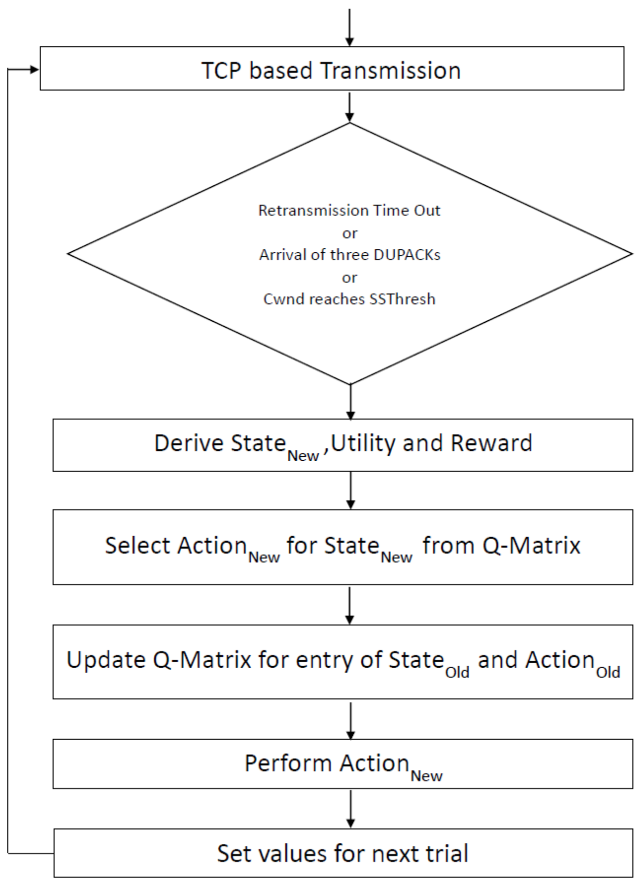

Figure 3 TCP-RLACC Flow

On any of these three conditions:

1. Retransmission Time Out,

2. Arrival of three DUPACKs,

3. *Cwnd* reaches *SSThresh*, follow below steps.

1. Derive $State_{New}$ as per section 3.2

2. Derive Utility and Reward as per section 3.4

3. Select $Action_{New}$ from Q-Matrix (Action is selected for $State_{New}$ with maximum value).

4. Update Q

   $T1 = (1 - \alpha) * Q\ [State_{Old}]\ [Action_{Old}]$

   $T2 = \alpha * (Reward + (\gamma * Q\ [State_{New}]\ [Action_{New}]\ ))$

   $Q\ [State_{Old}]\ [Action_{Old}] = T1 + T2$

5. Perform $Action_{New}$ as per section 3.3

6. Set values for next trial for learning

   $State_{Old} = State_{New}$

   $Action_{Old} = Action_{New}$

   $Utility_{Old} = Utility_{New}$

Algorithm 1 TCP-RLACC

4. RESULTS AND DISCUSSIONS

TCP-RLACC is implemented with NS 3.28 simulator and evaluated with large number of wireless network scenarios to compare its performance with TCP Westwood+ [32,33]. TCP Westwood+ has got wide acceptance due to its ability to improve performance significantly over wireless networks. TCP-RLACC is implemented with TCP NewReno which is AIMD based traditional TCP variant. This section includes details of experimental setup and performance analysis.

4.1. Experimental Setup

The experimental setup is divided into two parts: WiFi Networks and MANETs – Mobile Ad hoc Networks. WiFi Networks are infrastructure based without mobility of nodes whereas MANETs are infrastructure-less with node mobility. Table 3 shows parameters for network scenarios of experimental setup. In addition, MANETs have random mobility of nodes with speed of 5 meters / second.

Table 3 Experimental Setup – Parameters of Networks

No.	Parameter	Values
1	(No. of Nodes , TCP Flows)	(5,2), (10,5), (15,7)
2	Simulation Time (Second)	60, 120, 180
3	Loss Range (Meter)	70, 60, 50
4	Received Signal Strength (dBm)	-30, -40, -50

RESEARCH ARTICLE

Table 4 shows Experimental Setup summary. The total scenarios are setup with all possible combination of selected values of parameters mentioned in Table 3. All scenarios are evaluated twice (WiFi and MANETs). 1st with TCP Westwood+ and then with TCP-RLACC. Table 4 is summary for one TCP variant.

