Neural Network Based Reliable Transport Layer Protocol For MANET With High Degree Of Node Mobility

Abstract—**Use of traditional transport layer protocol to achieve reliability, such as TCP, over mobile ad-hoc network (MANET) is a challenging task due to unique features. The unique features includes absence of a base station, as it is in the case of cellular networks, node mobility, multi-hop communication over lossy and non-deterministic wireless mediums, similarity in traffic pattern experienced by neighboring nodes, etc. One of the main reasons for the poor performance of formal TCP version used in wired networks over wireless networks is that the, packet loss in wireless networks is not only due to congestion there are other problem inherent to wireless network which can cause packet loss. To address this issue, this paper proposes a neural network based congestion control technique for reliable data transfer over MANET, which recognizes and capture the mobility behavior of node. The captured mobility behavior is used to identify the cause of packet loss, in order to take action which increases the reliability of underlying MANET. We evaluate the performance of proposed protocol based on the simulation result, obtained using QualNet 7.4 network simulator. Our evaluation results show that proposed congestion control technique shows improvement in MANET with high degree of node mobility in terms of reliability, bandwidth usage and energy efficiency.**

I. INTRODUCTION

Mobile ad-hoc networks (MANET) are most popular communication paradigm for a variety of diverse applications [1]. Various deployments of ad-hoc multi-hop wireless network (WNs) such as city-wide connectivity, emergency response networking, community network, urban-scale sensor net, etc [2]. demand reliable data delivery during their operations. TCP is considered as the basic standard for the reliable data delivery [3]. To enable the reliable data delivery over wired networks, infrastructure-based wireless network and mobile ad-hoc network, several variants of TCP have been proposed [4]. In any case, generally little work has endeavored to address the challenge in adapting TCP to provide efficient support to the deployments of ad-hoc multi-hop wireless networks.

TCP in wired network scenarios is a transport layer protocol which is responsible for end-to-end reliable delivery of packet between source and destination [3]. Retransmission in an event of packet loss is one of the techniques adopted by TCP to achieve reliability [5]. TCP receive the information of packet loss either in the form of 3 duplicate acknowledgments (3-Dup-ACK) or expiration of time out interval. To distinguish the event of 3-Dup-ACK from retransmission time out, every TCP connection maintains the average deviation of round trip time of packet sent. In traditional wired network the basic reason of packet lost is high traffic rate resulting in congestion, and is resolved by streamlining the flow of data in network by regulating packet generation by different node in network.

Advancement in radio technology in recent decade made it possible to have long range multi-hop communication among devices, creating a network of wireless nodes. Wireless network community initially adopted the traditional wired network protocol stack for wireless network because similarity in its basic client-server approach used for communication. Compared to wired network a wireless communication network have many different features, such as radio wave spectrum in place of cables for communication between nodes, node with mobility capability, channel noise, etc. These wireless network features make it different from traditional wired network in term of bandwidth usage, end-to-end delay and reliability [6].

Due to these additional features in wireless networks, congestion control technique used in TCP, which were proposed for traditional wired network, are not well-suited for direct use in wireless networks. Because there can be other reasons of packet loss other than increased traffic rate. Common reason of packet loss other than increased traffic in wireless networks can be termed as: 1) lossy channel, 2)hidden and exposed station, 3)Path asymmetry, 4)Network partitions, 5)Route failures, 6)Power constraints and 7)Spectrum allocation [4].

In purposed work issue of packet loss due to link failure in MANET is discussed and a novel approach using neural network is presented. MANET is one of the desirable application field of wireless network, because of its capabilities of distributed and mobility assisted properties. Mobility is one of the attribute of MANET which equip it with the ability of movable nodes, which is well suited for application like military surveillance, data networks, sensor networks etc. Mobility in network along with its benefits and flexibilities also introduce additional issues like random topology, frequent link failure, decentralized control, hidden and exposed channel due to interference.

The aim of this proposal is to overcome the issue of bandwidth under utilization due to link failure caused by dynamic mobility behavior. In traditional network the dominating cause of packet loss is congestion; in response to that TCP layer either reduces the congestion window to half of current congestion window size or 1 maximum segment size (MSS) [7]–[9]. However in MANET link failure is also the cause of packet loss and frequency of such loss is high as compared to traditional network. So, adopting the same approaches as in the case of packet loss can cause the underutilization of bandwidth. A model is proposed to capture the behavior of node mobility pattern, which in later stage is used as an input to neural network to recognize the actual cause of packet loss.

