Bypass AODV: Improving Performance of Ad Hoc On-Demand Distance Vector (AODV) Routing Protocol in Wireless Ad Hoc Networks

Ahed M. Alshanyour
King Fahd University of Petroleum and Minerals
Computer Engineering Department
Dhahran-Saudi Arabia
shanyour@kfupm.edu.sa

Uthman Baroudi
King Fahd University of Petroleum and Minerals
Computer Engineering Department
Dhahran-Saudi Arabia
ubaroudi@kfupm.edu.sa

ABSTRACT

Bypass-AODV, a local recovery protocol, is proposed to enhance the performance of AODV routing protocol by overcoming several inherited problems such as unnecessary error recovery invocations, newly non-optimal reconstructed routes, high packet drop ratios, and high routing overheads. Bypass-AODV uses cross-layer MAC-notification to identify mobility-related link break, and then setup a bypass between the broken-link end nodes via an alternative node while keeps on the rest of the route. Therefore, Bypass-AODV enhances resource utilization by avoiding unnecessary error recovery cycles and consequently increases the network throughput. On the other hand, Bypass-AODV enhances route reliability; it avoids dropping packets by transmitting them over the constructed bypass. The simulation results show that when running 1-TCP connection, Bypass-AODV performs better than AODV. In particular, this behavior is rapidly changed with increasing the physical distance between the TCP connection end nodes beyond 2 hops. For example, when number of hops is equal to 6, goodput is enhanced by more than 100% compared to AODV for a 1-TCP connection and about 24% for multiple TCP connections. Further, the ratio of packet drop is reduced from 16% to 2%. Moreover, considering the hop count, the Bypass-AODV shows less sensitivity to the ongoing number of TCP connections.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols- Routing protocols.

General Terms

Algorithms, management, performance, reliability, local recovery

Keywords

Bypass routing, Bypass-AODV, Ad hoc networks, reliable routing, Cross-layer design.

AMBI-SYS 2008, February 11-13, Quebec, Canada

DOI 10.4108/ICST.AMBISYS2008.2895

1. INTRODUCTION

An ad hoc network is formed by a group of mobile devices that communicates without depending on any fixed infrastructure. In such environment, Neighbor nodes communicate directly with each others while communication between non-neighbor nodes performed via the intermediate nodes which act as routers. Furthermore, Because of node mobility and power limitations, the network topology changes frequently. Therefore, efficient routing protocols are necessary to organize and maintain communication between the nodes. Routing protocols for ad hoc networks can be categorized as proactive and reactive routing protocols. In proactive routing protocols [7], routing information about every possible destination is stored at each node. Any change in network topology trigger 'propagating' updates throughout the network in order to maintain a 'consistent' network view (heavy bandwidth utilization). On the other hand, reactive routing protocols [4][8] try to utilize network bandwidth by creating routes only when desired by the source node. Once a route has been established, it is maintained by some route maintenance mechanism as long as it is needed by the source node.

Wireless networks are prone to route breaks result from different sources such as: node mobility, signal interference, high error rates, fading environment, and packet collision. Mobility produces an actual route breaks while other sources produce a factious route breaks. MAC protocol, IEEE 802.11[2], translates unsuccessful packet transmission[1] as link failure. It has no ability to distinguish whether unsuccessful transmission has occurred due to mobility or something else. Routing protocols have one of the following three choices to deal with such link error:

- Do nothing, the source node will timeout waiting for a positive acknowledgment from the destination. Then after timeout occurrence the source node will start a new route discovery cycle.
- Report the error to the source node immediately by propagating a route error (RERR) message. The source node may then choose to re-initiate a route discovery for that destination if a route is still desired.
- Invoke some local recovery scheme to bypass the link in error. Some schemes concentrate on utilizing network resources by reducing the frequency of flooding such as

[1] In current IEEE 802.11 standards, a node will try to send a packet up to 7 times for smaller packets and 4 times for longer packets before deciding the transmission can't be completed.

multi-path routing [5][6][9][12]. Others [1][8][11] concentrate on reducing packet drop rather than utilizing network resources such as local repair schemes

Multipath routing provides fault-tolerance; it caches multiple routes to destination in a single route discovery cycle. When a link breaks, an alternative route can be used to route the packets. Although multipath schemes utilize network resources but they incur more packet drop and delay due to their dependency on stale routes. On other hand, local repair schemes introduce a special route maintenance method to repair broken routes. In Ad-hoc On Distance Vector (AODV) routing protocol [8], when a link and accordingly the route break, the upstream node decides either to repair the route via a limited broadcast or to send a route error (RERR) message to the source node based on its distance from the destination node. To repair the broken route, if the node is close to the destination, it sends a route request (RREQ) message with limited time-to-live (TTL) value. Otherwise, RRER message is propagated to the source node to start new route discovery process. After starting the repair process, the node waits for a discovery period. If the repair attempt fails, a RRER message is sent back to the source node. Otherwise, the node updates its routing entry. Local repair schemes are too bandwidth consumers, since even with a limited broadcast, flooding can deliver the RREQ messages to a large number of nodes, leading to high routing overheads. Additionally, routes maintained by local repair schemes are no longer being the optimal routes further along in time. Finally, in AODV, local repair scheme lacks an efficient way to handle link breaks that are close to the source node, those route breaks are handled by propagating an RERR message to the source node to start a new route discovery cycle which leads to further packet drop and bandwidth consumption. Our simulation results show that more than 50% of route failures are close to the source side than the destination side.

