ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse312942020.pdf
https://doi.org/10.30534/ijatcse/2020/312942020

Improving TCP Performance with Intelligent Loss Differentiation for Mobile Adhoc Networks

Hardik K. Molia[1,2], Dr. Amit D. Kothari[3]
[1]Gujarat Technological University – Ahmedabad, Gujarat, India, hardik.molia@gmail.com
[2]Government Engineering College – Rajkot, Gujarat, India
[3]Accenture – Pune, Maharashtra, India, amitdkothari@gmail.com

ABSTRACT

TCP-Transmission Control Protocol provides connection oriented and reliable delivery at the transport layer. The primary cause of packet losses in wired networks is the network congestion. Wireless networks may experience packet losses due to non congestion issues such as channel issues and route failures. TCP's default consideration of any loss as a cause of network congestion degrades the overall performance in wireless networks, due to unnecessary activation of congestion control in case of non congestion losses. In recent years, ML-Machine Learning based network protocol design has attracted the attention of the researchers. Reinforcement Learning is more suitable to propose a generalize and adaptable solution for the networks with dynamic topologies and traffic. In this paper, we propose TCP-ILD, TCP with Intelligent Loss Differentiation for MANETs-Mobile Adhoc Networks. TCP-ILD differentiates congestion loss, channel loss and route failure loss to act accordingly. The extensive simulation has been performed with NS- 3 simulator. A significant amount of performance improvement is found.

Key words : TCP, MANET, Congestion Loss, Channel Loss, Route Failures, Reinforcement Learning, Loss Differentiation

1. INTRODUCTION

TCP-Transmission Control Protocol [1,2] provides connection oriented and reliable communication at the transport layer. TCP ensures the reliable delivery with congestion control, flow control and error control. TCP was initially introduced for the wired networks where the primary cause of any packet loss is mainly the network congestion. TCP activates congestion control to retransmit the lost packet and to reduce the transmission rate. Wireless networks may experience packet losses due to some other issues such as channel issues and route failures. TCP's default consideration of any loss as a cause of network congestion activates the congestion control unnecessary in presence of channel loss and route failure loss. TCP should able to identify the cause of a packet loss and act accordingly [3,4,5].

Researchers have proposed various TCP variants for the wireless networks. Most of them are either strict rule based with limited scope to be generalize and adaptable for unknown or dynamic networks. These TCP variants are classified into layered and cross-layered approaches based on their implementation approaches. A detail discussion of these TCP variants is given in [6].In recent years, ML-Machine Learning based network protocol design has attracted the attention of the researchers. Reinforcement Learning is more suitable to propose a generalize and adaptable solution for the networks with dynamic topologies and traffic. A few of the recent ML based TCP variants are discussed in Section 2.

This paper proposes a TCP-ILD: TCP with Intelligent Loss Differentiation. TCP-ILD differentiates congestion loss, channel loss and route failure loss to act accordingly. TCP-ILD is discussed in Section 3. The implementation and simulation is performed with NS-3 simulator. The experimental setup and performance analysis are discussed in Section 4. This paper concludes with the future directions.

2. RELATED WORK

2.1 Machine Learning with Networks

ML-Machine Learning [7,8,9] techniques are used for the accurate and efficient predictions in various fields. The network protocols can be designed using ML techniques for more effective solutions. Such solutions can be generalize and adaptable to be suitable for the dynamic networks. The dynamic networks have rapidly changing topologies and unpredictable traffic patterns. Over the last few years, ML techniques are started being introduced for large scale networks to the adhoc networks, for the purpose of traffic analysis, resource management, performance prediction, network security and protocol design [10,11]. This section discusses some of the most recent ML based TCP schemes. A detail discussion of these TCP schemes is in [12].

2.2 TCP with Supervised Learning

Bayesian Packet Loss Detection for TCP [13] differentiates congestion loss and packet reordering event. A Bayesian framework based mechanism analyzes the RTT distribution to identify the cause of DACKs - Duplicate Acknowledgements.