The remaining of the paper is organized as follow: section II presents the problem definition, section III presents the literature survey over the proposals made to overcome the problem of bandwidth underutilization in MANET, section IV discusses the proposed model and finally in section V proposed work is validated through simulation result.

II. Problem definition

MANET is the collection of mobile nodes connected by wireless links, enabling them to move freely in any direction [10]. These dynamic and random movements lead to the formation of random and arbitrary topology in underlying network. Formation of random topology make it difficult to predict the source to destination route at any instance of time in underlying network, effecting the performance of network at different layer of communication stack. In the event of packet loss due to link failure or link break the situation become worse, because it is hard to identify the source of packet loss at the source node. Since in the event of packet loss due to link break or route failure does not resemble the state of heavy traffic in network, and adopting the traditional TCP approach of reducing the bandwidth in term of congestion window(C_win), even when there is no congestion in network, causes underutilization of network bandwidth [11].

Similar problem can occur during the local route-reestablishment, since it is possible that TCP source is not aware of the route reformation process and after a delay greater than RTT can launches its mechanism of back-off process causing long ideal delay. Also, old TCP connection may suffer if the new route formed after route re-establishment process have more number of node than the old route, causing the variation in RTT time value at source and actual RTT time [12].

In the proposed work we have designed a structure M_win, which capture the mobility behavior of each node in network and then utilizing it to estimate C_win size using single layer feed forward neural network [13] in the event of error reported either by 3-duplicate ACK or retransmission timeout. A number of work are proposed in literature to achieve the reliability but few have proposed the reliability in term of node mobility behavior [14]–[22].

III. Proposed Work

A. mobility behavior window

A data structure M_win is proposed to capture the behavior of each node mobility dynamics. The working of M_win is similar to left shift register [23]. An instance of M_win is presented in figure.1. Left shift register is an array in which if a value inserted at right rear index (last index of array) all the values shift toward left resulting in the drop of value at the left rear end. For example, in figure.1 if an element is inserted at index (n-1) the values at index (n-1) shift to (n-2), (n-2) to (n-3) and so on, and finally the value at index 1 shift to 0 resulting in the drop or loss of value at index 0.

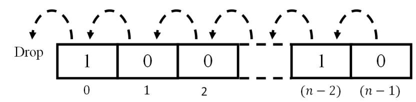

Fig. 1. M_win

Each mobile node maintains the M_win and update in each hello interval. Nodes update the values based on its entry in neighbor table received along with the hello message from one-hop neighbor node. The M_win can hold only binary values i.e. either 0 or 1. A value 1 is inserted if the entry is absent in the table else 0 is inserted. The main idea is that if a node x enter into the communication range of node y in between two consecutive hello interval, its value will not be present in the neighbor list of y. if a nodes movement behavior is regular and dynamic it will be jumping from the communication range of one node to another node, resulting in increased frequency or count of 1 in its M_win.

Conclusion of previous discussion is that if a node is moving too frequently, then the average weighted mean of the M_win of that node will be close to 1. Average weighted mean Wwin is calculated using formula presented by equation.1.

$$W\_win_{(n_i,t)} = \alpha(M\_win_{n_i}) + (1-\alpha)W\_win_{(n_i,t-1)} \quad (1)$$

Where W_win is the average weighted mean in previous hello interval and α is weight associated with data set, such that $(0 < \alpha \leq 1)$.

B. bandwidth estimation

Proposed congestion control mechanism make use of a single-layer, feed-forward, and biased neural network with reinforcement learning. The rate of error either due to congestion or link failure, cumulative moving average of M_win along the path from source to destination, and current C_win size are the inputs of neural network. The neural network computes the estimated congestion window size (est_c_win) as its output. The estimated congestion window size (est_c_wnd) may vary in three different ways-it may increase, decrease or remain same relative to the current c_wnd size.

