Towards efficient opportunistic communications: a hybrid approach

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

***Abstract*—Wireless mesh networks are well-recognised by their self-organising properties. End-to-end routing protocols are primarily responsible for achieving these advanced features. However, wireless link failures can cause a route to be invalidated and subsequently removed from the routing table in nodes along the path to destination. Once the route is not available, packets addressed for that destination will be dropped. To improve the packet delivery ratio of those end-to-end protocols, we propose a hybrid approach that integrates features of opportunistic protocols into traditional end-to-end routing protocols in mesh networks. The idea is to buffer packets for which there is no path to the destination and attempt to deliver these buffered packets when an alternative route is found or to pass them to neighbours who might eventually be able to establish a route to the destination. To demonstrate the concept, we present the AODV-OPP — an extension of the AODV protocol that uses opportunistic communication. AODV-OPP always prefers end-to-end route before attempting to send opportunistically to neighbours. Based on a number of systematic simulation scenarios, we observe that AODV-OPP consistently outperforms the original AODV, with a PDR gain greater than 8% most times and up to 45%.**

I. Introduction

As a result of the rapid growth in mobile device market, we see a significant growth in adoption of various wireless networking technologies. Among many promising wireless technologies, wireless mesh networks have been recommended as a complementary technology for offloading the ever increasing data traffic from cellular networks. In addition, around the world, wireless mesh networks have been deployed by the public safety sector as a way to establish essential communications for law enforcement personnel [1] and to provide city video surveillance [2].

Wireless mesh networks are recognised for their self-organising features - they respond to dynamic changes in network topology (due to device mobility or failure) and to dynamic changes in the channel quality. These networks are typically bundled with end-to-end routing protocols that are suitable for achieving those self-organising features. However, link failures are common in such networks due to mobility of wireless nodes and the ever increasing interference in dense networks. Link failures typically cause routing path to destination to be removed from the routing table, and if a new path cannot be established packets are dropped. Another well known approach to communication in dynamic wireless networks is opportunistic communication in which 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 end-to-end routing protocols if the end-to-end path from the source to the destination exists.

In this paper, we explore a hybrid approach that integrates the *store-and-forward* functionalities of opportunistic communication with the traditional end-to-end routing protocols as a means to improve the packet delivery ratio in highly mobile scenarios. In the literature, there exist solutions that attempt to combine the opportunistic and end-to-end protocols [3], [4], [5], [6], [7]. These solutions 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. In contrast, we propose a 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 it 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 the mobile network. By doing so, the hybrid approach not only improves the packet delivery ratio of end-to-end routing protocols, but also improves the efficiency of opportunistic protocols.

The proposed hybrid approach should be applicable to most end-to-end routing protocols (e.g., AODV, OLSR) due to the common similarities in most of these protocols. As an example, we present AODV-OPP, which is an extension of the AODV routing protocol with support for opportunistic communications. We evaluate the performance of AODV-OPP through a set of systematic experiments in NS2 simulations. These experiments include (i) four validation tests, which demonstrate AODV-OPP behaves according to the design specifications; (ii) synthesis trace tests, which show AODV-OPP outperforms the original AODV across different network densities and connectivities; and (iii) realistic trace tests, which validate the observations we see from the controlled experiments and show that AODV-OPP consistently outperforms AODV. In addition, AODV-OPP does not introduce significant overhead in the case when no packets are dropped due to no route to destination.

The remainder of the paper is organized as follows. Section II describes the general design principles of the hybrid approach. In Section III, we present AODV-OPP as an example of the proposed approach. The systematic evaluation of AODV-OPP

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

is presented in Section IV. A critical literature review of related work is presented in Section V. Finally, we conclude the paper and discuss future work in Section VI.

II. Towards efficient opportunistic communications

In wireless networks, link failures due to mobility of wireless nodes or interference in dense networks often lead to packet drop because of lack of route to the destination. Typically, end-to-end routing protocols provide mechanisms to repair the routing path should the route to the destination failed. These mechanisms try to find alternative routes to the destination, and packets will be dropped if alternative routes are not found within certain time windows.

