Proceedings of 2017 Tenth International Conference on Contemporary Computing (IC3), 10-12 August 2017, Noida, India

KNNR:K-Nearest Neighbour Classification based Routing Protocol for Opportunistic Networks

Deepak Kumar Sharma, Aayush, Abhishek Sharma, Jaideep Kumar
Division of Information Technology, Netaji Subhas Institute of Technology, University of Delhi, India
E-mails: dk.sharma1982@yahoo.com, {aayush.y1994, abhisharma.oms, jaideepkumar1936}@gmail.com

Abstract—**Opportunistic Networks (OppNets) are an extension of Mobile Adhoc Networks (MANETs), where no assumption is made regarding the preexistent path between the source and the destination node. Hence, the nodes in OppNets are required to rely upon intermediate nodes for successful message delivery. Therefore, the biggest challenge in OppNets for a carrier node is to make a decision whether the neighbour node will be a good carrier for the message in the future or not. Hence, in this paper a new routing protocol called K-Nearest Neighbour based Routing protocol (KNNR) is proposed which judicially forwards the message through intermediate nodes towards the destination. The proposed protocol initially stores the past behaviour of nodes in a dataset. Whenever a decision has to be made related to an intermediate node, the protocol studies this dataset and finds instances that closely resemble the intermediate node based on their network parameters using K-Nearest Neighhbour (KNN) algorithm. To evaluate the efficiency of the proposed protocol simulation results are compared with the existing routing protocols i.e. Epidemic, HBPR and ProPHET. It was observed that the KNNR protocol efficiently reduces average latency, overhead ratio and average hop count while at the same time increases the message delivery probability.**

Keywords— MANETs, ProPHET, KNN algorithm, Opportunistic network routing protocol, ONE simulator

I. INTRODUCTION

OppNets [1] have appeared as an evolution of the MANETs [2]. They are also a wireless based network and hence, they face various issues similar to MANETs such as frequent disconnections, highly variable links, limited bandwidth etc. [3]. In OppNets, nodes are always moving which makes the network easy to deploy and decreases the dependence on infrastructure for communication [4]. However, it also leads to frequent disconnections in the network which makes it difficult to acquire a stable communication between nodes. They make use of the node's mobility to acquire communication between disconnected components of network. Transfer of messages in OppNets is totally dependent on the contact opportunity between the nodes. Due to this reason, the nodes are allowed to follow the store-carry-forward method for routing the packets [5]. But in sparse networks if there is no forwarding opportunity, the message may be stored in the buffer for a long time. This increases the delay period and consequently increases the packet drop chances. Despite all the drawbacks and the challenges associated with OppNets there are various applications which can put up for these issues. The most popular applications are disaster recovery, wireless sensor networks, military applications such as coordinating fleet of airplanes moving at high speeds [6].

Most challenging task in OppNets is routing of message from source to destination as existence of end-to-end paths are unlikely [3]. Initial research was focused on flooding based routing protocols [5] like Epidemic, Spray and Wait etc. These protocols have high resource requirement and often lead to network congestion [7]. Thus researchers shifted their focus towards forwarding based techniques [5] which makes the decision to transfer a packet to the neighbour node after intelligently analysing the node's behaviour. These techniques have moderate delivery ratio but they use lesser resources. This in turn increases the efficiency of the network and therefore makes it more reliable.

After studying various protocols, the authors realized that it is important to keep incorporating previous behaviour of nodes in decision making relating to newer nodes or instances. To classify the current instance with the previous behavior the authors adopt the KNN approach. KNN model is a classifying model which tries to associate an unknown commodity to a known commodity by evaluating the similarities between them [8]. Hence, in this paper a new routing protocol called KNNR is proposed. It consists of two phases: training phase and application phase. In training phase information is collected based on the message delivery performance of the nodes. While during the application phase, decision of whether a node is well equipped to deliver the message is made by comparing them with the past records using KNN algorithm.

In the next section, various prominent routing protocols are discussed in detail. In section 3, the proposed protocol is discussed and in section 4, simulation setup is illustrated along with the results. Section 5 concludes the work and analyses the scope of future modifications and improvements.

