2014 International Conference on Advances in ICT for Emerging Regions (ICTer): 241 - 246

An Adaptive Routing Protocol with Congestion Avoidance for Opportunistic Networks

Thabotharan Kathiravelu[#1], Nalin Ranasinghe[*2]
[#] *Department of Computer Science, University of Jaffna,*
Jaffna, Sri Lanka
[1] `thabo@jfn.ac.lk`
[*] *University of Colombo School of Computing,*
35, Reid Avenue, Colombo 7, Sri Lanka.
[2] `dnr@ucsc.cmb.ac.lk`

Abstract--**Routing and forwarding of messages towards destinations in opportunistic networks is a challenging task due to the topological uncertainty caused by node mobility and frequent disconnections between node pairs. One of the recently proposed adaptive routing protocol [1] for opportunistic networks makes informed forwarding decisions based on the expected level of connectedness and the predictability of nodes as determined heuristically on the past history of contacts [1], [2]. Though this kind of forwarding in adaptive routing increases the final delivery probability, some nodes in the network will have to devote more of their resources than others as popular nodes often get congested with too many messages to store and carry, and be forced to drop incoming messages. In opportunistic networks, when congestion occurs at intermediate nodes messages get dropped and will not be forwarded towards their destination. In this paper we propose, implement and evaluate the performance of an enhanced congestion aware adaptive routing protocol for opportunistic networks and show that the proposed routing protocol outperforms many of the well-known routing protocols in the field.**

***Keywords*—congestion avoidance, opportunistic routing protocols, adaptive routing protocol**

I. INTRODUCTION

Routing in opportunistic networking has always been an interesting problem because of the non-existence of the end-to-end path between the source and the destination nodes. Routing proposals that range from flooding based approaches to probabilistic properties based approaches have been proposed by researchers over the years [3],[4],[5],[6],[7],[8],[9],[10],[11],[12]. The adaptive routing protocol proposed by Kathiravelu et al., [1] makes informed forwarding decisions based on the expected level of connectedness and predictability of nodes as determined heuristically on the past history of contacts. By using the predictability and connectedness information based on the past history, nodes can also predict their future contact opportunities with a good level of confidence. Experimental results indicate that the proposed adaptive routing protocol [1] outperforms three well known protocols in the field [4],[9],[8].

Though the forwarding approach employed in the adaptive routing protocol increases the final delivery probability, some nodes in the network will have to devote more of their resources than others for the overall benefit of the network. Even though the adaptive routing protocol takes in to account properties such as the reachability of a destination and number of hops needed to transfer a message etc., some nodes which are more popular than others will become congested and will be drained out of power. The heuristics considered so far do not consider the potential node's inability if any, to hold incoming messages. This will severely affect the final message delivery as popular nodes often get congested with too many messages to store and carry, and therefore will often have to drop them.