To improve the packet delivery ratio, we propose to integrate the *store-and-forward* functionalities of opportunistic communications with the end-to-end routing protocols. Analysing behaviour of end-to-end routing protocols (e.g., AODV, OLSR), we found similarities in the way they detect and handle link failures. They detect link failures either by the loss of periodic *Hello* messages or a mechanism called *Link-layer feedbacks*. When link failures occur and the protocols do not find alternative routes to the destination then the packets are dropped. In our solution we prevent these packets from being dropped, but do not intervene when an end-to-end route can be found as end-to-end provides much better chance of delivery.

When a packet is to be dropped due to the lack of route to the destination, the router will first check whether there is any one-hop neighbour. As shown in Fig. 1(a), if there are one-hop neighbours, a copy of the dropped packet will be sent to each neighbour and the $copy\_count$ is decreased. The use of $copy\_count$ is to provide a controlled flooding scheme, which will limit the number of packets to be sent to the network (therefore, will minimise the overhead). If there are no neighbours, the packet is stored in the queue with the remaining $copy\_count$ and $time\_TTL$ (i.e., time-to-live in time unit for the packet in the queue). The router then continues with the normal routing operations.

Another event that triggers the delivery of the buffered packets, as shown in Fig. 1(b), is when a node/router detects a new neighbour. When a router detects a new one-hop neighbour, it checks whether there is any packet in the queue. If the queue is empty, then the router continues with the normal routing processes. On the other hand if the queue is not empty, the router first checks whether an end-to-end route exists. In other words end-to-end routes are always preferred if they exist for each packet. When a packet is sent to the destination via an end-to-end route (or the one-hop neighbour is the destination), then this packet will be removed from the buffer queue. In the case when no end-to-end route exists, the packet is sent to the new neighbour and the associated $copy\_count$ is decreased. The same process is applied to every packet in the queue.

An end-to-end route may be created as a result of a node more than one-hop away creating the routing path to a given destination. Therefore, we introduce the third event trigger (as shown in Fig.1(c)) to try to send buffered packets using any new route. It is similar to the processes described in Fig.1(b). Except that if there is no route for a packet, then the next packet will be checked.

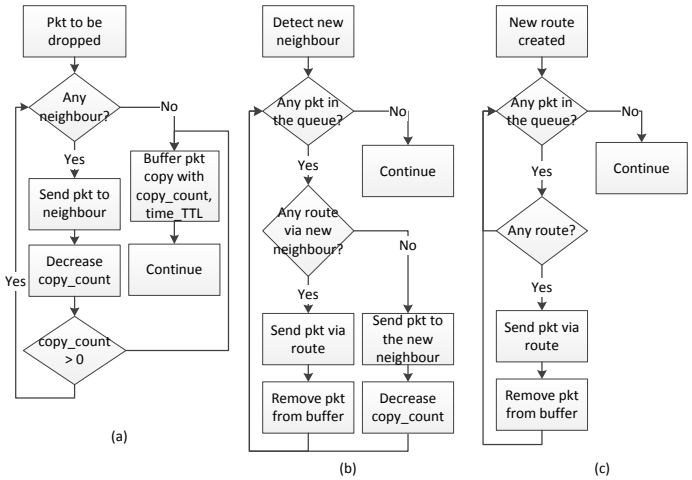


Fig. 1. Processes for handling packet drops.

These packet delivery operations can be applied to most of end-to-end routing protocols. As an example, in the next section we describe how these operations can be implemented as an extension of the AODV routing protocol.

III. Case study: AODV-OPP

In this section, we firstly explain the basics of AODV, and then describe an example of the hybrid approach toward efficient opportunistic communications — AODV-OPP.

A. AODV: the basic

AODV [8] is a reactive routing protocol designed for the Mobile ad-hoc networks (MANETs). AODV uses three message types to perform the functionalities of on-demand route discovery and route maintenance.