II. RELATED WORK

Epidemic routing protocol [9] is the basic routing protocol which employs the technique of flooding for delivering the messages. Whenever a node encounters another node, it transfers all the messages which the other node doesn't possess. Though this strategy ensures high delivery ratio, the network contains several copies of one message at a particular time. That's why bandwidth and buffer requirement for efficient implementation of Epidemic protocol are high.

One of the main limitations of Epidemic routing protocol is that it generates redundant message copies. Spray and Wait protocol [10] limits the copies of a message currently present in the network thus controlling the network congestion. This protocol works in two phases: Spray phase and Wait phase. In

Spray phase, source node creates the message and then transmits it over to L distinct nodes. If none of these relay node is the destination node, then the protocol enters in the Wait phase where all the relayed messages wait for the direct transmission. One of the main advantage apart from controlling congestion is its high scalability i.e. change in network size doesn't affect the efficiency of Spray and Wait protocol.

ProPHET [11] assumes that in real world, nodes do not move randomly but follow a certain predictable pattern. It uses probabilistic metric called *delivery predictability, P(a,b)* $\epsilon[0,1]$, which is maintained by each node *'a'* for every known destination *'b'*. It is the measure of possibility of message delivery from source to destination node. Delivery predictability is calculated considering the frequency of encounters, aging and transitivity. Nodes having high encounter frequency have higher delivery predictability. If a node does not get in contact for a long time then aging factor is taken into account and delivery predictability value is decreased. To forward the message, when two nodes encounter each other they exchange their delivery predictability values and node forwards the message only if the value of delivery predictability with respect to the destination is higher for the other node i.e. the other node has greater possibility to deliver the message to the destination.

History based Routing Protocol (HiBOp) [12] is a context based routing protocol which makes use of the context information stored by the node to find the path which is best suited for successful delivery of message in minimum time. Context can be thought of as information about the environment that the node is currently located in. HiBOp defines two tables: Identity Table and History Table. Using the information stored in these tables, HiBOp protocol determines how similar the context information stored by the neighbor node is with the Identity table of destination. If it is highly similar then message is forwarded to that node. HiBOp drastically reduces resource consumption and message loss rate but consumes a lot of memory by storing the tables.

History Based Prediction for Routing (HBPR) [13] is another context based routing algorithm. To predict next position of the node it uses past movement record of the node. This information can be used to anticipate the geographical location of the node and the neighbours which can be used to find out the proximity of a node to the destination. Using the context information stored in tables, HBPR protocol calculates a parameter called Utility Metric [13]. The Utility Metric is a consequence of three parameters i.e. Node's movement stability, prediction of direction of movement using Markov Predictor [13] and the perpendicular distance of the neighbouring nodes from the line of sight of source and destination. Along with the calculation of Utility Metric, a threshold is fixed. The message copy is given to the node if it has the Utility Metric value more than the threshold.

III. PROPOSED PROTOCOL: KNNR

A. Motivation

Routing in OppNets is always a challenge because of various challenges associated with them such as intermittent connections, random network topology, no direct connection between source and destination etc. But during literary research authors realised that even though various irregularities and challenges exist while designing a routing strategy, certain factors remain constant. Certain network parameters like available buffer size at neighbour node always affect the network performance. This constant behaviour of a node in certain cases motivate the authors to study various network parameters of both the host and the neighbour node to devise a strategy that can make the best decision based on how network responds in the past during similar network conditions. However, in order to make the network learn how to respond in the event of any network condition, an intelligent classifier needs to be implemented to take the decision. Through extensive research authors find that KNN classifier hasn't found its application in the field of OppNets till now. Hence, the authors proposed a new routing protocol called KNNR that uses KNN classifier to classify the response of routing strategy in case of different network settings. In the next subsection proposed protocol is explained in detail.

B. Decision Parameters

To implement the KNNR protocol six new parameters are introduced namely *Buffer Space Available, Time-Out-Ratio, Hop Count, Neighbour Node Distance from Destination, Interaction Probability and Speed of Neighbour*. These parameters define different useful characteristics of nodes which will help in making the final decision in application phase. Interaction Probability is also used to generate the dataset in training phase. In the following subsections, each decision parameter is explained in detail.