TCP+ Classifier [14] differentiates congestion loss and loss due to link error. A decision tree based classifier analyzes the delay and inter arrival times. TCP ex Machina's Remy program [15] produces computer generated congestion control algorithm. It analyzes the prior network assumptions and traffic for a specific objective such as to achieve higher throughput. EWMA - Exponentially Weighted Moving Averages of inter arrival times of ACKs, RTT ratio are used to represent the state. LP-TCP [16] - Loss Predictor based TCP predicts the loss probability for a packet being sent. A random forests based algorithm processes parameters like EWMA and TS-Time Series of inter arrival times of ACKs and EWMA of packet sending times.

2.3 TCP with Reinforcement Learning

A Machine Learning framework for TCP RTT-Round Trip Time estimation [17] uses experts framework method. The RTT value is derived as a weighted average of the values guessed by the experts. The weights decide the accuracy of the experts. This approach avoids pauses in transmission and unnecessary timeouts. i-TCP [18] intelligent TCP sets the value of Cwnd with a reinforcement learning based neural network. The inputs are Cwnd, number of consecutive timeouts and number of DACKs. Learning based and Data driven TCP [19] a reinforcement learning based solution to set value of Cwnd. Moving averages of inter arrival times between ACKs and packets, RTT-Ratio and SSThresh form the states. TCP-GVegas[20] is TCP Vegas with grey prediction to set Cwnd with reinforcement learning. Neural Network based Reliable Transport Layer Protocol [21] identifies the mobility pattern of the nodes to differentiate losses. RL-TCP [16] is a reinforcement learning based solution to set Cwnd. The EWMA of inter arrival times between ACKs, inter sending times between packets, Cwnd and SSThresh form the states. Q-TCP [22] Q-learning based TCP is a reinforcement learning based solution to set Cwnd. Average intervals between sending two packets and receiving two ACKs, average RTT are used to form the states.

3. TCP-ILD: PROPOSED SOLUTION

3.1 TCP-ILD Architecture

TCP-ILD: Intelligent Loss Differentiation is proposed with SARSA algorithm of Reinforcement Learning. TCP-ILD has three sections, The sensor, The learner and The actuator. The sensor derives the current state, utility and reward to be used for the learning purpose. The learner updates the Q table of SARSA algorithm. The actuator selects the action given the state. TCP-ILD is implemented independent of congestion control. It can be considered as loss differentiation module which is independent of congestion control. this enable us to implement TCP-ILD with multiple TCP variants easily.

3.2 The Sensor

TCP-ILD has 4 states and 3 actions. The 4 states are the TCP's states corresponding to its congestion control mechanism. These are Open (Normal State), Disorder (Duplicate ACKs), Recovery (Fast Retransmission) and Loss (Retransmission Time Out). The 3 actions are corresponding to congestion loss, channel loss and route failure loss. The utility function is designed with type of ACK and change of subsequent RTT value. The reward is calculated based on the change of utility function values. The detail of State, Utility and Reward is shown in Algorithm 1.

Algorithm – 1 – The Sensor (On Reception of ACK)

1: State Identification
CS ← TCP's Congestion State
if CS is Open **then** $State_{Next} \leftarrow 0$
else if CS is Disorder **then** $State_{Next} \leftarrow 1$
else if CS is Recovery **then** $State_{Next} \leftarrow 2$
else if CS is Loss **then** $State_{Next} \leftarrow 3$

2: Utility Function Calculation
$Utility_{Current} \leftarrow 0$
if IsDupAck() **then** $Utility_{Current} \leftarrow Utility_{Current} + 1$
if $RTT_{Previous} < RTT_{Current}$ **then** $Utility_{Current} \leftarrow Utility_{Current} + 1$