Table 4 Experimental Setup –Summary

.	Point	WiFi	MANETs	Total
1	No. of scenarios	81	81	162
2	No. TCP flows	378	378	756
3	Simulation (Hours)	162	162	324

4.2. Performance Analysis

As discussed in 4.1, performance of TCP-RLACC is evaluated with TCP Westwood+ for a large number of network scenarios differ in terms of number of nodes, TCP flows, simulation time and other parameters. Evaluating individual TCP flow and presenting observation in a tabular or graphical manner would have been very complex. At the same time, it would be difficult to justify the overall impact of TCP-RLACC.

Average performances of both TCP variants under evaluation are measured at different groups of network scenarios for comprehensive presentation. This leads us to observer some interesting characteristics of both approaches based on their unique features. Some of the important performance analysis and observations are shown in the subsequent tables and figures.

4.3. Average Throughput (Kbps)

Table 5 and Figure 4 represent results in terms of average throughput (Kbps) grouping network scenarios based on number of nodes. Table-6 and Figure-5 represent results in terms of average throughput (Kbps) grouping network scenarios based on simulation time.

Table 5 Performance Analysis -1 – Number of Nodes

No.	Average Throughput (Kbps)	TCP Westwood+	TCP-RLACC
1	WiFi (5 Nodes)	2.46	2.68
2	MANETs (5 Nodes)	2.22	2.43
3	WiFi (10 Nodes)	2.17	2.56
4	MANETs (10 Nodes)	1.87	1.93
5	WiFi (15 Nodes)	1.34	1.43
6	MANETs (15 Nodes)	1.24	1.28

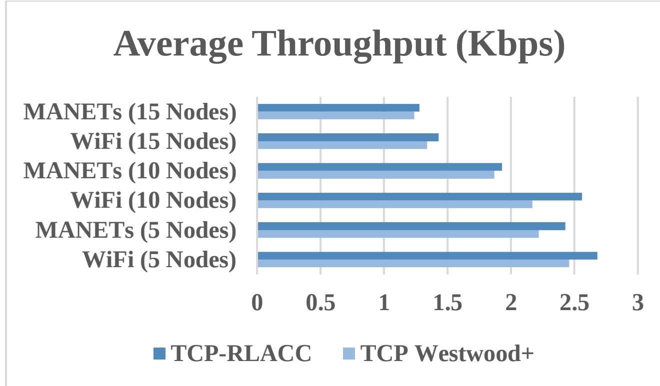

Figure 4 Performance Analysis -1 – Number of Nodes

Table 6 Performance Analysis -3 – Simulation Time

No.	Average Throughput (Kbps)	TCP Westwood+	TCP-RLACC
1	WiFi (60 Seconds)	2.65	2.55
2	MANETs (60 Seconds)	2.53	2.48
3	WiFi (120 Seconds)	2.78	2.89
4	MANETs (120 Seconds)	2.59	2.67
5	WiFi (180 Seconds)	2.82	2.95
6	MANETs (180 Seconds)	2.64	2.83

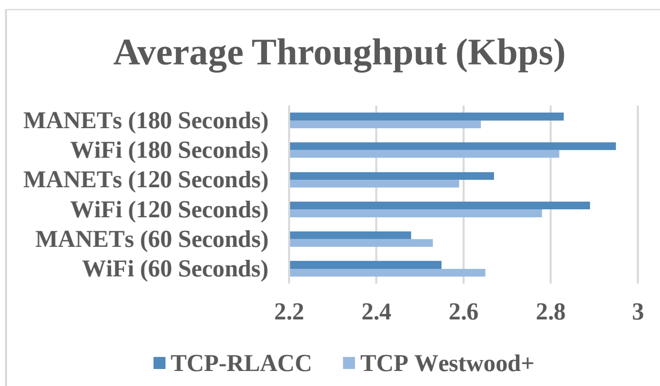

Figure 5 Performance Analysis -3 – Simulation Time

4.4. Packet Delivery Ratio (%)

Table-7 and Figure-6 represent results in terms of packet delivery ratio (%) grouping network scenarios based on number of nodes. Table-8 and Figure-7 represent results in terms of packet delivery ratio (%) grouping network scenarios based on simulation time.