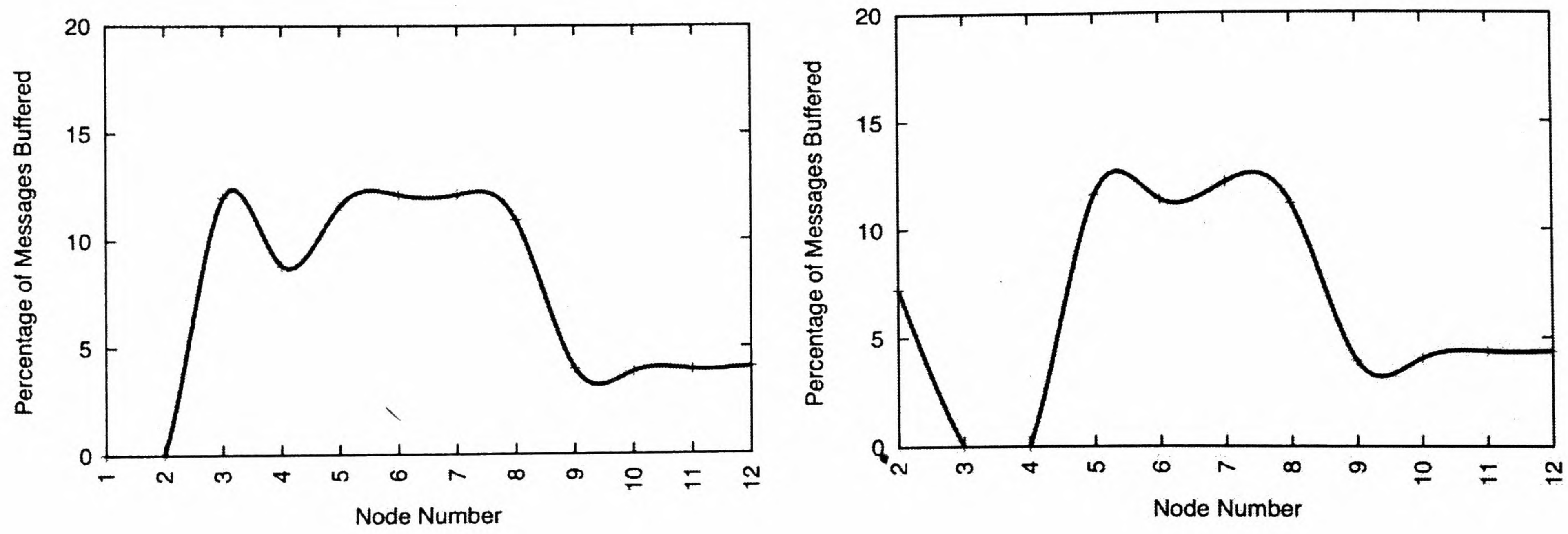
Conventional congestion control approaches in MANETs simply drop messages subject to a replacement policy and this results in messages being completely lost in transit [13],[14]. Participating nodes in opportunistic networks too, often apply cache replacement policies, in order to allocate buffer space for newly arriving messages, which results in dropping of a significant number of messages especially at the heavily utilized nodes. In opportunistic networks, due to the non-employment of an end-to-end path and the presence of frequent disconnections between nodes, source nodes have no way of being reactively made aware of the congestion at the intermediate nodes and the resultant packet loss. When congestion occurs at intermediate nodes, they refrain themselves from accepting messages for forwarding, the network becomes highly partitioned and messages will not get delivered. Therefore the conventional approaches for congestion avoidance and control which are built based on the assumptions of an end-to-end feed-back loop are not applicable for such partitioned, intermittently connected dynamic networks. Our initial investigations in the application of traditional message drop policies have indicated us that the node congestion has a significant impact on the network performance and more controlled approaches for congestion avoidance are therefore needed.

In this paper we propose, implement and show by simulation based test studies that a predictability and connectedness based [1] congestion aware adaptive routing protocol outperforms other well known routing protocols in message delivery. Here our enhanced congestion aware protocol enables nodes to utilize the inherent properties of predictability and connectedness to make well informed decisions on accepting messages to forward while the node is in the state of congestion. We also show by simulation based tests that our protocol achieves the best message delivery per unit of power expended.

II. CONGESTION EFFECTS AND THE ADAPTIVE ROUTING PROTOCOL

In [1], authors have described the design and analysis of an adaptive routing protocol for opportunistic networks, and have shown how the protocol outperforms three other

978-1-4799-7732-1/14/$31.00 ©2014 IEEE



(a). For the case of 50P50C (b). For the case of 90P50C

Fig. 1. Messages buffered at each node as a percentage of overall buffered messages by nodes running Adaptive Routing Protocol with buffer sizes ranging from 20 to 100 messages for the cases of 50P50C and 90P50C [1] with Just Drop buffer management policy

wellknown protocols namely the Epidemic [4] Prophet [9] and Hibop[8] routing protocols.