3: Reward Calculation
if $Utility_{Current} > Utility_{Previous}$ **then** $Reward_{Current} \leftarrow -1$
else if $Utility_{Current} = Utility_{Previous}$ **then** $Reward_{Current} \leftarrow 0$
else if $Utility_{Current} < Utility_{Previous}$ **then** $Reward_{Current} \leftarrow 1$

$Utility_{Previous} \leftarrow Utility_{Current}$

Algorithm 1: The Sensor

3.3 The Learner

TCP-ILD learns using SARSA algorithm. We have simplified SARSA algorithm by selecting next action based on maximum value rather than following ε-greedy approach. The learning rate - σ is set to 0.9 and the discount factor – ϒ is set to 0.5. The steps followed by the learner are shown in Algorithm 2.

Algorithm – 2 – The Sensor (On Reception of ACK)

1: Receive $State_{Next}$ and $Reward_{Current}$ from the Sensor
2: $Action_{Next} \leftarrow$ Select_Action_Max($State_{Next}$)
3: $Temp_1 \leftarrow [1- \sigma] * Q[State_{Current}] [Action_{Current}]$
4: $Temp_2 \leftarrow \sigma * [Reward_{Current} + [\Upsilon * Q[State_{Next}] [Action_{Next}]]$
5: $Q[State_{Current}] [Action_{Current}] \leftarrow Temp_1 + Temp_2$
6: $State_{Current} \leftarrow State_{Next}$
7: $Action_{Current} \leftarrow Action_{Next}$

Algorithm 2: The Learner

3.4 The Actuator

TCP-ILD avoids immediate retransmission and reduction of transmission rate on a packet loss. To achieve this, the actuator predicts the type of loss during recovery phase and retransmission phase. $Action_{Next}$ 0, 1 and 2 are considered as consideration of loss as a cause of congestion, channel issues and route failure. At present, the decision making is implemented with two phases.

During fast retransmission phase, if the $Action_{Next}$ is 2, it is considered as indication of route failure and TCP will exit from the fast retransmission phase without retransmission and without reduction of the transmission rate. During recovery phase, if the $Action_{Next}$ is 0, congestion control is continued. TCP-ILD will perform retransmission, reduction of transmission rate as per the base TCP variant. Otherwise, TCP will exit from the recovery phase immediately.

4. EVALUATION

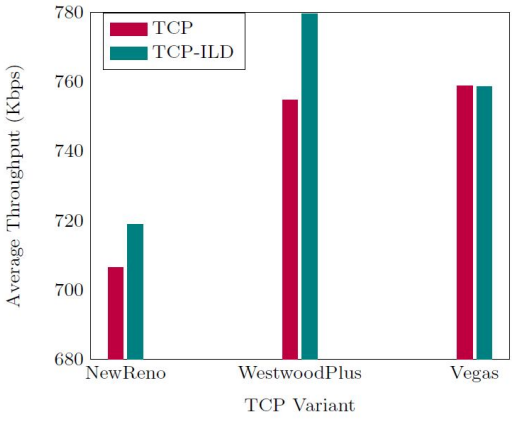
4.1 Experimental Setup

TCP-ILD is implemented with NS 3.28 simulator. We have performed its evaluation with large number of long running network scenarios. These scenarios differ in terms of the number of nodes, mobility speed, number of TCP flows, transmission parameters such as RSS etc. Based on our earlier evaluation of existing TCP variants, TCP WestwoodPlus, TCP Vegas and TCP NewReno performed better. So TCP-ILD is evaluated with these three TCP variants. The results of one main experimental setup is discussed in this paper. We are discussing two end-to-end performance metrics: Average Throughput (Kbps) and Total Received Data (MB).

4.2 End-to-End Performance

This experimental setup has 10 networks each of different nodes (05,10,15,….45,50). Each network has a single TCP flow. Each of these networks is evaluated for 300, 600 and 900 seconds. Due to large number of TCP flows, the results are evaluated on average scale. Figure 1, Figure 2 and Figure 3 show the average performances for the evaluation of these scenarios with simulation time of 300, 600 and 900 seconds.

