

Research Article

Robust On-Demand Multipath Routing with Dynamic Path Upgrade for Delay-Sensitive Data over Ad Hoc Networks

Sunil Kumar,¹ Siddharth Khimsara,¹ Kashyap Kambhatla,¹ Kalyani Girivanesh,¹
John D. Matyjas,² and Michael Medley²

¹ *Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182, USA*

² *Air Force Research Laboratory, Rome, NY 13441, USA*

Correspondence should be addressed to Sunil Kumar; skumar@mail.sdsu.edu

Received 10 December 2012; Accepted 6 February 2013

Academic Editor: Liansheng Tan

Copyright © 2013 Sunil Kumar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Node mobility in mobile ad hoc networks (MANETs) causes frequent route breakages and intermittent link stability. In this paper, we introduce a robust routing scheme, known as ad hoc on-demand multipath distance vector with dynamic path update (AOMDV-DPU), for delay-sensitive data transmission over MANET. The proposed scheme improves the AOMDV scheme by incorporating the following features: (i) a routing metric based on the combination of minimum hops and received signal strength indicator (RSSI) for discovery of reliable routes; (ii) a *local path update* mechanism which strengthens the route, reduces the route breakage frequency, and increases the route longevity; (iii) a *keep alive* mechanism for secondary route maintenance which enables smooth switching between routes and reduces the route discovery frequency; (iv) a *packet salvaging* scheme to improve packet delivery in the event of a route breakage; and (v) low HELLO packet overhead. The simulations are carried out in ns-2 for varying node speeds, number of sources, and traffic load conditions. Our AOMDV-DPU scheme achieves significantly higher throughput, lower delay, routing overhead, and route discovery frequency and latency compared to AOMDV. For H.264 compressed video traffic, AOMDV-DPU scheme achieves 3 dB or higher PSNR gain over AOMDV at both low and high node speeds.

1. Introduction

Wireless mobile ad-hoc networks (MANETs) are usually bandwidth limited and characterized by time-varying error-prone channels due to node mobility, resulting in the link loss, route breakage, and short route lifetime. These characteristics pose a serious challenge in designing efficient routing schemes, especially for supporting the delay-sensitive traffic such as video streaming.

The proactive routing protocols (e.g., destination sequenced distance vector (DSDV) [1] and optimized link state routing (OLSR) [2]) generate too much overhead in MANET because they find paths in advance for all source and destination pairs and periodically exchange topology information to maintain them, even when no data is being transmitted [3, 4]. In contrast, the reactive routing protocols (e.g., ad-hoc on-demand distance vector (AODV) [5] and dynamic source routing (DSR) [6]) have lower overhead as they find a route from the source to destination as and when required [3, 4].

Many on-demand routing protocols proposed for MANET in the past do not sufficiently address the issue of frequent route breakage due to high node mobility. Two approaches employed to overcome this problem are (i) frequent link updates along the route to notify the sender and receiver about poor intermittent links and carry out path maintenance and (ii) enabling route diversity by discovering multiple routes between the same source and destination pair to avoid complete route breakdown.

Multipath ad-hoc on-demand routing protocols (e.g., ad hoc on-demand multipath distance vector (AOMDV) [7]—a multipath extension of AODV [5]) discover and maintain more than one route between a source and destination pair [3, 4]. They achieve lower average end-to-end packet delay and higher throughput than AODV by using the alternate routes between source and destination when the primary route fails. However, high node mobility may cause the routes to break frequently forcing the source to switch between routes (if available) or rediscover new routes, which increases

both packet losses and latency. To counter it, the use of better routing metrics has been proposed in the literature, including the received signal strength indicator (RSSI) along with hop count [8]. Various route maintenance mechanisms have also been investigated in the literature, which avoid the need to use costly route discovery [9]. Many of these major schemes have been reviewed in Section 2.

In this paper, we propose a robust on-demand multipath routing scheme for transmission of delay-sensitive data over MANET, which we refer to as AOMDV with dynamic path update (AOMDV-DPU). Our scheme incorporates the following features in AOMDV: (i) it uses a routing metric based on the combination of minimum hops and RSSI for the discovery of robust and reliable routes; (ii) it uses a low-overhead *local path update* mechanism, which strengthens the primary route, reduces its breakage frequency, and increases its lifetime; (iii) it uses a very low-overhead *keep alive* mechanism for secondary route maintenance, which reduces the route discovery frequency and enables smooth switching between routes; (iv) it uses a *packet salvaging* scheme to improve packet delivery in the event of a route breakage; and (v) it reduces HELLO packet overhead by controlling their frequency.

We have significantly improved the above-mentioned features in our AOMDV-DPU scheme. To the best of our knowledge, none of the existing on-demand multipath routing schemes has all these features. These features enable our scheme to significantly improve the performance of AOMDV in terms of throughput, end-to-end packet delay, route discovery frequency, route discovery latency, and routing overhead for delay-sensitive as well as delay-agnostic data traffic. For H.264 compressed video traffic, our AOMDV-DPU scheme achieves approximately 3 dB or higher PSNR gain over AOMDV at both low and high node speeds.

The rest of the paper is organized as follows. Section 2 summarizes the existing on-demand routing protocols. Section 3 describes the proposed AOMDV-DPU routing scheme, including the routing metric, local path update, HELLO packet frequency, packet salvaging, and *keep alive* message features. Section 4 describes the simulation setup and the performance evaluation for the delay-sensitive, delay-agnostic, and H.264 video traffic. Finally, Section 5 discusses the conclusion and future work.

2. Related Work

In this section, we briefly review the on-demand multipath routing schemes available in the literature. AOMDV [7] uses the information available in the routing messages to find multiple disjoint and loop-free routes during its route discovery process. The link disjoint feature of this protocol ensures that no two parallel routes have a common link (link disjoint) or common node (node disjoint). In the event of a route breakage due to link instability, the source simply chooses the next available route in order to continue packet transmission to the destination. Further, every node broadcasts HELLO packets for the purpose of maintaining local link connectivity like in AODV. This provides a better fault tolerance in the

sense of faster and efficient recovery from route failures, at the cost of high HELLO packet frequency. AOMDV also does not ensure the longevity of discovered routes at high node mobility.

In AODV-BR (AODV with backup route) [10], a node overhears its neighboring nodes and records them as the next hop to the destination in an alternate route table. When the node detects a route breakage in the primary route, it forwards the data packet to the best neighbor from its alternate route table. The neighbor node, in turn, will check its own route table and find its best next hop node to forward the packet, thus creating a localized detour. AODV-AP (AODV with accessibility prediction) [11] follows the same paradigm by keeping a track of all “accessible” nodes. Multipath and multipoint relays- (MPR-) based AODV (MMDV) [12] is a hybrid variation of AODV and the link state OLSR protocol. It finds multiple paths, uses a dynamic MPR flooding, and also accumulates paths to reduce the overhead while improving the packet delay. NDMR [13] is a similar extension to AODV, which finds multiple paths by virtue of path accumulation via the route request (RREQ)/route reply (RREP) packets and additionally discovers node disjoint paths such that there is no common node between two distinct paths. The split multipath routing (SMR) protocol [14] finds multiple paths which are maximally disjoint. Ad hoc Distance Vector Multipath (AODVM) [15] extends AODV to find multiple node-disjoint paths with an additional feature of enabling intermediate nodes to respond to RREQs to enhance their routing tables with intermediate paths. AODVM, however, does not significantly improve the performance of AODV. DYMOM [16] is a multipath extension to the dynamic MANET on-demand (DYMO) protocol which finds multiple node disjoint paths.

