A Performance Study of Hybrid Protocols for Opportunistic Communications

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

***Abstract*—Typically, many multi-hop wireless networks rely on routing protocols to form end-to-end routes to deliver data packets from the sources to the respective destinations. In the case of link failures occur due to interference or node mobility, these protocols will try to reroute. When no alternative route is found, packets will be dropped as a result. As a way to improve the packet delivery ratio, these protocols can be extended to support the store-carry-forward features. We propose a hybrid approach to enable dynamic switching between mode of communication according to the link conditions. This is different from other existing solutions where the switching occur only when the end-to-end route fails, and packets will be delivered using another communication mode for the rest of the routing path. In this paper, we present the implementation of incorporating the dynamic switching into two routing protocols (AODV and OLSR) and conduct a performance study for the two hybrid protocols in a number of simulation based scenarios.**

I. Introduction

In many multi-hop wireless networks, data packets are delivered from source nodes to their respective destinations using end-to-end routes. Routing protocols play an important role in discovering and maintaining these routes according to the ever changing topology. Whenever a route fails due to changing link conditions these protocols try to repair the route. In the case when alternative routes can not be found, a typical router action is to drop all the packets that rely on the failed route. In [1], we proposed to extend the capability of end-to-end routing protocols to support the *store-carry-forward* features. With the extended functionalities, the traditional protocols are able to dynamically switch between end-to-end routing and opportunistic or delay-tolerant network (DTN) type's hop-by-hop routing depending on the link conditions. The hybrid protocols are able to improve the packet delivery ratio (PDR) in the case of link failure and also reduce the packet delay (compared to opportunistic delivery) when an end-to-end route is possible.

In this paper, we extend the basic idea proposed in our previous paper and apply the hybrid communication mode to two of the most well-known routing protocols in multi-hop wireless networks: *AODV* [2], a reactive distance vector protocol, and *OLSR* [3], a proactive link state protocol. As a result, we present our design and implementation of the hybrid protocols, AODV-OPP and OLSR-OPP respectively. In addition, we explore the *pros* and *cons* of these hybrid protocols by conducting a number of simulations with various evaluation metrics.

The related work on adding the store-carry-forward features to traditional end-to-end roting protocols can already be found in the literature. However, the primary goal of these described extensions is to solve the problems related to link failure by switching to packet delivery in a hop-by-hop opportunistic fashion for the lifetime of the packet flow. For example, SF-BATMAN [4] presents a preliminary attempt to extend BATMAN (a reactive protocol similar to AODV) with the store-and-forward functionality. The evaluations presented in the paper are very preliminary. In [5], Ott et al. proposed an approach to extend AODV to support opportunistic routing when path to the destination breaks and cannot be repaired. The switching from AODV to opportunistic communication is applied in the source node and the packets are delivered opportunistically to their destination.

In contrast, we propose a dynamic hybrid protocol in which packets that would be lost due to route failures 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 potential partial end-to-end routes that can be created in wireless networks.

The contributions of this paper are: (i) an overview of the hybrid approach for efficient opportunistic communication; (ii) a description of the design and implementation of the hybrid AODV-OPP and OLSR-OPP protocols; (iii) a systematic evaluation of the two hybrid protocols.

The remainder of the paper is organized as follows. Section II provides a general functionality overview of the two end-to-end routing protocols for multi-hop wireless networks. This is followed by the description of the hybrid protocol concept in Section III. The implementation details of the AODV-OPP and OLSR-OPP hybrid protocols are presented in Section IV. Section V discusses the evaluation methodology and results. Finally, we conclude the paper in Section VII.

II. Background

There exist many routing protocols for multi-hop wireless networks, and each with different characteristics in terms of

978-1-4799-0540-9/13/$31.00 ©2013 IEEE

route discovery and maintenance. In this paper, since our goal is to demonstrate the feasibility of developing a hybrid protocol, we focus on two well-known protocols, AODV and OLSR, as they respectively represent two classes of routing protocols: reactive protocols and proactive protocols. In this section, we provide a brief overview of the functionality of these two protocols and discuss their differences that have impact on the proposed hybrid protocols.

A. AODV: the basics

AODV [2] is a reactive protocol, which establishes a route on-demand when a node needs to send a packet and the route is not present in the routing table. 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 a packet should 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 destination 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 becomes active. RREQs that exceed their lifetime are discarded. Each routing entry in the routing table 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 the periodic Hello messages (i.e., missing Hello messages). 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 are sent along the affected path to invalidate the routing entry in all the affected nodes.