Justification for using feed-forward NN is as follow: Cumulative moving average of M_win(C_M_win) of all the nodes from source to destination is conveyed to the source node by each incoming ACK packet. C_M_win is used to measure the dynamics related to node mobility along the path between source to destination, and represented by equation 2 .

$$\begin{aligned} C\_M\_win_{n_i,t} &= C\_M\_win_{(n_j,t-\gamma)} \\ &+ \frac{W\_win_{(n_i,t)} - C\_M\_win_{(n_j,t-\gamma)}}{j-2} \end{aligned} \quad (2)$$

The value of C_M_win closer to 1 represent the probability of packet loss due to link breakage, on the other hand value closer to 0 represent the chances of packet loss due to high traffic rate in network (congestion). C_M_win is used as input to NN along with error rate to estimate the c_win in the event of packet loss. Rate of error is measured using rate of consecutive time or duplicate ACK. The key idea is that if rate of error and C_M_win input are high the probability of packet loss due to link failure is more; otherwise the probability of packet loss due to congestion is more. Figure shows our design of the single-layer and feed-forward NN for TCP. The NN has one input layer, one hidden layers, and one output layer. The input layer consists of three neurons (t_out, dack, and c_win) to separately process three different inputs. All neurons in the input layer feed their outputs to all neurons in the hidden layer. The hidden layer determines the relative strength of recommendation for each type of update to be the optimal one to be applied on the current c_win size. Here, we consider three types of updates increase, decrease, and keep the current c_win size unchanged. Three different neurons (incr, decr, and same) separately determine the relative strengths of recommendations for the three types of update. All of these three neurons take inputs from each neuron in the input layer and a Bias (bias) value as input and passed their outputs through activation function to a single neuron (est_c_win) in the output layer. The bias is always set to 1.

1) Adjustments of weights: There are two type of weight that need to be adjusted:

- Fixed Weight
- Dynamic Weight

We continuously tune the dynamically adjusted weights using reinforcement learning to suitably incorporate the response of the underlying network.

Fixed Weight:In Fixed Weight adjustment value of the current c_win size directly to determine the relative strength of recommendations for different types of updates, the extent of the optimal update, and the next c_win size. Therefore, we assign a constant value of 1.0 to all the weights associated with the flows.

Dynamically adjusted weights: Three neurons incr, decr, and same determines the relative order of three types of updates depending on the response from the network to the current c_win size. We calculate a weighted sum of Rate of error, C_M_win, and current c_win size in incr, decr, and same neurons. Here, we utilize the weighted sum to determine the relative strengths of recommendations for three different types of updates.

To consider the contribution of current c_win size in the weighted sum, we have set the weights associated with the current c_win size to 1.0. We have to dynamically adjust the weights associated with the Rate of error and the M_win, as the c_win size varies dynamically during data transmission. For this we define sensitivity factor γ to dynamically adjust the weights. Here, depends on the current c_win size. We set the value of γ as current c_win size γ, where α is the sensitivity granularity. We use a threshold T_error and T_C_M_win corresponding to Rate of error and C_M_win.

2) Choices of activation functions: The most critical part in our design of the NN is the choice of activation functions for the neurons in the hidden layers. We attempt to choose

TABLE I
WEIGHTS IN NEURAL NETWORK

Weight (fixed)	Value	Weight (dynamically adjusted)	Value
Wt	1	Wt	-4*γ,or -2*γ
Wd	1	Wd	-3*γ,or -2*γ
Wc	1	Wc	1

activation functions in all the neurons to be consistent with the choices of the weights.

If the Rate of error and the M_win value grows high, then the possibility of congestion occurring becomes more appealing and activation function act like as additive nature.

If both Rate of error and Rate of C_M_win continuously occur beyond the threshold value T_error and T_ C_M_win then activation function f_incr act as additive nature (figure.2).

if $If(min(rate\_error) > Th\_error \&\& min(C\_M\_win) > T\_C\_M\_win)$ **then**
  $f\_incr = min(min(rate\_error), T\_rate\_error) + min(1 + log(min(C\_M\_win)), T\_C\_M\_win);$
else
  $f_incr = max(min(rate_error), (1 + log(min(C_M_win))))$f_incr=max (min (rate_error),(1+log(min(C_M_win))))
end if