The multipath adaptive load balancing (MALB) [17] protocol introduces a general framework of extending any ad hoc routing protocol to perform load balancing to distribute the traffic among multiple paths dynamically, by measuring the path statistics so as to better use the network resources resulting in reduced congestion and end-to-end delay. AODV with break avoidance (AODV-BA) [18] scheme proposes a modification in the route maintenance phase of AODV such that any intermediate node that detects a link failure likelihood immediately notifies the upstream node. AODV with preemptive local route repair scheme [19] avoids route failures by preemptively and locally repairing the routes when a link break is about to occur, by using mobility extensions and HELLO messages. AODV with received signal strength (AODV-RSS) [20] uses the RSSI changing rate to predict the link available time and thus find the paths with longer life time. The link heterogeneity-aware on-demand routing (LHAOR) scheme [8] is an improvement over AODV and considers the link heterogeneity, in terms of link quality and mobility. It proposes a local update mechanism by using an RSSI-based link metric to make AODV adaptive to the topology variation and link quality changes. We adopt the local path update using the RSSI-based link metric of LHAOR into our AOMDV-DPU to improve the lifetime of our selected paths.

To reduce the number of route discoveries due to route failures in AOMDV, a scheme is presented in [21] to maintain all the paths by means of periodic update packets. These update packets measure the RSSI in each hop along the alternate paths and only the path with the strongest received signal strength is used for data transmission. The scheme discussed in [22] builds two paths between source and destination and creates backup paths during route reply, route maintenance, and local recovery process to improve the data transfer and fault tolerance. In AOMDV-APLP [23], multiple paths are generated by accessibility prediction, from which the route with the strongest signal strength is selected depending on the link life value predicted by a link breakage prediction technique. Enhanced AOMDV [24] reduces the route failure by preemptively predicting link failures based on the signal strength at a receiver. In MANETs, packet transmission is impaired by radio link fluctuations. The channel-aware AOMDV (CA-AOMDV) [25] uses a received signal strength threshold of the channel to select stable links for path discovery and applies a preemptive handoff strategy to maintain reliable connections by exploiting the channel state information. Several other on-demand multipath protocols were also reviewed in [7, 26].

3. Proposed AOMDV-DPU Routing Scheme

The proposed AOMDV-DPU scheme, including its important features, is discussed below.

3.1. Routing Metrics. Hop count is used as a routing metric in most routing schemes. However, the shortest-hop routes may contain weak links which are likely to break due to change in link quality in the presence of node mobility. The AOMDV-DPU scheme adopts the *node-to-end RSSI metric*, in addition to hop count, to select more stable and efficient routes [8]. We assume that the RSSI value in a wireless link could range from -50 dBm to the noise floor (-90 dBm). Since the RSSI variation is not linearly correlated to the link stability, we have nonlinearly distributed the RSSI values into different bands as shown in Table 1.

3.2. Route Setup Phase. In order to transmit packets to the destination, the source node broadcasts a route request (*RREQ*) packet to discover a route to the destination and waits for a route reply (*RREP*) packet. Each intermediate node rebroadcasts the received *RREQ* and sets up a backward route called *reverse route* towards the source [5]. Redundant *RREQ* packets received from the source via the same first hop are discarded. This is done to guarantee the link disjoint routes [7]. The next hop and last hop information is also used for checking node disjoint routes. When the first *RREQ* packet reaches the destination node, it sends a *RREP* back to the source using *reverse routes* stored in each intermediate node. The destination generates a *RREP* for each *RREQ* received from a unique last hop node. When any node receives a *RREP*, it will look for an unused next hop on the *reverse route* and unicast it via that hop to the source. This forms multiple loop-free disjoint *forward routes* (*FRs*) to the destination from all

Packet type (8 bits)	RT flags (8 bits)	Path entry: hop count (4 bits)	Path entry: RSSI (4 bits)	Reserved (8 bits)
HELLO originator IP				
Routing entry: destination IP				
Routing entry: next hop IP				
Routing entry: last hop IP				
Routing entry: dest. seq. no.				

FIGURE 1: HELLO_{LU} packet format.

the nodes that receive the *RREP* [7]. The first route is labeled as the *primary route*, and the other two routes are labeled as the *secondary routes*.

We have added an *RSSI* field to the *RREQ* and *RREP* packets to store the minimum *RSSI* value for a route. The source maps the minimum *RSSI* value of a route to the corresponding *RSSI metric* shown in Table 1. Our scheme uses the minimum *RSSI* metric because the route may contain one or more intermediate links with low *RSSI* which would affect its quality.

The first *RREP* packet received at the source initiates the forwarding of data packets, and each node on the selected *FR* to the destination invokes a local path update (*LPU*) mechanism to improve the path as explained below.

3.3. Local Path Update (LPU). Our local path update mechanism is adapted from a similar mechanism used for AODV protocol in [8]. Our scheme extends the idea to establish multiple routes using the *RSSI* metric obtained during the route setup stage, in order to generate more stable and shorter routes. This multipath adaptation required substantial improvements to the local update mechanism proposed in [8]. The details are given below.

When an *RREP* packet travels from the destination to source, each intermediate node along the *FR* marks itself as an *FR* node. When the first *FR* node on the *primary route* receives a data packet from the source, the *LPU* process is triggered by broadcasting a HELLO_{LU} packet over 1-hop length. An “*IComeFromFR*” flag is set high in the HELLO_{LU} packet to indicate that it was originated by an *FR* node. This packet contains path information from its routing table as illustrated in Figure 1. When the immediate one-hop neighbors, labeled *surrogate route* (*SR*) nodes (which are not the *FR* nodes on any of the multiple paths), receive this local route update, they add a new route entry (or update an existing route entry) in their own routing table. The *SR* neighbors also use the path information (i.e., hop count and *RSSI*) to determine if they are able to provide a better path toward the same destination. The *SR* nodes respond by rebroadcasting the HELLO_{LU} packet over one hop with the *IComeFromFR* flag set low. This packet contains new neighbor information that helps the *FR* nodes on the primary route to locally decide if they should switch to the new route with better links. Though the size of HELLO_{LU} packet in AOMDV-DPU is larger than the HELLO packet in AOMDV, it is

TABLE 1: Mapping node-to-end RSSI value to the metric.

RSSI value (dBm)	≥ -50	≥ -55	≥ -60	≥ -65	≥ -70	≥ -75	≥ -80	≥ -85	Noise floor
RSSI metric	1	2	3	4	5	7	10	14	20

a controlled one hop broadcast initiated only by the *FR* nodes compared to the HELLO packets, which are broadcasted by even those nodes in AOMDV which are not in the vicinity of an established route.

The routing table of a node is updated whenever a HELLO_{LU} is received as it also carries routing metric information of the neighboring nodes. We retain information about only the best routes in our table. HELLO_{LU} from an *FR* to *SR* node informs other *FR* nodes around it that the originating node is an *FR* node (using an *IComeFromFR* flag) and vice versa; this information is used for making routing decision(s) by each node. The *FR* node first checks if the number of hops to the destination via the new *SR* node is less than or equal to the current primary route. It then checks if the new RSSI metric is greater *than or equal* to that of the current primary route. If both conditions are satisfied, the new route through the *SR* node has a lower cost and is added to the routing table. Once the *SR* node is added to the *primary route*, it is labeled as an *FR* node. The discarded *FR* node is labeled as an *SR* node after it completes forwarding all the packets in its buffer to its next hop on the primary route.

The LPU process is illustrated through an example in Figure 2. Nodes S_{rc} , I , J , K , L , and D_{st} are *FR* nodes, and the surrounding nodes V , T , U , X , and W are *SR* nodes. The *FR* node L broadcasts a HELLO_{LU} packet with “*IComeFromFR*” flag set high causing other neighboring *FR* nodes to discard this packet. This keeps the links and nodes on the primary and secondary routes disjoint. The *SR* nodes U , X , and W in the transmission range of L hear this HELLO_{LU} packet and either add a new route entry or update the existing route information in their routing table. The *SR* nodes respond to L by broadcasting another HELLO_{LU} but with the “*IComeFromFR*” flag low. Other neighboring *FR* nodes (e.g., K) accept this HELLO_{LU} broadcast and add new path(s) in their routing table or update the existing paths. A HELLO_{LU} packet is broadcasted at a prespecified frequency in terms of packets/sec whenever an *FR* node receives a data packet as discussed in Section 3.4. In Figure 2, the *FR* node J finds that *SR* node U provides a better link to node L with the same hop count of 2 and a higher minimum RSSI of 1. It therefore switches from the current primary route $J \rightarrow K \rightarrow L$ (shown in solid line) to a new route $J \rightarrow U \rightarrow L$ (shown in dotted line). After receiving the first data packet, the *SR* node U labels itself as an *FR* node. Meanwhile, K stops receiving data packets from its previous *FR* hop J and, after forwarding all the packets in its buffer to *FR* node L , it labels itself as an *SR* node.