In AODV, Link-layer detection is another approach to detect link failure. As shown in Fig. 1, the node registers a callback function for each link a node has with its neighbours. When the link-layer reports route failure through this function, the protocol will use the *local repair* mechanism (the link-layer detection feature needs to be 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.

B. OLSR: the basics

OLSR [3] is a proactive (table-driven) protocol, which maintains up-to-date link state information of nodes in the network. The routing table has route information for any destination whenever it is needed. A technique, called *Link Sensing*, is employed to distribute the link state information (using periodic HELLO and Topology Control messages) of each node to the neighbouring nodes. Alternatively, link-layer feedback is

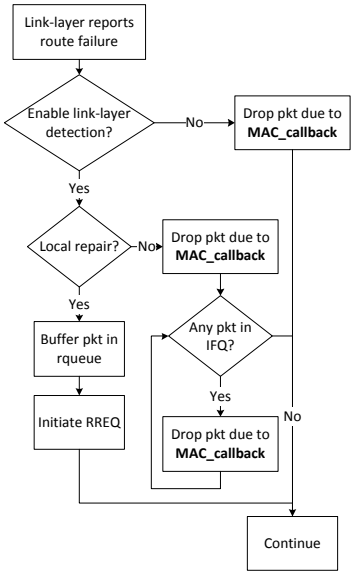


Fig. 1. The link-layer detection in AODV.

another way to populate the local link set. The link state information needs to be flooded through the network to keep each node's routing tables up-to-date. In large networks, if every node frequently sends the topology information, it will dramatically increase the protocol overhead. To reduce the overhead, OLSR nodes delegate the task of exchanging topology information (in the form of Topology Control messages) to a set of multi-point relies (MPRs). Each node will choose MPRs (supported by a MPRs selection algorithm) from its one-hop neighbours that have *symmetric* connectivities to the node. Also, the MPRs are those neighbours that are able to completely cover the set of the two-hops neighbours of the node. The role of MPRs are to disseminate the topology information between other MPRs for the members of the MPRs. For these two types of control messages, HELLO messages are sent only to the one-hop neighbours, but the TC messages are forwarded by the MPRs in order to flood the entire network with topology information [6]. OLSR achieves optimal efficiency when the MPR set is as small as possible. Based on this topology information, any node in the network can compute the next-hop required by the routing table using the shortest path algorithm.

C. AODV and OLSR: the comparison

In highly mobile networks, AODV could perform better than OLSR, since it does not actively maintain routes for the entire network (i.e., smaller protocol overhead). Due to mobility, most routes might not be valid when they are needed. In addition, in OLSR any change in part of the network will cause a global update in every node's routing table. However, OLSR outperforms AODV for some metrics, especially in terms of delay. As OLSR exchanges topology information with all nodes in the network, the route is ready for use whenever a node has packets to send. The responsiveness of OLSR is entirely up to the interval settings of the two link state messages (HELLO

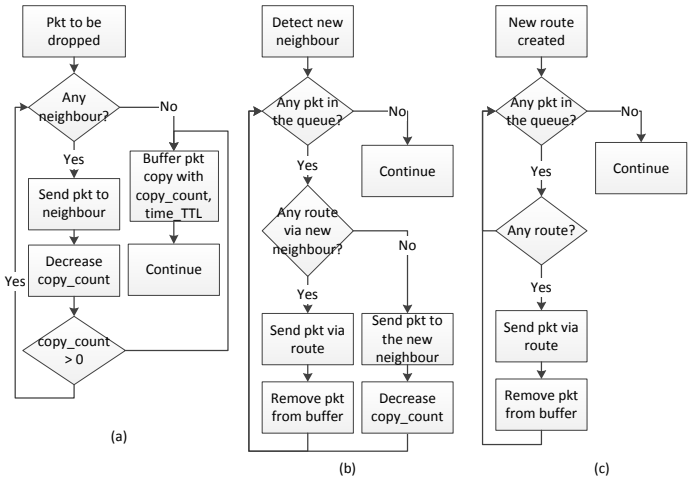


Fig. 2. Processes for handling packet drops.

and TC). In comparison, AODV requires the time to initiate the route discovery process if the route does not exist. This potentially increases the packet delay.