The main contribution of this paper is to propose a new maintenance scheme, Bypass-AODV, which improves the performance of an existing on-demand routing protocols, specifically AODV. Bypass-AODV routing protocol reduces bandwidth consumption and increases the network throughput by increasing the route reliability for highly-mobile ad hoc wireless networks. The Bypass-AODV initially follows the route discovery mechanism of AODV. Then, in case of failure in the primary route, the proposed protocol uses cross-layer MAC-notification to identify mobility-related packet loss, then it setups a bypass between the node at which the route failure occurred and its previous successor via an alternative node while keeps on the rest of the route. The proposed mechanism can significantly reduce the routing overhead and lead to corresponding increase in throughput by avoiding unnecessary invocation of error recovery mechanism. Bypass construction is independent on the location of route failure and it increases the route reliability by decreasing the number of packet drops, packets can be salvaged by redirecting them over the bypass. The ns-2 simulator [2] is used to study the proposed protocol. For analyzing the performance of Bypass-AODV, we use different performance measures for ad hoc networks: goodput, packet delivery ratio, packet drop ratio and routing overhead. Results of extensive simulations show that Bypass-AODV enables fast recovery of broken routes, reduces the number of unnecessary route discoveries initiated by the source node and increases the delivery ratio while maintaining acceptable routing overhead.

The rest of the paper is organized as follows. We discuss related work in Section 2. In Section 3, we describe the Bypass-AODV mechanism. In Section 4, we present the simulation model used for performance analysis, and discuss the workload model and parameters used in the simulation. In section 5, the results of the performance evaluation are presented and discussed. Finally, conclusions are given in Section 6.

2. RELATED WORK

Several protocols implement solutions to the flooding problem in on-demand routing protocols by enhancing route recovery mechanisms [8][11].

In AODV with Backup Routing (AODV-BR) [11], nodes overhear route reply messages of their neighbors to create their own alternate routes to destination. When a node detects a broken route, it broadcasts the packet to its neighbors hopefully that one of them has a valid route to the destination and at the same time sends a RERR message to the source to initiate a route rediscovery. The reason for reconstructing a new route instead of continuously using the alternate path is to build a fresh and optimal route that reflects the current status of the network. AODV-BR concentrates on increasing route reliability by decreasing packet drop rates but it suffers from two main problems: stale routes and duplicate packet transmission.

In Neighborhood-aware Source Routing (NSR) protocol [10], each node has a partial topology that covers in addition to the 2-hop neighborhood, the links in requested paths to destinations. Link state information is maintained by broadcasting periodic HELLO messages. In case of route failure, an intermediate node tries to repair the route if either the link to the next hop has failed or the link headed by the next hop on the path to be traversed has failed. RERR message is propagated to the source node if an intermediate node uses a completely new route to destination or it has no alternate route to destination. HELLO messages in NSR incur excessive overhead to maintain the partial topology of the network. Additionally, stale route problem may affect the performance of NSR.

The Dynamic Source Routing protocol (DSR) is suitable for networks with relatively small diameters and in which the mobile nodes move at a moderate speed with respect to packet transmission latency [4]. It potentially cashes multiple routes to a destination and provides a route salvaging option that enables intermediate nodes to recover from route failure locally by searching for an alternate route. Even with successful salvaging, intermediate nodes immediately send an RERR message back to the source to notify it about route failure. Then the source node can check its cache for another valid route. If such route is found, route reconstruction does not need to be invoked. But if there are no additional routes to the destination in the source node's cache, route discovery must be reinitiated. DSR is not scalable to large networks. Additionally, when the failure occurs far away from the sender and close to the destination and no alternate routes are available, the fact that the packet succeeded in traversing most of the path is not exploited. This increases the overall packet delivery time and the network resources used by the routing protocol. Furthermore, DSR incurs more packet drop and delay due to its dependency on stale routes.

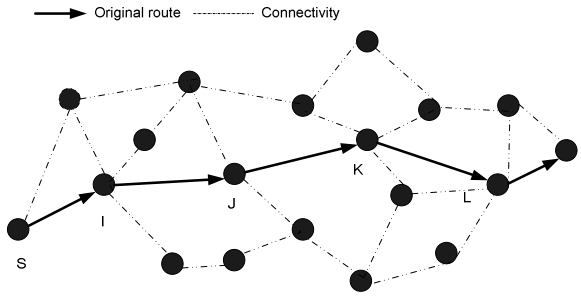
Bypass routing [1], a local recovery protocol that aims to reduce the frequency of route request floods triggered by broken routes through localizing the reaction to route failures using on-demand local recovery and a novel cache invalidation mechanism. The proposed mechanism uses link-state information to find a patch between one of the neighbors and a node along the route to the destination. This proposed mechanism is suitable for source routing protocols where complete route information are stored for each route entry. When a link between two nodes is broken, the node that detects the failure tries to patch the route by looking for a bypass route that connects the node with any of the downstream nodes of the broken route. If such route is unavailable, node triggers a local query to its neighbors hopefully that one of them has a valid route to any of the downstream nodes of the broken link. Bypass routing is an optimistic routing protocol assuming it can repair the route. If neither the intermediate node nor its neighbors has an alternate route, then bypass routing is equivalent to DSR but with further overhead and delay increase. Bypass routing does not propose any solution for the stale route problem exists in DSR. Furthermore, bypass routing is only applicable to on-demand routing protocols where complete route information is included in the transmitted data packet.