Figure 4 shows the growth of throughput and received data with respect to the time. As reinforcement learning based solution needs time to build an efficient model, the results are improved for long running scenarios. The purpose of evaluating TCP-ILD on the networks with different number of nodes and for large simulation durations is to identify weather it works as an adaptable and generalize solution or not.



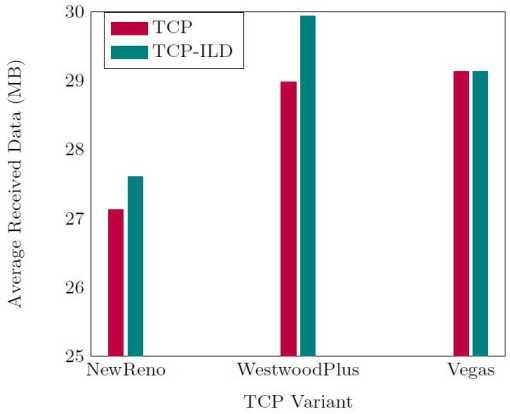
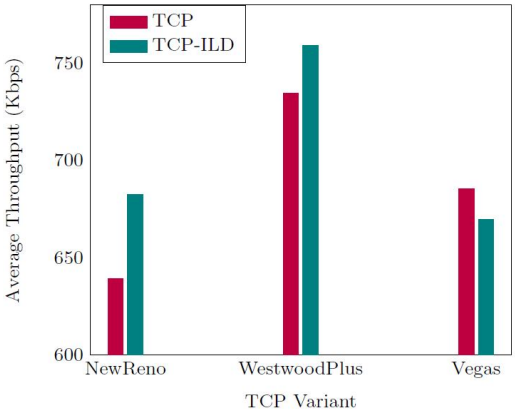


Figure 1: Average Performance – 300 Seconds



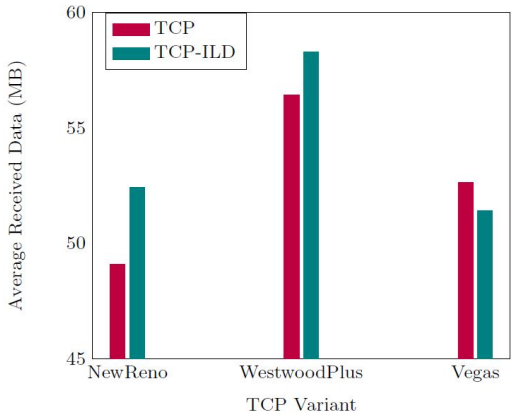
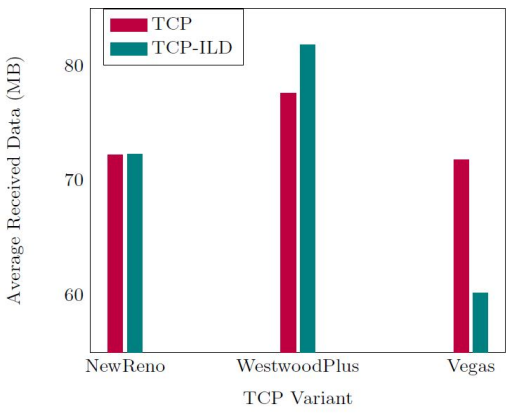