Thus, the LPU scheme preemptively removes the possibility of link breakage and causes the route to converge to a more efficient route. It therefore helps the source-destination pair to cope with node movements and deteriorating links. Also, this process gradually builds up a rich database of surrogate nodes and routes and increases the route lifetime.

3.4. HELLO Packet Interval. Conventionally, all the network nodes periodically broadcast HELLO packets to update the link information in AOMDV [7]. This introduces high overhead and packet loss due to interference with the data transmission on the same and neighboring route(s). In the AOMDV-DPU scheme, the HELLO_{LU} packets are generated at 1 packet/s by the *FR* nodes participating in data forwarding. We have also employed the following scheme to further reduce the number of HELLO messages without affecting the packet delivery ratio: (i) if an *FR* node does not receive any data or control packet (except the HELLO_{LU} packets from neighboring nodes) for a specified time period T_1 , we assume that it is no longer participating in the data transmission and its HELLO_{LU} packet frequency is reduced to half; (ii) further, if no data or control packets are received till a time period T_2 , ($T_2 > T_1$), the node stops transmitting the HELLO_{LU} packets. The node would restart transmitting HELLO_{LU} packets as soon as it receives a data or control packet.

3.5. Enhanced Link Layer Failure Handling: Packet Salvaging.

A link failure is assumed when the link layer does not receive (i) a clear-to-send (CTS) packet even after retransmitting the request-to-send (RTS) packet for a short retry limit, or (ii) an ACK packet in response to a data packet for a long retry limit. This is reported to the routing layer through a call-back mechanism, and a packet salvage process is initiated. With the help of the LPU process discussed in Section 3.3, each node possesses a recently updated set of its one hop *SR* neighbors which can forward the packets to the destination. Thus, the link layer failure handling is enhanced in AOMDV-DPU scheme due to availability of multiple next hops (i.e., in terms of *SR* nodes). Meanwhile, the upstream nodes continue to forward packets to the node with the broken link until they are notified about it through a route error (RERR) packet. Figure 3 illustrates that when the link between nodes K and L fails, K has another path to destination via the *SR* node W , which has the same number of hops. Therefore, K now starts forwarding packets to W . Node W has two paths to node D , with one hop and with two hops (via L) both having the same RSSI value. The path selection algorithm chooses the least hop path to D .

Unlike AOMDV [7], where packet salvaging starts as soon as the link failure is detected, a node in our scheme generates a *route error* (RERR) packet for packet salvaging only when it has exhausted all possibilities of finding a next hop node to the destination as discussed above. Recall that this RERR packet travels upstream towards the source causing all intermediate nodes, including the source, to delete the link-failed node from their routing table. The source then switches to the best available secondary route from the set of multiple paths established in the route setup stage; otherwise it enters the route setup phase explained in Section 3.2. The proposed

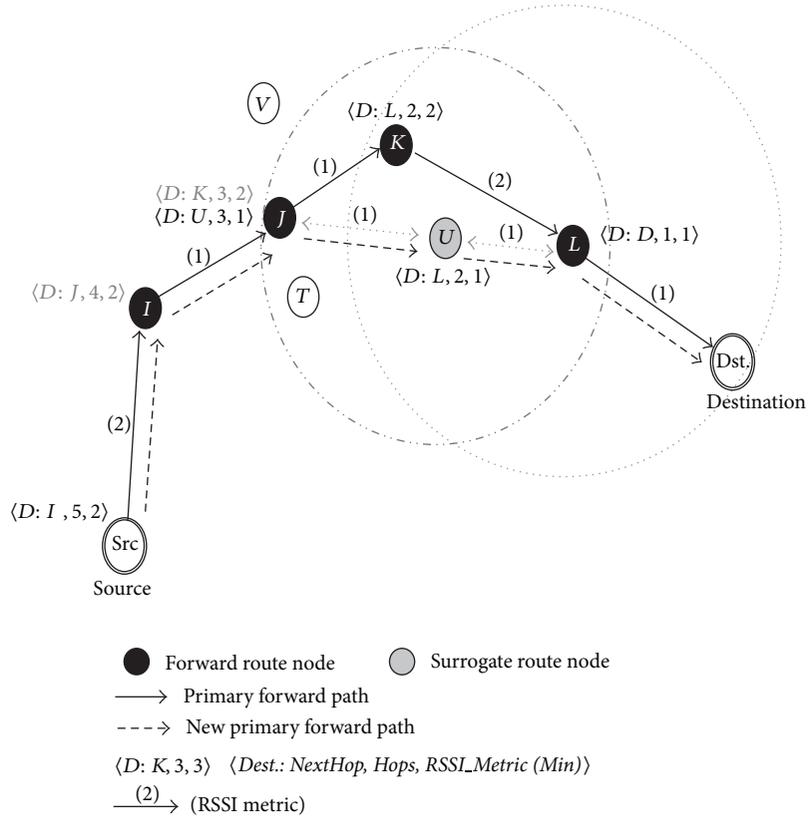


FIGURE 2: Illustration of the local route update (LPU) process.

packet salvaging scheme thus improves packet delivery and also reduces the route discovery frequency.

3.6. Keep Alive Mechanism for Secondary Route Maintenance.

The schemes mentioned in the previous sections increase the primary route lifetime. Since AOMDV does not maintain or repair the secondary routes, they often break due to node mobility and may not even be available when the primary route breaks down. We have designed a simple secondary route maintenance scheme, which performs local repair and update of the secondary routes with very low overhead. For this, the *FR* nodes on the secondary routes transmit *keep alive (KA)* packets in a unicast manner to search and repair any link failures along the secondary routes. When any *FR* node does not find a valid next hop, it initiates a *one-hop local repair* asking its neighboring *SR* nodes for a route via the same last hop. The neighboring *SR* nodes having paths to the destination via the same last hop respond to the query. Thus, the *FR* node adds the *SR* node with the best routing metric as a new next hop in the routing table. Figure 4 shows the *KA* packet format. It carries the link metric information along with the last hop and the first hop of the route. To reduce the interference, the *KA* packet frequency is kept lower than the HELLO packet frequency of 1 packet/sec. We do not employ the LPU process on secondary routes because it is too expensive in terms of the HELLO_{LU} broadcast overhead.

4. Performance Evaluation

We have simulated the AOMDV-DPU and AOMDV [7] schemes using the discrete event network simulator ns-2.32. In this section, their performance is compared for the delay-agnostic and delay-sensitive traffic, and H.264 video for varying node speeds and traffic loads.