3. BYPASS-AODV

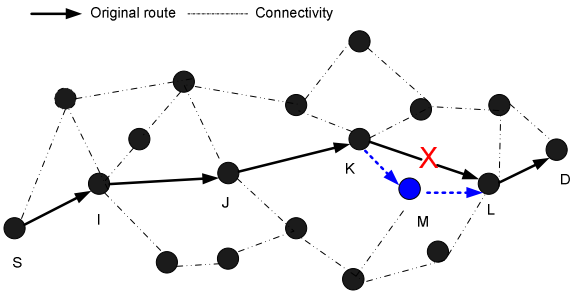
Bypass-AODV uses cross-layer MAC-notification to identify mobility-related packet loss then triggers the routing layer to start local repair, which allows upstream node of a broken link to do the repair by setting up a bypass between it and the downstream node via an alternative node. MAC-notification messages are used to distinguish between mobility-related packet loss and other sources-related (signal interference, high error rates, fading environment and packet collision) packet loss. Bypassing mechanism works with a specified *TTL* to limit the area of route search. Our simulation results show that to bypass a broken link, a *TTL* value of two is more than enough. Therefore, route bypassing will minimize routing overhead. In contrary to AODV local repair mechanism, Bypass-AODV has the ability to repair the broken route regardless of break location and consequently it will minimize packet losses. Packet losses occur when route bypassing does not work, for example, the distance between upstream and downstream nodes of a broken link is more than *2* hops or power depletion of the downstream node. In such cases, no bypass can be constructed with the downstream node and an RERR message after a timeout value is propagated toward the source node. Route bypassing results in an unnecessary increase in hop count metric. This increase depends on the frequency of route bypassing. To handle this issue, the bypassed-route is a temporary route which lasts for a period which is enough to guarantee that packets left their source node will reach their final destination.

3.1 An Illustration of Bypass-AODV

Figure 1-a gives a brief illustration of route bypassing in AODV. Initially, the flow from source S to destination D goes through I, J, K and L nodes. A link break between K and L will be detected by K. Then node K will initiate a limited route discovery cycle to search for a route that links it with node L (node K is a *bypass-source* and node L is a *bypass-destination*). Neighbors of node K will receive the route request message and rebroadcast it to their neighbors. Assuming the distance between *bypass-source* and *bypass-destination* did not change more than one hop; *bypass-destination* will receive the route request message and start unicasting a route reply message toward *bypass-source*. Figure 1-b shows a situation where the route reply message is unicasted to *bypass-source* via node M. Simulation results show that in most cases, *bypass-destination* receives the route request message directly from *bypass-source* which means that the detected route failure is a factious one. Factious failure results due to congestion. In such cases no need to bypass the original route.



a. - Routing example



b. Route maintenance using Bypass-AODV

Figure 1: Route maintenance example

3.2 Bypass-AODV Implementation

As mentioned above, Bypass-AODV uses two mechanisms that work together to allow efficient recovery from route failures. The next two subsections present these two mechanisms in detail.

3.2.1 Enhancing MAC Mechanism

One of the main problems in mobile ad hoc networks involves how to distinguish the cause of packet loss at the MAC layer and then correspondingly enforce the routing and transport layers to react properly. The MAC protocol translates unsuccessful packet transmission, packet loss, as link failure. It has no ability to distinguish whether packet loss has occurred due to mobility, signal interference, high error rates, fading environment or collision. To enhance the response of the MAC layer to packet losses, we propose a simple prediction mechanism, which considers any two successive packet losses as mobility-related failure. The modified-MAC layer protocol defines three different states (*ON, RETRY, OFF*) for the channel. The channel resides in *ON* state if it does not encounter any packet loss. It switched to the *RETRY* state if it encounters a packet loss and switched to the *OFF* state if it encounters two successive packet losses. The channel returns from *RETRY* state back to *ON* state if it encounters a successful transmission in the second retry. When channel enters *OFF* state it generates and sends a route failure notification (*RFN*) message to the routing layer. Figure 2 shows the state diagram for enhancing MAC layer reaction to packet

losses. Packet retransmission results in more delivery delay, but actually the cost of packet retransmission is negligible compared to the cost of new route discovery.

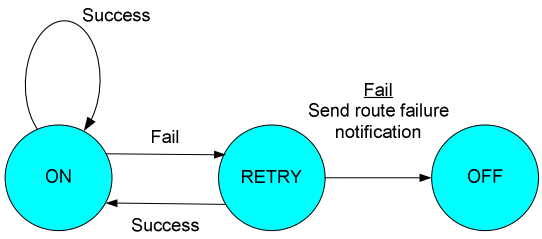


Figure 2: Channel state transition diagram

3.2.2 Route Bypassing

For implementation purposes, two types of RREQ and RREP messages are used. The first type is similar to that used in AODV, which are used for route discovery cycle initiated by the source node. The second type of RREQ and RREP messages are specifically named *bypass-RREQ* and *bypass-RREP*, which are distinguished by setting one of RREQ/RREP reserved bits. *Bypass-RREQ* and *bypass-RREP* are used for route bypassing purposes. To keep on AODV implementation, each node has in addition to its routing table a *bypass-routing-table*. *Bypass-routing-table* is used to store routing entries of bypassed-routes. When a node receives an RREQ or an RREP message, it directly checks its *bypass-routing-table* to invalidate the corresponding route entries.