B.1) Buffer Space Available

All nodes in the network maintain a constant sized FIFO data structure which contains the copies of the messages that need to be transferred by that particular node. Consequently if there is no space available in the buffer and there is an incoming request for message delivery, the oldest message in the queue is dropped. This is not an ideal scenario where important messages may be dropped especially when network has high traffic probability. On the other hand, it ensures that if the message has been sitting in the queue for a long time, it might hinder the possibility of other messages being delivered. Hence, examining the buffer space available can guide the message towards a more suitable node.

A node in an OppNet stores two parameters related to the buffer namely $Buffer_{total}$ and $Buffer_{used}$. Available Buffer space can be calculated as follows:

$$Buffer_{available} = {Buffer_{total} - Buffer_{used}}/{Buffer_{total}} \quad (1)$$

In training phase $Buffer_{available}$ is a critical parameter. If the buffer is relatively filled, it is better to look for a better node as it might cause substantial delay in message delivery. Hence, buffer space available helps in determining the efficiency of node to deliver messages.

B.2) Time-Out Ratio(TOR)

Due to constant buffer size, whenever it gets full, the oldest message gets dropped. So it is beneficial to make sure that the message does not stay in the buffer for too long as it might be dropped. For this, *TOR* of the message is studied to make sure that if the message has been sitting in buffer for too long, it is transferred to another node in hope that the other node would be more equipped to deliver the message. Here message maintains a parameter, $Buffer_{time}$ which is the amount of time the message has spent sitting in the buffer. Similarly the node maintains a parameter $Time_{maximum}$ which is the maximum amount of time for which the message can stay in the buffer. Percentage of time spent in buffer is given by:

$$TOR = {}^{Buffer_{time}}/_{Time_{maximum}} \tag{2}$$

If *TOR* is high it means that the message has been in the buffer for an extended period of time and it would be advantageous to transfer the message, as it would be more probable for the other node to deliver the message and also maintain a fairly empty buffer for delivery of new messages.

B.3) Hop Count

In OppNets transmission of message between source and destination node occurs hop by hop. So a parameter which keeps track of total number of hops travelled by the message can be very useful. Hop count is a parameter which counts how many intermediate nodes the message has passed through, before reaching the destination node. Messages which have high hop count value ensures high successful delivery ratio and faster message delivery to the destination [15]. But such messages will have redundant copies in the network and this increases resource consumption. Messages with low hop count value relates to direct transmission. It will require messages to stay in node buffers for longer time. Hence, a value neither very high nor very low is expected for the hop count field. So, the KNNR protocol will ensure a desirable value for the hop count is achieved to maximise the successful delivery ratio as well as minimising the resource consumption.

B.4) Neighbour Node Distance from Destination

While delivering a message it is required that with every hop, message should get as close as possible to the destination, because a node which is closer to the destination at current instant may have better chance for successful delivery in the future. Hence, this parameter plays a vital role in minimising the transmission delay. The proposed protocol calculates the distance of neighbour from the destination by assuming that the simulation area is a 2D plane. The KNNR protocol defines a new standardised distance formula as shown below:

$$\sigma = Distance(C, D) - Distance(N, D) \tag{3}$$

where *C*, *D* and *N* denote the carrier, destination and neighbour node respectively and σ is the standardised distance. *Distance(X_1, X2)* is the distance between X_1 *and* X_2.

B.5)Interaction Probability

This parameter defines the odds of a node to successfully carry out a message delivery towards a specific node. It is based on the probability of encounter between two nodes. Nodes that frequently encounter each other have higher probability of encounter as compared to nodes that rarely meet. When a node X_1 encounters a node X_2 then the probability of encounter is calculated as [11]:

$$P(X_1, X_2) = P(X_1, X_2)_{old} + (1 - P(X_1, X_2)_{old}) * P_{int} \tag{4}$$

where P_{int} is the initialisation constant.

If two nodes have had no interaction for a long time, their probability must age as it is quite possible that nodes may have moved farther away from each other in that time. Aging factor is calculated as [11]:

$$P(X_1, X_2) = P(X_1, X_2)_{old} * \gamma^s \tag{5}$$

where *s* is the total number of time slice units since the last update.