We evaluate the performance in terms of the following five performance metrics used in AOMDV [7]: (i) *%packet loss rate (PLR)* defined as the ratio of the number of data packets dropped in the network to the total packets transmitted. Here the packet drops consist of the data packets lost due to collisions and those dropped from the queue in the network; (ii) *average latency* measured as the time a packet leaves the routing layer of the source node to the time it reaches the routing layer of the destination node. This delay includes queuing delay at the source and intermediate nodes, MAC back-off delay when the channel is busy, and propagation and retransmission delays; (iii) *average route discovery delay (D_{RD})* measured as the average amount of time for all sources in the network for which a data packet has to wait in the source buffer until the first RREP packet is received either during the initial route setup stage or the intermittent route rediscovery stage when all the multiple paths are broken; (iv) *average route discovery frequency (F_{RD})* measured as the average rate at which RREQ packets are generated every second in the network. It gives a measure of the adaptability

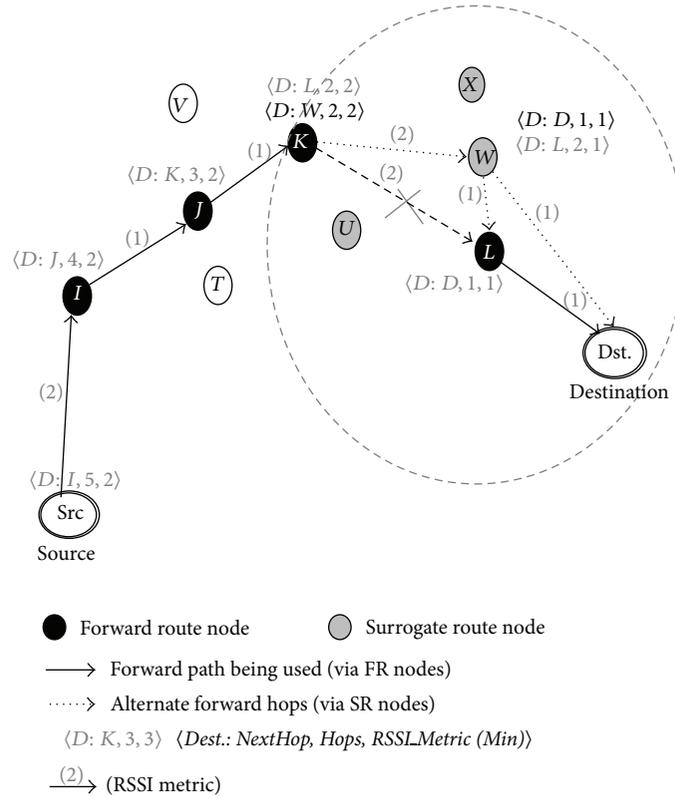


FIGURE 3: Illustration of the packet salvaging scheme.

Packet type (8 bits)	RT flags (8 bits)	Path entry: hop count (4 bits)	Path entry: RSSI (4 bits)	Reserved (8 bits)
HELLO originator IP				
Routing entry: destination IP				
Routing entry: next hop IP				
Routing entry: last hop IP				

FIGURE 4: The *keep alive* packet format.

of a routing scheme to the network topology changes due to node speeds, node positions, and link losses. A lower F_{RD} frequency signifies that the protocol is very stable, and as the frequency increases the PLR and average latency also increase; and (v) *routing overhead* (O_R) is the total number of route setup, update, and maintenance packets (i.e., RREQ, RREP, RERR, HELLO_{LU}, and KA packets) transmitted per hop per second over the network.

4.1. Simulation Scenario. Table 2 summarizes the constant network and protocol parameters used in our simulation setup, which are the same as used in AOMDV [7]. We use the time-to-live (TTL) value of 250 ms for delay-sensitive data traffic. Each node in the network is equally likely to act as source, destination, and intermediate router. Table 3 provides

TABLE 2: Simulation parameters.

Parameters	Values
Network size	1000 m × 1000 m
Number of nodes	100 (25 for video traffic)
Transmission range	250 m
Simulation time	1000 seconds
Traffic type	CBR @ 512 bytes/packet
Time to live (TTL)	250 ms for delay-sensitive data
MAC protocol	IEEE 802.11
Channel transmission rate	2 Mbps
Interface queue length	Drop tail of length 50 packets
Mobility model	Random way point model

TABLE 3: Scenario variations.

Varying parameters	Constant parameters
Average speed = 1–20 m/sec	Connections = 50, packet rate = 1 packet/sec
Connections = 1–50	Average speed = 5 m/sec, packet rate = 2 packets/sec
Packet rate = 0.25–2 packets/sec	Average speed = 5 m/sec, connections = 50

the set of parameters used during the simulation of various scenarios. We study the performance for up to 50 connections

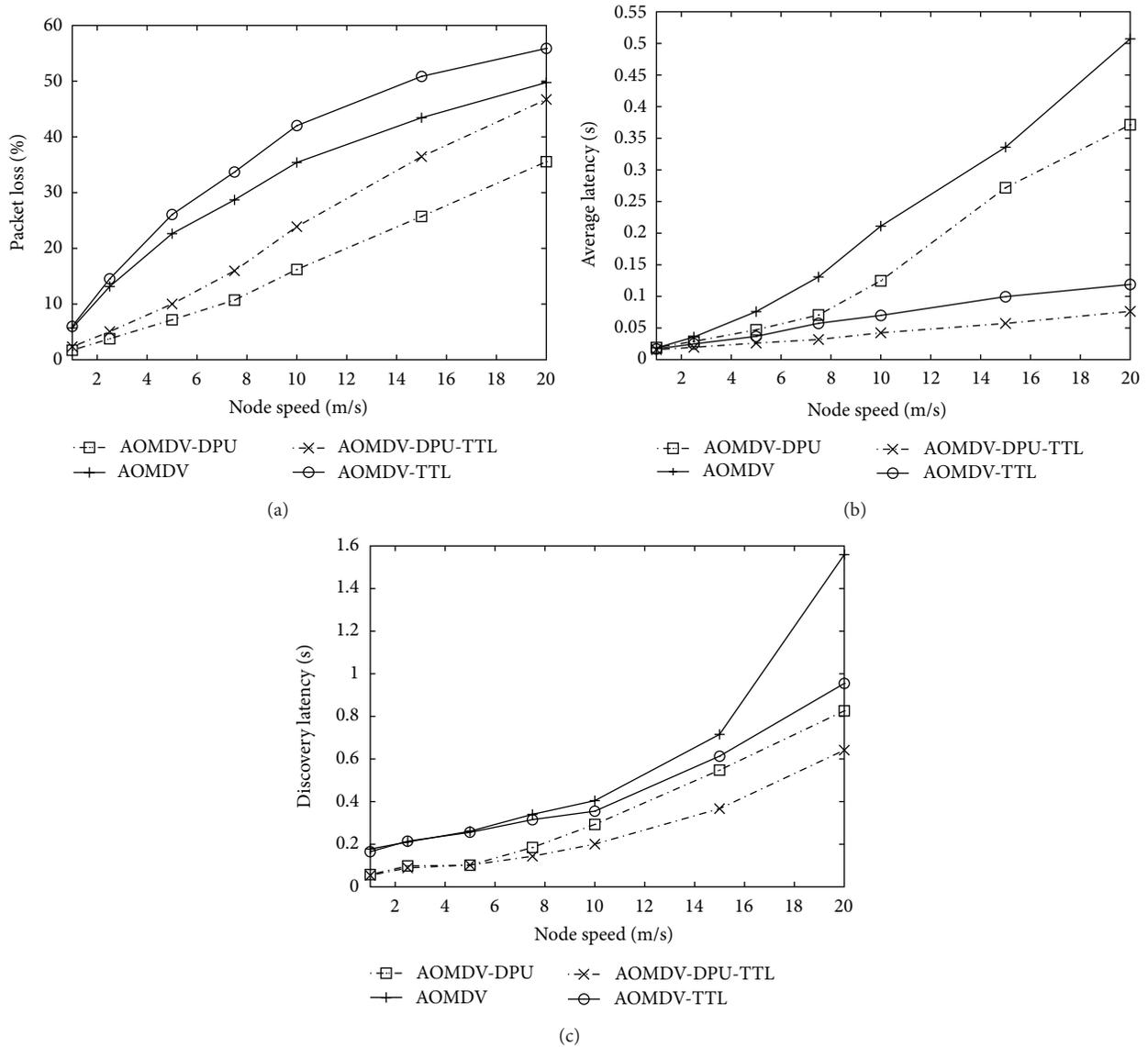


FIGURE 5: Performance with varying node speeds (a) packet loss, (b) average packet delivery latency, and (c) route discovery latency.

and a maximum node speed of 20 m/s compared to 10 m/s in AOMDV [7]. Each simulation is run for 1000 seconds, and the initial 250 seconds are ignored to allow the network to stabilize. In order to average out the randomness in the movement pattern, we repeat each scenario 10 times (i.e., ten unique traffic movement patterns) and compute the mean of each performance parameter.

