PACKET LOSS DIFFERENTIATION OVER MANET BASED ON A BP NEURAL NETWORK

Dimitris Kanellopoulos[1], Pratik Gite[2]

[1]*ESD Lab, Department of Mathematics, University of Patras, Patras, Greece, d_kan2006@yahoo.gr*

[2]*Institute of Engineering and Sciences, IES-IPS Academy, Rajendra Nagar Indore, 452012, India pratikgite135@gmail.com*

Contribution to the state of the art

DOI: **10.7251/JIT1801018K** UDC: **551.588:551.506**

Abstract: An adaptive distributed routing algorithm is essential in MANETs, since there is no central routing system. Actually, there is no central point of coordination; each node is responsible for forwarding data packets to other nodes, thereby acting as router and host. A packet might travel through multiple intermediary ad hoc nodes in order to arrive to its destination, while the nature of wireless multi-hop channel is bringing in various types of packet losses. This paper focuses on three main reasons of online packet losses in MANETs: (1) losses due to wireless link errors; (2) losses due to congestion; and (3) losses due to route alteration. It proposes a deep learning-based algorithm for packet loss discrimination. The algorithm uses the backpropagation neural network (BPNN) concept. We performed simulation experiments for evaluating the performance of the proposed loss discrimination algorithm under different network configurations. Through simulation results, we confirmed that the proposed algorithm improves packet loss discrimination and route alteration in the network. It also reduces congestion and increases network throughput.

Keywords: MANET, packet loss discrimination, multicast congestion detection, backpropagation neural network, deep learning.

INTRODUCTION

Mobile ad-hoc networks (MANETs) can be deployed in many applications such as disaster recovery, search and rescue operations, military service, and vehicular networks. A MANET is a group of autonomous nodes that form a dynamic, multi-hop radio network in a decentralized way [3]. MANET nodes can be a variety of mobile devices such as mobile phones, laptops or handheld devices, which present various computational and bandwidth capabilities. Nodes themselves implement the network management in a cooperative fashion. They operate on a multi-hop basis, while they are detecting routes and forwarding data packets. Since the channel is broadcast in nature, multiple nodes contend for the channel simultaneously. The wireless channel is also error-prone, and this situation becomes even worst because packet losses occur due to route alteration and congestion [8]. MANETs have peculiar characteristics including dynamic network topology, asymmetry, multi-hop communication, and limited bandwidth and energy resources. These characteristics complicate quality of service (QoS) provision, and impose various challenges in the design of congestion control [11]. Wired transport layer protocols tend to achieve congestion control, flow control, and end-to-end delivery of data packets.

TCP ensures reliability by using acknowledgments (ACK); for every transmitted TCP packet it waits for an ACK. *Additive-Increase/Multiplicative-Decrease* (AIMD) is the feedback control algorithm used in TCP congestion control. AIMD combines linear growth of the *congestion window* with an exponential reduction when a congestion occurs. The window size is increased by one segment in every round-trip-time (RTT), when no packet losses occur. In case of the reception of a duplicate acknowledgment, a TCP sender will first assume that some packet reordering has occurred in the network. But upon reception of the fourth copy of an ACK (Triple Duplicate ACK, TDACK) a congestion loss is assumed. In this case, the missing segment is repeated and the window size is cut in half. Additionally, TCP uses a timeout that depends on the measured RTT of the connection. If this retransmission timeout (RTO) elapses without an acknowledgment, TCP concludes severe congestion. Then, the window size is reduced to one and the unacknowledged segment is sent again. The timeout until the next retransmission attempt is doubled, if still no acknowledgment arrives. Thus, this timeout grows exponentially. During the first phase of a connection and after a timeout, the "*slow start*" mechanism is employed that allows for a faster convergence to the correct window size [2]. While "*slow start*" is active, the window size is not increased by one segment size for every RTT, but instead for every received acknowledgment. This means that during this phase, the window size grows exponentially.