Figure 2: Average Performance – 600 Seconds



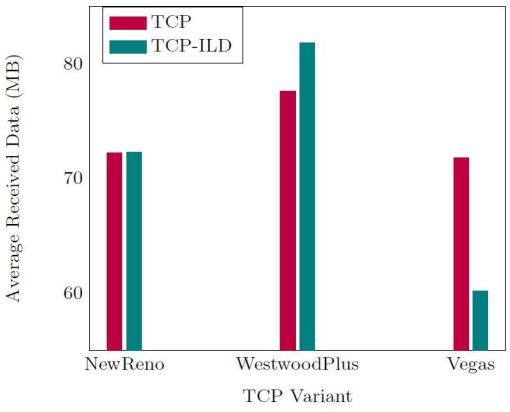


Figure 3: Average Performance – 900 Seconds

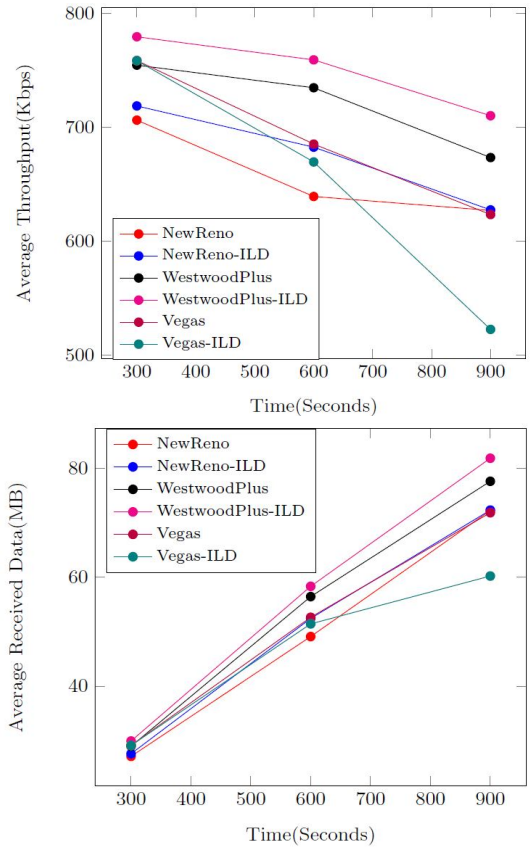


Figure 4: Average Performance With Respect to Time

4.3 End-to-End Performance

The purpose of our evaluation is to compare performances of existing TCP variants and with their TCP-ILD implementations. TCP-ILD is evaluated with 3 existing TCP Variants: TCP NewReno, TCP WestwodPlus and TCP Vegas. The selection of these three base TCP variants is done based on our prior evaluation of 14 existing TCP variants. TCP-ILD is implemented with these 3 TCP Variants so total 6 TCP Variants are evaluated.

Based on the results of evaluation, it is observed that the performance of TCP WestwoodPlus is better than TCP NewReno and TCP Vegas. The same observation is found for their respective TCP-ILD implementations. TCP-ILD performs better for TCP WestwoodPlus and TCP NewReno. It is observed that TCP WestwoodPlus-ILD performs better out of all the six TCP Variants. The evaluation is continued with some more complex networks too. The same observation is found.

5.CONCLUSION

Dynamic networks such as MANETs have random topologies and unpredictable traffic flows. Reinforcement learning based solutions perform better as they learn via the trials without need of dataset for prior learning. TCP-ILD is reinforcement learning based, adaptive and generalized solution to differentiate losses for MANETs. We have performed extensive simulations and found TCP-ILD performs well with TCP-WestwoodPlus and TCP-NewReno. TCP-ILD is decoupled from the TCP's congestion control. This enables it to be implemented with any TCP variant easily.

FUTURE DIRECTIONS

TCP-ILD can be evaluated with real world network scenarios. The learning parameters and the exploration – exploitation trade-off of reinforcement learning can be analyzed to identify the scope of performance improvement.

REFERENCES

[1] Behrouz Forouzan. Tcp/ip protocol suite. McGraw-Hill, 2009.

[2] W. Richard Stevens. Tcp/ip illustrated, vol. 1: The pro-tocols. Addison-Wesley Professional Computing Series, 2000.

[3] Noor Mast and Thomas J Owens. A survey of performance enhancement of transmission control protocol (tcp) in wireless adhoc networks.EURASIP Journal on Wireless Communications and Networking-Springer, 2011.