4.2. Varying Node Speed. Figure 5 shows the performance of the AOMDV-DPU and AOMDV schemes for the delay-sensitive (represented as TTL scheme) as well as delay-agnostic traffic with varying average node speeds at a constant load of 1 packet/s per connection for 50 connections (i.e., source destination pairs). This corresponds to an offered load of about 200 kb/s.

In Figure 5(a) the PLR increases with node speed because the corresponding link failure probability also increases.

Overall, the delay-sensitive traffic has higher PLR than the delay-agnostic traffic because the packets are discarded if they cannot reach destination before their TTL expiry. The AOMDV-DPU scheme has up to 20% lower PLR compared to the AOMDV scheme for both types of traffic. This reduction corresponds to less than half the PLR of AOMDV scheme for both types of traffic at 1–10 m/s node speeds. Though AOMDV would switch to an alternate secondary route in the event of a link failure, the packets already queued in the nodes of the broken route may not be efficiently salvaged. Moreover, the life time of secondary routes also decreases at higher node speeds. Another contribution to packet loss in AOMDV is the network wide periodic HELLO broadcasts to maintain the link connectivity. This collides with data packet transmission leading to MAC layer backoffs. On the other hand, the LPU mechanism in AOMDV-DPU increases the longevity of primary route, the use of KA messages updates

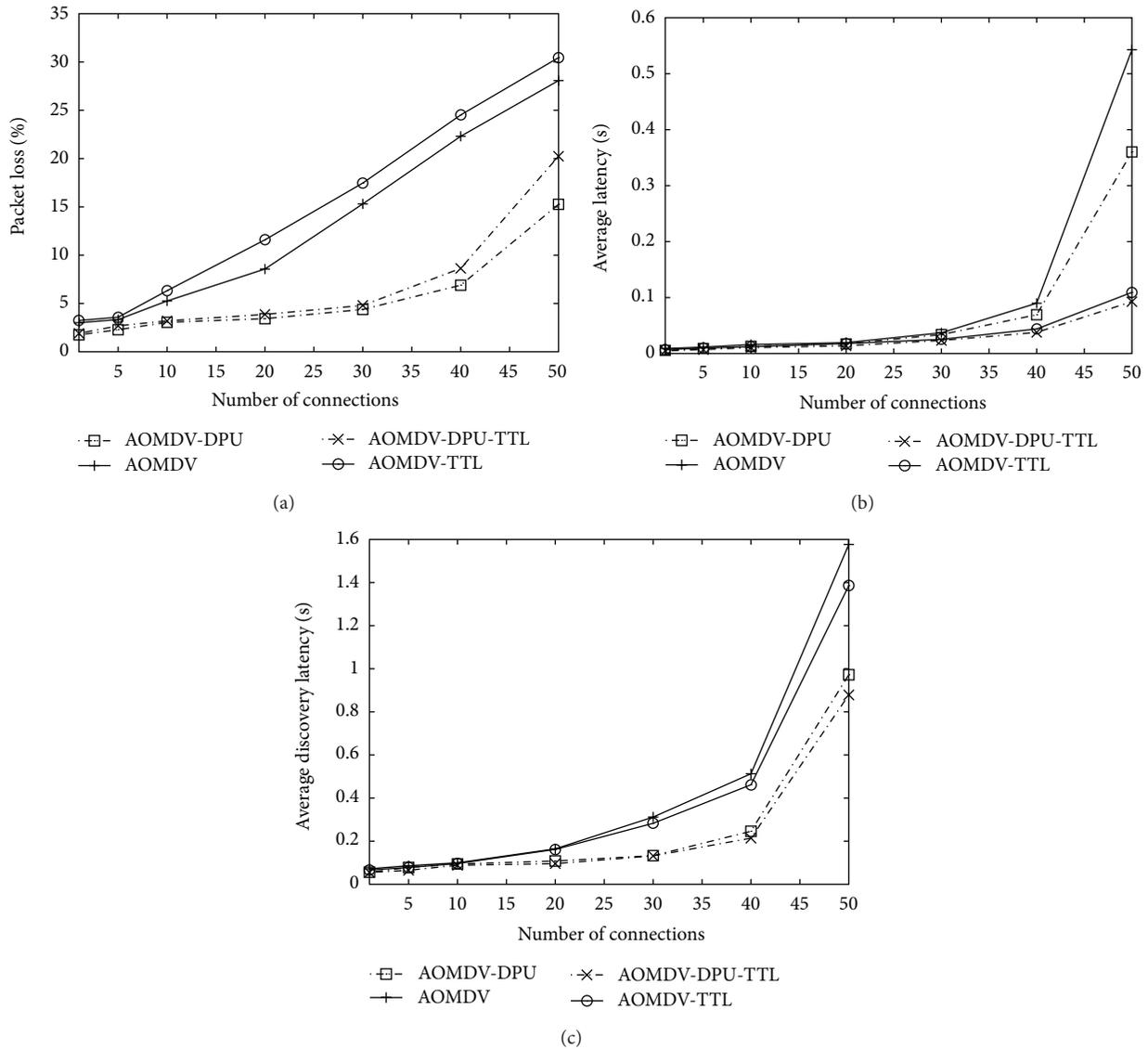


FIGURE 6: Performance with varying number of connections (a) packet loss, (b) average packet delivery latency, and (c) route discovery latency.

the secondary routes, the use of lower frequency HELLO_{LU} broadcasts decreases the collisions with data packets, and packet salvaging increases the packet transmission efficiency. These features enable the AOMDV-DPU scheme to deliver more packets in comparison to AOMDV.

Figure 5(b) shows that the average end-to-end delay experienced by the packets at higher node speeds is up to 150 ms less in AOMDV-DPU as compared to the AOMDV, which can be attributed to the AOMDV-DPU features discussed in the previous paragraph. The difference in delay for both schemes is less than 50 ms for delay-sensitive traffic because the TTL is fixed at 250 ms.

Figure 5(c) illustrates that the average route discovery latency (D_{RD}) increases with node speed. The value of D_{RD} is lower by at least 100 ms for AOMDV-DPU compared to AOMDV scheme for both types of traffic and the difference in their values increases with node speed. This is due to

the combination of increased route lifetime achieved by the LPU process, maintenance of secondary routes, and packet salvaging. Also the routing overhead is higher in AOMDV due to HELLO messages, which increases the collisions in the network.

4.3. Varying Connections. Figure 6 shows the performance of the AOMDV-DPU and AOMDV schemes for both traffic types for varying number of connections. The average node speed and packet generation rate are constant at 5 m/s and 2 packets/s for each connection. Here the offered load varies from very light (i.e., 8 kb/sec for 1 connection) to heavy (i.e., 400 kb/s for 50 connections).

Figure 6(a) shows that the PLR increases with number of connections. AOMDV-DPU achieves substantially lower PLR than AOMDV for both types of traffic, especially when