In AODV, two route states are proposed *UP* and *DOWN*. In *UP* state, packets are transmitted while in *DOWN* state, the received packets are dropped. To implement the local repair mechanism, a *REPAIR* state is added. *REPAIR* state is proposed for handling route maintenance; the node that initiates a route maintenance process puts its route in *REPAIR* state and then it will buffer all incoming packets in addition to the dropped packet. For Bypass-AODV, two more states are added, *BYPASS* and *WAIT*. Router uses *BYPASS* state to indicate that the route is bypassed. While, source node use *WAIT* state to stop any further transmission over the bypassed route, destination node use *WAIT* state to transmit the last-transmitted-packet's acknowledgment. To handle the transition between different route states, two control bits may be added to the data packet's IP header for TCP applications or a control message with at least two control bits may be proposed for UDP applications (due to the absence of ACK data packets). Those two control bits are set to *00, 01, 10*, or *11*. In this paper, Bypass-AODV will be presented for TCP applications. The state diagrams for the proposed scheme are shown in Figure 3.

UP state

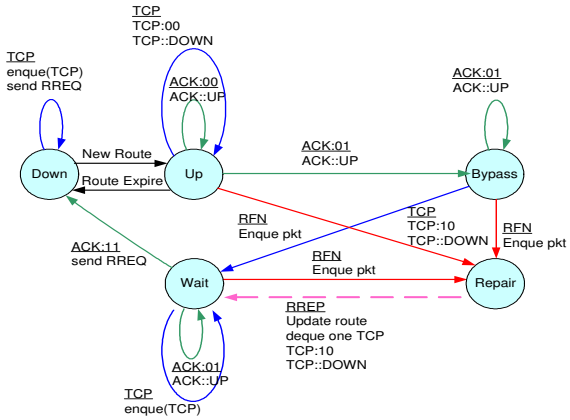
In this state, control bits of all transmitted packets are set to *00*.

DOWN state

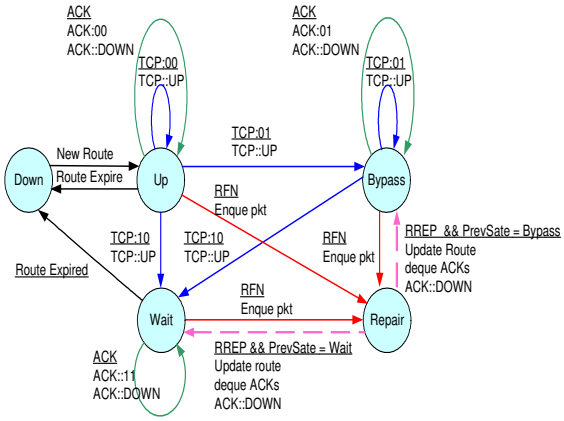
In this state, route is inactive. Upon receiving a new packet, source node will initiate a route discovery process.

REPAIR state

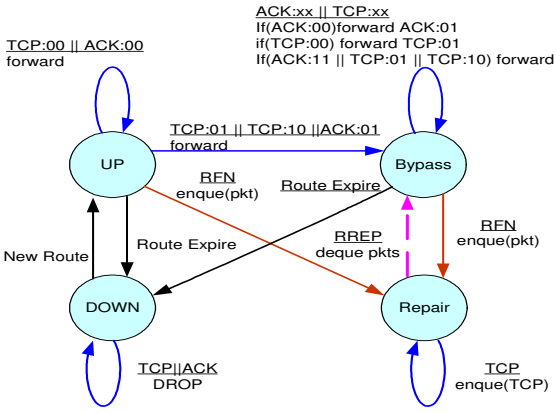
The router switches the route to this state as it receives the *RFN* message. At this state, router starts bypassing mechanism by broadcasting a *bypass-RREQ* message with a limited *TTL* value to discover a bypass to the broken downstream node. The *bypass-RREQ* message defines *bypass-source* and *bypass-destination*. Data packets received during this state are buffered in the router's buffers to transmit them later over the constructed bypass. Figure 3 (a), (b) and (c) show the transition from the *REPAIR* state to the other states for the source node, the destination node and the intermediate node respectively.



a. Route transition state diagram (source side)



b. Route transition state diagram (destination side)



c. Route transition state diagram (Intermediate node)

Figure 3: State Diagrams

Moreover, the proposed mechanism is enhanced by giving the router the ability to distinguish between false route failures and true ones. False route failure is detected by receiving a data packet or an *RREP* message directly from the *bypass-destination*. In such

situation, route is switched to the previous state and the *RREP* message is discarded. This enhancement is not shown in the state diagrams for drawing simplicity.

BYPASS state

There are two cases for switching the route to *BYPASS* state: router receives *bypass-RREP* message that has a number of hops equals *2* or a packet whose control bits are *01* or *10*. In *BYPASS* state, router uses the *bypass-route-table* to forward the received data packets. Control bits of the forwarded packets are set as shown in the corresponding state diagrams

WAIT state

WAIT state is a special state for source and destination nodes only. In source node side, *WAIT* state is used to prepare the source node for transmitting the last data packet over the bypassed route and buffer any subsequent packets received from the above layer. In destination side, *WAIT* state is used to indicate the reception of the last transmitted data packet.

