Protocol for Efficient Opportunistic Communication

Ranjana Pathak*†, Peizhao Hu†, Jadwiga Indulska*† and Marius Portmann*†
*The University of Queensland,
School of Information Technology and Electrical Engineering
†National ICT Australia (NICTA)
Email: {Firstname.Lastname}@nicta.com.au

Abstract—**In typical wireless networks, end-to-end routing is a usual way to deliver data packets from source nodes to destination nodes. In the case of link failures when no alternative route is found, the routing protocols will drop these packets. As a way to improve the packet delivery ratio, an integration of the store-carry-forward features with the traditional end-to-end communication has been already proposed. The existing solutions propose one-time only switching from one communication mode to another should the link failures occur. In this paper, we propose a hybrid protocol to support the dynamic switch between the two modes of communication should the link conditions changed. That is, the protocol utilises the ability to buffer packets when end-to-end routes are not possible, and leverages the end-to-end routes whenever they become available to ensure performance. We evaluate the proposed protocol using a set of comprehensive simulation scenarios to systematically demonstrate its significant improvement in packet delivery over one of the best representative routing protocol for end-to-end routing.**

I. INTRODUCTION

Due to various reasons (including mobility and interference) wireless networks are vulnerable to changes of link conditions and therefore can experience link failures. Typical routing protocols (e.g., AODV, OLSR) are designed to discover and maintain the route between a source and a destination. They are responsible for evaluating and acting upon the changes of link conditions. When link failures occur, these routing protocols often attempt to reroute (or repair the route) and drop the subsequent data packets if no alternative route exists.

Another way of communication in wireless networks is opportunistic networking which shares the communication concept with delay-tolerant networking. In these networks, data packets are delivered to neighbouring nodes on encounter between mobile devices and packets travel in this manner hop-by-hop until they reach the destination. The performance of opportunistic protocols is much lower than routing protocols which find an end-to-end route if such a path from the source to the destination exists.

There already exists research on combining the opportunistic and end-to-end protocols [1], [2], [3], [4]. These approaches tend to switch over to the opportunistic communication paradigm for the lifetime of the packet flow when the packets are dropped due to link failures. For example, SF-BATMAN [3] is an attempt to extend BATMAN (a reactive protocol similar to AODV) with the store-and-forward functionality. However, only a preliminary design is presented and the evaluations are very preliminary. In [1], Ott et al. proposed an approach to extend AODV to support DTN routing when path to the destination breaks and cannot be repaired. The switching from AODV to DTN is always at source nodes and the switching back from DTN to AODV is not supported. In contrast, we propose a truly hybrid protocol in which the packets that would be dropped due to route failure are delivered *opportunistically* through the network until they reach a node that is able to create an end-to-end path to the destination. Therefore the approach leverages the potential partial end-to-end routes that can be created in wireless networks. By doing so, this hybrid approach not only improves the packet delivery ratio compared to end-to-end routing protocols, but also shows efficiency improvement when compared to opportunistic protocols.

In our previous paper [5], we demonstrated the initial concept of the hybrid protocol by extending AODV[1]. We named the extended protocol — AODV-OPP. When a link failed and no alternative route existed, AODV would drop all subsequent packets. However, AODV-OPP buffers these packets and delivers them at a later time to all its neighbours. Whenever packets arrive at a node that has end-to-end route to destination, these packets will be sent using the route. The switching between mode of communications is managed dynamically. As the protocol forwards packets to all one-hop neighbours, overhead is certainly a concern. In this paper, we propose a new AODV-OPP+ protocol that balances the trade-off between delivery ratio and overhead and retains the ability to dynamically switch between communication modes if necessary. The main contributions of this paper are: (i) a metric to evaluate neighbours of a node and to identify the neighbours that have the best chance of forwarding the packets toward the destination; (ii) a low-overhead algorithm to support the dynamic switching between communication modes; (iii) extensive simulation evaluations (including synthetic) of the proposed hybrid protocol; and (iv) performance comparison of the proposed protocol with the AODV [6] protocol.