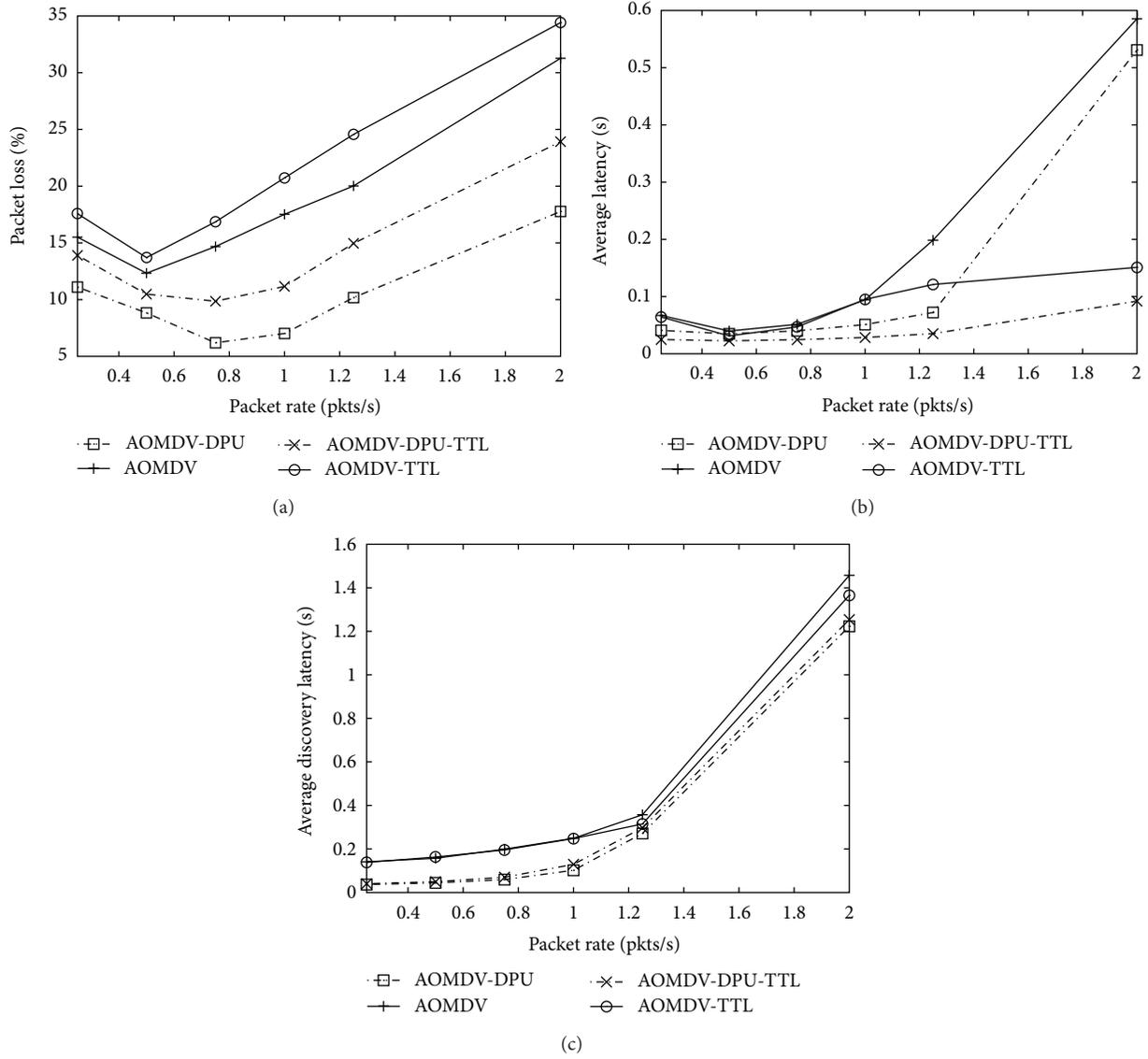


FIGURE 7: Performance with varying packet rate (a) packet loss, (b) average packet delivery latency, and (c) route discovery latency.

the number of connections is more than 10. While the AOMDV-DPU scheme achieves less than 5% PLR for up to 30 connections, the corresponding PLR for the AOMDV scheme increases to 15%. PLR achieved by both schemes increases more rapidly for more than 30 connections due to higher offered load.

Figure 6(b) shows that the average packet delivery latency of both schemes is lower than 50 ms for up to 30 connections. Though average latency of both schemes for delay-agnostic traffic increases rapidly for more than 30 connections, the AOMDV-DPU scheme experiences lower latency. Note that the delay-sensitive traffic experiences lower delay than TTL value of 250 ms.

Figure 6(c) shows that the D_{RD} for both schemes is about 100 ms for up to 10 connections. For more than 10 connections, the AOMDV-DPU scheme achieves lower D_{RD} than AOMDV and their difference increases with the number of connections. When the traffic in the network increases,

HELLO broadcasts become abundant and collisions increase in AOMDV.

4.4. Varying Packet Rates. In this set of experiments, the packet generation rate varies from 0.25 to 2 packets/s for each connection, the average node speed is 5 m/s, and 50 source destination pairs are activated in order to compare the results to those shown in [7]. The minimum offered load is 50 kb/s and the maximum load is 400 kb/s. As shown in Figure 7(a), the AOMDV-DPU scheme achieves significantly lower PLR than the AOMDV scheme for all the packet generation rates with maximum difference of 15% and 12% at 2 packets/s for the delay-agnostic and delay-sensitive traffic, respectively. For the proposed scheme, the PLR is relatively high at low packet generation rates because the HELLO packets are also generated at a low rate and therefore tracking SR nodes becomes difficult. As a result, the route breakage

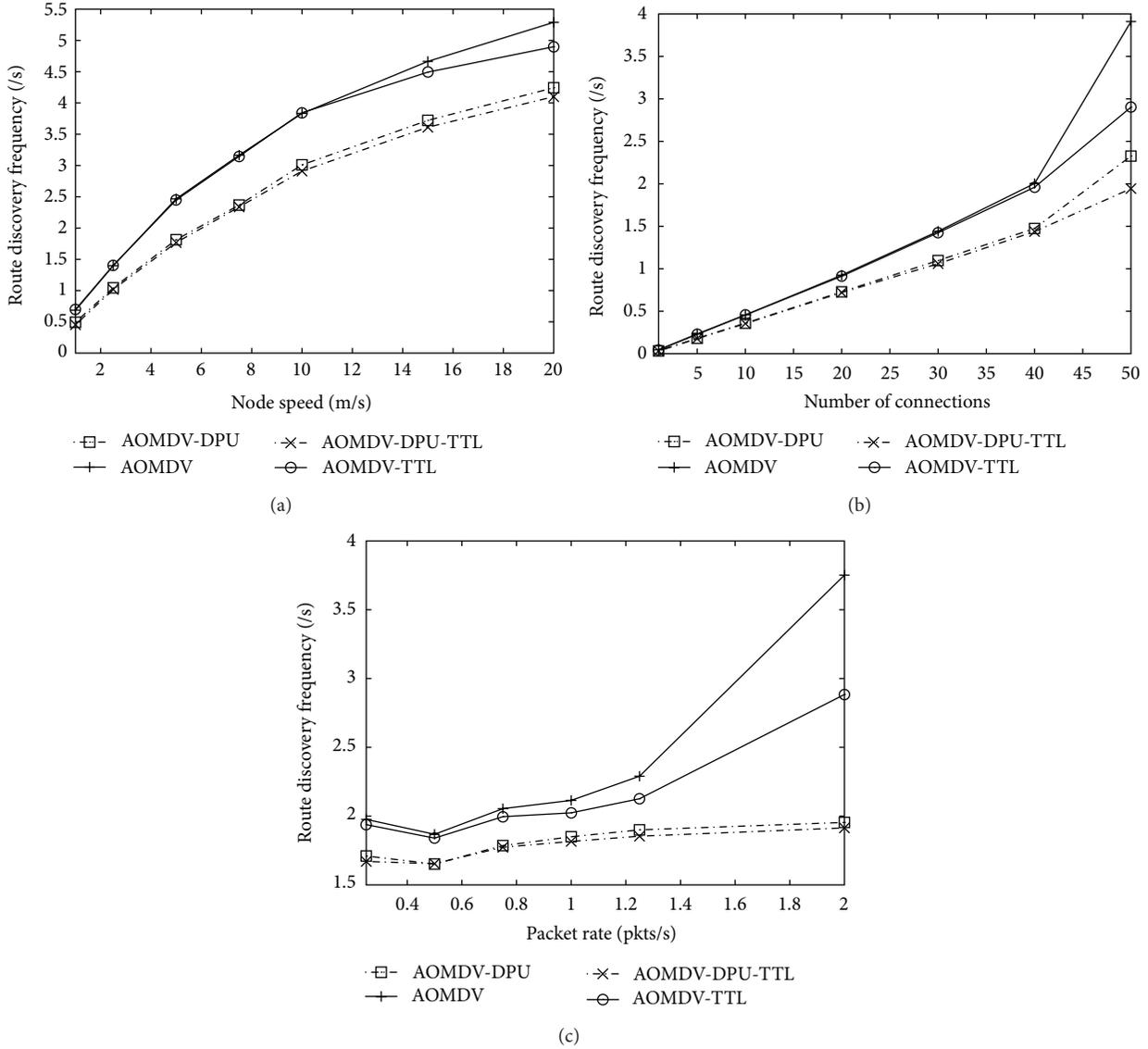


FIGURE 8: Route discovery frequency (F_{RD}): (a) F_{RD} versus node speed, (b) F_{RD} versus number of connections, and (c) F_{RD} versus packet generation rate.

probability increases. As the packet rate increases, the LPU process becomes more effective and PLR decreases. When the packet rate exceeds 0.5 packet/s for AOMDV and 0.75 packet/s for AOMDV-DPU scheme, the PLR starts increasing correspondingly.