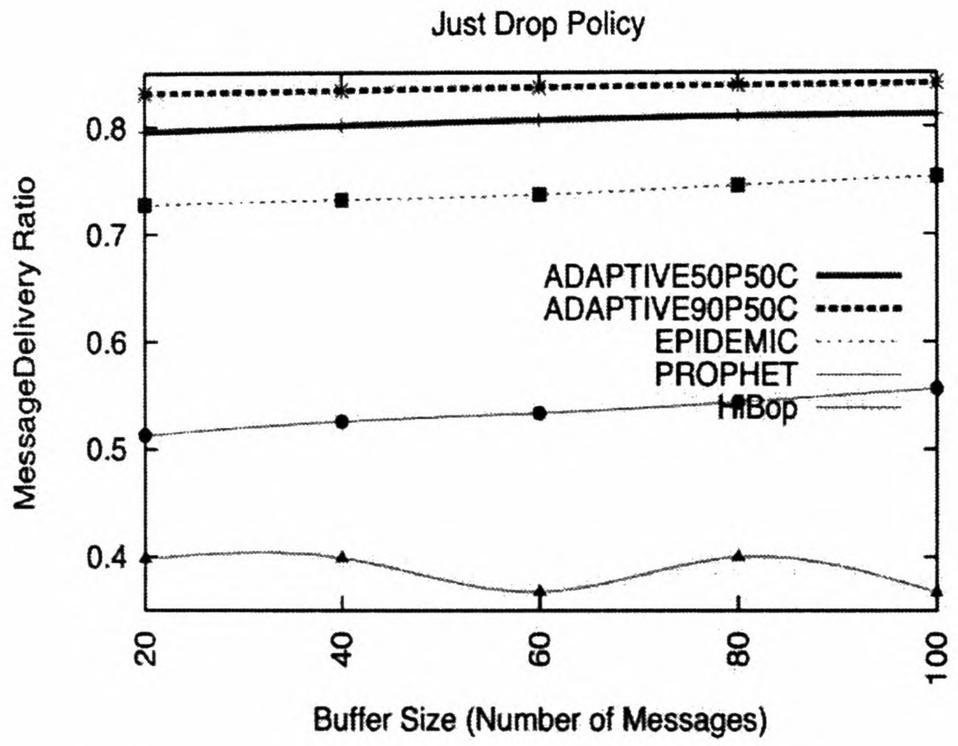


Fig. 2 Message delivery ratio of the Adaptive routing protocol compared with the Epidemic [4], Prophet [9] and Hibop [8] routing protocols.

We wanted to investigate the performance of the adaptive routing protocol [1] in the presence of congestion in an opportunistic content distribution scenario. We have considered the same evaluation scenarios as described in [1] in this regard. We compare the performance of the adaptive routing protocol against Epidemic, Prophet and Hibop routing protocols where, no buffer management policies have been enforced, which is referred to as Just Drop policy [15], and received messages are dropped when the buffer becomes full. In these studies the buffer size was varied from 20 messages to 100 messages with a step wise increment of 20 and the Message Delivery Ratio (MDR) was used as the performance metric [1]. From Fig. 2 we can clearly observe that for the above cases, the adaptive routing protocol outperforms all three routing protocols in MDR. A further observation that can be made is that, it shows a higher MDR for a predictability of 0:9 than when the predictability is at 0:5. We can also observe that except for the Hibop routing protocol, the Adaptive, Epidemic and Prophet routing protocols show an increase in their message delivery ratios when the buffer size gets increased, even though the increased MDR in the Epidemic and Prophet routing protocols are still lower than the MDR of the adaptive routing protocol.

A further analysis of the collected connectivity information data from this simulation study reveals that a few nodes are more popular than the others and have been committing much of their buffer space for the benefit of the others. In Fig. 1(a) and Fig. 1(b) we plot the average percentage of buffer usage for the two cases of 50P50C and 90P50C of the adaptive routing protocol. These plots indicate that even though the adaptive routing protocol helps achieve a higher message delivery, it does not eliminate congestion at most popular nodes. In Fig. 1(a) we can see that node 3 gets most of the messages buffered, nodes 6 and 7 also buffer a large amount of messages, where as nodes 1 and 2 do not buffer even a single message. Similarly, for the case of 90P50C, in Fig. 1(b) we can see that node 1 gets most of the messages buffered, nodes 5 and 7 also buffer a large amount of messages, whereas nodes 3 and 4 do not buffer even a single message. This clearly indicates that it is the estimated values of the predictability and connectedness information which determine the future contacts a node makes with its neighboring nodes. As such there is a need to further investigate and propose policies for forwarding buffered messages while minimizing congestion in the network nodes.