The remainder of the paper is organized as follows. Section II describes the overall design of the hybrid protocol — AODV-OPP+, including the new *reachability* metric and algorithm. This is followed by the evaluation results and discussions on the performance of the proposed protocol. Finally, we conclude the paper in Section IV.

---

[1] The idea can be easily applied to other routing protocols, such as OLSR.

978-1-4799-0537-9/13/$31.00 ©2013 IEEE

II. AODV-OPP+: METRIC AND ALGORITHM

In this section, we describe the core of AODV-OPP+, including its *reachability* metric and a new packet forwarding algorithm. Although we describe the approach based on AODV, most parts of the proposed concept are applicable to other routing protocols, such as OLSR.

A. Metric

In our previous approach (as described in [5]), AODV-OPP will buffer the packets being dropped by AODV and forward these packets to **ALL** one-hop neighbours. Sending buffered packets to all neighbours may lead to higher delivery probability, but also results in significantly higher overhead. In this paper, we propose a new protocol that has a much lower overhead but maintains a similar level of delivery probability. To achieve this goal, we propose the *reachability* metric, which measures the probability of a node having connection to a desired destination. With the reachability metric, the data packets are forwarded only when a neighbour node has higher reachability than the current node. This way the data packets are likely to travel towards the desired destination.

In opportunistic communications data packets are forwarded hop-by-hop until they reach the destination and the reachability metric is usually computed as the *direct contact time* between any two nodes. In our hybrid approach, a node can have communication with the desired destination either by direct encounter (a timer records the time between when detecting a new neighbour until the neighbour leaves) or via an end-to-end route (a timer records the time between a route for a destination is created until this route is deleted from the routing table). Therefore, each node will have two reachability metrics for a desired destination node, namely $R_{encounter}$ and R_{route}. The highest one of the two values will be used as the node's reachability to the destination, that is $max(R_{encounter}, R_{route})$.

The reachability R (either $R_{encounter}$ or R_{route}) of a node are computed separately as

$$R = (1 - \alpha) * R_{old} + \alpha * R_{measured} \tag{1}$$

where $R_{measured}$ is the respective probability in the last measurement window; R_{old} is the historical probability (initialized to zero when a node first bootups); and α is an adjustable parameter, which controls the weight between the history and new measurements. In the current implementation, we set the α to 50%. The investigation of optimal α value is left as future work.

To measure the reachability $R_{measured}$ of a node to other nodes in the network, we use the following equation.

$$R_{measured} = \frac{\sum T_{connection\_duration}}{T_{window}} \tag{2}$$

where $\sum T_{connection\_duration}$ is the sum of the duration (in time unit) the two nodes stay connected (with respect to having direct contact or having connection via a route) within a period of time T_{window}. T_{window} represents the measurement window and is a tunable parameter depending on the node mobility in a particular scenario. When the network is relatively mobile, then T_{window} needs to be relatively small to cope with the rapid changes in the topology. We plan to investigate on T_{window} selection in the future.

Our goal is to maximise delivery probability and in this paper we describe a threshold-based approach to determine the best set of neighbours to forward the buffered data packets. The idea is to specify a reachability threshold in the packets and broadcast the packets to its one-hop neighbours. Upon receiving the packets, each neighbour will check its own reachability to the desired destination. Neighbours will only contribute in the packet forwarding when their reachability is higher than the threshold specified in the packets.

B. Algorithm

Having described the *reachability* metric, in this section we explain how we use this metric in neighbour selection for packet forwarding.