A *Route Request* (RREQ) is broadcast from the source node when it has a message to be sent to a given destination and the routing table does not have a path to the destination. Upon receiving the RREQ, all one-hop neighbours of the source node will create a reverse route to the source node and forward the RREQ further. When a RREQ reaches the destination (or a node that has a route to the destination), the respective node will generate *Route Reply* (RREP) as the response that is sent back to the source node along the reverse path. After receiving the RREP, the route that was previously created by the RREQ will becomes active. RREQs that exceed their lifetime are discarded silently. Each routing entry has a lifetime, which will be updated every time a packet passes through the route. When the lifetime of a route expires, the route is invalidated and subsequently removed from the routing table.

To detect link failures, AODV uses either the periodic *Hello* messages (i.e., from missing hello messages) or the link-layer feedback detection. After detecting link failure, a local repair mechanism can be invoked. If an alternative route can not be created within a time window, *Route Error* (RERR) messages

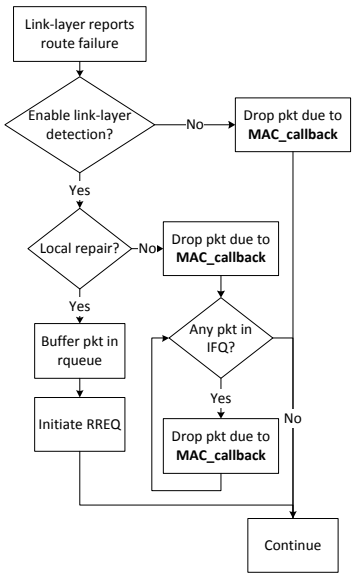


Fig. 2. The link-layer detection in AODV.

are sent along the affected path to invalidate the routing entry within all affected nodes.

B. The making of AODV-OPP

There are four crucial components in the AODV-OPP protocol:

1) Detecting route and link failures: AODV supports two ways to detect the failure of a route or a link. Once a packet drop is detected, the methods of handling packet drop as described in Section II are applied. The most common approach is by periodic exchange of *Hello* messages between neighbours. Upon receiving these heartbeat messages, a node will refresh the time-to-live timer of the respective neighbours. Any neighbour that does not refresh its timer will be removed from the node's neighbour list. Subsequently, an AODV RERR message will be propagated to all the nodes along the affected routing path. Any route that relies on the removed node will be invalidated. As the result, packets that depend on these routes will be dropped. At this point, the method of Fig. 1(a) is applied to handle the packets rather than dropping them.

Link-layer detection is another approach to detect link failure. As shown in Fig. 2, for each link a node has with its neighbours, the node registers a callback function. When the link-layer reports route failure, it will try to perform the *local repair* mechanism if the link-layer detection feature is enabled. Otherwise, the packet is dropped. Similarly, if local repair is not supported, all packets within the interface priority queue (IFQ) will be dropped as a result. During the local repair, the packet will be buffered in the *rqueue* (designed to store packets that are dropped due to route failures), and a route discovery is initiated.

In the cases when a packet is dropped due to the *MAC_callback*, the same method as for the Helloing approach, shown in Fig. 1(a), is applied.

2) Buffering dropped packets: As shown in Fig. 1(a), the copy_count of a dropped packet will decrease by one after sending to a neighbour. After this packet has been sent to all neighbours, if the copy_count is still greater than zero (i.e., this dropped packet is still allowed to be sent), this packet will be put into the *BufferQueue* (a link list) in order to be sent when a new neighbour is detected (as shown in Fig.1(b)). In addition to storing the packet ID and other forwarding information, each packet in the BufferQueue also keeps information about the *copy_count* and the *time_TTL* (time-to-live). Packets with expired time_TTL are purged from the BufferQueue.

3) Engaging route discovery: Since AODV is a reactive routing protocol, it will not initiate route discovery unless there is a packet to be sent from a source node (or a packet being buffered in the intermediate node due to no route). As shown in Fig. 1(b), when a new neighbour is detected an end-to-end route that could be created via the new neighbour is preferred because end-to-end routes provide better guarantee of delivery compared to hop-by-hop delivery. Therefore, an approach is needed to initiate the route discovery processes when a new neighbour is detected.