III. INCORPORATING CONGESTION AVOIDANCE IN THE ADAPTIVE ROUTING PROTOCOL

We propose a congestion aware adaptive routing protocol, where a sending node makes forwarding decisions in an adaptive manner also taking into consideration, the space needed for incoming messages in the message buffer. When the receive buffer of a node gets filled up and the node has a limited buffer size, then there is a need to decide on which messages need to be accommodated and which need not, the congestion aware algorithm considers the number of hops that an arriving message needs to be forwarded further in order to reach its destination. If the number of hops to be traversed is higher, then that message is accommodated in the buffer. Here we consider a message which still needs to be forwarded on many hops to reach its destination as a higher priority message, than a message which has only a few hops yet to reach its destination. The rationale behind this approach is that when a node receives a message that has many number of hops yet to reach the destination, it will notice that the predictability and connectedness values stored for a particular destination of the message to be lower than that for meeting other destinations by the same message because it has no means of meeting the destination directly or in a few hops, and therefore will not prefer to receive that message. This will create a situation where nodes that receive messages that have lower predictability and connectedness values, and have many hops to travel towards their destinations, will simply be ignored. In addition, assuming that a message has already traversed many hops means that somehow it could have reached its destination or is very close to the destination when it gets forwarded in a hop-by-hop manner, and therefore it can be given a lower priority over the other messages that have traversed a few number of hops [15]. We want to avoid such a situation and ensure that even the messages with many hops to traverse will also get forwarded by the intermediate nodes.

Alternatively, if the number of hops already traversed by a message was found to be higher, it could be preferentially accommodated too. The rationale behind this is to accommodate the message which has already traversed many number of hops and not to simply ignore the cooperative effort put forward by all the nodes in an opportunistic networking environment. The consequence of this approach has already been discussed in the MOFO (evict the MOstFOrwarded packet first) buffer management policy [15]. Accommodating a message, which has traversed many nodes, will prevent a message, which has been forwarded only a few times, from being forwarded by its neighbors. This will prevent messages that have just been forwarded by only a few nodes, from being forwarded by intermediate nodes without considering the number of hops that these messages need yet to traverse to reach their destinations. As a result many messages which have just begun their journey will not be preferred by the intermediate nodes and will not be able to reach their intended destinations.

When two nodes meet, they exchange their summary vectors, as described in [1], which would include each node's willingness to receive messages based on its available buffer space, and based on the receiving node's willingness to receive messages for other nodes, the sending node will prepare a list of messages in its buffer and will then forward the list towards the receiving node. When the receiving node receives the list, it will check if it is in direct contact with the forwarder nodes and if so, then it will receive such messages and shall forward them directly towards the forwarder nodes. If the forwarder node is not directly reachable according to the predictability table of the receiving node, then it will keep the messages in the buffer until it meets the best forwarder node.

When a node receives a message and wants to store it in its buffer, it will first check the buffer space occupancy. As long as the buffer occupancy is below one half of buffer size, the node can accommodate an arriving message without much problem [16]. When the buffer space fills up and the occupied buffer space goes above a set threshold, say one half of the total buffer size, our congestion avoidance algorithm takes control in deciding which messages are to be kept in the buffer. Here the algorithm considers the available buffer space under two cases: in the first case it checks if the occupancy of the buffer is above a "safety" threshold margin, which is dependent on the traffic generation rate of the node and the current occupancy of the buffer.

Algorithm 1: message Acceptance Logic – Receiver. The node which is in contact with a new node will execute this algorithm and shall exchange its buffer space availability to accept messages from the node it has just met.

```
Input:   Buffer size $B_{capacity}$
Input:   Current buffer occupancy $B_{occ}$
Input:   Traffic Generation Rate $T_{rate}$ gives as the
         number of packets generated by a node per unit
         time
Input:   The ordered list of predictability information of
         node's neighbours
Input:   $\delta_t$ the worst case expected time duration till the
         next contact
Define:  Available free space in the buffer $B_{free}$
Define:  Expected occupancy requirement $B_{req}$
Define:  List of nodes reachable in the descending order
         of the number of hops to reach $List_{best\_nodes}$

Begin
    Set $B_{req} \leftarrow min(B_{occ} + (T_{rate} + \delta_t), B_{capacity})$
    If $B_{occ} \leq ½ + B_{capacity}$ then
        Send the list of messages that can be
          accepted within the buffer based on
          $B_{free} \leftarrow (B_{capacity} - B_{req})$
        Set $B_{occ} \leftarrow B_{req}$
        Receive messages from the neighbour
        Set $B_{occ} \leftarrow B_{occ}$ + number of messages
          received
        Set $B_{free} \leftarrow B_{free}$ – number of messages
          received
        Set $List_{best\_nodes} \leftarrow$ nodes in the
          descending order of hops
    Else if $(B_{occ} \geq ½ + B_{capacity})$ and
      $(B_{occ} \leftarrow B_{req})$ then
        While $B_{occ} \leq B_{capacity}$ do
            Receive a message from the neighbour
            Accommodate this message in the buffer
            Set $B_{occ} \leftarrow B_{occ} + 1$
            Set $B_{free} \leftarrow B_{free} - 1$
            Set $List_{best\_nodes} \leftarrow$ nodes in the
              descending order of hops
end
```