Probabilities defined for two nodes are also transitive in nature i.e. if nodesX_1 and X_2 meet frequently and node X_2 and X_3 also meet frequently then consequently node X_1 will have high probability of interaction with node X_3 through node X_2.

$$P(X_1, X_3) = P(X_1, X_2)_{old} + (1 - P(X_1, X_3)_{old} * P(X_1, X_2) * P(X_2, X_3)) \tag{6}$$

If two nodes wants to interact with each other in the proposed protocol, the carrier node requests probability of interaction which is interpreted as a probability of carrying out a successful message delivery to the destination.

B.6) Speed of Neighbour Node

Every node in OppNets is usually in motion. This leads to an unreliable connection between the nodes because regular movement results in the nodes moving out of transmission region, leading towards connection loss. Each node in a OppNet records its past speed and calculates the $Speed_{average}$ parameter. If the current speed of the neighbouring node is more than its $Speed_{average}$ it represents unstable behaviour of a neighbour node. In these type of cases even if neighbouring node is good carrier of message, it is better not to transfer the message because unstable behaviour of node will eventually cause connection loss and hence, message will also be lost. When two nodes interact with each other then, the carrier node requests for two parameters i.e. $Speed_{current}$ and $Speed_{average}$. The neighbour node sends these parameters and carrier node calculates $Speed_{difference}$:

$$Speed_{difference} = Speed_{average} - Speed_{current} \tag{7}$$

$Speed_{difference}$ is negative when the $Speed_{current}$ of node is higher than $Speed_{average}$.

C. Phases of KNNR

This section describes the two phase namely Training phase and the Application phase of KNNR.

C.1) Training Phase:

For training the network KNNR protocol requires a dataset which is created by simulating the network for *'A'* units of time and recording the values of network parameters as mentioned above. Since KNN is a classification algorithm, obtained dataset needs to be classified into different classes. The KNNR protocol defines two classes: “Class 0” and “Class

1". "Class 0" depicts a scenario when the message is not transferred while "Class 1" represents the scenario when the message is transferred to the neighbour node. In OppNets it is very difficult to decide whether transferring the message to a neighbour node is the optimal choice for successful delivery towards destination. This is overcome by using interaction probability parameter defined above because message delivery in OppNets depends on opportunity of encounter between two nodes and interaction probability parameter describes how frequently two nodes encounter each other. So it provides a good measure to classify the decision. When a carrier node encounters a potential intermediary node it compares the node's interaction probability with its own value with respect to the destination. It helps the carrier node to classify the given dataset. Equation (8) mathematically defines the classification of obtained dataset.

$$D = \begin{cases} 1\,, if\ IP(N, destination) > IP(C, Destination) \\ 0, if\ IP(N, destination) < IP(C, destination) \end{cases} \quad (8)$$

here C and N represents carrier and neighbour node respectively. $IP(X,Y)$ is the interaction probability between X and D is the decision to transfer the packet. The packet will be transferred if $D = 1$ otherwise, if $D = 0$ then packet will not be transferred.

Another challenge in training phase is the random nature of nodes which leads network parameters to change constantly with time, so it is necessary to update the dataset periodically. Hence, dataset is continuously updated with new instances by training for *'A'* units of time after every *'B'* units of time until the routing protocol is relaying the messages. Dataset is refreshed at the beginning of every training phase which not only results in low overhead but also helps in adapting to changing environment.

C.2) Application Phase:

In training phase a dataset is created by recording latest instances and their decisions. This dataset is used to make decision in application phase. Previous instances are considered for evaluating the decision because it is likely that the real users will repeat their paths and their movement can be somewhat predictable. Hence, in the application phase, KNN algorithm classifies current network instances by using a specialised distance function with respect to all the previous instances available in the dataset. In other words when the carrier node interacts with a neighbour node it asks for the decision parameters from the neighbour node. For this instance of network parameters, the distance function is used to calculate its deviation with already existing instances in the dataset. Distance function is a function which is used to determine the degree of closeness between two instances. Distance function used in this work is mathematically represented as:

$$D = \sum_{i=1}^{6}(N_i - C_i) * Z_i \quad (9)$$

where Z_i is the standardisation constant for i_{th} parameter. D is the distance between current instance and the instance in the dataset. C_i represents value of i^{th} decision parameter for current node and N_i represents the value of i^{th} decision parameter for dataset instance. Value of Z_i is -1 or 1 depending on if it helps or hinders message delivery.