RESEARCH ARTICLE

Table 7 Performance Analysis -2 – Number of Nodes

No.	Packet Delivery Ratio (% - Percentage)	TCP Westwood+	TCP-RLACC
1	WiFi (5 Nodes)	97.34	97.78
2	MANETs (5 Nodes)	96.53	96.88
3	WiFi (10 Nodes)	97.23	97.45
4	MANETs (10 Nodes)	95.34	96.31
5	WiFi (15 Nodes)	95.12	95.56
6	MANETs (15 Nodes)	94.63	94.77

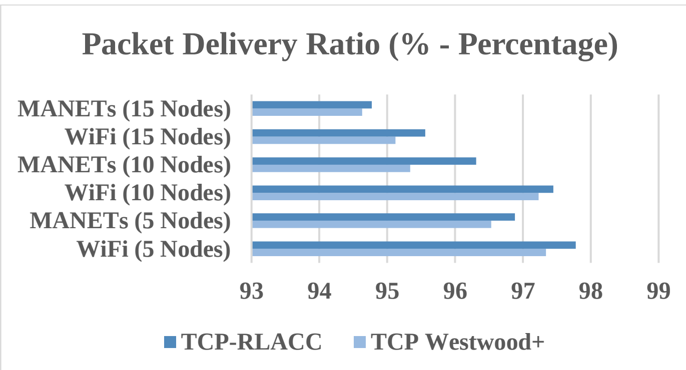

Figure 6 Performance Analysis -2 – Number of Nodes

Table 8 Performance Analysis -4 – Simulation Time

No.	Packet Delivery Ratio (% - Percentage)	TCP Westwood+	TCP-RLACC
1	WiFi (60 Seconds)	95.34	95.47
2	MANETs (60 Seconds)	94.89	95.12
3	WiFi (120 Seconds)	95.76	95.94
4	MANETs (120 Seconds)	95.18	95.36
5	WiFi (180 Seconds)	96.69	96.95
6	MANETs (180 Seconds)	96.54	96.82

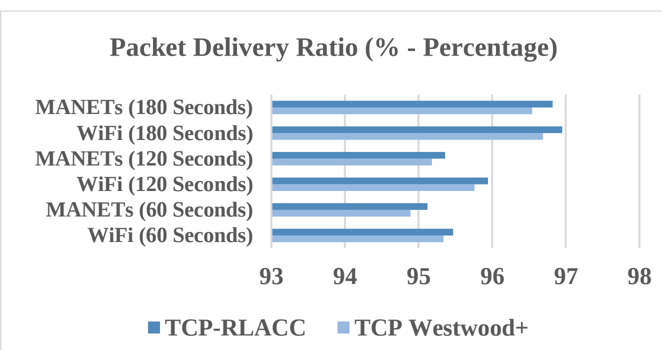

Figure 7 Performance Analysis -4 – Simulation Time

4.5. Discussions

Performance analysis of TCP-RLACC with significantly large number of network scenarios enabled us to accept it as an improvement over TCP Westwood+. The adaptive nature of RL algorithm and diverse network scenarios in terms of number of nodes, number of TCP flows, simulation time, and other relevant parameters made our analysis more generic. Following are some key observations derived through this research work.

1. TCP Westwood+ is one of the most widely accepted TCP variants for wireless networks. TCP-RLACC has shown a significant improvement over it. Average performances are measured and compared over subsets of diverse network scenarios for a more comprehensive analysis. The average performances are taken into result analysis as they represent overall impact of various solutions and useful to deduct more generic observations, rather than deducting observations from a few network scenarios with very less number of TCP connections.

2. Two categories of wireless networks have been implemented to evaluate performance of TCP-RLACC: WiFi Networks and MANETs. The observation was that average throughput and packet delivery ratio are higher in WiFi Networks compared to MANETs for same scenarios. This is mainly due to the infrastructure-less nature and node mobility support of MANETs.

3. Any RL algorithm needs time to develop an accurate policy through interaction with its environment. Over the time, RL algorithm gets more opportunities to fine-tune its decision making policy for convergence. The same is reflected in our work as performances of same network scenarios over different simulation times. TCP-RLACC performance is more improved for long running scenarios.