TCP-friendly congestion control schemes for the wired channels provide smooth sending rates [17]. These schemes include the window-based schemes and the rate-based schemes, which can be further classified into probe-based and equation-based schemes. Such schemes cannot be applied to wireless scenarios [19]. Rate-based congestion control schemes control the transmission rate and generate a smoothed flow by spreading the data transmission across a time interval. Thus, the burstiness induced by the window-based mechanisms is avoided. Equation-based congestion control enables bandwidth estimation, based on statistics of RTT and packet loss probability. In response to the bandwidth estimates, the source adjusts the transmission rate to prevent congestion. A typical equation-based protocol is the *TCP-friendly rate control* (TFRC) designed for wired link, and thus it accounts wireless losses also as congestion losses. TFRC adjusts its transmission rate in response to the level of congestion, as estimated based on the calculated loss rate. Multiple packet drops in the same RTT are considered as a single loss event by TFRC leading to a more slow congestion control strategy.

Wired transport protocols cannot be migrated directly to MANETs, due to some of the issues associated with it [10]. The TCP congestion control mechanism cannot handle the special properties of a shared wireless multi-hop channel well. Various situations may lead to packet loss like path break due to mobility, hidden terminal problem, high error-prone wireless links, node failure due to low battery etc. [6]. Route changes and the error-prone wireless channel result in unsteady packet delivery delays and packet losses. Thus, packet losses must not be misinterpreted as congestion losses. Other mechanisms are required to identify network congestion instead of packet loss and retransmission timeout. The transport protocol needs to decrease the sending rate only when there is congestion in the network. Also, the transport layer in the wireless network must periodically interact with lower layers to adapt the changing network conditions often and the frequent changes in network topology.

An enhanced transport protocol should reliably handle loss, minimize errors, manage network congestion and transmit efficiently. An enhanced transport protocol for MANETs should support *packet loss differentiation and estimation*. Traditional transport protocols assume that a packet loss is encountered due to congestion in the network. This assumption may lead to performance degradation in MANETs where the packet losses due to transmission errors are more probable in error-prone links. Actually, packets might be missing due to multipath fading, link-layer contention and channel errors Doppler shift, node's buffer excess. Conclusively, the transport layer mechanism must distinguish between the packet losses experienced due to network congestion and packet losses due to wireless link errors.

Deep Learning for Packet Loss Discrimination in MANETs

In MANET, an accurate packet loss discrimination system in the transport layer is often cooperated with the flow-control protocol in order to improve the overall network performance. Deep learning can be applied in MANET for discriminating packet losses due to wireless errors, congestion, and route alteration. Many studies have shown that deep learning entirely surpasses other machine learning approaches for solving similar problems. When a loss discrimination algorithm is utilized with a deep learning neural network (NN), then it can be more accurate and effective than other discrimination algorithms utilized with other machine learning algorithms. Deep learning is a branch of machine learning methods based on learning data representations [14]. Learning can be supervised, partially supervised or unsupervised. Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to various fields including signal processing, communication systems, and adaptive control systems etc. The hardware implementations of deep learning algorithms are often exploiting the advantage of deep learning characteristic, called '*parallelism*'. A deep learning method stores knowledge inside inter-neuron connection through the support of the neuron weights. Deep learning contains a huge amount of processing elements, which are used to yield inputs and then processing takes place and after that it provides response to the inputs. A huge Network contains a huge number of processing elements. Deep learning is complete up of an amount of layers. Layers are organized by a quantity of consistent nodes, which contain an *activation* function. Patterns are characterised in the network as the *input layer*, which takes links to one or additional hidden layers. Then, the input layer is connected with the middle layer, which is identified as *hidden layer*. In the hidden layer, processing is completed by a number of weighted influences. The hidden layer straight connected to an *output layer*, which has the ability to provide response to completely the inputs. Deep learning works similar an input since the user and previously processing is taking place. Then, the processing takes place. Deep learning has the capability to originate the meaning after the difficult data and is used to extract patterns and detect leanings that are further multifaceted to announcement. A trained deep learning (based on backpropagation neural network - BPNN) can be supposed as an expert that analyse specified information. *Hyperbolic tangent sigmoid* is the activation function used in deep learning. It processes the input (which might have some value among plus and minus immensity). It is used in multi-layer neural networks and performs the training by using the BP algorithm, and thus this function is differentiable.