The proposed mechanism

As *bypass-source* receives an *RFN* message, it attempts to repair the broken link by broadcasting a *bypass-RREQ* massage with *TTL* value of *2* to the downstream node (*bypass-destination*) and then switches the broken route to *REPAIR* state. In addition to IP addresses of *bypass-source and bypass-destination*, current *bypass-source's* sequence number and broadcast ID, the *bypass-RREQ* also contains the IP addresses of the primary route's end nodes. Intermediate nodes rebroadcast the *bypass-RREQ* and keep track of the received *bypass-RREQ*s by caching their source IP addresses and broadcast IDs. Multiple *bypass-RREQ*s with same source address and broadcast ID are discarded. *Bypass-RREP* is sent only by the *bypass-destination*. As soon as *bypass-destination* receives the *bypass-RREQ*, it adds to its *bypass-routing-table* a new routing entry, sets the corresponding route state to *BYPASS*, invalidates the same route entry in the original route table, and unicasts *bypass-RREP* message to *bypass-source*. Intermediate nodes that receive *bypass-RREP* add to their *bypass-routing-tables* new route entries for the broken route's source and destination nodes of instead of *bypass-source* and *bypass-destination* nodes. *Bypass-RREP* message is forwarded by intermediate nodes to *bypass-source* using routing entries available in the original routing tables. As bypass-*RREP* is received by *bypass-source*, it adds a new routing entry to its *bypass-routing-table*, sets its route state to *BYPASS*, and starts transmitting buffered packets over the constructed bypass.

4. SIMULATION ENVIRONMENT

We implement a simulation model in ns-2 [2] to evaluate the performance of TCP used in conjunction with the Bypass-AODV listed earlier. The distributed coordination function (DCF) defined in the IEEE 802.11 standard [3] is used at the MAC layer. The radio model is a ground radio propagation model with a minimum signal to noise ratio (SNR) equals to *10dB*, a nominal bit-rate of *2Mb/sec* and a nominal radio range of *250* meters. The performance metrics that we are interested in are:

- The routing overhead ratio, which is the ratio of the amount in bytes of control packets transmitted to the amount in bytes of data packets received.
- The packet drop ratio, which is the ratio of the amount in bytes of data packets dropped to the amount in bytes of data packets transmitted.
- The packet delivery ratio, which is the percentage of data packets received at the destinations as compared to the number of data packets generated by TCP sources.
- The absolute goodput of TCP, which is the number of sequenced bits that a TCP receiver received per second.
- The "goodput ratio" which is the TCP goodput observed with a Bypass-AODV strategy as compared with the standard AODV routing strategy.

The simulations region is 1500m x 500m. In each simulation-iteration, a random scenario is generated; a number of source-destination pairs are randomly chosen and TCP connections are established between the pairs. These TCP connections begin sequentially. The initiation instances of consecutive TCP connections are separated by 3 seconds. The simulation time lasts for 150 seconds and is then terminated. The simulation results reported in next Section represent the average results over 700 different scenarios. The results were collected as average values over 10 runs of each simulation setting. TCP New-Reno is used in all our simulations. The length of each TCP packet is 1060 bytes. In our simulations with mobile nodes, we use the random waypoint model to simulate node mobility. In random waypoint model, each node randomly selects a position, and moves toward that location with a speed between the minimum and the maximum speed. Once it reaches that position, it becomes stationary for a predefined pause time. After that pause time, it selects another position and repeats the process.

Since our objective is to study the Bypass-AODV routing protocol on both long (in terms of hop-count) and short TCP connections, we would need to keep the physical distance[2] between the source and the destination of a TCP connection relatively unchanged during a simulation run in order to classify the connection as either long or short. In our simulations, we thus make the TCP end nodes static, while all the other nodes are allowed to move in accordance with the mobility model.

5. PERFORMANCE EVALUATION

In our simulation, we evaluate the performance of Bypass-AODV compared to AODV routing protocol and then we study the impact of node density and mobility on Bypass-AODV compared to AODV.

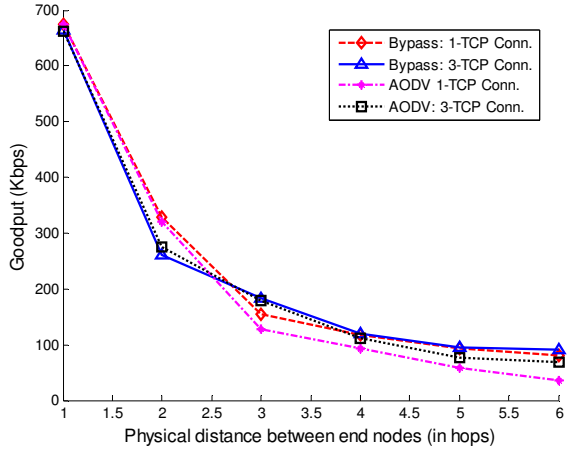
5.1 Impact of Distance between End Nodes

To evaluate the performance of Bypass-AODV compared to AODV, we placed *60* nodes in *1500m x 500m* region. The pause time was set to zero to keep on continuous mobility. The velocity for each node is uniformly distributed over [*0*, *20m/s*]. The physical distance between TCP connection end-nodes was varied between *1* and *6* hops.