Here we consider the traffic generation rate in order to accommodate the node's own generated messages in case it could not find any potential forwarders. Therefore when the occupancy of the buffer is above a "safety" threshold margin, the algorithm will not put any external messages in to the buffer in order to accommodate its self generated messages. In the second case the node checks if the occupancy of the buffer is above one half of the buffer size but below the "safety" threshold margin. In such case, since the receivingnode maintains a record of the number of hops that needs to be traversed towards the destination by each message, it will accept only the messages (which the node names as best nodes) that have the higher number of hops to reach their destination. By maintaining a safety margin and receiving messages according to buffer space availability of the node, each node will try its best, to keep itself from being congested.

Algorithm 1 and Algorithm 2 describe how each node limits its willingness to accept new messages whenever its buffer is occupied more than one half of its capacity. Each node chooses the best nodes it will potentially meet in the near future and shall advertise about these nodes to its current neighbors. Therefore the node which receives this information can refrain itself from overloading the intermediate node and thereby avoid congestion at that node. Even when nodes have larger space available in their buffer, they reserve a small portion of it for their own messages and then advertise the rest of it to other nodes.

Algorithm 1 describes how a receiving node limits its willingness to receive packets when it senses self congestion.\

Algorithm 2: Message Sending Logic – Sender. The algorithm will be executed by the intermediate node which receives the willingness information from a neighbour it had just met and shall prepare messages to be sent.

```
Input:  The received $B_{free}$ information from the neighbour
            just met. If there are two neighbouring nodes
            that want to send at the same time then choose
            one of them randomly.
Input:  The $List_{best\_nodes}$ just received from a neighbour
Begin
    Fetch buffered messages to be forwarded
        according to $B_{free}$ and $List_{best\_nodes}$ )
    When two or more messages have the same hop
        value, choose the node with the minimum
        predictability value
    While $List_{best\_nodes}$ is not empty do
        Forward the message to the neighbour
        Set $List_{best\_nodes} \leftarrow List_{best\_nodes} - 1$
end
```

Algorithm 2 describes how a sending node prepares its list of messages to be forwarded to a potential neighbor whom it had just met.

We incorporate our congestion aware adaptive approach into the adaptive routing protocol given in [1] and enhance it to make a congestion avoidance mechanism. We name it as the Congestion Aware Adaptive Routing Protocol (CAARP). In addition to having the properties of contact predictability and connectedness, and while exchanging this information with neighboring nodes, in CAARP nodes will also advertise their buffer space availability and their willingness to store, carry and forward messages for other nodes.

IV. SIMULATION SETUP

We use simulations to evaluate and compare the performance of the Adaptive routing protocol described in [1], the proposed CAARP protocol, the Epidemic routing protocol [4], the Prophet routing protocol [9] and the Hibop routing protocol [8] in the presence of congestion. We use the Jist/SWANS discrete event driven simulator [17] to model these protocols and to carry out simulations and collect statistics. For our simulation studies we consider the MOFO buffer management policy as described in [15]. With the assigned simulation set up parameters and considering each buffer management policy separately, we ran each of our simulation test scenarios for a time period of six hours and have collected packet delivery statistics. We use the Message Delivery Ratio (MDR) as the performance metrics defined below:

Message Delivery Ratio (MDR): We define message delivery performance as the system's ability to deliver messages of application data in the presence of intermittent connectivity. This is expressed as the ratio of the messages delivered to the totally generated messages of each type.

$$MDR = \frac{number\ of\ messages\ received\ at\ destination}{number\ of\ messages\ generated\ by\ source}$$

In all simulations, we follow a similar setup and traffic is generated as described in [1], and we implement connectivity modelling to model intermittent node connectivity of the network in the simulator [18], [19]. As in [1], at the beginning of the simulation, the network is assumed to be connected, that is, each device is assumed to have a path either multihop or one-hop to any other node. As the time progresses, connectivity between nodes becomes intermittent. We also assume that only the links between nodes disconnect and not the nodes themselves. We further assume that each network's full bandwidth is shared by all the nodes and there is non-negligible queuing delay in the originating or intermediate nodes. We also model two clusters of rescue nodes and a single cluster of head quarters nodes as there are different traffic generation requirements between different types of nodes as described in [19]. Each cluster is assumed to consist of four nodes and we assume that all nodes have a radio range of 200 meters and assume a single channel data rate of 2Mbps for intra-cluster and inter-cluster communications. We also assume that the time to live values of packets of different types are the same. A period of 30 minutes is used as the start up time so that the system can come to a stable state before any data collection is done, then data is collected for a simulation run time of six hours. In all our simulation work we consider a case where the contact and inter-contact durations vary in the time interval of 3 to 5 minutes since in typical rescue scenarios it has been observed that the rescue team members are expected to make regular contacts for every 3 to 5 minutes.