In our approach, we use the first packet of a destination in the BufferQueue to trigger the processes of sending a RREQ message for that destination. After a RREQ for the packet's destination is sent, this packet is then sent to the new neighbour. Subsequent packets will also be sent in a hop-by-hop fashion until an end-to-end route is found. By doing so, we maximize the number of packets to be sent to nodes that are closer to the destination, rather than waiting for the RREQ timeout of 10s by default. In addition, since these packets are already in the neighbouring node, they will get a higher priority to be sent should a route be created. If an end-to-end route is possible, then the remaining packets in the BufferQueue will be sent using the route as shown in Fig. 1(b).

In large-scale networks, there could be many traffic flows sending packets to the same destination. When this happens, there will be a significant number of RREQ messages created for the same route. To reduce the overhead, a list of RREQ messages sent to the same destination was introduced.

4) Avoiding routing loop: Sequence numbers are used to avoid routing loops in AODV. As AODV-OPP incorporates the store-and-forward feature of opportunistic protocols we introduced another component — *PacketHistory*, to keep trace of all packet IDs that a node seen over the last time window. Currently it is a linked list but will be converted to more efficient data structures (hash table) in the future to improve performance.

IV. Evaluation

We developed AODV-OPP as an extension of the NS2 implementation of the AODV protocol. To validate the correctness and to evaluate the AODV-OPP performance we used three types of NS2 simulation scenarios that are described in this section. These simulations evaluate the AODV-OPP performance from different aspects, including overhead, functionalities and improvement in packet delivery ratio.

TABLE I
SIMULATION PARAMETERS

Copy_count	10 copies
Packet TTL	400 s
Simulation Time	500 s
Traffic start time	10 s
Traffic end time	120 s
Data rate	4 packets per second
Tx range	250 m
IFQ length	50 pkts
BufferQueue size	unlimited

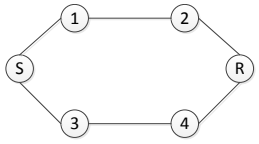


Fig. 3. Validation tests topologies

All simulations use the parameters listed in Table I, unless they are discussed in the respective simulation scenarios.

A. Validation tests

These tests aim to verify the basic operations of AODV-OPP. For example, the protocol should not introduce additional overhead when channel quality is good and no packet is dropped. For these validation tests, we use the six nodes topology as shown in Fig 3, which allows us to define five different test scenarios. Each scenario aims to test one specific aspect of the proposed protocol. For each scenario, we run the simulation 100 times and compute the average.

Case 1, all nodes are static: These are baseline tests, which aim to verify that AODV-OPP does not introduce unnecessary overhead when normal AODV operations should be used for packet delivery. In this scenario, the sender (Node S) tries to send traffic to the receiver (Node R).

Case 2, link breaks causing rerouting: In this scenario, we first verify that the traffic from the sender (Node S) to the receiver (Node R) traverses the route S-3-4-R. Then we move Node 4 out of range, which causes AODV to reroute the traffic through alternative path S-1-2-R. Except Node 4 other nodes are static, therefore, AODV should be able to repair the route quickly.

Case 3, no alternative route causing packet drop: This scenario emulates the situation when Node 1 moves out of range from the route S-1-2-R at around 50 s after node 4 moves out of range from the route S-3-4-R. These will cause AODV to reroute via the alternative route, but the second link break will cause packet drops. As a result, the delay-tolerant mechanisms in the AODV-OPP should buffer a copy of these packets for delivery at a later time. The goal of this scenario is to confirm AODV-OPP is able to correctly buffer all the packets being dropped.

Case 4, route can be re-established: This scenario tests AODV's ability to re-establish the route when a node on the routing path moves within range again, and all the buffered packets are delivered successfully through the new route. In this scenario, we move Node 4 back to its original position after 50 seconds. Due to Case 3, there are packets buffered at Nodes S and 3. A new route established through S-3-4-R will cause packets to be delivered via this route, as shown in Fig 1(b).

TABLE II
ANALYSIS OF DROPPED PACKETS

	Buffered	Received	Lost
Case 1	0	0	0
Case 2	1	0	1
Case 3	73	0	73
Case 4	73	49	25

Fig. 4 shows the averaged results of all the simulation tests. In addition, Table II details (in one test run) the number of buffered packets (being dropped due to no route) that are *buffered* in the BufferQueue, *received* at the destination, and *lost* due to various reasons (e.g., IFQ being full). As we have expected, we see that AODV-OPP achieves exactly the same PDR as the original AODV and no packets have been buffered when nodes are static (as in Case 1). This means AODV-OPP does not generate additional overhead when AODV is capable to handle the traffic.