Key-Contributions

In this paper, we assume congestion control as a transport layer issue and propose a Loss Discrimination Algorithm (LDA) that distinguishes the packet losses due to congestion, route alteration, and wireless link errors in MANETs. The proposed deep learning algorithm for packet loss discrimination is based on a backpropagation neural network (BPNN) architecture, while semi-supervised learning is applied. So, we exploited the feature reduction capability of the deep learning for pre-training. A deep learning algorithm is very powerful, when it is used with semi-supervised learning that reduces the training time. The key contributions of our work can be summarized as follows:

1. Experimental simulation results demonstrated that the proposed packet loss discrimination algorithm improves the accuracy under different network configurations and high-level network congestion conditions. This is achieved as an introduced congestion control process is invoked, when packet losses/failures (due to congestion) are detected.
2. The proposed packet loss discrimination algorithm can be implemented by a similarity circuit, which is fast and insignificant in size. Thus, the algorithm is appropriate to MANETs, where time and space are critical and near-optimal resolution is suitable.
3. In our framework, deep learning is executed on MANET for observing the regularity of '*hello*' messages of the AODV routing protocol in MANET. The Ad hoc On-Demand Distance Vector (AODV) is a well-established reactive routing protocol that is used for MANETs [7]. The term "*reactive*" implies that routing does not depend on periodic exchange of routing information or route calculation. When a route is required, the node must start a route discovery process. AODV uses classification numbers, where sequence statistics regulate the 'freshness' of routing information and to avoid routing loops. When an active link is broken, AODV initiates a finding process for a new route.

The rest of this paper is organized as follows. In the next Section, we discuss related work. Then, we present the proposed LDA algorithm. After that, we analyze the performance of our LDA algorithm. In the last Section, we provide our conclusions and further work.

Related Work

The main causes for packet loss in MANET are mobility, channel error, and congestion. Reducing packet loss involves congestion control functioning on top of an adaptive routing protocol at the OSI-RM Network layer that supports errors due to mobility and failures. In the current designs, routing is not congestion-adaptive. Routing may let a congestion happen which is detected by congestion control, but dealing with congestion in this reactive manner results in longer delay and unnecessary packet loss and requires significant overhead, if a new route is required. This problem becomes more visible especially in large-scale transmission of heavy traffic such as multimedia data, where congestion is more probable and the negative impact of packet loss on the service quality is of more significance. Habbal and Hassan [5] introduced a model that analyses the factors that impact on TCP congestion control. Furthermore, their model points to those critical factors that must be addressed by researchers in order to improve TCP performance over MANETs.

Tran and Raghavendra [16] argued that routing should not only be aware of, but also be adaptive to, network congestion. They proposed a routing protocol (CRP) with such properties. CRP improves the packet loss rate and end-to-end delay, while enjoying significantly smaller protocol overhead and higher energy efficiency as compared to AODV and DSR. Designing an efficient routing protocol for MANET is a challenging task because of the dynamic environment of the network topology and resource limitations. Multipath routing can offer consistent communication in MANETs. Mallapur et al. [12] introduced an efficient routing technique called the *Multipath Load Balancing technique for Congestion Control* (MLBCC) for MANETs to efficiently balance the load among multiple paths by reducing congestion. MLBCC introduces a congestion control mechanism and a load balancing mechanism during the data transmission process. The congestion control mechanism detects the congestion by using an arrival rate and an outgoing rate at a particular time interval *T*. The load balancing mechanism is the selection of a gateway node by using the link cost and the path cost to efficiently distribute the load by selecting the most desirable paths. For an efficient flow of distribution, a node availability degree standard deviation parameter is introduced. Simulation results, under the network simulator-2 (NS-2), show that MLBCC improves the performance of FLMB and AOMDV in terms of the control overhead, packet delivery ratio (PDR), average delay and packet drop ratio. The results also show that MLBCC efficiently balances the load of the nodes in the network.

De Oliveira and Braun [4] investigated the use of fuzzy logic theory for assisting the TCP error detection mechanism in MANETs. They presented an elementary fuzzy logic engine as an intelligent technique for discriminating packet loss due to congestion from packet loss by wireless induced errors. They also introduced the architecture of the proposed fuzzy-based error detection mechanism. Their full approach, for inferring the internal state of the network, relies on Round Trip Time (RTT) measurements only. Hence, their end-to-end scheme requires only end nodes cooperation. Preliminary simulation evaluations showed how viable their approach is.