With regard to handling packet drops, AODV introduces a buffering feature, i.e., the *rqueue*, to give a node the time to repair the route by its route discovery process. In contrast, in OLSR if a packet cannot be sent due to no route to the destination, this packet will be dropped.

III. Concept of the Hybrid protocol

In wireless networks, link failures due to mobility of wireless nodes or interference lead to packet loss if an alternative route to the destination cannot be established. In our previous paper [1], we proposed a generic hybrid approach that integrates the *store-carry-forward* functionalities with the end-to-end protocols. In addition, we demonstrated the feasibility of the concept by developing AODV-OPP that is an extension of the AODV protocol. In this section, we revisit briefly the concept of the hybrid protocol.

A. Handling packets drop

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. 2(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 opportunistic packet queue (OP_Queue) with the remaining $copy\_count$ and $time\_TTL$ (i.e., time-to-live in time unit for the packet in the OP_Queue). The router then continues with the normal routing operations.

B. Detecting new neighbour

Another event that triggers the delivery of the buffered packets, as shown in Fig. 2(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 OP_Queue. If the OP_Queue is empty, then the router continues with the normal routing processes. On the other hand if the OP_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 OP_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 OP_Queue.

C. Detecting new route

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.2(c)) to try to send buffered packets using any new route. It is similar to the processes described in Fig.2(b). Except that if there is no route for a packet, then the next packet will be checked.

IV. Implementation

There are common similarities of both classes of protocols (*proactive* and *reactive*), the generic concept of the hybrid protocols is presented (in the previous section) to summarise the common processes required to support dynamic switching in the hybrid protocols. In this section, we discuss the implementation differences of AODV-OPP and OLSR-OPP in order to share our experience in developing the hybrid protocols.

We develop AODV-OPP as an extension of the NS2 implementation of AODV (the built-in CMU version) protocol, whereas the OLSR-OPP is an extension of the UM-OLSR [7] protocol. The implementation of both protocols are developed in *ns2.34*.

A. Common parts of AODV-OPP and OLSR-OPP

As highlighted in the generic concept of the hybrid approach, there are common parts required to develop the hybrid protocols.

The first common element that is in both protocols is the extended packet structure for storing the dropped packets with meta-information. In the NS2 simulation environment, we extend the node class with a link list (OP_Queue) to store those buffered packets. Each buffered packet in the OP_Queue not only stores the original data packet, but also records the *copy_count* and *time_TTL* respective to this packet. Fig. 3 shows the structure of the OP_Queue and the buffered packets that are stored in it.

The *copy_count* is used by the packet delivery processes to limit the number of duplicates forwarded to the next hop. To remove the stale packets from the OP_Queue, we introduce a timer event that will periodically check the *time_TTL* of each packet inside the OP_Queue. Expired packets will be removed.

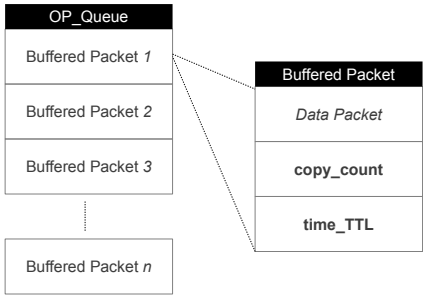


Fig. 3. OP_Queue and Buffered Packet structure.

Algorithm 1 Route probing for destinations

```
if OP_Queue.contains(Packets p) then
  for p in Packets do
    Destination d = p.getDst();
    if !RREQList.contain(d) then
      sendRREQ(p, d);
      RREQList.add(d);
    end if
  end for
end if
```

To prevent routing loop, many routing protocols use sequence number, which increase by one for every one further hop towards the destination. However, due to the packet drops and these packets may be stored for delivery at a later time. In this case, packets are delivered not in order, therefore sequence number does not work. To address this problem, we introduce a structure (currently implemented as link list) to record the packets IDs the node has seen in the last time window.

B. Implementation of AODV-OPP

Through the development of AODV-OPP [1], we identify a number of scenarios that are specific to AODV and the generic concept of hybrid protocol does not address.