In the second case, when node 4 moves out of range, AODV will reroute the traffic flow from S-3-4-R to S-1-2-R. We observe that most of the traffic goes through this alternative route. However we also notice that one packet was at node 4 when the node moved out of range. As a result, this packet is dropped and buffered on node 4 and lost due to TTL expiry.

In Case 3, when nodes 1 and 4 move away, there are no routes to the destination R. We observe around the same PDR for both protocols, because no alternative route exists after nodes 1 and 4 move out of range. In AODV-OPP, there are 73 packets that have been buffered in the BufferQueue in nodes S and 3.

In Case 4, AODV achieves the same PDR as in Case 3, this is because the simulation traffic was stoped (at 120 s as shown in Table I) when nodes 1 and 4 were out of range, and there is no new packet generated when node 4 moves back to its original position. In addition, we notice a significant PDR gain of around 12% achieved by AODV-OPP. This demonstrates the store-and-forward mechanism works well in AODV-OPP. Although 49 packets arrived at the destination among the 73 packets that are dropped and buffered, as shown in Table II, 25 packets are lost. When we investigate the traces, we uncover that there are 24 packets lost due to IFQ being full and one packet lost due to expired TTL. The reason for the loss due to IFQ is because of our small IFQ length (allowing only 50 packets). A larger IFQ can reduce the packet loss for this cause, but a very large IFQ is unrealistic. We argue that for a protocol to be practical, it should work under the practical settings.

B. Synthesis traces

In this section, we aim to evaluate our hybrid protocol using scenarios with different network characteristics (e.g., node connectivities). We use a mobility model generator — BonnMotion [9] to generate these diverse scenarios. All generated scenarios

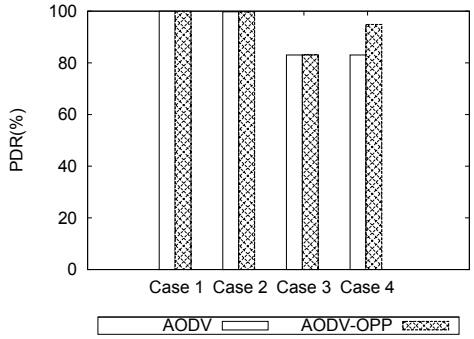
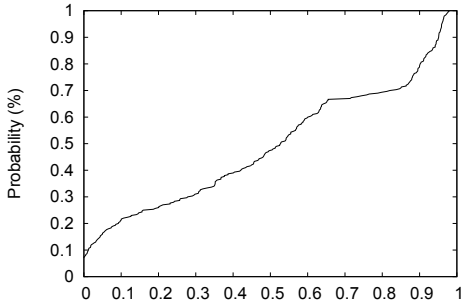


Fig. 4. Performance comparison of the validation tests. (increase font size)

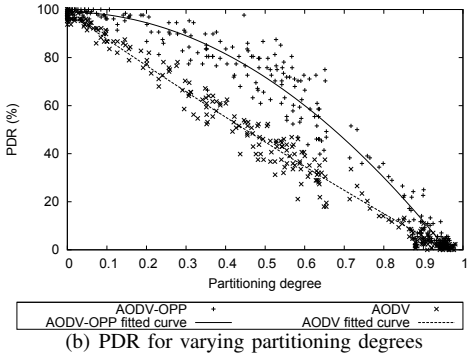
conform to the random way-point model. We argue that a large set of these randomly generated scenarios should be representative for most of the application scenarios (including corner cases). In addition to the mobility model generation, BonnMotion also supports scenario analysis. It computes different 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 likely is it that two randomly chosen nodes are connected at any point in time*) [9]. For the synthesis 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: 0.34-0.66; PD high: 0.67-1). To achieve statistical confidence for our results, we run 100 simulations for each partitioning degree range. To generate these 300 scenarios, we use BonnMotion to generate 2000 scenarios with different area sizes. Then we randomly select 100 scenarios for each partitioning degree range. From Fig. 5(a), we see that these 300 scenarios are relatively uniform distributed across the whole range of partitioning degree values. In all our simulations, we use 50 mobile wireless nodes and allow a maximum of 10 concurrent traffic flows between any randomly selected sender-receiver pairs.