Yang et al. [18] proposed an explicit loss discrimination scheme (F-ECN) to discriminate, if a packet loss is due to congestion or to a wireless link fault. F-ECN is based on a fuzzy logic controller that uses queue length and packet arrival rate in order to measure congestion and achieves a tradeoff between queue stability and responsiveness. The performance of F-ECN takes a tradeoff between the throughput and the delay time. Papanastasiou and Ould-Khaoua [13] developed a TCP variant that adjusts the sending rate increase to achieve competitive throughput for TCP connections. Extensive simulation experiments indicate that a slower sending rate increase, during the congestion avoidance phase of TCP, leads to improved performance for TCP Reno, while eliminating the negative effects inherent in restricting the maximum sending window size. Their work discusses the applicability of their TCP oriented solution to the hidden terminal effect.

For effective load balancing and congestion control, routing metrics need to accurately capture the load in various locations of the network. Ali et al. [1] presented a congestion adaptive multipath routing protocol to increase the throughput and avoid congestion in MANETs. In their approach, when the average load of an existing link increases beyond a defined threshold and the available bandwidth and residual battery energy decreases below a defined threshold, traffic is distributed over fail-safe multiple routes to reduce the traffic load on a congested link. Through simulation results, they showed that their approach achieves better throughput and PDR with reduced delay for constant bit rate (CBR) traffic, when compared with QMRB (viz., a protocol using mobile routing backbones).

Sliwa et al. [15] presented a simple passive decentralized load balancing scheme for MANET routing protocols. In contrast to existing load balancing schemes, the nodes consider only local knowledge and no additional communication or coordination is required. The proposed scheme can easily be applied to increase the reliability of existing routing protocols. Their simulative evaluation showed that three examined routine protocols (B.A.T.M.A.N, B.A.T.Mobile, G-OLSR) were able to achieve significant PDR gains through integration of the proposed load balancing approach. By distributing the packets over multiple suitable links, packet collisions are less probable and the reliability is increased. The probability for losses of routing packets is lowered, which leads to a higher consistency of the routing tables and avoids occurrences of drastic PDR drops. Additionally, the transmission queues of the forwarding nodes are relieved and queuing-related packet drops occur less often.

Khan et al. [9] designed a new routing algorithm using the combination of AODV and cross layer design approach. It is referred as *Congestion Control AODV* (CCAODV) *approach*. It is used to avoid link break in MANET. Received signal strength is used as cross layer design parameter. The CCAODV protocol creates strong and stable route by using signal strength of node. The signal strength mainly depends on the parameters like transmission power of node and distance between two nodes. The Cross layer design approach is tested by using Ns 2.35 simulator and compared with the AODV routing protocol.

The Packet Loss Differentiation Algorithm (LDA)

In the proposed LDA, a BPNN is employed to classify the causes of packet losses. The proposed technique is based on a deep learning algorithm that adopts the BPNN method into MANETs. Backpropagation (BP) is a method used in neural networks to calculate the error contribution of each neuron after a batch of data is processed. The proposed technique achieves multi-metric cooperative decision at the receivers (nodes) to distinguish the three main reasons of packet losses.

The BPNN architecture

The ratios of the evaluation indicators are used as input features of the deep learning-based BPNN packet loss Classifier, located in the transport layer. These end-to-end evaluation indicators are the following ones:

- *X1*: *The comparative one-way expedition time,*
- *X2: The inter-arrival time of packets fore-and-after the losses, and*
- *X3: The amount of out-of-order packets.*

These input variables *X1, X2, X3* are used as measures to predict congestion. In our BPNN architecture (Figure 1), we have three continuous variables as output variables (*Yl, Y2,* and *Y3*). These outputs vary at range [0, 1].

- *Y1*: It represents that network is at normal conditions. If the output were *Y1*, the MANET would be at normal conditions.
- *Y2*: It represents that network is at congestion conditions (packet loss due to congestion).
- *Y3*: It represents that in the network, this TCP connection is experienced link bit error just now (packet loss due to link error).