5. CONCLUSION

Over the years, improving the performance of TCP for wireless networks has remained an active area of research. Recently, many solutions have begun to incorporate machine learning for more accurate decision-making. Our proposed solution, TCP-RLACC, is a reinforcement learning-based approach for adaptive congestion control. It has been evaluated against TCP Westwood+, a widely accepted solution for wireless networks. The performance analysis was conducted for two types of wireless networks: WiFi networks and MANETs. A large number of network scenarios were designed, diverse in terms of the number of nodes, TCP flows, simulation time, and other relevant parameters. Based on the comprehensive performance analysis of TCP-RLACC across these scenarios, it is evident that TCP-RLACC offers substantial improvements over TCP Westwood+. The level of improvement depends on network scenarios, traffic conditions, and various other parameters, making it difficult

RESEARCH ARTICLE

to define a single value to represent the improvement percentage. Still based on our experimental setups covering wide range of network scenarios, there is 5.2% improvement in end to end throughput and 0.3% improvement in end to end packet delivery ratio.

This research work can be further extended for analyzing TCP-RLACC with other type of wireless networks and comparing its performance with other TCP variants. TCP-RLACC is a layered approach for end-to-end transport layer. Evaluation is performed with two of the most important end-to-end transport layer measures: Average Throughput and Packet Delivery Ratio. This work can be further extended by incorporating cross-layer approach design to measure various other parameters like delay, jitter, energy consumption etc.

REFERENCES

[1] M. Wang, Y. Cui, X. Wang, S. Xiao and J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities," in IEEE Network, vol. 32, no. 2, pp. 92-99, March-April 2018.

[2] Ahmad, I., Shahabuddin, S., Malik, H., Harjula, E., Leppänen, T., Loven, L., ... & Riekki, J. (2020). Machine learning meets communication networks: Current trends and future challenges. IEEE Access, 8, 223418-223460.

[3] Lorincz J, Klarin Z, Ožegović J. A Comprehensive Overview of TCP Congestion Control in 5G Networks: Research Challenges and Future Perspectives. Sensors. 2021; 21(13):4510.

[4] Y. Sun, M. Peng, Y. Zhou, Y. Huang and S. Mao, "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3072-3108, Fourth quarter 2019.

[5] Zhang, T., & Mao, S. (2020). Machine learning for end to end congestion control. IEEE Communications Magazine, 58(6), 52-57.

[6] Parween, S., & Hussain, S. Z. (2023). TCP Performance Enhancement in IoT and MANET: A Systematic Literature Review. International Journal of Computer Networks and Applications, 543-568.

[7] Ahmad R, Wazirali R, Abu-Ain T. Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues. Sensors. 2022; 22(13):4730.

[8] Messaoud, S., Bradai, A., Bukhari, S. H. R., Quang, P. T. A., Ahmed, O. B., & Atri, M. (2020). A survey on machine learning in Internet of Things: Algorithms, strategies, and applications. Internet of Things, 12, 100314.

[9] Kanagarathinam, M. R., Singh, S., Sandeep, I., Kim, H., Maheshwari, M. K., Hwang, J., ... & Saxena, N. (2020). NexGen D-TCP: Next generation dynamic TCP congestion control algorithm. IEEE Access, 8, 164482-164496.

[10] Kumar, S., Andersen, M. P., Kim, H. S., & Culler, D. E. (2020). Performant TCP for Low-Power wireless networks. In 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20) (pp. 911-932).

[11] Olmedo, G., Lara-Cueva, R., Martínez, D., & de Almeida, C. (2020). Performance analysis of a novel TCP protocol algorithm adapted to wireless networks. Future Internet, 12(6), 101.

[12] Joseph Auxilius Jude, M., Diniesh, V. C., & Shivaranjani, M. (2020). Throughput stability and flow fairness enhancement of TCP traffic in multi-hop wireless networks. Wireless Networks, 26(6), 4689-4704.

[13] Wei, W., Xue, K., Han, J., Xing, Y., Wei, D. S., & Hong, P. (2020). BBR-based congestion control and packet scheduling for bottleneck fairness considered multipath TCP in heterogeneous wireless networks. IEEE Transactions on Vehicular Technology, 70(1), 914-927.

[14] Grazia, C. A., Klapez, M., & Casoni, M. (2020). Bbrp: improving tcp bbr performance over wlan. IEEE Access, 8, 43344-43354.

[15] Pawale, S. S., Vanjale, S. B., Joshi, S. D., & Patil, S. H. (2020). Performance Improvement of TCP Westwood by Dynamically Adjusting Congestion Window in Wireless Network. Journal of University of Shanghai for Science and Technology. ISSN, 1007-6735.

[16] Saedi, T., & El-Ocla, H. (2021). TCP CERL+: Revisiting TCP congestion control in wireless networks with random loss. Wireless Networks, 27, 423-440.