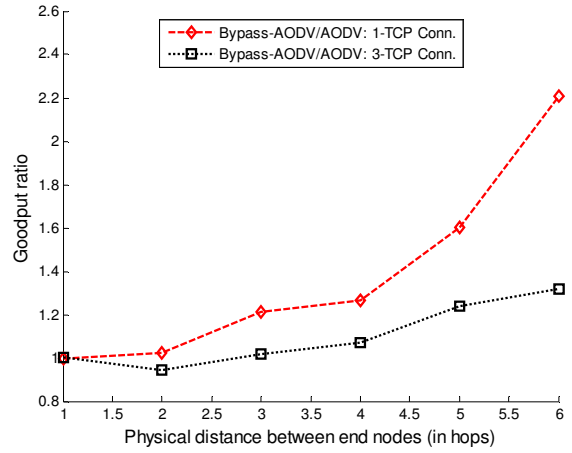
All the nodes in ad hoc network share the same transmission medium; if a node is transmitting, other nodes within a certain range of the transmitting node cannot transmit. There are two ranges defined by 802.11 MAC and used in our simulation: the

[2] The minimum distance between TCP connection end nodes in terms of the number of hops, assuming nodes use their maximum transmission range (250m).

transmission range and the sensing range. The transmission range is the maximum distance between two nodes such that the signal transmitted by one node can be received and decoded correctly by the other node. Sensing range is defined as the maximum distance between two nodes such that signal transmitted by one node can be received by the other node but cannot be decoded correctly. The sensing range is much larger than the transmission range. In 802.11 MAC the transmission range is defined to be 250m while the sensing range is assumed to be 550m. 802.11 MAC protocol ensures that while a node is transmitting, other nodes within the sensing range of that node cannot transmit.



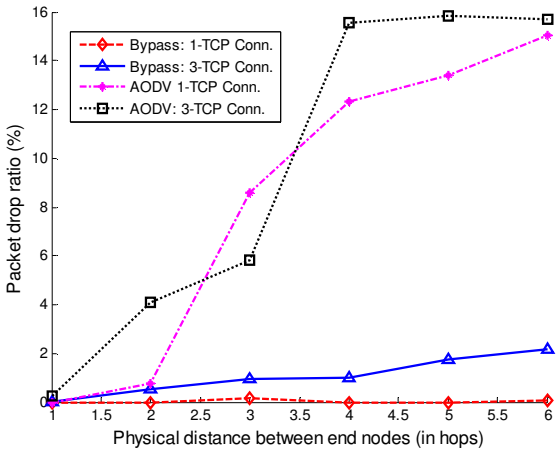
a. Goodput



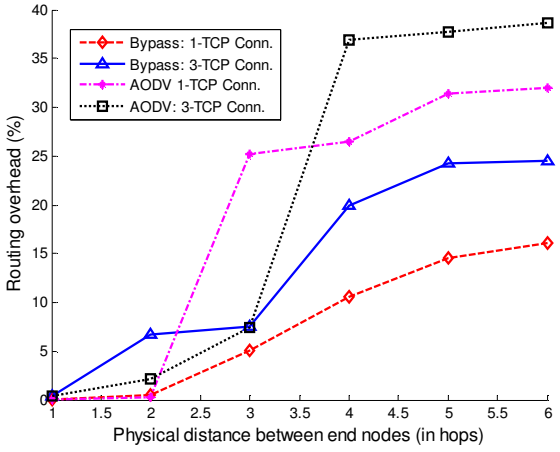
b. Goodput ratio

Figure 4: TCP Goodput and Goodput ratio

From Figure 4.a, for single TCP connection, bypass-AODV and AODV have similar TCP goodput when the two end nodes are close to each other. When the physical distance between the two end nodes is one hop, the two end nodes are in direct communication. Since the TCP connection end nodes are static, there is no possibility of link failure. Therefore, Bypass-AODV and AODV routing protocols are equivalent. As the physical distance between the two end-nodes becomes 2 hops, the two end nodes are communicated via an intermediate node. In such scenario, all communicating nodes are within the sensing range of each others and thus only one transmission is allowed at a given time. Therefore, any link failure is mobility-related. At this physical distance, Bypass-AODV starts showing slight enhancement for TCP goodput. In AODV such failure is always handled by dropping the buffered data packets and initiating a new route discovery process because link failure is close to the source than the destination. Whereas, Bypass AODV effectively minimizes packet drop by buffering the data packets for subsequent transmission after doing route bypassing. Figure 5.a shows how Bypass-AODV minimizes packet drop compared with AODV. But bypassed route is a temporary route that lasts for a period of time which is enough to forward the buffered packets and then a new route mechanism will start. Therefore, it is expected that routing overhead in Bypass-AODV will be increased compared to AODV as shown in Figure 5.b. from our simulation results, the maximum deviation from the average goodput were found to be less than 8%.



a. Packet drop ratio



b. Routing overhead

Figure 5: Packet drop ratio and routing overhead

When the physical distance between the TCP connection end nodes is greater than or equal to 3 hops, there is a possibility of simultaneous contention on the transmission medium; collision. Collision causes unsuccessful packet transmission. 802.11 MAC translates unsuccessful packet transmission into link failure.