After calculating all the distances from previous instances, K nearest distances are chosen to determine the decision for current instance. To decide whether the neighbour node is the favourable next carrier or not, average decision of K nearest neighbours recorded above is taken. This average decision calculated by the proposed protocol is then compared with the threshold. If value of the average decision parameter is greater than or equal to the threshold then the neighbour node is selected as the next carrier node. An important step is to determine the value of K (number of nearest neighbours taken). A high value of K ensures more precise results but will increase computational complexity. While a low value of K won't give reliable results.

IV. Simulation Setup and Results

1) Simulation Setup

Simulation of Benchmark protocols and the KNNR protocol is done using the ONE simulator [14]. ONE simulator is run on a machine with Intel(R) Core(™) i7-4700MQ processor with clock speed of 2.40GHZ, 8GB of RAM and running on Windows 8.1 Pro. The simulation scenario defined for the proposed work includes 3 tram groups containing 2 nodes in each and 3 pedestrian groups having 40 nodes in each by default. Default size of the buffer is 50MB for Tram groups and 15MB for pedestrian groups. Transmission interface chosen for Tram groups is High Speed Bluetooth Interface which has transmission range of 1500M and 10MB/sec transmission speed while Bluetooth Interface is used for pedestrian group which has transmission range of 20m and 250KB/sec transmission speed. Interval between message generation is set to be 25-35 sec while Time-to-Live(TTL) is kept at 300min. Walking speeds of tram and pedestrian range from 5-6.5 Km/h and 0.5-1.5 Km/h respectively. The simulation run time is 43200 sec. Table I defines the values of constant used in the proposed work.

After simulation, comparison between the KNNR protocol, Epidemic, ProPHET and HBPR is made based on the results obtained. During evaluation of the KNNR protocol, TTL is varied from 100 to 300 minutes with increments of 50min. each time. Similarly the number of nodes is varied from 66 to 186 in the steps of 30 which helps in thorough comparison of results. When these above mentioned parameters are varied, rests of the parameters are kept at their default values as stated above. Evaluation of the performance is done by recording message delivery ratio, average latency time, average hop count and overhead ratio at each instance. In the next subsection, various results are discussed in detail.

TABLE I. PARAMETER DEFINED IN KNNR PROTOCOL

Parameter	***Value***
K	50
Class Threshold	1
A	1000
B	10000
P_{int}	0.75

2) Simulation Results

Figures 1-4 shows the comparison of message delivery probability, average latency, average hop count and overhead ratio when the number of nodes is varied. From Figure 1, it can be inferred that the KNNR protocol performs better and has substantially greater message delivery probability than other protocols. This is directly related to the use of KNN which helps in selection of a better intermediary node. Due to the expanding dataset, decision making capability of the protocol keeps improving with time resulting in better future decisions. In Figure 2, it can be observed that proposed protocol has a lower average latency than the other protocols due to the careful selection of the intermediary node which leads to decrease in the time gap between creation and delivery of the message. In Figure 3, it is shown that the KNNR protocol has smaller average hop count which is consistent with the fact that with better decision making the path chosen is optimal, hence, requiring less number of nodes to deliver the message. Similarly, in Figure 4 it can be observed that the proposed protocol has better performance in terms of overhead ratio as the selected intermediary node is highly capable of guiding the message towards the destination which leads to fewer numbers of copies being created.

Figures 5-8, show the comparison of message delivery probability, average latency, average hop count and overhead ratio when the TTL is varied. In Figure 5, it can be inferred that the KNNR protocol has better delivery probability due to better selection of intermediary node. But with increase in TTL, messages are getting more time to wait for the optimal node. This act as an anchor for other messages as after certain limit messages start to block other messages by occupying the buffer space. In Figure 5 it is clearly evident that the message delivery ratio is first increasing and then decreasing. From Figure 6 it is evident that average latency is slightly more when compared with other protocol but it is still within acceptable range. This is because of the fact that with increasing TTL nodes are getting more time in their buffer to wait for the more optimal node. This sometimes results in the increase in the latency time before the delivery of message. From Figure 7 and Figure 8, it can be observed that performance of the KNNR protocol is better in terms of average hop count and overhead ratio. This is due to the protocol's ability to choose optimal intermediary node which leads to optimal path of delivery leading to less nodes being travelled and hence, less copies of message are created.