Figure 1 shows an example of the AODV-OPP+ broadcast-based forwarding mechanism. To explain the mechanism, we first assume that a link from node 1 to a destination failed and an alternative route could not be found resulting in packet drop. As shown in Figure 1(a), node 1 will buffer subsequent packets and broadcast them to its one-hop neighbours (nodes 2, 3, 4) attaching to the packets its reachability (R_1) to the destination. Node 1 marks itself as in the "broadcast" state and waits to overhear rebroadcast from its neighbours. Upon receiving the broadcast packets, as shown in Figure 1(b), nodes 2, 3, 4 will receive the packets if themselves are the destination or will forward the packets if they have a route to the destination. Otherwise, they will check their respective reachability to the destination (R_2, R_3 and R_4). Nodes with greater reachability than R_1 will buffer the packets in their *BufferQueue* and rebroadcast these packets with its own reachability. We assume nodes 3 and 4 satisfy the conditions and broadcast the packets. Node 2 will simply ignore the packets. Because nodes can only participate in the packet forwarding if and only if they have greater reachability to the destination, the buffered packets are likely heading towards the destination. When node 1 overhears the rebroadcast packets from nodes 3 and 4, it adds nodes 3 and 4 into the forwarder list and reduces the retry count accordingly. The retry count is introduced to limit the number of copies allowed to be disseminated from a particular node. In the same way, the buffered packets will be forwarded from node 6 to node 8 until they arrive at the destination or their TTLs expire, as shown in Figure 1(c). In this round, nodes 3 and 4 were the senders of the packets. They reduce their respective retry count upon receiving the rebroadcast packets from nodes 5, 6 and 7. In AODV-OPP+, we prefer end-to-end routes for their performance. Therefore, if any node in the forwarding path has end-to-end route to the destination, packets will be delivered using the route.

III. EVALUATION

In this section, we show the evaluation results for the aforementioned tests and analyse the performance differences

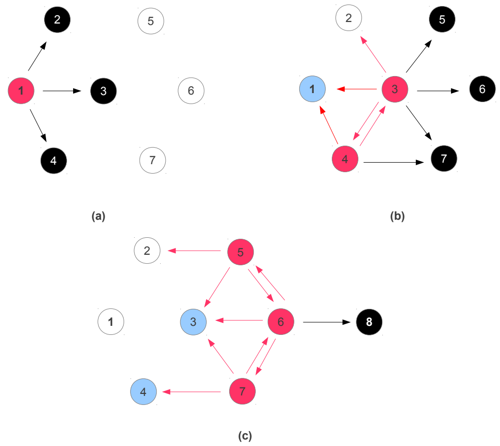


Fig. 1. Example of AODV-OPP+ broadcast-based forwarding mechanism.

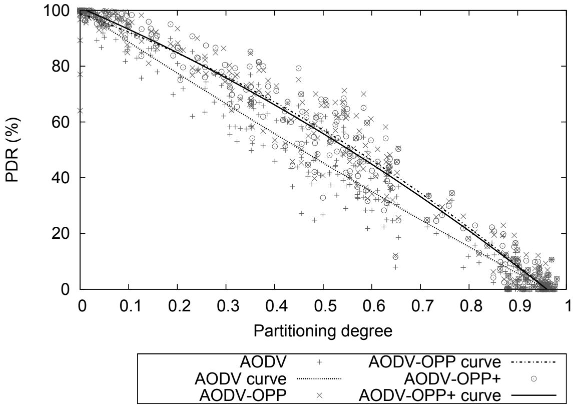


Fig. 2. Performance of AODV-OPP+ against AODV and AODV-OPP.

of AODV-OPP and AODV-OPP+ against AODV.

A. Evaluation scenario

The synthetic tests are designed to evaluate the hybrid protocol using a set of random scenarios that represent all possible network characteristics (density or node connectivity). We use a mobility model generator — BonnMotion [7] to generate these random scenarios. All generated scenarios conform to the random way-point model. In addition to the mobility model generation, BonnMotion also supports scenario analysis. It computes different characteristics of a given scenario; for example, the average node degree (*to how many other nodes is one node connected*) and the partitioning degree (*how unlikely is it that two randomly chosen nodes are connected at any point in time*) [7]. For the synthetic simulations, we use the partitioning degree (a value normalised to 0-1) to characterise the network scenarios from dense to sparse. We divide the partitioning degree (PD) into three equal ranges (PD low: [0-0.33); PD medium: [0.33-0.66); PD high: [0.66-1]). To achieve statistical confidence in our results, we generate 100 different scenarios for each partitioning degree range. That is, in total we need to generate 300 scenarios for the entire PD range. To generate these 300 scenarios, we first use BonnMotion to generate 2000 random scenarios with different area sizes. Then we randomly select 100 scenarios for each partitioning degree range. These 300 scenarios are uniformly distributed across the whole range of partitioning degree values. We argue that this set of randomly generated scenarios should be representative for most of the application scenarios (including corner cases). It should be noted that we have fewer samples between PD value of 0.65-0.85. This means we have not as much scenarios for this PD range as the other ranges. However, the whole point of systematic evaluation is that we investigate the performance of each protocol using randomly selected scenarios. Therefore, we do not want to artificially change the set of scenarios for the evaluation. By evaluating our proposed protocol against these randomly selected scenarios, we should be able to analyse how the protocol performs under different characteristics of the network and the evaluation results should be comprehensive.