Therefore, there is a need for an efficient MAC mechanism that distinguishes mobility-related failures from other sources-related failures. Existence of such mechanism will reduce the frequency of route mechanism invocation and correspondingly minimize routing overhead and packet drop. Bypass-AODV has such mechanism; cross-layer MAC-notification mechanism. Figure 4.a and Figure 4.b show a clear improvement in the TCP goodput and TCP goodput ratio respectively. Also, Figure 5.a and Figure 5.b show obvious enhancement in minimizing packet drop ratio and routing overhead. For example, at a physical distance of 6 hops, Bypass-AODV improves the TCP goodput from 36Kbps to 78Kbps which is more than 100% improvement.

For multiple TCP connections, the probability of unsuccessful packet transmission is increased even at short physical distances due to collisions and high interference levels, route bypassing in such scenarios minimizes packet drop ratios but produces high routing overheads. Figure 5.b shows how routing overheads is much larger than that produced by AODV but the ratio of packet drop is minimized. This behavior is rapidly changed when the physical distance between the communicating end nodes becomes more than 3 hops. For long TCP connections (in terms of hop count), all performance metrics are improved. For example, at physical distance of 6 hops, improvement in TCP goodput is reached 24% and packet drop ratio is less than 2%. Whereas packet drop ratio at the same physical distance is 16% in AODV.

5.2 Impact of Node Density

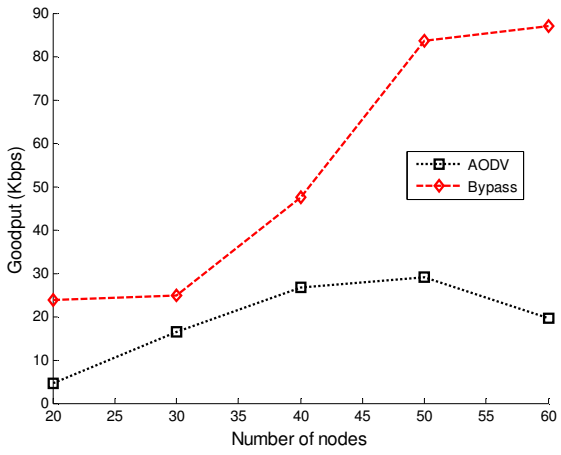
To evaluate the impact of node density on Bypass-AODV and compare it to AODV, number of nodes is varied between *20* and *60* nodes placed in *1500m x 500m* region. The pause time was set to zero to keep on continuous mobility. The velocity for each node is uniformly distributed over [*0*, *20m/s*]. The physical distance between TCP connection end-nodes was fixed at *6* hops.

Bypass-AODV shows continuous enhancement in performance as node density increases. Whereas, AODV shows enhancement in performance for medium node densities but at high densities, its performance starts decreasing. Figure 6-(a) and Figure 6-(b) show that Bypass-AODV performance outperforms AODV at different node densities. At low densities, node connectivity is low and the network may suffer from partitioning. Both routing protocols show low goodput. At medium densities, packet drop ratio for both routing protocols decreased but it is still large for AODV compared to Bypass-AODV. Finally, at high densities, packet drop ratio for bypass-AODV is relatively small while it starts increasing for AODV. This enhancement is attributed to success of cross-layer MAC-interaction mechanism in distinguishing related-mobility failures from other sources-related failures.

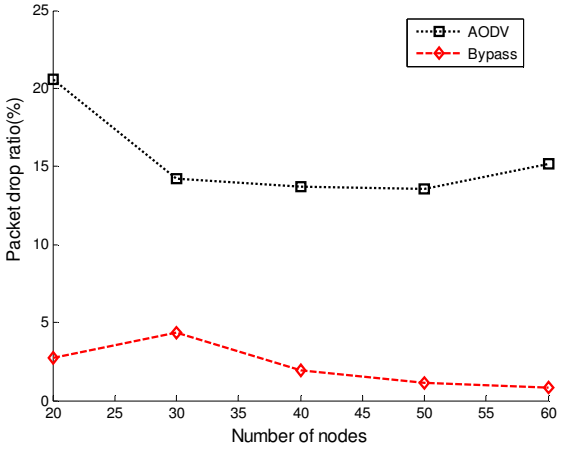
5.3 Impact of Node Mobility

we placed *60* nodes in *1500m x 500m* region. The pause time was set to zero to keep on continuous mobility. The maximum velocity for each node is varied between *0* and *30m/s*. The physical distance between TCP connection end-nodes was fixed at *6* hops.

The delivery ratio for AODV and Bypass-AODV are shown in Figure 7-(a) and Figure 7-(b). Both figures show that Bypass-AODV under different mobile speeds outperforms AODV routing protocol. For single TCP connection, Bypass-AODV delivery ratio is almost converging to 100%. However, AODV routing protocol starts delivering at high percentage of the original data packets but it drops at high speeds. Moreover, for multiple TCP connections, this trend becomes very clear with even lower delivery ratio at slow speeds. These findings illustrate the ability of Bypass-AODV to minimize packet drop by buffering data for subsequent transmission over the bypassed route. Whereas, AODV drops data packets when its repair mechanism not working.



a. Goodput



b. Packet drop ratio

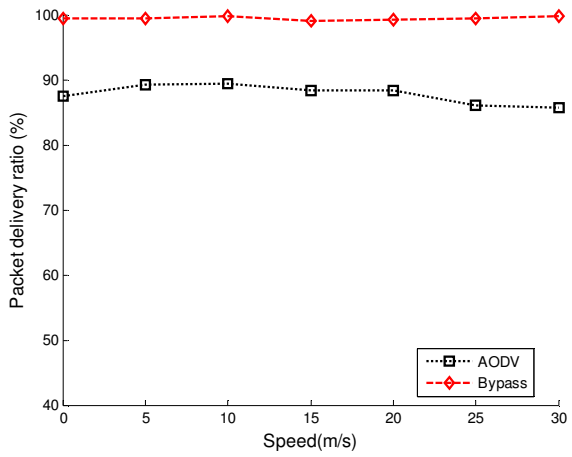
Figure 6: Delivery ratio vs. node density, number of TCP connections=1, physical distance between end nodes=6 hops, 1500mx500m region, and pause time=0.