[4] Ammar Mohammed Al-Jubari, Mohamed Othman, Borhanuddin Mohd Ali, and Nor Asilah Wati Abdul Hamid. Tcp performance in multi-hop wireless ad hoc networks challenges and solution. EURASIP Journal on Wireless Communications and Networking-Springer, 2011.

[5] Vassilis Tsaoussidis and Ibrahim Matta. Open issues on tcp for mobile computing. Journal on Wireless Communications and Mobile Computing, 2:3–20, 2002. https://doi.org/10.1002/wcm.30

[6] Hardik K. Molia and Amit D. Kothari. Tcp variants for mobile adhoc networks: Challenges and solutions. Wire-less Personal Communications - Springer, 100:1791– 1836, June 2018.

[7] S. Wang, W. Chaovalitwongse, and R. Babuska. Machine learning algorithms in bipedal robot control. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(5):728–743, 2012.

[8] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies), pages 3–24, 2007.

[9] F. Woergoetter and B. Porr. Reinforcement learning. Scholarpedia, 3(3):1448, 2008.

[10] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang. Machine learning for networking: Workflow, advances and opportunities. IEEE Network, 32(2):92–99, 2018. https://doi.org/10.1109/MNET.2017.1700200

[11] Raouf Boutaba, Mohammad A. Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar M. Caicedo. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications - Springer, 2018.

[12] Hardik K. Molia and Amit D. Kothari. Tcp with machine learning - advances and opportunities. International Journal of Advanced Trends in Computer Science and Engineering, 8:3526–3534, December 2019. https://doi.org/10.30534/ijatcse/2019/132862019

[13] N. Fonseca and M. Crovella. Bayesian packet loss detection for tcp. IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.,3:1826–1837, 2005.

[14] Ibtissam El Khayat, Pierre Geurts, and Guy Leduc. Enhancement of tcp over wired/wireless networks with packet loss classifiers inferred by supervised learning. Wireless Networks-Springer, 16:273–290, 2010.

[15] Keith Winstein and Hari Balakrishnan. Tcp ex machina: Computer-generated congestion control. Proceedings of the ACM SIGCOMM 2013 Conference, pages 123–134, 2013.

[16] Yiming Kong, Hui Zang, and Xiaoli Ma. Improving tcp congestion control with machine intelligence. Proceedings of the 2018 Workshop on Network Meets AI & ML - NetAI'18, pages 60–66, 2018.

[17] Bruno Astuto Arouche Nunes, Kerry Veenstra, William Ballenthin, Stephanie Lukin, and Katia Obraczka. A machine learning framework for tcp round-trip time estimation. EURASIP Journal on Wireless Communications and Networking-Springer, 2014.

[18] A. B. M. Alim Al Islam and Vijay Raghunathan. itcp: an intelligent tcp with neural network based end-to-end congestion control for ad-hoc multi-hop wireless mesh networks. Wireless Networks-Springer, 21:581–610, 2015.

[19] W. Li, F. Zhou, W. Meleis, and K. Chowdhury. Learning- based and data-driven tcp design for memory-constrained iot. International Conference on Distributed Computing in Sensor Systems (DCOSS)-IEEE Explore, pages 199– 205, May 2016. https://doi.org/10.1109/DCOSS.2016.8

[20] Hong Jiang, Ying Luo, QiuYun Zhang, MingYong Yin, and Chun Wu. Tcp-gvegas with prediction and adaptation in multi-hop ad hoc networks. Wireless Networks-Springer, 23:15351548, July 2017.

[21] P. Kumar, S. Tripathi, and P. Pal. Neural network based reliable transport layer protocol for manet. 4th International Conference on Recent Advances in Information Technology IIT(ISM), Dhanbad-IEEE Explore, pages 1– 6, March 2018.

[22] W. Li, F. Zhou, K. R. Chowdhury, and W. M. Meleis. Qtcp: Adaptive congestion control with reinforcement learning. IEEE Transactions on Network Science and Engineering, 2018.