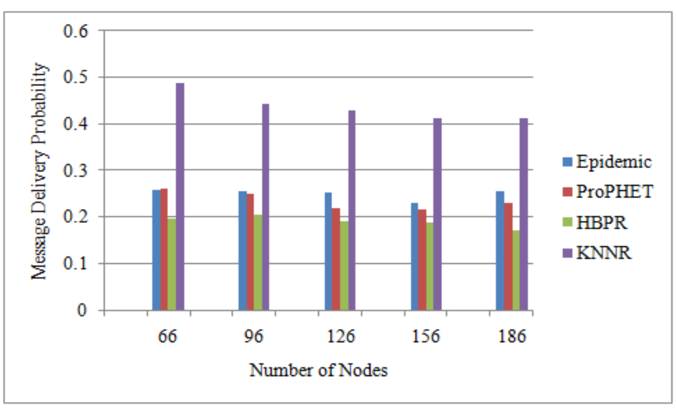

Figure1. Message Delivery Probability vs Number of Nodes

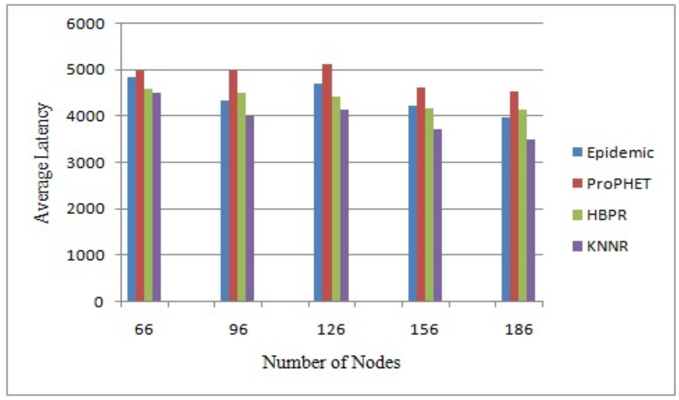

Figure 2. Average Latency vs Number of Nodes

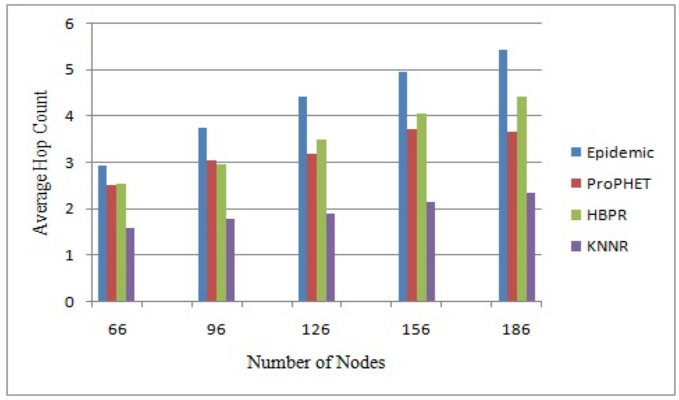

Figure3. Average Hop Count vs Number of Nodes

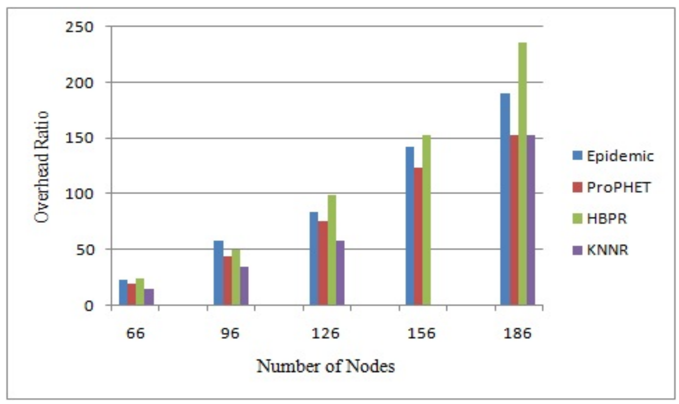

Figure4. Overhead Ratio vs Number of Nodes

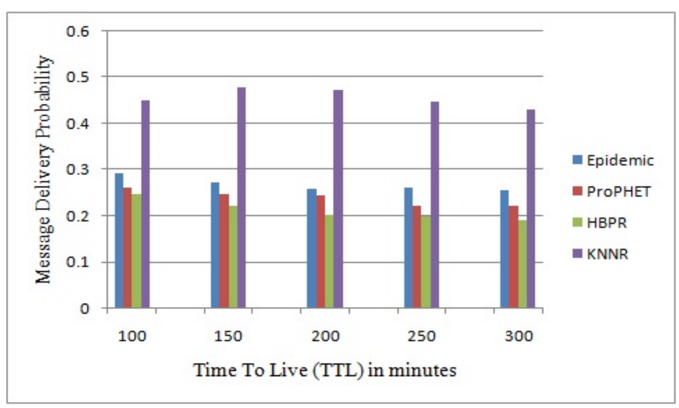

Figure 5. Message Delivery Probability vs Time-to-Live

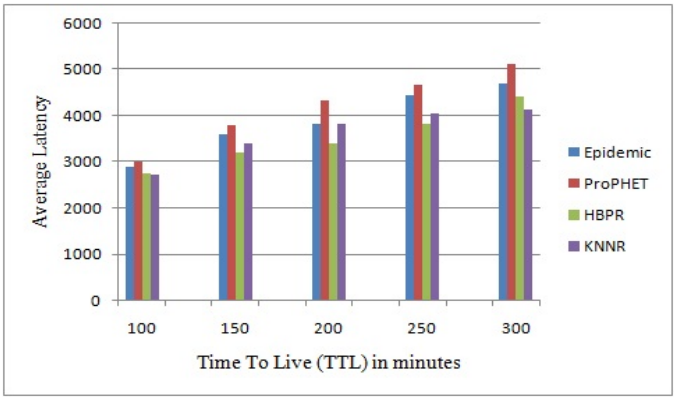

Figure 6. Average Latency vs Time-to-Live

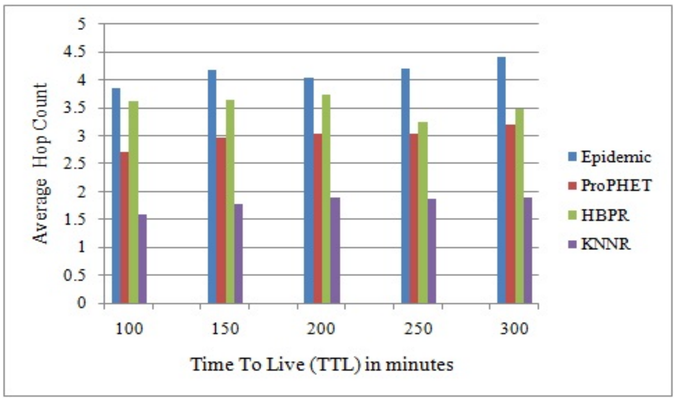

Figure 7. Average Hop Count vs Time-to-Live

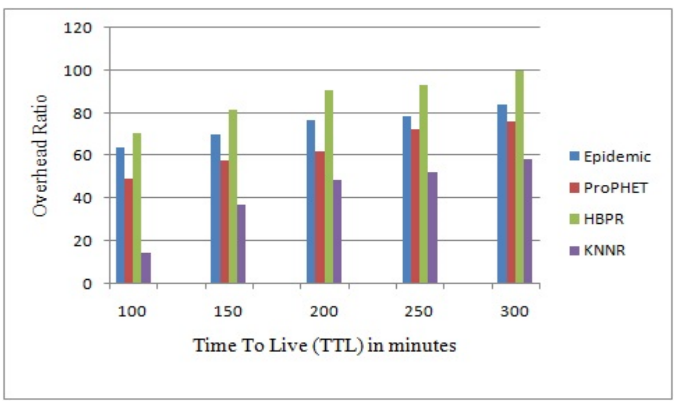

Figure 8. Overhead Ratio vs Time-to-Live

V. CONCLUSION AND FUTURE WORK

In this paper KNNR protocol for OppNets has been proposed. Due to unpredictable nature of nodes, it becomes important to keep integrating the node's past behaviour in decision making relating to new nodes or instances. The proposed protocol consists of two phases: training and application. In training phase dataset is created by recording different network parameters namely buffer space available, Time Out Ratio, hop count, neighbour node distance from destination, speed of neighbours, interaction probability. After building the dataset the protocol enters in the application phase where decision regarding message transfer to the neighbour node is made based on the class it belongs. Through simulations, it is observed that the KNNR protocol is superior in terms of message delivery probability, average latency, average hop count and overhead ratio when compared with Epidemic, ProPHET and HBPR protocols due to restriction imposed on the succeeding hop selection process by this protocol.