6. CONCLUSION

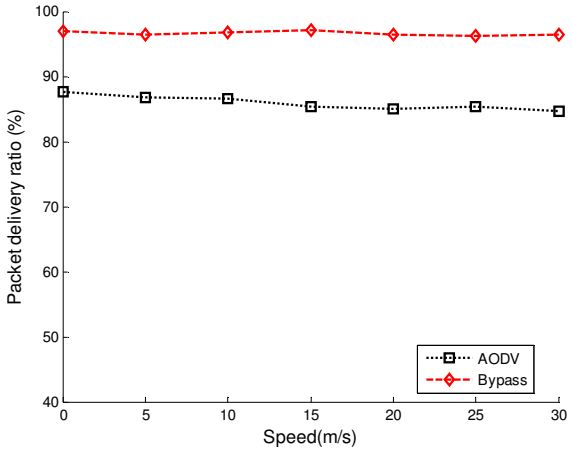
In this paper, Bypass-AODV scheme is proposed to improve an AODV routing protocol. Bypass-AODV uses a specific strategy of cross-layer MAC-interaction to identify mobility-related link breaks, and then setup a bypass between the broken link end nodes via an alternative node. By restricting the bypass to a very small topological radius, routing overheads are minimized considerably.

Simulation results show clear enhancement in route reliability by reducing the frequency of route maintenance and the amount of packet drop. By comparing the performance of Bypass-AODV

with AODV, results show that for long TCP connection (in term of hop count), Bypass-AODV enhances the TCP goodput by more than 100% for single TCP connection and more than 24% for multiple TCP connections. Further, the percentage of packet drop is reduced considerably. Furthermore, Bypass-AODV performance outperforms AODV for different node densities. Moreover, Bypass-AODV goodput is insensitive to the change in mobile speeds. This feature makes the proposed routing protocol very attractive to VANET applications. As a future work, we are planning to investigate the impact of different mobility models on the performance of Bypass-AODV. Additionally, we are planning to design more efficient MAC-detection algorithm that can distinguish mobility-related packet loss from other sources-related packet loss.



a. Number of TCP connections=1



b. Number of TCP connections=3

Figure 7: Delivery ratio vs. mobility, physical distance between end nodes=6 hops, 60 nodes, 1500mx500m region, and pause time=0.

7. ACKNOWLEDGMENTS

The authors acknowledge King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia for its support.

8. REFERENCES

[1] Cigdem, Sengul and Robin, Kravets. 2006. Bypass routing: An on-demand local recovery protocol for ad hoc networks. Ad Hoc Networks. 4, 3 (January 2006), 380-397.

[2] Fall, K. and Varadham, K. 1997. The ns manual http://www.isi.edu/nsnam/ns/ns-documentation.html/.

[3] IEEE Standards Department. 1997. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11.

[4] Johnson, D. B., Maltz, D. A., Hu, Y. C. and Jetcheva, J. G. 2002. The dynamic source routing protocol for mobile ad hoc networks (DSR). *Draft-ietf-manet-dsr-07.txt*.

[5] Marina, M K. and Das, S. R. 2001. On-demand multipath distance vector routing in ad hoc networks. In Proceedings of IEEE International Conference on Network Protocols (ICNP), 14—23.

[6] Nasipuri, A., Castañeda, R. and Das, S. R. 2001. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. ACM/Baltzer Mobile Networks and Applications (MONET) Journal-6, 4, 339—349.

[7] Perkins, C. E and Bhagvat, P. 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. Conference on Communications architectures, protocols and applications (SIGCOMM '94), 24, 4 (October 1994), 234-244.

[8] Perkins, C. E., Royer, E. M. and Das, S. R. 1999. Ad Hoc On-Demand Distance Vector Routing. IETF Internet Draft. http://www.ietf.org/internet-drafts/draft-ietf-manetaodv-03.txt.

[9] Raju, J. and Garcia-Luna-Aceves, J. J. 1999. A new approach to on demand loop-free multipath routing. In IEEE International Conference on Computer Communications and Networks (ICCCN).

[10] Spohn, M. and Garcia-Luna-Aceves, J. J. 2001. Neighborhood aware source routing, in Proceedings of 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc).

[11] Toh, C. K. 1997. Associativity-based routing for ad hoc mobile networks. *Wireless Personal Communications Journal, Special Issue on Mobile Networking & Computing Systems*, 4, 2 (March 1997), 103–109.

[12] Ye, Z., Krishnamurthy, S. V. and Tripathi, S. K. 2003. A framework for reliable routing in mobile ad hoc networks. In Proc. IEEE INFOCOM.