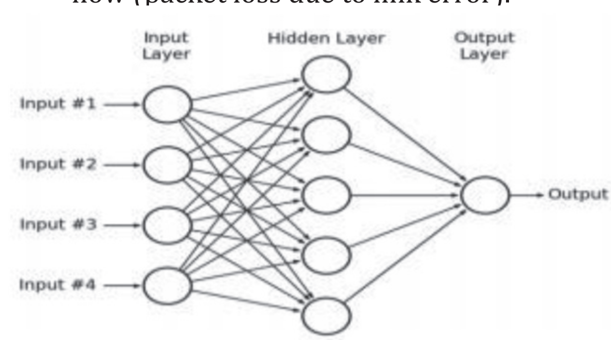

Figure 1. *The BPNN architecture*

Hereafter, the proposed packet loss discrimination algorithm follows:

Algorithm 1: *The LDA algorithm for MANETs*

ALGORITHM 1: *Load dissemination using the multiple paths in deep leaning adopting the BPNN method.*

Step 1: *a) Process for load distribution and choose the INPUT parameters for deep learning.*

b) Choose the OUTPUT parameter after the deep learning.
Step 2: *Compute the training pairs of INPUT and OUTPUT using deep learning.*
Step 3: *Train the Deep learning using the training pair's congestion detected. After training, acquire the trained Deep learning.*
Step 4: *IF (congestion is identified OR accumulation contextual traffic exists)*
THEN
Step 5: *Acquire the weights and biases after the trained Deep learning and input it to the BPNN program of deep learning.*
Step 6: *Implement the BPNN program for the definite input, weights, and biases for deep learning.*
PACKET LOSS DISCRIMINATION ALGORITHM:
Step 7: *Evaluate node accessibility of path using Step 2*
Step 8: *Estimate entire node accessibility of path using Step 3.*
Step 9: *Estimate entire average node accessibility of path.*
Step 10: *Evaluate the normal node accessibility degree consuming the normal deviation using Step 5.*
Step 11: *ENDIF*
Step 12: *Based on the value found of opening node, it selects implementation of deep learning- based reactive routing protocol.*
Step 13: *Observe the outcomes using NS2.*
Step 14: *Repeat Steps 2 to 6 for dissimilar INPUTs and confirm the OUTPUTs.*
Step 15: *Link the outcomes with NS2-based deep learning.*
Step 16: *IF (outcomes are acceptable) THEN Load BPNN program of deep learning. The default path and the additional normal path to distribute the traffic complete the multiple paths using Step 12.*

Congestion control

The lost packets in MANET often trigger retransmissions. In particular, control packets are retransmitted in order new routes to the destination node to be found. That means that even more packets are sent into the network. Therefore, network congestion can severely deteriorate network throughput. If no congestion control is performed, this can lead to a network collapse, where almost no data is successfully delivered. In traditional route discovery approaches, nodes use control packets (e.g., RREQ, RREP) to find out a new route to the destination. For designing a congestion control mechanism, we must use a hybrid approach for applying the different state of data preprocessing with received signal asset parameter.

In the proposed routing algorithm, routing is done on demand, and BPNN is used to save information about the link status and observing the route state. Each node in MANET can share its information with its neighboring nodes. A query processing for particular routing information is originated to complete the search between ad hoc nodes. All nodes are linked through one additional forwarding a request message from one node to the next ad hoc node until a search for the particular information processing is completed. The information is processed on ad hoc basis from the source node to that node that can satisfy the particular requirements. The information passes through numerous ad hoc nodes until the suitable node is resolute. The proposed BPNN uses this information as input information to find the decision for repeating the information into the nodes. The decision is made by assembling the input parameters of respectively node and by yielding these input parameters into the deep learning algorithm.

The routing algorithm for discovering the best route path

The proposed routing algorithm is based on the deep learning algorithm discussed. It finds the most reliable route path set. A flooding technique is used for route discovery. In the proposed algorithm, routing is done on demand and BP is used to save link status information and route state observing. We propose a congestion control technique that detects the congestion through candidate nodes by using the influx rate and the leaving rate. For load balancing reasons, our deep-learning learning technique allocates the existing traffic over numerous route paths by using gateway nodes. Algorithm 2 follows.