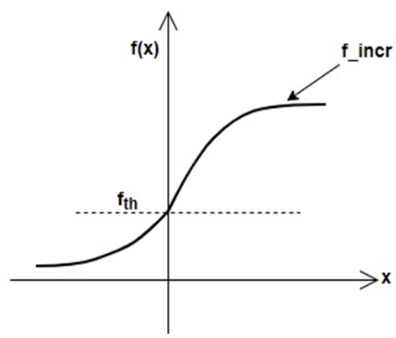

Fig. 2. activation function f_incr

If either Rate of error or C_M_win goes beyond the threshold value then activation function acts as comparative nature.

if $(min(rate\_error) > T\_rate\_error || min(C\_M\_win) > T\_C\_M\_win)$ **then**
  $f\_decr = max(min(min(rate\_error), T\_rate\_error), min(1 + log(min(C)\_M\_win)), T\_C\_M\_win));$
else
  $f\_decr = max(T\_rate\_error, T\_C\_M\_win)$
end if

If the value of both rate_error and C_M_win are close to 0 then next congestion window size will be approx to current congestion window. So activation function f_same can be define as: $f\_same = max(c\_wnd\_size) * B^(- min(c\_wnd\_size))$; where B is exponent factor and its value

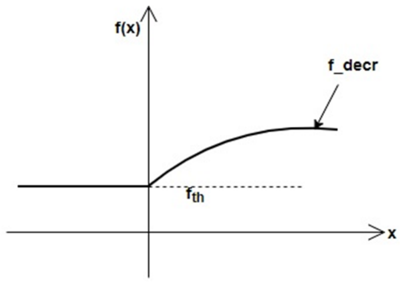

Fig. 3. activation function f_decr

TABLE II
NETWORK PARAMETERS

Sr. no	network parameter	value
1	Routing Protocol	AODV
2	Maximum allowed hello loss	2
3	Maximum buffered packet	100
4	Router type	Cisco-7600-MSFC2
5	TCP variant	Reno,SACK,Tahoe, Lite
6	Sending buffer size	16384
7	Receiving buffer size	16384
8	Maximum segment size	512 byte

is taken as 2.

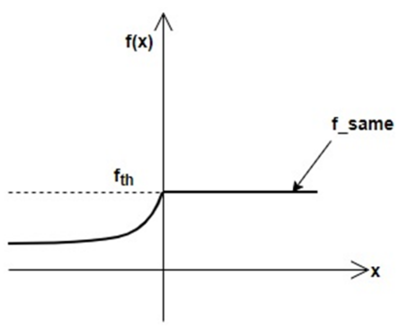

Fig. 4. activation function f_same

Finally We define f_res as final Activation function which is given as: $f\_res = min(max(f\_incr, f\_change), max(f\_same, max(c\_wnd\_size)))$.

IV. Simulation

The simulation results are presented to validate the problem definition, neural network model and network performance under different network parameters. Simulation for problem definition validation and network performance analysis are done using QualNet-7.4 network simulator, while the adapted NN-model is validated using MATLAB.

A. simulation parameter

Common QualNet-7.4 simulation parameter are presented in table?. However, the value for number of static and mobile nodes are varied to analyze the performance of network under varying number of nodes.

B. Simulation Result

First we have validated our problem definition by comparing different mobility models based on the speed and position granularity of the mobile nodes. Three different mobility models are considered and are: 1) Random way-point, 2) Random-walk and 3) Gauss Markov mobility model. Simulation result over number of packet drop, for a network of 50 static node and 20 mobile nodes in a region of [300x300] ,due to link failure with an increase in speed is presented in figure?. From result presented in figure we can analyze that with an increase in mobility speed from [20-50] ms to [40-70] ms the rate of packet drop also increases, from which we can conclude that link failure is also a dominating cause of packet loss along with congestion due to high traffic rate.

the results presented in figure.? network performance is measured based on the packet transmission error. As we can see the network performance curve approaches to the best performance which leads to minimum error in packet transmission. That means we get maximum amount of reliability. In next simulation result presented in figure?,show that