In the hybrid approach, if a packet is dropped by an intermediate node due to no route to the destination, the packet is then stored on this node. When the intermediate node encounters other nodes in the network, they will exchange the buffered packets. However, they do not initiate route discovery, which might discover new routes that consist of the encounting nodes. As a result, we might miss the opportunity to deliver some packets using the end-to-end route. To address this problem, we incorporate a process to probe for potential route for every destination using the first buffered packet in the OP_Queue. Algorithm 1 shows the Pseudo-codes of this process. If the OP_Queue has packets, then for each unique destination (gather from the packets) we send a route request for that destination and add the destination address to the RREQList (to avoid sending duplicate route request for the same destination). This is a process executed in addition to sending the packet to the one-hop neighbours. The idea is that rather than waiting for the route reply or the expiry of RREQ (default value is 10 s), we initiate the route discovery process with the first packet. Therefore, subsequent packets will have chance to use the end-to-end route if there is one.

C. Implementation of OLSR-OPP

In OLSR, the status of a link is managed very carefully. A link is considered to be bi-directional. When a node received HELLO messages from a neighbour, it creates an entry in the neighbour information set (*nb_tuple*) to indicate the present of the neighbour. However, exchanging of packets happens after these two nodes created an entry in the link information set (*link_tuple*). Once the *link_tuple* is created, it can be either *asymmetric* if HELLO messages are received from one direction, or *symmetric* if exchanging a sequence of HELLO message containing information of both nodes. Every time when an OLSR node receive a control packet, the routing table will be recomputed after processing the non-duplicate message. A control packet could be (i) *HELLO message* about the local neighbour information, (ii) *TC message* that will update the topology information set, or (iii) *MID (Multiple Interface Declaration) message* that advertises the information about the node's interface association. Therefore, whenever a change of neighbour or link, every node in the network will get an update of the routing table. This update can be very frequent. Because of these characteristics in OLSR, the implementation of OLSR-OPP is slightly different from the implementation of AODV-OPP. For example, in OLSR-OPP the processes, shown in Fig. 2(b) is triggered by update on the *link_tuple*, rather than changes of the *nb_tuple*. Furthermore, the processes for handling new routing entry update is not necessary in OLSR-OPP. There are two reasons: (i) routing table update is very frequent and majority of the new routes found are not for the buffered packets inside the OP_Queue, and (ii) neighbour set update and recomputation of routing table are triggered by the received HELLO messages, but updating the neighbour set is always done first.

V. Evaluation

In this section we describe the NS2 simulation scenarios that we used to validate the correctness and to evaluate the performance of AODV-OPP and OLSR-OPP. These simulations evaluate the performance of both protocols for a variety of aspects, including overhead, functionalities and improvement in packet delivery ratio.

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

A. Evaluation methodology

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 [8] 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

TABLE I
SIMULATION PARAMETERS

Copy_count	5 copies
Packet TTL	400 s
Simulation Time	500 s
Traffic flow	one per node
Traffic duration	10 s
Data rate	4 packets per second
Tx range	250 m
IFQ length	50 pkts
BufferQueue size	unlimited

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*) [8]. 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 need 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. From Fig. 4, we see that 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. Comparison of AODV and OLSR

The first set of simulations is to investigate how AODV and OLSR perform in the 300 scenarios with different partitioning degrees. Fig. 5 shows the performance of two protocols, in terms of packet delivery ratio (PDR). In this figure, we also plot the fitted curves (labelled as *curve*) of both protocols using second degree polynomial. As we have expected, both protocols achieve lower PDR as the partitioning degree increases (i.e., the network becomes sparse). As highlighted by the fitted curves

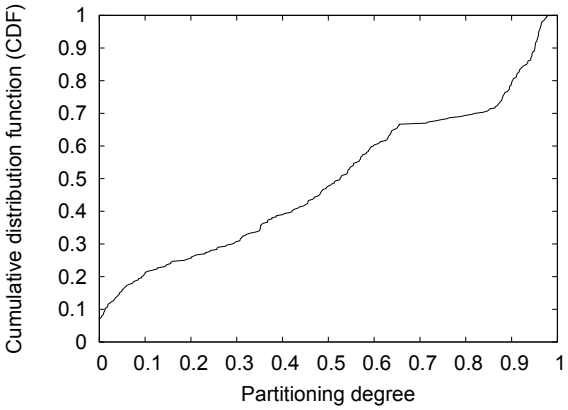


Fig. 4. Scenario distribution.