Algorithm 2: *The proposed routing algorithm for discovering the best route path*
Step 1: *Initialize the input parameters*
Step 2: *Select the data for processing*
Step 3: *Define the information by setting the input parameters (required delay, required PDR, and re-*

quired energy)
Step 4: *Select the nodes for replication this information for processing.*
Step 5: *Find the node's position using the BPNN classifier*
Step 6: *IF (Output=0) THEN STOP: Information processing is completed.*
ELSE Output=1: stay on the network: more node information processing is required.
ENDIF
Step 7: *Form the group of nodes (viz. the route path) by selecting the next ad hoc node to be examined.*
Step 8: *Try to succeed until a suitable node is reached in the random replication.*

This routing algorithm achieves objective one route finding to compute the separate path set. Thus, it creates less number of control packets compared with the conventional AODV routing algorithm that achieves numerous route findings. For every pair of the network parameters (required delay and required PDR), the total number of the generated control packets differs. The number of nodes for information processing is 100 with the intention of specifying the number of control packets (*m*) generated by the new routing algorithm process. The routing algorithm of AODV was simulated in this work for comparison reasons.

PERFORMANCE EVALUATION

Simulation Setup

We created and configured a small size MANET that contains 20, 40, 60, 80 and 100 nodes. This MANET uses AODV as a routing protocol. We defined the locations of nodes manually in TCL script. Also, we used the Omni antenna model: a wireless transmitting or receiving antenna that radiates or interrupt radio-frequency (RF) electromagnetic fields equally well in all horizontal directions in a flat, two-dimensional (2D) geometric plane. The parameter of "*Topography area*" indicates the area where the nodes can be moved in all directions. Finally, we used the "*Two Ray ground*" propagation model to predict the received signal power for each packet. The "*random mobility model*" was used to simulate the mobility of nodes. Table 1 shows the simulation parameters for our experiment.

Table 1. *Simulation parameters*

Parameter	Value
Simulator	Ns-2.34
Simulation time	100 seconds
Channel type	Wireless
Number of nodes	100
Topography area:	500 x 500 (sq.m.)
Pause time	20 sec
Packet size	512 byte
Bandwidth	40 MHz

The simulation results were gained using NS-allinone-2.35.tar.gz simulator. NS2 is discrete event simulator. System language is C++ and scripting language is OTcl. Authors deliberate performance indices as stated in equation. For the experiments performed, a variable-size network of size 500*500 (sq.m.) was randomly generated with number of nodes afterward accomplishment its destination, the node silences for a definite time period, and then it chooses a different random location and continues the method again.

In our experiments, every node pauses at the current position for 10 sec, while movement speed of separate nodes ranges from 0 to 20 m/s. Simulations for networks have been path with 100 mobile hosts, effective at transmission ranges changing from 150 to 500 m. The Max_hop property is set to 5 as an initial value. The behaviour of the replicated network Relationships of reliability, lifetime, amaount of paths and above is compared to existing algorithms. Similarly, the computational complexity of the proposed algorithm is assessed and evaluated to separate AODV routing protocol. Control packets (as overhead) are generated in order to compute separate path sets. In this segment, the above performance of the proposed algorithm, in relationships of produced control messages, is linked to the AODV routing algorithm.

RESULTS ANALYSIS AND DISCUSSION

In order to analyse the performance of an intermediary/gateway node, we calculated for each node, the *average Throughput* and PDR for the following two cases:

- **Case 1:** TCP *Multiplicative Decrease*-MD is applied (without LDA) over AODV.

- **Case 2:** TCP MD cooperates with the new LDA algorithm over the new introduced routing scheme. In this case, the new routing scheme (Algorithm 2) is used.

During the simulation process, we compared PDR and average Throughput for both cases. We evaluated our algorithm under numerous network configurations and we observed that it provides high accuracy under most types of packet losses. Moreover, in order to evaluate the accuracy of our BPNN classifier, we defined two types of errors:

- EC (*Error in discriminating packet loses due to Congestion*): The probability that the BP Network classifier misclassifies a congestion error as a link error.
- EL (*Error in discriminating packet loses due to Link errors*): The probability that the BP Network classifier misclassify a link error as a congestion error.

During the evaluation of the proposed algorithm, the network size and speed were changing. The simulation results (Figures 2, 3) showed that Case 2 outperforms Case 1. In a MANET having 100 nodes, we found the generated and received packets, when no LDA algorithm is used over AODV (the TCP MD is used without LDA over AODV). This is **Case 1**. Then, we found the generated and received packets, when the new routing scheme (Algorithm 2) cooperates with our packet loss discrimination algorithm (**Case 2**). Afterward, for both cases, we specified *average Throughput* and *PDR* and compared them.