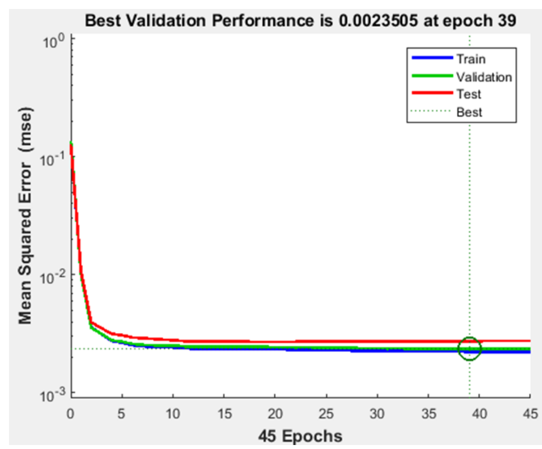

Fig. 5. Network Performance

the expected congestion window size perfectly fit the 'Y=T' regression line. In successive training of Neural Network all data point converge to 'Y=T' regression Line which get maximum reliability and minimum packet drop. Proposed ANN model try to achieve minimum packet loss in network .Following figure? show the in successive transmission and training process packet drop near about to zero error line. Proposed model can also handle the vast traffic of network and try to achieve minimum failure. The following figure? show that network stability at different rate of packet transmission.

V. Conclusion

The proposed work presents a variant of TCP specialized for MANET with high degree of node mobility. The proposal captures the node mobility behavior in data structure M_win. Since link failure due nodes random movement is also a dominating cause of packet loss, other than congestion due high traffic rate, using traditional TCP approach of either reducing

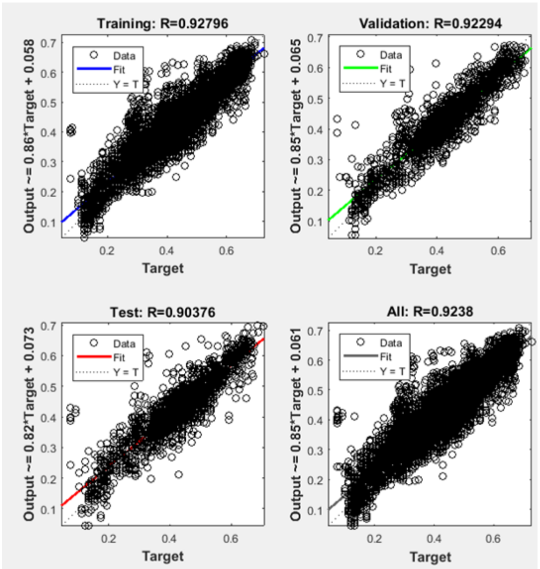

Fig. 6. Expected Congestion Window size

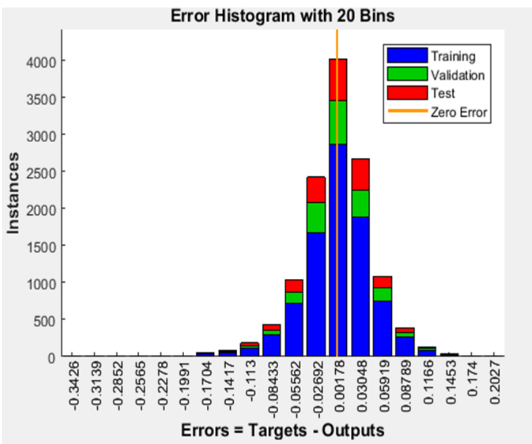

Fig. 7. Packet Drop Rate

current congestion size to half or 1-MSS is not a efficient approach. Because, in case of packet loss due to link failure the desired action should be to maintain the current congestion window size. However, traditional TCP approaches treat both packet losses due to 1) link failure, 2) high traffic data rate, as an indication of congestion. The proposed approach uses the cumulative moving average of nodes along the path from source to destination to identify the reason of packet loss. Cumulative moving average along with error rate and current congestion window size is provided to NN to estimate the C_win in event of error. Future work involves identifying other more efficient machine learning approach to estimate C_win size more accurately.

References

[1] S. Taneja and A. Kush, “A survey of routing protocols in mobile ad hoc networks,” *International Journal of innovation, Management and technology*, vol. 1, no. 3, p. 279, 2010.