In future, comparison between other routing protocols that implements similar decision making process for message transfer like CRPO Protocol [16] can be made based on their efficiency. It would be interesting to observe how the proposed protocol would perform when other network parameters like available bandwidth or node power are considered.

REFERENCES

[1] L Lilien, ZH Kamal, V Bhuse, A Gupta. "Opportunistic networks: the concept and research challenges in privacy and security". In: Proceedings of NSF international workshop on research challenges insecurity and privacy for mobile and wireless networks (WSPWN 2006), Miami, March 2006, pp 134–147.

[2] C-K Toh. "Ad hoc mobile wireless networks: protocols and systems". Prentice Hall PTR, Englewood Cliffs, 2002

[3] S.K. Dhurandher, D.K. Sharma, I Woungang, HC Chao, "Performance evaluation of various routing protocols in opportunistic networks". In Proc. of IEEE GLOBECOM workshop 2011, Houston, Texas, 5-9 Dec 2011, pp 1067-1071.

[4] S.K. Dhurandher, D.K. Sharma, I Woungang, R Gupta, S Garg, "GAER:genetic algorithm-based energy-efficient routing protocol for infrastructure-less opportunistic networks", The Journal of Supercomputing, Springer, 2014.

[5] C-M Huang, K-C Lan and C-Z Tsai, "A Survey of Opportunistic Networks", 22nd International Conference on Advanced Information Networking and Applications Workshops, 2008, pp.1672-1677.

[6] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks", IEEE Communications Magazine, Vol. 44, No. 11, Nov. 2006, pp. 134-141.