NS-2 was used to generate the learning sample set. In order to evaluate our packet loss Classifier we divided the learning sample set database into two parts: (1) a learning sample that was used to learn the model, and (2) a test sample on which the resulting classifier was tested. In particular, we collected 1000 positive and 1000 negative samples correspondingly to make the training set. For the testing set, we used 200 positive and 200 negative samples to assess the accurateness. As depicted in Figure 2, the average Throughput in Case 2 is increasing, especially when the number of nodes is greater than 70.

The receiver uses the training deep learning model to automatically identify the reason of the current packet loss. This technique is an endwise solution and does not require support. The proposed packet loss Classifier cooperates with the TCP MD algorithm. The TCP MD algorithm is executed upon detecting a packet loss. Thus, we can say that we have a new TCP variant for MANETs, after using the proposed Loss Differentiation Algorithm (LDA).

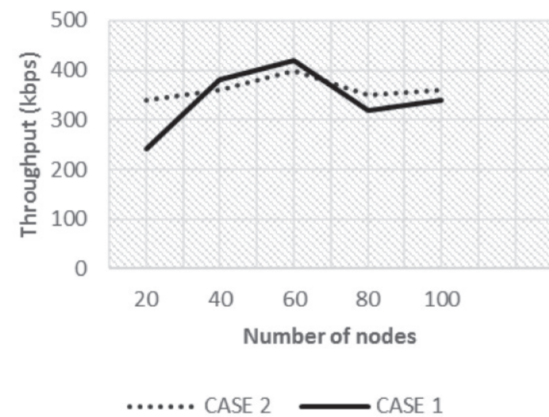

Figure 2: *Comparison of average Throughput (Case 1 vs. Case 2)*

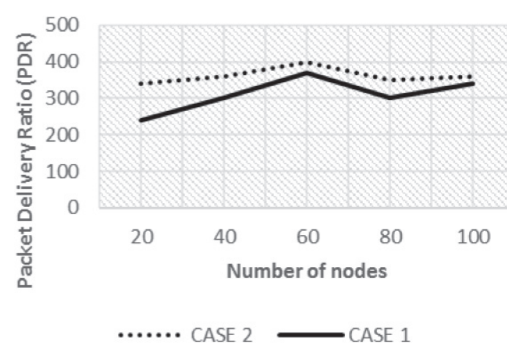

Figure 3. *PDR comparison (Case 1 vs. Case 2)*

Conclusion

In this paper, we proposed a deep learning–based algorithm for packet loss discrimination in MANETs. The proposed algorithm uses the back propagation neural network (BPNN) concept. Based on this loss discrimination algorithm (LDA), we also introduced a new route maintenance scheme that reduces the overall routing overhead of control packets in the network. Such reduction is obtained as the new routing scheme is changing the problematic, "weak" associations inside a link with more robust associations. Obviously, in "weak" associations, a lot of packet failures are observed. We performed simulation experiments for evaluating the performance of the proposed LDA under different network configurations. Through simulation results, we confirmed that the proposed LDA algorithm improves packet loss discrimination and route alteration in the network. It also reduces congestion and increases network throughput. By using simulation, in the near months, we will compare our case/TCP variant (Case 2) with other TCP variants (e.g., TCP New Reno, TCP Vegas, TCP Westwood) over AODV.

REFERENCES

[1] Ali M., Stewart B. G., Shahrabi A. and Vallavaraj A. (2012, March) "Congestion adaptive multipath routing for load balancing in mobile ad hoc networks," In *Innovations in Information Technology (IIT), 2012 International Conference on* (pp. 305-309). IEEE.

[2] Allman M., Paxson V. and Blanton, E. (2009) *TCP Congestion Control*, RFC 5681, Sept. 2009, <https://www.rfc-editor.org/info/rfc5681>.

[3] Basagni S., Conti M., Giordano S. and Stojmenovic I. (Eds.) (2013) *Mobile ad hoc networking: The cutting edge directions* (Vol. 35). John Wiley & Sons.

[4] De Oliveira R. and Braun T. (2004, March) "A delay-based approach using fuzzy logic to improve TCP error detection in ad hoc networks," In *Wireless Communications and Networking Conference*, 2004. WCNC. 2004 IEEE (Vol. 3, pp. 1666-1671). IEEE.