[17] Li, T., Zheng, K., Xu, K., Jadhav, R. A., Xiong, T., Winstein, K., & Tan, K. (2020, July). Tack: Improving wireless transport performance by taming acknowledgments. In Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (pp. 15-30).

[18] Tang, J., Jiang, Y., Dai, X., Liang, X., & Fu, Y. (2022). TCP-WBQ: a backlog-queue-based congestion control mechanism for heterogeneous wireless networks. Scientific Reports, 12(1), 3419.

[19] Sarkar NI, Ho P-H, Gul S, Zabir SMS. TCP-LoRaD: A Loss Recovery and Differentiation Algorithm for Improving TCP Performance over MANETs in Noisy Channels. Electronics. 2022; 11(9):1479.

[20] Manikandan, A., Ramprasad, O. G., Rohra, H. A., & Vardhan, U. S. A. (2023, December). An Improved Cross Layer Design For TCP Optimization With Flow Control and Acknowledgment Aggregation. In 2023 International Conference on Next Generation Electronics (NEleX) (pp. 1-6). IEEE.

[21] Li, W., Zhou, F., Chowdhury, K. R., & Meleis, W. (2018). QTCP: Adaptive congestion control with reinforcement learning. IEEE Transactions on Network Science and Engineering, 6(3), 445-458.

[22] Xiao, K., Mao, S., & Tugnait, J. K. (2019). TCP-Drinc: Smart congestion control based on deep reinforcement learning. IEEE Access, 7, 11892-11904.

[23] Emara, S., Li, B., & Chen, Y. (2020, July). Eagle: Refining congestion control by learning from the experts. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications (pp. 676-685). IEEE.

[24] Chaudhry, A. U. (2021). Using machine learning to find the hidden relationship between RTT and TCP throughput in WiFi. EURASIP Journal on Wireless Communications and Networking, 2021(1), 200.

[25] Shi, H., & Wang, J. (2023). Intelligent TCP Congestion Control Policy Optimization. Applied Sciences, 13(11), 6644.

[26] Zhang Z, Li S, Ge Y, Xiong G, Zhang Y, Xiong K. PBQ-Enhanced QUIC: QUIC with Deep Reinforcement Learning Congestion Control Mechanism. Entropy. 2023; 25(2):294.

[27] Feng, Y., Wang, Y., Zhang, B., & Zhou, L. (2023, November). Deep Reinforcement Learning Based TCP Congestion Control in UAV Assisted Wireless Networks. In 2023 International Conference on Wireless Communications and Signal Processing (WCSP) (pp. 862-867). IEEE.

[28] Cao, Y., Nie, J., Fan, Y., Shao, X., & Lei, G. (2023, December). DRLFcc: Deep Reinforcement Learning-empowered Congestion Control Mechanism for TCP Fast Recovery in High Loss Wireless Networks. In GLOBECOM 2023-2023 IEEE Global Communications Conference (pp. 4345-4350). IEEE.

[29] Leon, J. P. A., de la Cruz Llopis, L. J., & Rico-Novella, F. J. (2023). A machine learning based Distributed Congestion Control Protocol for multi-hop wireless networks. Computer Networks, 231, 109813.

[30] Welzl, M., Islam, S., & Von Stephanides, M. Real-Time Tcp Packet Loss Prediction Using Machine Learning. Available at SSRN 4813494.

[31] Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning. Journal of Cognitive Neuroscience, 11(1), 126-134.

[32] Gangadhar, Siddharth, et al. "TCP Westwood (+) protocol implementation in ns-3." Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques. 2013.

[33] Mascolo, S., Grieco, L. A., Ferorelli, R., Camarda, P., & Piscitelli, G. (2004). Performance evaluation of Westwood+ TCP congestion control. Performance evaluation, 55(1-2), 93-111.

RESEARCH ARTICLE

Author

Dr. Hardik Molia is working as an Assistant Professor in Computer Engineering department of Government Engineering College-Rajkot. His areas of interest are Computer Networks, Machine Learning, Compilers and Databases.

How to cite this article:

Hardik K. Molia, "Reinforcement Learning based Adaptive Congestion Control for TCP over Wireless Networks", International Journal of Computer Networks and Applications (IJCNA), 11(5), PP: 607-616, 2024, DOI: 10.22247/ijcna/2024/39.