V. RESULTS AND DISCUSSION

In Fig. 3, we present MDR for the Adaptive [1], and the CAARP protocols for the cases of 50P50C and 90P50C. We can observe that the CAARP protocol outperforms the adaptive routing protocol especially when the buffer size is *100*. When the buffer sizes are smaller, the adaptive routing protocol performs better than the CAARP protocol.

Another observation which can be made is that when the buffer sizes are smaller, the MDR is very low for the CAARP protocol in both the cases of *50P50C* and *90P50C*. This is because the CAARP protocol allocates at least half of its buffer space to buffer its own traffic and only will then accommodate the messages that are being forwarded by other nodes. As a result of this, when the buffer sizes are smaller, the MDR becomes low. In order to further analyse the effect of the smaller buffer space of each node on the MDR, we have further varied the buffer sizes from *1* to *20* and have observed the performance of the Adaptive and CAARP protocols.

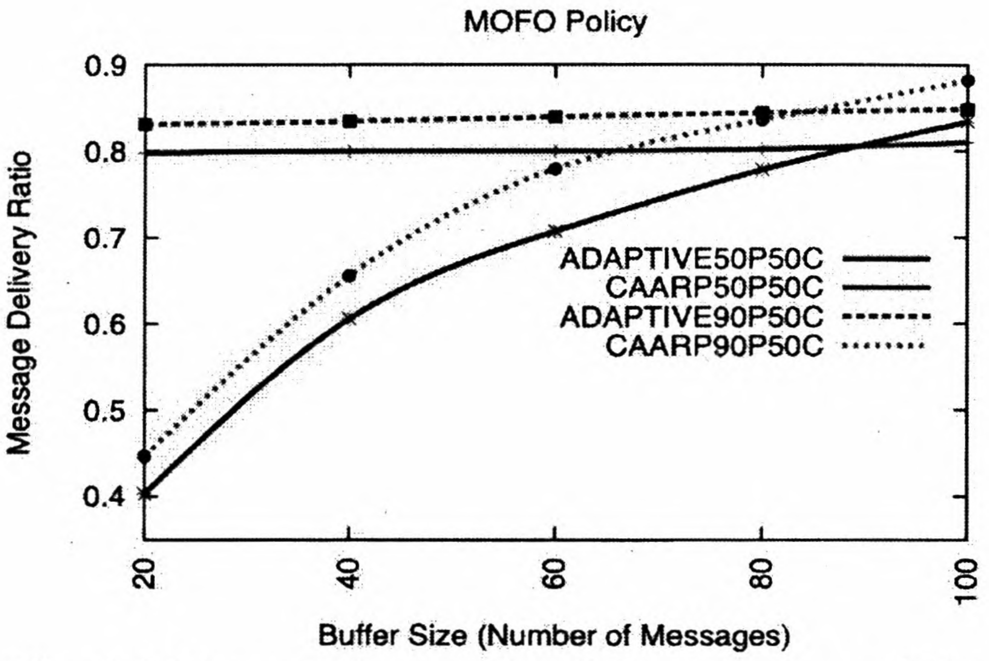


Fig. 3 Message delivery ratio of the Adaptive and the CAARP routing protocols.

In Fig. 4 we compare the performance of the Adaptive, CAARP, Epidemic, Prophet and Hibop routing protocols. A general observation that could be made is that the CAARP is very much dependent on the buffer size. It exhibits a rapid increase in the MDR as the buffer size increases, where as all other protocols show an almost flat behaviour in MDR with increasing buffer size.