[7] T. Anindya, D. Rajkumar, K. Humayun, and C. S. Gholamali, "TTL based routing in opportunistic networks," Journal of Network and Computer Applications, vol. 34, no. 5, 2011, pp. 1660–1670 .

[8] B.V. Dasarathy. Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, 1991.

[9] A Vahdat, D Becker, "Epidemic routing for partially connected ad hoc networks". Technical report S-2000-06, Duke University, Durham.

[10] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Net-works", Proc. of SIGCOMM Workshop on Delay-Tolerant Networking, PA, USA, Aug. 22-26, 2005, pp. 252-259.

[11] A Lindgren, A Doria, O Schelen, "Probabilistic routing in intermittently connected networks," ACM SIGMOBILE Mobile Computing and Communication, 2003 Review vol.7,Issue 3, 2003, pp 19-20.

[12] C Boldrini, M Conti, I Iacopini, A Passarella, "HIBOP: a history based routing protocol for opportunistic networks," In Proc of IEEE WOWMOM 2007, Espoo, Finland, 18-21 June 2007, pp 1-12.

[13] S.K. Dhurandher, D.K. Sharma, IsaacWoungang, S. Bhati, "HBPR: History Based Prediction for Routing in Infrastructure-less Opportunistic Networks," in Proc of 27th IEEE AINA 2013, Barcelona, Spain, pp 931-936.

[14] A Keranen (2008)"Opportunistic network environment simulator special assignment report". Helsinki University of Technology, Dept. of Communications and Networking, May 2008.

[15] S. Bayhan, E. Hyyti¨a, J. Kangasharju, and J. Ott., "Analysis of hop limit in opportunistic networks by static and time aggregated graphs", In IEEE ICC, 2015.

[16] A. Gupta, A. Bansal, D. Naryani, D.K. Sharma, "CRPO: Cognitive routing protocol for opportunistic networks", in Proc. of HP3C-2017, Kuala Lumpur, Malaysia, 2017.