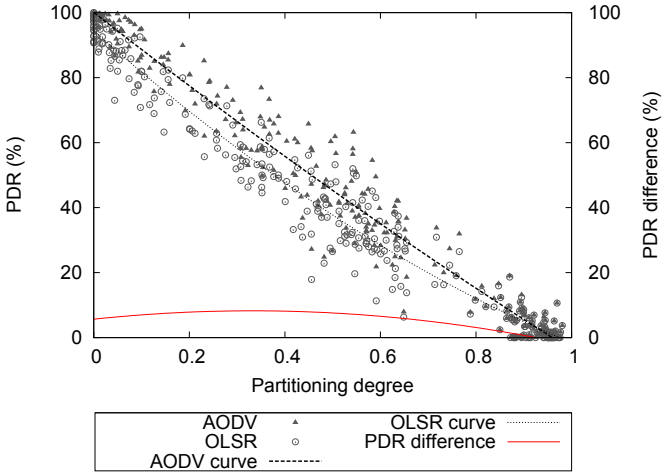


Fig. 5. Performance of AODV and OLSR.

and the PDR difference, AODV outperforms OLSR more than 5% in all the scenarios from the low to medium PD ranges. In these scenarios the network is relatively dense, therefore frequent HELLO and TC messages exchange in OLSR could become the source of interference that prevents nodes from successfully sending data packets, resulting in lower PDR than AODV.

We conjecture the overhead in OLSR is the cause of this PDR difference. The excessive overhead is understandable as OLSR frequently exchanges control information to keep the routing tables of every node in the network up-to-date. Fig. 6(a) shows the Normalised Routing Load (NRL) [9] of both protocols for all the scenarios. The NRL is defined

$$NRL = \frac{N_{routing}}{N_{data}} \quad (1)$$

where $N_{routing}$ is the number of routing control packets sent or forwarded; N_{data} is the number of data packets received at the destination. In other words, the NRL is *the number of routing packets transmitted per data packet received at the destination*. As shown in the figure, OLSR generates significantly higher load of control packets for every successfully delivered data packet. Also, this routing load increases dramatically when the

network becomes very sparse, as shown in Fig. 6(a). From the Cumulative Distribution Function (CDF) graph, as shown in Fig. 6(c), we can see that the overhead generated by AODV is almost negligible compared to OLSR (about 5% of the cases with NRL more than 1000). The saving in overhead in AODV is due to its on-demand route discovery when a node has packet to send, whereas OLSR needs frequent exchange of control information to maintain the up-to-date neighbour, link and topology information.

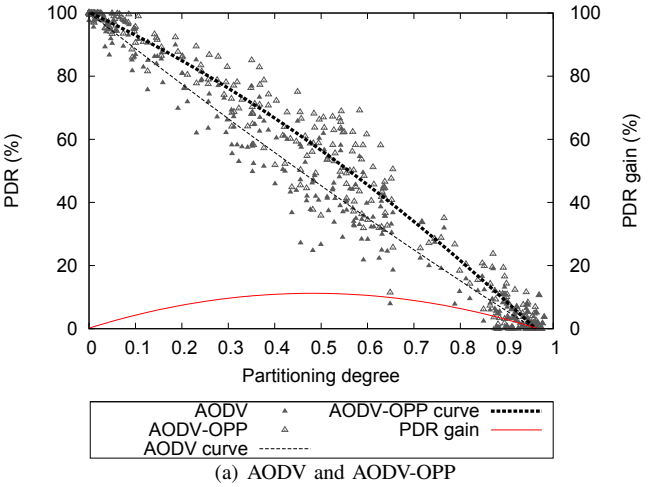
In addition to the overhead, Figs. 6(b) and (d) show comparison of the packet end-to-end delay of the two protocols and the corresponding CDF analysis. The figures show the average end-to-end delay of AODV is significantly higher than OLSR. In about 30% of cases the averaged end-to-end delay is more than 200 ms, whereas the delay for OLSR is almost negligible. These results validate the discussion about the two protocols in Section II; that is, in OLSR every node always knows how to route a packet if the route exists, but in AODV the node will have to initiate the route discovery processes.