Figure 7(b) shows the performance of average packet delivery latency with varying packet generation rate. It generally increases when packet rate exceeds 0.75 packet/s since more packets lead to more collisions and therefore retransmissions at the MAC. At 1.25 packets/s, AOMDV-DPU shows a considerably lower latency of 67 ms compared to the latency of 209 ms achieved by AOMDV. Our scheme benefits from the better route maintenance and optimization made to the $HELLO_{LU}$ messages. Beyond 1.25 packets/s, the latency increases significantly due to higher offered load of up to 400 kb/s and a multihop network.

Figure 7(c) shows the variation of D_{RD} with respect to the packet generation rate. The AOMDV-DPU scheme achieves lower D_{RD} compared to AOMDV. It increases rapidly when the packet rate exceeds 1.2 packets/s because RREQ and RREP messages experience higher interference due to a higher offered load in the network.

4.5. Overall Route Discovery Frequency (F_{RD}). The F_{RD} indicates the total number of RREQ packets generated per second by all the sources to find new routes. Figures 8(a), 8(b), and 8(c) illustrate F_{RD} variation for different network parameters, that is, average node speed, number of connections, and packet generation rate, respectively. We observe that the AOMDV-DPU consistently shows a lower value of F_{RD} than AOMDV for both traffic types. This is because AOMDV-DPU, unlike AOMDV, strengthens the primary route and

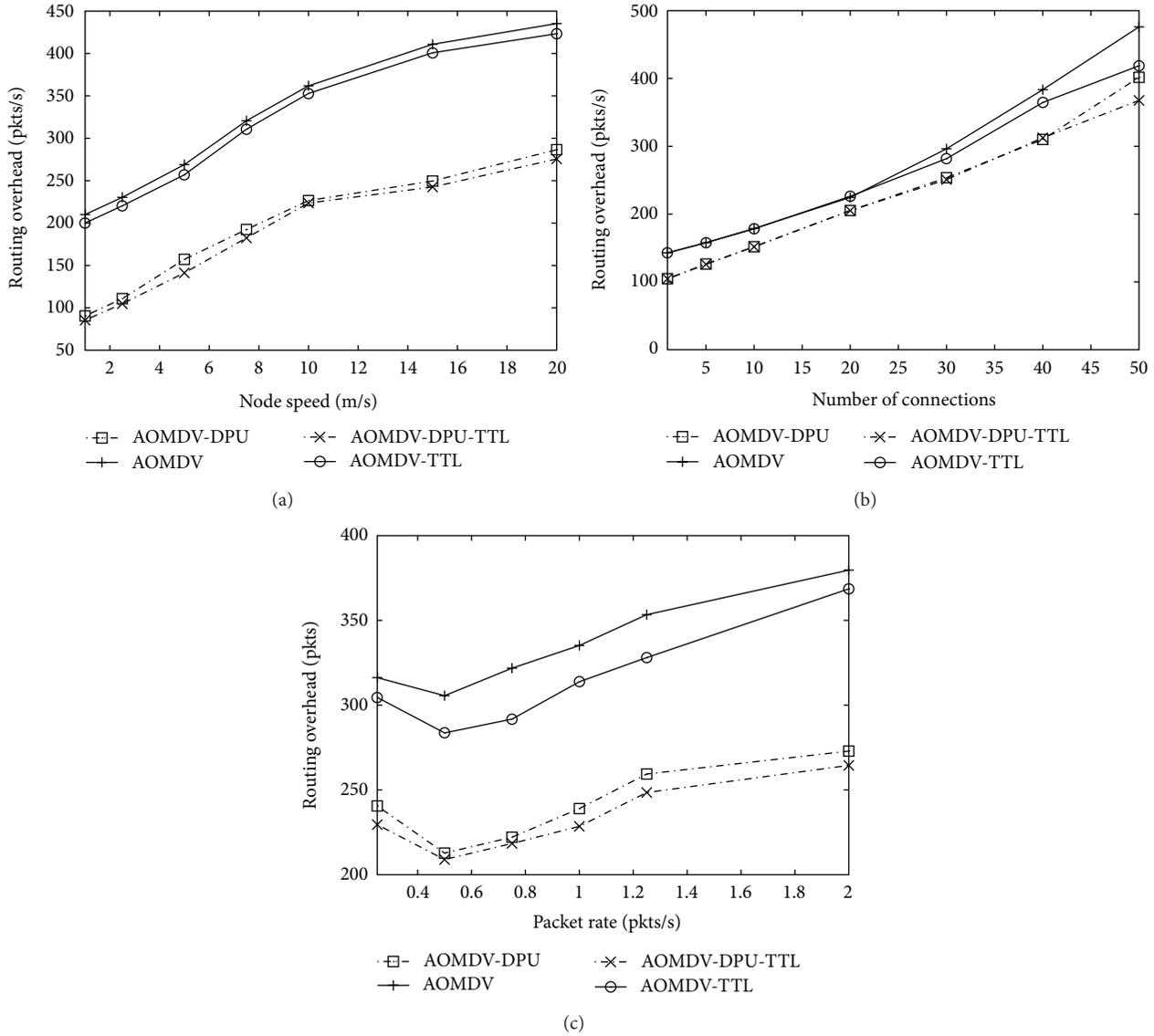


FIGURE 9: Overall routing overhead (O_R): (a) O_R versus node speed, (b) O_R versus number of connections, and (c) O_R versus packet generation rate.

also maintains the secondary routes by using the *KA* packets. This increases the primary route lifetime and improves the probability of finding an active secondary route, when the primary route fails.

4.6. Overall Routing Overhead (O_R). Figures 9(a), 9(b), and 9(c) show the total O_R measured in terms of the number of routing control packets transmitted per second, for average node speed, number of connections, and packet generation rate, respectively. The AOMDV-DPU has a lower value of O_R compared to AOMDV due to the following reasons: (i) it reduces the number of HELLO_{LU} packets by limiting them locally along the primary routes compared to AOMDV where every node broadcasts HELLO packets periodically; (ii) the *LPU* and *KA* algorithms of AOMDV-DPU strengthen the primary route and secondary routes, respectively. As a result,

the frequencies of RREQ, RREP, and RERR messages are reduced considerably. For varying packet rate in Figure 9(c), the routing overhead for 0.25 packet/s is higher than 0.5 packet/sec because the HELLO packets are also generated at a low rate and therefore tracking *SR* nodes becomes difficult. As a result, the route breakage probability increases.

4.7. Performance for H.264 Video Transmission. In this section, we compare the performance of AOMDV-DPU with AOMDV for transmitting videos over the network. We have used the same network parameters as discussed in Table 2, except using a lower node density of 25 nodes. We used the JM14.2 reference source code of H.264/AVC [27, 28] for encoding Silent and Foreman CIF video sequences (with 352×288 pixel resolution @24 bits/pixel) at the bit rate of 256 kbps. These videos have different motion and texture

TABLE 4: Average PSNR (in dB) for Silent and Foreman video encoded at 256 kbps.