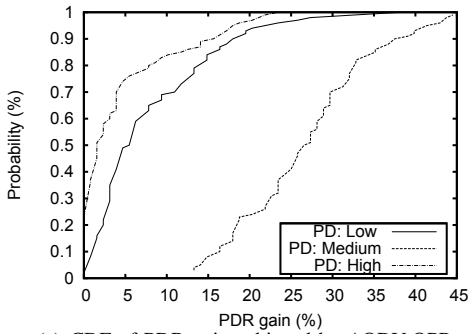
Fig. 5(b) shows all the PDR of AODV and AODV-OPP for the 300 scenarios (in points) and outlines the relationship between the partitioning degree and PDR (in fitted curves). The first observation is that both protocols achieve lower PDR as the partitioning degree increases. This is what is expected as the network becomes sparse. Another observation is that AODV-OPP outperforms AODV in most cases, and the PDR gains in the medium PD are significantly higher than the other two ranges. The cumulative distribution function (CDF) of PDR gains achieved by AODV-OPP for each partitioning degree range, as shown in Fig. 5(c), provides the same observation. For example, for achieving PDR gain greater than 20%, there is about 80% of chance in the medium PD range (but only around 4% and 10% in the high PD and low PD ranges). In addition, it shows that AODV-OPP outperforms AODV over all 300 scenarios, with a maximum improvement of 45% in medium PD cases (around 13% improvement even in the worst case). Surprisingly, AODV-OPP is able to outperform the original



(a) Scenario distribution



(b) PDR for varying partitioning degrees



(c) CDF of PDR gain achieved by AODV-OPP

Fig. 5. Performance comparison by synthesis traces.

AODV over 5% in PDR with about 50% chance in low PD cases and about 25% chance in high PD cases.

C. Realistic traces

The tool BonnMotion allows to generate synthesised traces that conform to a particular characteristic of the scenario (e.g., different densities or node connectivities), and hence allows carrying out systematic tests for the proposed hybrid protocol.

In this section, we conduct a performance evaluation for AODV-OPP using realistic mobility patterns gathered by the GPS devices mounted on the San Francisco city cabs [10]. The use of these realistic traces serves as a means to validate observations from previous tests and to demonstrate how AODV-OPP performs in real life scenarios.

We convert the GPS traces into the NS2 simulation and use these traces as the node mobility model. As for the traffic model, we generate 500 different traffic models (e.g., each with different sender-receiver pairs and connections are formed at

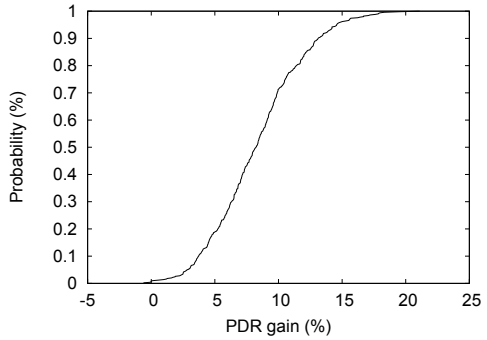


Fig. 6. The CDF of PDR gain for AODV-OPP over AODV

random time). The simulations involve 116 mobile wireless nodes.

Fig. 6 shows CDF of the PDR gain achieved by AODV-OPP over the original AODV. As shown in the figure, more than 99% of the 500 different scenarios, AODV-OPP outperforms the original AODV. The overall average gain in PDR is around 8%. There is around 70% chance that AODV-OPP can achieve less than and equal to 10% of PDR gain. We also see very small cases (less than 0.8%) where AODV performs better than AODV-OPP. The negative PDR gain may be caused by the interference introduced when sending buffered packet hop-by-hop in AODV-OPP. These additional transmissions might compete with the data sent by AODV.