C. Performance of AODV-OPP and OLSR-OPP

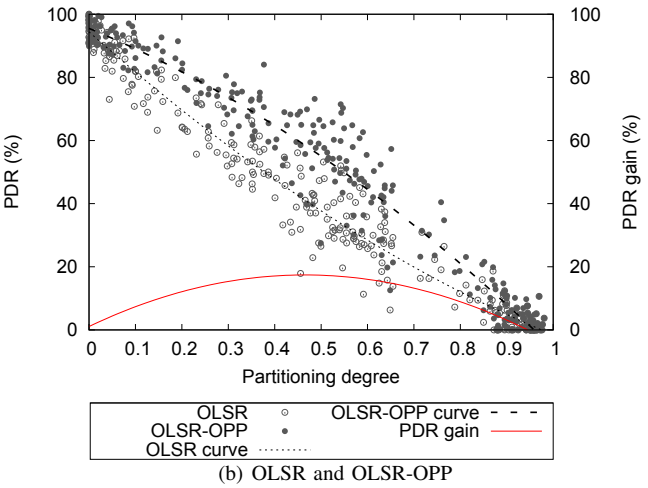
From the baseline comparison between AODV and OLSR, we observe that AODV achieves better PDR than OLSR in almost all 300 scenarios with different PD values. Using these 300 scenarios, we conduct the same set of simulations to study the performance of the AODV-OPP and OLSR-OPP. In this subsection, we present the results for both protocols.

As shown in Fig. 7(a), we can see that AODV-OPP improves the PDR over AODV for a maximum of 10% (as highlighted by the PDR gain curve). However, a greater PDR gain by the hybrid approach is achieved by OLSR-OPP over OLSR. As shown in Fig. 7(b), the maximum PDR gain is as high as 17%. We observe in both figures that the hybrid protocols achieve the maximum PDR gain over their respective protocols in the medium PD range. This is because in low and high PD scenarios, nodes are either too dense or too sparse for the new extension to play a part in improving the PDR. For example, if the network is too dense (i.e., low PD value), nodes are connected by end-to-end routes most of the time and the chance of finding an alternative route when route failure occurs is high. Therefore, both the original protocols and the hybrid protocols will achieve similar PDR. Following this argument, when the network is relatively sparse (i.e., medium PD value), node mobility not only causes a lot of route failures, but also creates the opportunities for the hybrid protocols to demonstrate their *store-carry-forward* capability. Fig. 7(c) shows the CDF for the PDR gain of the two hybrid protocols. We notice that for 70% of cases OLSR-OPP achieves more significant PDR improvement (over OLSR) than AODV-OPP (over AODV).

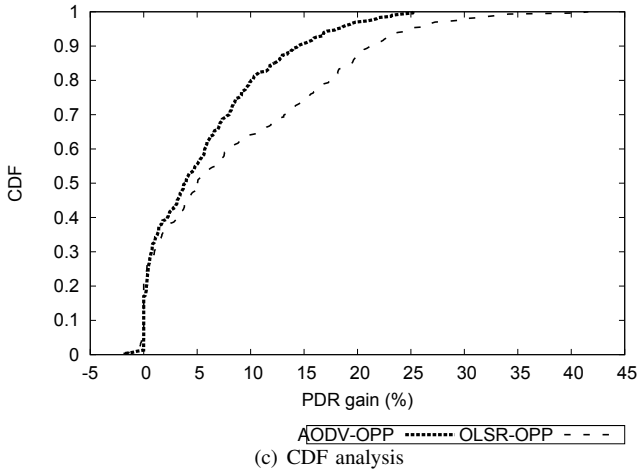
To compare the performance of all protocols under the same set of scenarios, we put the fitted curves of all four protocols in Fig. 8. It is surprising to see that the hybrid protocol OLSR-OPP is able to achieve similar PDR as the AODV-OPP for the same set of simulation scenarios, except the cases when the network is relatively dense. In dense networks (low PD range), AODV-OPP outperforms OLSR-OPP by about 10%. We conjecture that the difference in PDR is caused by the overhead of OLSR, this overhead not only increases the NRL, but also increases the interference and therefore reduces the success rate of packet delivery.