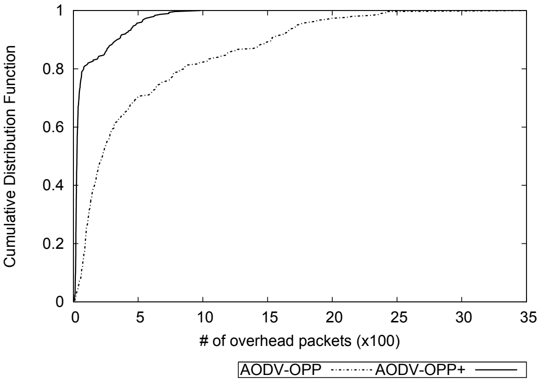
In all our synthetic tests, we use 50 mobile wireless nodes. Each of these 50 nodes are allowed to form connections with any one other node in the network. These connections will be formed randomly at different time during the simulation. For each of the 300 scenarios, we run the simulation 10 times and compute the average.

B. Performance of AODV-OPP+: first glance

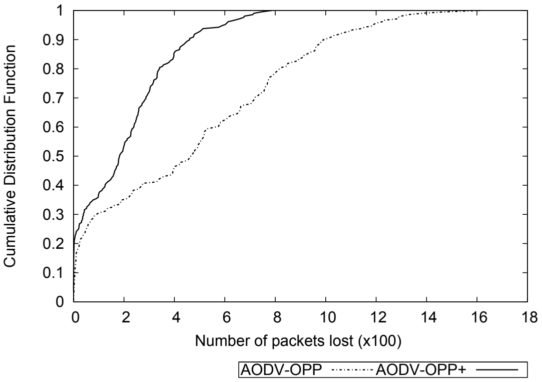
Before the systematic evaluations, we conduct the same set of validation tests to verify the basic operations of AODV-OPP+, as described in [5]. The results from the validation tests confirm that AODV-OPP+ is performing as expected.

To study the performance of AODV-OPP+, we first conduct a set of simulations to compare AODV-OPP+ against AODV and our previous proposal AODV-OPP. By using the 300 scenarios (varying in PD), we compute the average PDR (over 10 simulations for each PD value) of the respective protocols. As highlighted by the fitted curves (using second degree polynomial, hereafter we label them as *curve* in the figures) in Fig. 2, both AODV-OPP and AODV-OPP+ outperform the original AODV across all different network densities. As expected, these protocols achieve lower PDR when the network become sparse, since they rely on end-to-end routes. We also noted that AODV-OPP, which broadcasts the buffered packets to all one-hop neighbours, achieves slightly high PDR in some cases, as compared to AODV-OPP+. However, the overhead AODV-OPP generates is significantly higher when the network load increases, as discussed later.

As the mobile nodes need to forward those buffered packets to their neighbours, overhead is one of the concerns in the proposed idea. In the previous approach, AODV-OPP broadcasts the buffered packets to all one-hop neighbours, which can result in significant increase in overhead. AODV-OPP+ is a new protocol that is proposed to reduce the overhead by selectively disseminating buffered packets to neighbours that are more likely to have connections (or be part of a route connecting) to the desired destination. In this paper, we define overhead as the number of additional packets forwarded in the network for every packet successfully delivered to the



(a) CDF analysis of Overhead



(b) CDF analysis of packet lost

Fig. 3. Overhead and packet lost analysis.

destination. Therefore, we calculate the overhead O as

$$O = \frac{N_{forwarded}}{N_{received}} \qquad (3)$$

where $N_{forwarded}$ is the number of additional copies of the buffered packets forwarded in the network; $N_{received}$ is the number of buffered packets received at the destination.