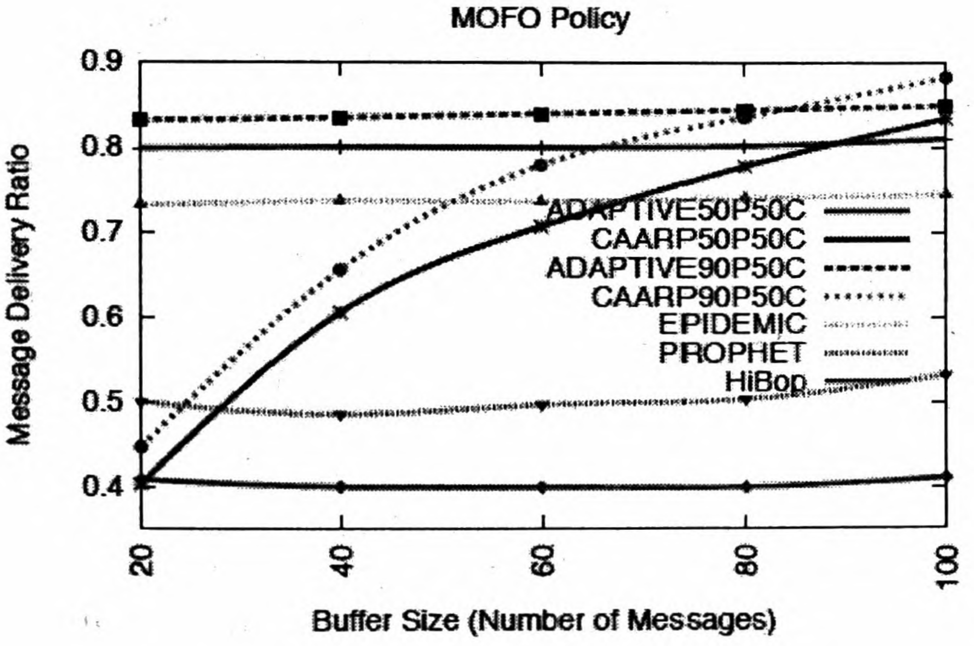


Fig. 4 Message delivery ratio of the Adaptive, CAARP, PRoPHET, Hibop and Epidemic routing protocols.

Our observations indicate that the CAARP achieves a higher MDR to that of Adaptive routing protocol when the buffer sizes are larger. When the buffer sizes are smaller, the adaptive routing protocol [1] shows a much higher MDR than the CAARP. As we have discussed in the above paragraphs this is due to the fact that the CAARP protocol allocates at least half of its buffer space for its own traffic which prevents the CAARP protocol in receiving packets from its neighbours. As the buffer size reaches to around *100*, the CAARP finds it comfortable to accommodate more packets from its neighbours and therefore shows a higher MDR. Towards investigating the energy expended on message exchanges we have considered the following energy consumption values as given in TABLE I for the Transmission, Receiving, Sleeping and Idle states [20] for a 2Mbps communicating radio. We have adapted the method described by Friedman *et al.*, [21] in measuring the energy consumed for each activity by monitoring the state changes in nodes for each activity in the Jist/SWANS simulator [17].

TABLE I
POWER CONSUMPTION IN THE RX, TX, IDLE AND SLEEP MODES

T_X	R_X	**Idle**	**Sleeping**
1400mW	1000mW	830mW	130mW

We have then estimated the best MDR achieved per unit of power expended for the Adaptive and the CAARP protocols for the cases of *50P50C* and *90P50C*. TABLE II shows the empirically estimated values per unit of energy expended to achieve the best MDR for these two routing protocols. From this table we can clearly observe that the CAARP protocol achieves the best MDR per unit of power expended.

TABLE III
AN ESTIMATION OF BEST MDR ACHIEVED AND BEST MDR PER UNIT OF POWER EXPENDED FOR THE CASES OF 50P50CAND 90P50C

Routing Policy	**Best MDR**	**Best MDR per unit of power expended**	**Energy Savings by CAARP**
CAARP (50P50C)	0.82120	5.91E-05	
Adaptive (50P50C)	0.80148	4.94E-05	19.59%
CAARP (90P50C)	0.89384	6.40E-05	
Adaptive (90P50C)	0.82612	5.10E-05	20.31%

Therefore it is evident that the higher message delivery ratio of the adaptive routing protocol is achieved at the expense of an increased consumption of energy. As indicated by many researchers, increased energy consumption can lead to node withdrawal from the opportunistic content distribution scenarios and can bring the network to a more disconnected state [20], [22], [23]. Lower power consumption and a higher delivery ratio with the assistanceof a larger buffer space is achieved by the CAARP protocol in an opportunistic networking environment.