(a) AODV and AODV-OPP

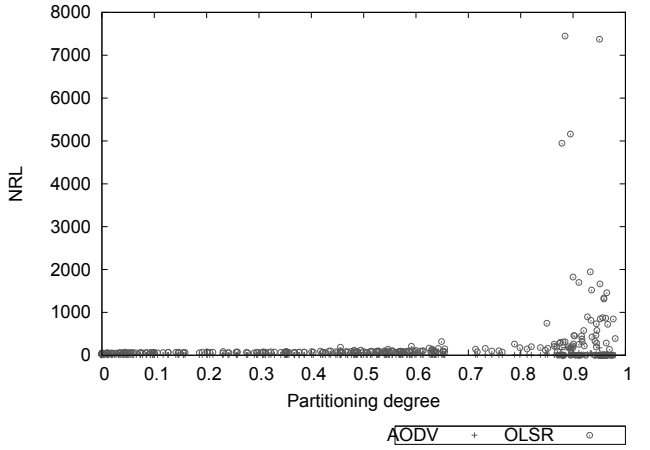


(b) OLSR and OLSR-OPP

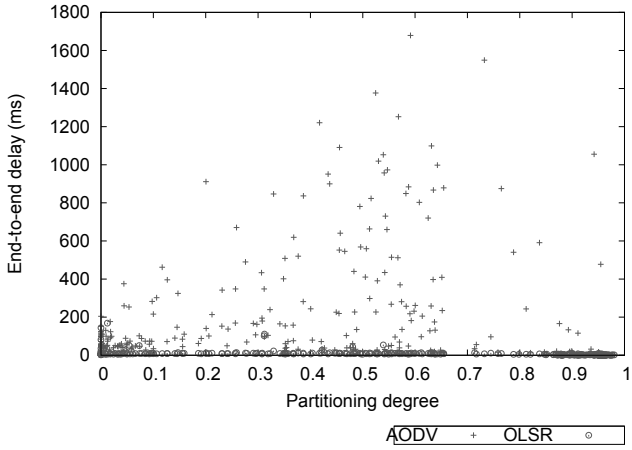


(c) CDF analysis

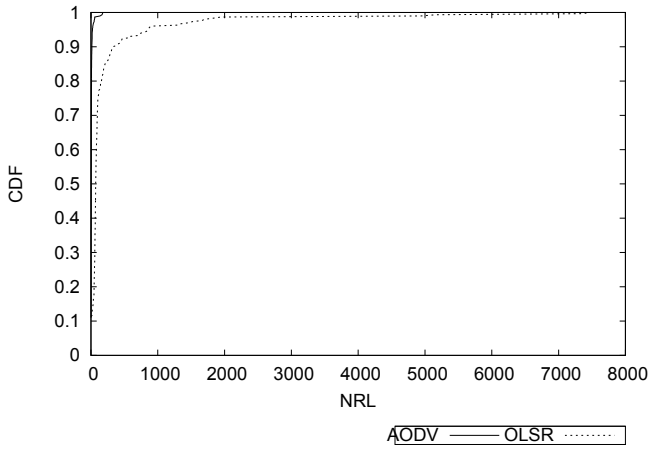
Fig. 7. Performance of AODV-OPP and OLSR-OPP.



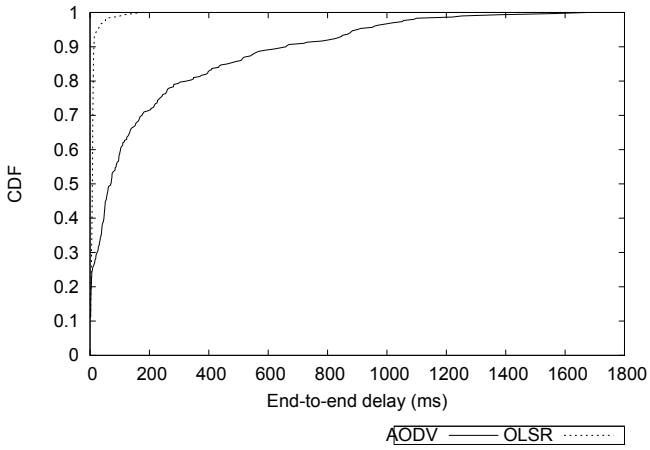
(a) Normalised routing load (NRL)



(b) End-to-end delay



(c) CDF analysis of NRL



(d) CDF analysis of Delay

Fig. 6. Overhead and delay comparison of AODV and OLSR.