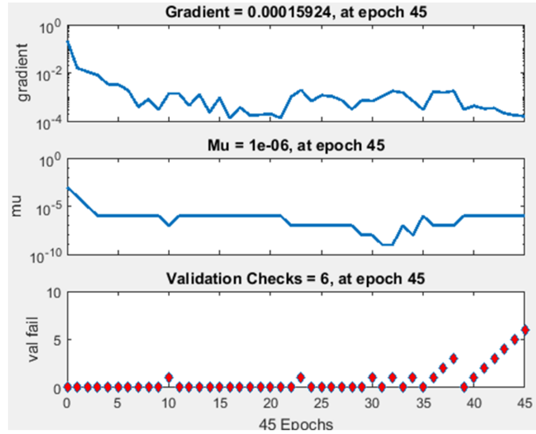

Fig. 8. Network stability

[2] J. Hoebeke, I. Moerman, B. Dhoedt, and P. Demeester, “An overview of mobile ad hoc networks: applications and challenges,” *Journal-Communications Network*, vol. 3, no. 3, pp. 60–66, 2004.

[3] J. Postel, “Transmission control protocol,” 1981.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving tcp performance over wireless links,” *IEEE/ACM transactions on networking*, vol. 5, no. 6, pp. 756–769, 1997.

[5] W. R. Stevens, “Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms,” 1997.

[6] S. Kurkowski, T. Camp, and M. Colagrosso, “Manet simulation studies: the incredibles,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 50–61, 2005.

[7] F. Bai, N. Sadagopan, and A. Helmy, “Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks,” in *INFOCOM 2003. Twenty-second annual joint conference of the IEEE computer and communications. IEEE societies*, vol. 2, pp. 825–835, IEEE, 2003.

[8] G. Lin, G. Noubir, and R. Rajaraman, “Mobility models for ad hoc network simulation,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, IEEE, 2004.

[9] B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, “Impact of node mobility on manet routing protocols models,” *JDIM*, vol. 5, no. 1, pp. 19–23, 2007.

[10] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[11] M. Allman, V. Paxson, and E. Blanton, “Tcp congestion control,” tech. rep., 2009.

[12] G. Holland and N. Vaidya, “Analysis of tcp performance over mobile ad hoc networks,” *Wireless Networks*, vol. 8, no. 2/3, pp. 275–288, 2002.

[13] D. Davidian, “Feed-forward neural network,” Aug. 1 1995. US Patent 5,438,646.

[14] A. Al Hanbali, E. Altman, and P. Nain, “A survey of tcp over ad hoc networks.,” *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1-4, pp. 22–36, 2005.

[15] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, “The impact of multihop wireless channel on tcp throughput and loss,” in *INFOCOM 2003. Twenty-second annual joint conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, pp. 1744–1753, IEEE, 2003.

[16] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, “A survey of energy efficient network protocols for wireless networks,” *wireless networks*, vol. 7, no. 4, pp. 343–358, 2001.

[17] S. K. Das and S. Tripathi, “Adaptive and intelligent energy efficient routing for transparent heterogeneous ad-hoc network by fusion of game theory and linear programming,” *Applied Intelligence*, pp. 1–21, 2017.

[18] S. K. Das and S. Tripathi, “Energy efficient routing formation technique for hybrid ad hoc network using fusion of artificial intelligence techniques,” *International Journal of Communication Systems*.

[19] S. K. Das and S. Tripathi, "Intelligent energy-aware efficient routing for manet," *Wireless Networks*, pp. 1–21, 2016.

[20] A. K. Yadav, S. K. Das, and S. Tripathi, "Efmmrp: Design of efficient fuzzy based multi-constraint multicast routing protocol for wireless ad-hoc network," *Computer Networks*, vol. 118, pp. 15–23, 2017.

[21] S. K. Das, A. K. Yadav, and S. Tripathi, "Ie2m: Design of intellectual energy efficient multicast routing protocol for ad-hoc network," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 670–687, 2017.

[22] S. Pandey, P. Pal, and A. Mukherjee, "Irf-nmb: Intelligent route formation technique in ad hoc network using node mobility behaviour," *National Academy Science Letters*, vol. 38, no. 3, pp. 213–219, 2015.

[23] S. W. Golomb, *SHIFT REGISTER SEQUENCES: Secure and Limited-Access Code Generators, Efficiency Code Generators, Prescribed Property Generators, Mathematical Models*. World Scientific, 2017.