As shown in Fig. 3(a), the CDF graph for overhead clearly shows that AODV-OPP+ generates much less overhead than AODV-OPP. For around 80% of the cases, AODV-OPP+ generates less than 40 additional packets (with the worst case of around 800 packets). In contrast, AODV-OPP needs up to 1000 additional packets for 80% of the times (with the worst case of more than 3000 packets). In addition to the overhead, Fig.3(b) shows the CDF graph for the number of packet lost (for the whole network) due to TTL timeout or lost at the end of the simulations. These lost packets remain in the buffer; that is, they are taking up the resources at the nodes. We argue that the fewer of these lost packets, the better is resource usage. As shown in the figure, AODV-OPP+ has significantly smaller number of packet lost, with less than 450 packets in 90% of the cases and AODV-OPP will have up to 1000 packets.

C. Benefit of overhead reduction

Typically, reduction in overhead means there will be more capacity for the actual data packets. To verify that more data packet can be injected into the network due to the reduction in overhead, we conduct additional experiments to vary the number of connection per node to create scenarios with increase in network load. For these experiments, we randomly pick a scenario (with partitioning degree of 0.34) from the 300 scenarios. We increase the number of connections a node is allowed to have with other nodes in the network, from 5 to 25. Fig. 4 shows that both protocols achieve similar PDR when the network load is relatively low. However, when the number of connections increases to around 13, we see the performance difference between the two approaches. When the maximum number of connections allowed is at 25, AODV-OPP+ is able to outperform AODV-OPP by around 20% in PDR. This confirms the reduction in overhead can ultimately increase the PDR gain for normal data packets when network load is high.

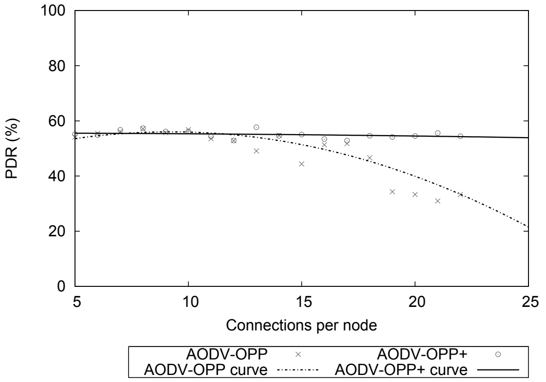


Fig. 4. Performance when increase in network load.

IV. CONCLUSION

In this paper, we proposed a new hybrid protocol — AODV-OPP+, which incorporates (i) a metric that ranks neighbours of a node by their probability to have connection to the destination, and (ii) a forwarding algorithm that provides a number of advanced features to improve on PDR and at the same time reduce the number of overhead packets and packet loss. Through a number of comprehensive simulations using synthetic mobility traces, we demonstrate the superior performance of AODV-OPP+.

REFERENCES

[1] J. Ott, D. Kutscher, and C. Dwertmann, “Integrating dtn and manet routing,” in *Proc. of ACM SIGCOMM workshop CHANTS*, 2006.

[2] J. Lakkakorpi, M. Pitkänen, and J. Ott, “Adaptive routing in mobile opportunistic networks,” in *Proc. of MSWiM2010*, Bodrum, Turkey, October 2010, pp. 101–109.

[3] L. Delosiéres and S. Nadjm-Tehrani, “Batman store-and-forward: the best of the two worlds,” in *Proc. of PerNEM 2012 (PerCom workshop)*, Lugano, Switzerland, March 2012, pp. 727–733.

[4] J. Whitbeck and V. Conan, “Hymad: Hybrid dtn-manet routing for dense and highly dynamic wireless networks,” *Comput. Commun.*, vol. 33, no. 13, pp. 1483–1492, Aug. 2010.

[5] R. Pathak, P. Hu, J. Indulska, M. Portmann, and W. Tan, “Towards efficient opportunistic communications: a hybrid approach,” in *Proc. of PerMoby 2013 (PerCom workshop)*, San Diego, CA, USA, March 2013.

[6] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (aodv) routing,” IETF. RFC 3561, July 2003.

[7] “Bonnmotion,” http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/.