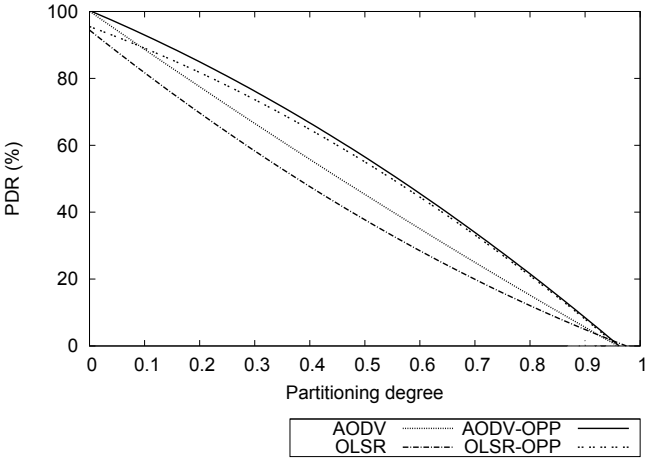


Fig. 8. Performance of all protocols.

VI. Related Work

As far as a pure performance study between AODV and OLSR is concerned, there are a number of related approaches in the literature [10], [11], [12]. For example, Huhtonen [10] presented a qualitative comparison between the two protocols. Most of the conclusions (about the overhead and delay), discussed by the author, were quantitatively evaluated in this paper using comprehensive mobility scenarios and the results confirmed the conclusion. Rastogi et al. [12] conducted a comparative study of AODV and OLSR on the Orbit testbed. However, the paper focused on the use of Orbit testbed for the performance study. The performance study is rather simplified and the evaluation metrics are limited.

The difference between this paper and the existing work is twofold: (i) the focus of this paper is not on evaluating AODV and OLSR but on using their performance as benchmark for the hybrid protocols, and (ii) we use a systematic approach in our simulation based evaluation. The 300 synthetic traces that are generated by the BonnMotion tool are randomly generated, but carefully selected to represent network scenarios with different characteristics (i.e., different densities and mobility patterns).

Another type of related work is with regard to the proposed hybrid approach. As discussed in Section I, there exist approaches [4], [5], [13], [14] that extend end-to-end routing protocols to support the carry-store-forward capabilities. However, most of these existing approaches only address the problem of

packet loss due to link failure, but cannot dynamically switch between end-to-end to carry-store-forward and back to end-to-end if the route becomes available for a packet on its way to destination.

VII. Conclusion

In this paper, we addressed the concept of hybrid routing protocols, which can dynamically switch between end-to-end and DTN type's hop-by-hop routing in multi-hop wireless networks. The goal of hybrid protocols is to reduce packet drops due to the lack of end-to-end routes caused by interference or node mobility. We discussed the concept of hybrid routing protocols and its implemention in the best known protocols from the proactive (OLSR) and reactive routing protocol (AODV) classes. The resulting hybrid protocols are AODV-OPP and OLSR-OPP. Through a number of comprehensive simulations using the synthetic traces, we conducted a performance study of these two protocols. The results showed significant PDR improvement (AODV-OPP achieves a maximum of 10% and OLSR-OPP achieves a maximum of 17% PDR gain) in the two hybrid protocols.

As dissemination of dropped packets to all one-hop neighbours cause an overhead in the network we plan to address this issue in our future work. To reduce the overhead, one solution is to selectively forward the dropped packets to only those neighbours that have better opportunity than other neighbours to deliver the packets to the destination.

References

[1] 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.

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

[3] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," IETF. RFC 3626, October 2003.

[4] 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.

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

[6] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance of multipoint relaying in ad hoc mobile routing protocols," in *Networking 2002*, Pisa, Italy, 2002.

[7] F. J. Ros, "Um-olsr ns2 implementation," http://masimum.inf.um.es/fjrm/?page_id=116.

[8] "Bonnmotion," http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/.

[9] C. Perkins, E. Royer, S. Das, and M. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," *Personal Communications, IEEE*, vol. 8, no. 1, pp. 16–28, 2001.

[10] A. Huhtonen, "Comparing aodv and olsr routing protocols," Seminar on Internetworking, 2004.

[11] J. Härri, F. Filali, and C. Bonnet, "Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns," in *Med-Hoc-Net 2006*. IFIP, Jun. 2006.

[12] D. Rastogi, S. Ganu, Y. Zhang, W. Trappe, and C. Graff, "A comparative study of aodv and olsr on the orbit testbed," in *IEEE MILCOM2007*, 2007, pp. 1–7.

[13] 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.

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