VI. CONCLUSIONS

In this paper we have analysed the existing work on congestion avoidance in opportunistic networks and presented the CAARP routing protocol for opportunistic networks. Our congestion control mechanism uses the predicted connectivity information, the number of hops needed for a given message to be forwarded and the amount of buffer space available in a node in order to avoid congestion at popular nodes. We have incorporated this mechanism in to our adaptive routing protocol and through simulation studies we have evaluated the performance of the proposed enhanced congestion aware adaptive routing protocol, CAARP. Our results indicate that the CAARP shows a higher MDR than adaptive routing when the buffer sizes are relatively larger. When the consumed energy is also taken in to account, for any given buffer size and a message delivery ratio, we have found that the CAARP has an overall superior performance with lower energy utilization than adaptive routing protocol.

REFERENCES

[1]. T. Kathiravelu, N. Ranasinghe, and A. Pears, "Towards designing a routing protocol for opportunistic networks," in Proceedings of the International Conference on Advances in ICT for Emerging Regions (ICTer2010), Colombo, Sri Lanka, September 2010, pp. 56–61.

[2]. J. Lakkakorpi, M. Pitkänen, and J. Ott, "Adaptive routing in mobile opportunistic networks," in proceedings of the MSWiM'10, Bodrum, Turkey, October 2010, pp. 101–109.

[3]. C. E. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 1999, pp. 90–100.

[4]. A. Vahdad and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep., April 2000.

[5]. D. B. Johson, D. A. Maltz, and J. Broch, DSR: The dynamic source routing protocol for Multihop wireless ad-hoc networks. Addison-Wesley, 2001.

[6]. Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05), 2005.

[7]. M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay tolerant mobile networks," IEEE Transactions on Mobile Computing, July 2008, pp. 246–260.

[8]. C. Boldrini, M. Conti, I. Jacopini, and A. Passarella, "Hibop: A history based routing protocol for opportunistic networks," in Proceedings of the (WoWMoM 2007), 2007.

[9]. A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in Proceedings of the Fourth ACM International Symposium on Mobile AdHoc Networking and Computing (MOBIHOC 2003). ACM, June 2003.

[10]. T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." in Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking, August 2005, pp. 252–259.

[11]. H. A. Nguyen and S. Giordano, "Routing in Opportunistic Networks," International Journal of Ambient Computing and Intelligence, vol. 1, no. 3, 2009, pp. 19–38.

[12]. K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay tolerant mobile networks: Controlled flooding in sparse mobile networks," in Proceedings of Networking, Waterloo, Ontario, Canada, May 2005.

[13]. V. K. Sharma and S. S. Bhadauria, "Agent based congestion control performance in mobile ad-hoc network: a survey paper," (IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Wireless and Mobile Networks, vol. 197, pp. 324–333, 2011.

[14]. S. Jain, S. Kokate, P. Thakur, and S. Takalkar, "A study of congestion aware adaptive routing protocols in manet," Computer Engineering and Intelligent Systems, vol. 3, no. 4, pp. 63–74, 2012.

[15]. A. Lindgren and K. S. Phanse, "Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks," in Proceedings of the First International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE 2006), January 2006, pp. 1–10.

[16]. J. Lakkakorpi, M. Pitkänen, and J. Ott, "Using buffer space advertisements to avoid congestion in mobile opportunistic dtns," in Proceedings of the 9th IFIP TC 6 International Conference on Wired/Wireless Internet Communications, ser. WWIC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 386–397.

[17]. R. Barr, Z. J. Haas, and R. van Renesse, "Jist: An efficient approach to simulation using virtual machines," Software Practice & Experience, vol. 35, no. 6, pp. 539–576, May 2005.

[18]. T. Kathiravelu, A. Pears, and N. Ranasinghe, "Connectivity models: A new approach to modelling connectivity in opportunistic networks," in Proceedings of the 8th international information technology Conference IITC2006, Colombo, Sri Lanka, October 12-13 2006.

[19]. ——, "Evaluation of the impact of opportunistic networking on command and control system performance," in Proceedings of the Next Generation Wireless Networks (NGWS 2009), Melbourne, Australia, October 12-13 2009.

[20]. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," Wireless Networks, vol. 8, no. 5, pp. 481–494, Sep. 2002.

[21]. R. Friedman and A. Kogan, "Power aware management middleware for multiple radio interfaces," in Middleware '09: in Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware. New York, NY, USA: Springer-Verlag New York, Inc., 2009, pp. 1–20.