V. Related Work

In the literature, there are attempts to merge traditional end-to-end routing protocols with opportunistic communications.

SF-BATMAN [6] is the most recent attempt to extend BATMAN (a reactive protocol similar to AODV) with the store-and-forward functionality. It has a similar goal to our proposed solution. However, the paper did not discuss solutions for problems introduced by the extension (e.g., routing loop), which can potentially affect the protocol performance. From the published simulation results, we see that a consistent PDR gain of 0.1% is achieved by this extension. This is a relatively low PDR gain compared to the overhead the solution introduces. Furthermore, the proposed protocol is only evaluated in a single and simple simulation scenario with only limited number of measurement samples. Therefore, it is difficult to see whether the presented improvement will hold for different scenarios. HYMAD [7] is a hybrid approach that operates in groups. Within a group, nodes use end-to-end protocols; whereas between disconnected groups, edge nodes of each group use a DTN protocol to exchange packets. In HYMAD, each node shares the knowledge of what packets they have for all the edge nodes. Whenever the edge nodes of two groups meet, they will check whether there are packets that need to be sent to a node in the other group. If so, the edge node will notify the group member about the opportunity. The grouping scheme of this solution means that that the protocol works better in scenarios in which mobility within a group is relatively low.

Other proposed protocols are much more different from our proposal; most of them propose either selecting between the two paradigms at the start of communication or switching from end-to-end protocols to opportunistic protocols if link failures occur. Ott et al. [3] propose an approach to extend AODV to support DTN routing when path to the destination breaks and cannot be repaired. The opportunistic delivery is not from the node where the path broke but from the source node (i.e., switching to DTN at source). In [5], the authors propose to select mode of communication in advance. The selection of either end-to-end or opportunistic depends on metrics which indicate the estimated lifetime of the link and the required time for successfully completing the transmission. Once the mode of communication is identified, data will be sent only using that communication method. In the case of link failure, the selection processes will be re-evaluated again.

Our goal is to develop an adaptive protocol, which can dynamically switch between end-to-end and opportunistic routing according to the network situations, as discussed in Section I.

VI. Conclusion

In this paper, we proposed a hybrid approach to routing in wireless mesh networks that integrates features of delay-tolerant (opportunistic) protocols with the traditional end-to-end routing protocols (e.g., AODV, OLSR). We described a generic approach for integrating end-to-end routing protocols with opportunistic communication. We also demonstrated how this generic approach can be applied to integrate AODV with opportunistic communication. The resulting protocol, AODV-OPP, buffers packets that would be dropped by AODV due to path break, for later delivery. We validated AODV-OPP's operations against its design specifications and evaluated the performance of the protocol using both synthesised and realistic traces. The results show that AODV-OPP consistently outperformed the original AODV protocol with packet delivery ratio (PDR) gain up to 45%.

References

[1] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *Communications Magazine, IEEE*, vol. 43, no. 3, pp. 123–131, March 2005.
[2] Firetide, http://www.firetide.com/innercontent.aspx?taxid=16&id=3216.
[3] J. Ott, D. Kutscher, and C. Dwertmann, "Integrating dtn and manet routing," in *Proc. of ACM SIGCOMM workshop CHANTS*, 2006.
[4] A. Petz, A. Bednarczyk, N. Paine, D. Stovall, and C. Julien, "Madman: A middleware for delay-tolerant mobile ad-hoc networks," Technical Report TR-UTEDGE-2010-010, Tech. Rep., 2010.
[5] J. Lakkakorpi, M. Pitkänen, and J. Ott, "Adaptive routing in mobile opportunistic networks," in *Proceedings of MSWiM2010*, Bodrum, Turkey, October 2010, pp. 101–109.
[6] L. Delosiéres and S. Nadjm-Tehrani, "Batman store-and-forward: the best of the two worlds," in *Proceedings of PerCom2012 workshop, PerNEM 2012*, Lugano, Switzerland, March 2012, pp. 727–733.
[7] 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. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.03.005
[8] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," IETF. RFC 3561, July 2003.
[9] "Bonnmotion," http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/.
[10] "The san francisco cab traces," http://cabspotting.org.