[5] Habbal A. M. M. and Hassa S. (2011, November) "Delay-based loss discrimination mechanism for congestion control in wireless ad-hoc network," In *International Conference on Informatics Engineering and Information Science* (pp. 689-700). Springer, Berlin, Heidelberg.

[6] Iren S., Amer P. D. and Conrad P. T. (1999) "The transport layer: Tutorial and survey," *ACM Comput Surv*, 31(4): 360-404.

[7] Jhaveri R. H. and Patel N. M. (2015) "Mobile ad-hoc networking with AODV: A review," *International Journal of Next-Generation Computing*, 6(3): 165–191.

[8] Kanellopoulos D. (2017) "QoS routing for multimedia communication over wireless mobile ad hoc networks: A survey," *International Journal of Multimedia Data Engineering and Management*, 8(1):42-71.

[9] Khan M. (2016, December) "Cross layer design approach for congestion control in MANETs," In *Advances in Electronics, Communication and Computer Technology* (ICAECCT), *2016 IEEE International Conference on* (pp. 464-468). IEEE.

[10] Klemm F., Ye Z., Krishnamurthy S.V. and Tripathi S.K. (2005) "Improving TCP performance in ad hoc networks using signal strength based link management," *Ad Hoc Netw*orks, 3(2):175-191.

[11] Lochert C., Scheuermann B. and Mauve M. (2007) "A survey on congestion control for mobile ad hoc networks," *Wireless Communications and Mobile Computing*, 7(5): 655-676.

[12] Mallapur S. V., Patil S. R. and Agarkhed J. V. (2015, November) "Load balancing technique for congestion control multipath routing in mobile ad hoc networks," In *TENCON 2015-2015 IEEE Region 10 Conference* (pp. 1-6). IEEE.

[13] Papanastasiou S. and Ould-Khaoua M. (2004) "TCP congestion window evolution and spatial reuse in MANETs," *Wireless Communications and Mobile Computing*, 4(6): 669-682.

[14] Schmidhuber, J. (2015) "Deep learning in neural networks: An overview," *Neural Networks*, 61: 85–117.

[15] Sliwa B., Falkenberg R. and Wietfeld C. (2017) "A simple scheme for distributed passive load balancing in mobile ad-hoc networks," In *Vehicular Technology Conference* (VTC Spring), 2017 IEEE 85th 10.1109/VTC-Spring.2017.8108553 Sydney, Australia.

[16] Tran D. A. and Raghavendra H. (2006) "Congestion adaptive routing in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, 17(11): 1294-1305.

[17] Widmer J., Denda R. and Mauve M. (2001) "A survey on TCP-friendly congestion control," *IEEE Network*, 15(3):28-37.

[18] Yang X., Ge L. and Wang Z. (2009) "F-ECN: A loss discrimination based on fuzzy logic control," In *Proc. Sixth IEEE Int. Conf. on Fuzzy Systems and Knowledge Discovery 2009*, Vol. 7, 2009, pp. 546-549.

[19] Zhu P., Zeng W. and Li C. (2007) "Cross-layer design of source rate control and congestion control for wireless video streaming," *Advances in Multimedia*, vol. 2007, http://dx.doi.org/10.1155/2007/68502>

Submitted: March 28, 2018
Accepted: May 14, 2018

ABOUT THE AUTHORS

Dimitris Kanellopoulos is a member of the Educational Software Development Laboratory in the Department of Mathematics at the University of Patras, Greece. He holds a PhD in Electrical and Computer Engineering from the University of Patras. His research interests include: multimedia networking, MANETs, and intelligent information systems.

Pratik Gite holds a PhD in Computer Science and Engineering from the Pacific Academy of Higher Education and Research University, Udaipur (R.J.). He serves as an Assistant Professor in IES-IPS, Indore. He has participated in many workshops/seminars/conferences. His research interests include: wireless networks, MANETs, software engineering and software testing.

FOR CITATION

Dimitris Kanellopoulos, Pratik Gite, Packet loss differentiation over MANET based on a BP neural network, *JITA – Journal of Information Technology and Applications*, PanEuropien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 8(2018) 1:18-26, (UDC: 551.588:551.506), (DOI: 10.7251/ JIT1801018K), Volume 8, Number 1, Banja Luka, june 2018 (1-44), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004