Routing scheme and node speed	Silent video sequence			Foreman video sequence		
	Number of connections					
	2	3	4	2	3	4
AOMDV-DPU @ 5 m/sec	31.4	28.4	24.2	30.2	27.6	23.7
AOMDV [7] @ 5 m/sec	28.1	24.5	20.1	27.4	23.8	19.5
AOMDV-DPU @ 10 m/sec	30.4	27.5	23.7	29.4	26.6	23.0
AOMDV [7] @ 10 m/sec	26.8	23.4	19.8	25.1	22.9	19.2
AOMDV-DPU @ 20 m/sec	28.9	25.9	22.3	28.0	25.3	21.6
AOMDV [7] @ 20 m/sec	25.3	21.6	19.8	23.8	21.1	18.6

contents. Other encoder parameters were frame rate of 30 frames/second, GOP size of 20 frames, GOP frame pattern: IDR B P . . B, two reference frames for motion estimation, 150 byte slice size, and a robust decoder with the error concealment scheme used in JM. The H.264/AVC video slices were aggregated into 600 byte packets, which were interfaced with EvalVid in ns-2 [29]. The time-to-live value for a video packet was 350 msec.

Table 4 shows the results for 2, 3, and 4 video sources (i.e., connections) at three different node speeds of 5 m/sec, 10 m/sec, and 20 m/sec. Note that these connections present a source load of 512 kbps for 2 connections (moderate), 768 kbps for 3 connections (heavy), and 1 Mbps for 4 connections (very heavy). We observe that the video PSNR decreases with the increase in the number of connections and node speed. However, our scheme achieves approximately 3 dB or higher video PSNR gain over AOMDV for each case. Note that a 3 dB PSNR improvement generally corresponds to a twofold increase in video bit rate.

5. Conclusion

We introduced a robust AOMDV-DPU routing scheme for MANET with the following features: (i) a routing metrics based on the combination of minimum hops and RSSI provided robust and reliable routes; (ii) *local path update* mechanism strengthened the route, reduced the route breakage frequency, and increased the route longevity; (iii) *keep alive* mechanism for secondary route maintenance enabled smooth switching between routes and reduced the route discovery frequency; (iv) *packet salvaging* scheme improved packet delivery in the event of a route breakage; and (v) reduced HELLO packet overhead was achieved by controlling the HELLO packet frequency.

AOMDV-DPU scheme achieved significantly higher throughput, lower end-to-end packet delay, lower routing overhead, and lower route discovery frequency and latency compared to AOMDV for the delay-sensitive as well as delay-agnostic traffic under varying node speeds, number of sources, and traffic load conditions. For H.264 video traffic, the AOMDV-DPU achieved approximately 3 dB or higher PSNR gain, which is equivalent to sending about two times the data rate on the channel.

Our AOMDV-DPU scheme provides the scope for quality of service support to real-time video applications. It can also be extended to support multichannel MAC schemes where primary routes use channels with more idle time and secondary routes use channels with less idle time. The path selection and route update algorithms could be further enhanced by considering other network-based metrics.

Acknowledgments

This work was supported in part by the U.S. Department of Defense under Grants nos. FA8750-08-1-0078 and FA8750-11-1-0048. The study was approved for public release; Distribution Unlimited: 88ABW-2012-6578, 18 December 2012. The authors would also like to acknowledge Abhishek Madpur, Sandep Nanda, and Seethal Paluri at San Diego State University for their support in running the simulations and H.264 video coding.

References

- [1] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, 1994.
- [2] T. Clausen and P. Jacquet, "Optimized link state routing (OLSR) RFC," RFC 3626, 2003, <http://www.ietf.org/rfc/rfc3626.txt>.
- [3] J. Tsai and T. Moors, "A review of multipath routing protocols: from wireless ad hoc to mesh networks," in *Proceedings of the ACoRN Early Career Researcher Workshop on Wireless Multihop Networking*, Australia, 2006.
- [4] C. Mbarushimana and A. Shahrabi, "Comparative study of reactive and proactive routing protocols performance in mobile ad hoc networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINAW '07)*, pp. 679–684, May 2007.
- [5] C. Perkins, E. Belding Royer, and S. R. Das, "Ad-hoc on-demand distance vector (AODV) routing," RFC 3561, 2003, <http://www.ietf.org/rfc/rfc3561.txt>.
- [6] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, ch 5, Kluwer Academic Publishers, 1996.

- [7] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.
- [8] S. Tang, B. Zhang, M. Watanabe, and S. Tanaka, "A link heterogeneity-aware on-demand routing (LHAOR) protocol utilizing local update and RSSI information," *IEICE Transactions on Communications*, vol. 88, no. 9, pp. 3588–3597, 2005.
- [9] J. Novatnack, L. Greenwald, and H. Arora, "Evaluating ad hoc routing protocols with respect to quality of service," Tech. Rep. DU-CS-04-05, Department Computer Science, Drexel University, Philadelphia, Pa, USA, 2004.
- [10] S. J. Lee and M. Gerla, "AODV-BR: backup routing in ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1311–1316, Chicago, Ill, USA, September 2000.
- [11] H. U. Rehman and L. Wolf, "Performance enhancement in AODV with accessibility prediction," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, pp. 1–6, October 2007.
- [12] A. Mtibaa and F. Kamoun, "MMDV: multipath and MPR based AODV routing protocol," in *Proceedings of the IFIP 5th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net '06)*, pp. 137–144, 2006.
- [13] X. Li and L. Cuthbert, "Node-disjointness-based multipath routing for mobile ad hoc networks," in *Proceedings of the 1st ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '04)*, pp. 23–29, New York, NY, USA, October 2004.
- [14] S. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," 2001, <https://citeseer.ist.psu.edu/lee01split.html>.
- [15] G. Koltsidas, S. Karapantazis, G. Theodoridis, and F. N. Pavlidou, "A detailed study of dynamic manet on-demand multipath routing for Mobile Ad hoc Networks," in *Proceedings of the 4th IEEE and IFIP International Conference on Wireless and Optical Communications Networks (WOCN '07)*, July 2007.
- [16] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 270–280, April 2003.
- [17] S. Yin and X. Lin, "MALB: MANET adaptive load balancing," in *Proceedings of the IEEE 60th Vehicular Technology Conference (VTC '04)*, pp. 2843–2847, September 2004.
- [18] M. Tauchi, T. Ideguchi, and T. Okuda, "Ad-hoc routing protocol avoiding route breaks based on AODV," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05)*, Washington, DC, USA, January 2005.
- [19] S. Crisóstomo, S. Sargento, P. Brandão, and R. Prior, "Improving AODV with preemptive local route repair," in *Proceedings of the International Workshop on Wireless Ad-Hoc Networks*, pp. 223–227, June 2004.
- [20] R. S. Chang and S. J. Leu, "Long-lived path routing with received signal strength for ad hoc networks," in *Proceedings of the 1st International Symposium on Wireless Pervasive Computing*, pp. 1–6, January 2006.
- [21] P. Sambasivam, A. Murthy, and E. Belding-Royer, "Dynamically adaptive multipath routing based on AODV," in *Proceedings of the IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net '04)*, Bodrum, Turkey, June 2004.
- [22] M. Khazaei and R. Berangi, "A multi-path routing protocol with fault tolerance in mobile ad hoc networks," in *Proceedings of the 14th International CSI Computer Conference (CSICC '09)*, pp. 77–82, October 2009.
- [23] A. Chintawar, M. Chatterjee, and A. Vidhate, "AOMDV-APLP: an enhanced algorithm with accessibility and link breakage prediction," in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology (ICWET '11)*, pp. 841–845, February 2011.
- [24] S. V. Mallapur and S. Terdal, "Enhanced ad-hoc on demand multipath distance vector routing protocol (EAOMDV)," *International Journal of Computer Science and Information Security*, vol. 7, no. 3, pp. 166–170, 2010.
- [25] X. Chen, H. M. Jones, and D. Jayalath, "Channel-aware routing in MANETs with route handoff," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 108–121, 2011.
- [26] N. S. Kulkarni, B. Raman, and I. Gupta, "On demand routing protocols for mobile ad hoc networks: a review," in *Proceedings of the IEEE IAAC*, pp. 586–591, Patiala, India, March 2009.
- [27] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [28] H.264/AVC Reference Software JM14.2, ISO/IEC Standard, <http://iphome.hhi.de/suehring/tml/download/>.
- [29] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid—a framework for video transmission and quality evaluation," in *Proceedings of the International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, pp. 255–272, 2003.

