

# Modified route-maintenance in AODV Routing protocol using static nodes in realistic mobility model

Rakesh kumar<sup>1</sup>, Siddharth Kumar<sup>2</sup>, Sumit Pratap Pradhan<sup>3</sup>, Varun Yadav<sup>4</sup>

Department of Computer Science and Engineering,  
Madan Mohan Malaviya Engineering College, Gorakhpur, India

## ABSTRACT:

One of the popular wireless network architectures is Mobile Ad Hoc Network (MANET). It can be easily deployed in every environment, without any underlying backbone and infrastructure support. MANET is a wireless network that consists of nodes which are mobile in nature. Each node in the network acts as a router which provides the flexibility to the topology of the network.

In this paper, an extended AODV routing protocol is proposed for Ad-hoc networks which typically suits to resolve the realistic model problems. This proposed protocol may improve the performance of regular AODV routing protocol. It has all the features of AODV routing protocol, as it follows all the steps of the route discovery algorithm of AODV routing protocol. The main difference arises when it comes to the route maintenance approach. We assume that the network is also consist of the static nodes, which helps to recover the path while route maintenance. An assumption reveals that it may increase the efficiency of regular AODV routing protocol in terms of the metrics: End-to-end delay, Packets dropped, Routing overhead.

**Keywords:** AODV (Ad-hoc On-demand Distance Vector), ROUTE-REQ (Route Request), ROUTE-REPLY (Route Reply), ROUTE-ID (Route Identification), ROUTE-ERR (Route Error).

## 1. INTRODUCTION:

The idea of ad-hoc networking is gaining popularity due to sudden increase in the mobile computers such as laptops and palmtops. Its minimal configuration, absence of infrastructure and quick deployment make Ad Hoc Networks convenient for emergency operations, especially for military operations. Since host nodes are mobile in nature, so it causes frequent and unpredictable topological changes in the network, dealing with the formation and maintenance of Ad Hoc Network is very difficult as compared to the wired networks. Sometimes it may possible that the source node or the destination node moved out of the network, or cause change in the topology of the network.

Ad Hoc Routing Protocols are classified into Proactive and Reactive routing protocols. Proactive routing protocols use the periodic tables at each participating node to get the information about the current topology, for example DSDV, WRP, DBF. Reactive routing protocols uses on-demand basis to create the route from source to destination, for example DSR, AODV, etc. No protocol is superior to the other as every protocol has its limitations on various metrics, but in general the reactive protocols exhibit better performance than the proactive protocols.

This very protocol consists of static nodes[1]. These static nodes are nothing but those computers (nodes) that are static in any network. Since, most of the user access to the network from their offices or from their houses. So, these systems do not move from a place to another. So we can keep the changes stored in the static nodes in the network in order to provide better performance. The protocol suitably deals with the real world problems that occur in the regular AODV routing protocol.

The remainder of this report is organized as follows: Section 2 provides the previous work done in this area. Section 3 describes the design of the proposed protocol. Section 4 present an example in order to understand the behaviour of proposed protocol. Section 5 describes the possible advantages of the proposed protocol over regular AODV. At last flow chart is also explained.

## 2. RELATED WORKS AND THEIR SHORTCOMINGS:

In this paper, the proposed protocol attempts to modify the route maintenance procedure in the AODV routing protocol. The AODV (Ad-hoc On-demand Distance Vector) Protocol is a Reactive routing Protocol that uses the concept of On-demand routing approach. AODV routing protocol consist of both the features of Proactive as well as Reactive routing i.e. DSDV[2] and DSR[3] respectively. The route discovery and maintenance process in AODV is similar to that in DSR. The periodic use of Hello messages for checking the neighbour nodes and using sequence numbers for route refreshness is borrowed from DSDV protocol[4]. AODV can deal with any kind of mobility rates and variety of data traffic. A number of studies and comparisons were carried out by many authors in order to analyze the performance and behaviour in terms of metrics such as route latency, throughput and bandwidth utilization, and found that AODV exhibits much better performance than DSDV and DSR[5,6].

### *ROUTE DISCOVERY:*

In AODV routing, when a source has data to transmit to a new destination, it broadcast a RREQ for that destination to its neighbours. A node on receiving the RREQ checks if it has not received the same request before using the ROUTE-ID. It is not the destination and does not have a current route to the destination, it rebroadcasts the RREQ and at same time backward route to the source is created[7]. If the receiving node is the destination or has a current route to the destination, it generates a RREP. The RREP is unicast in a hop-by-hop fashion to the source. As the RREP propagates, each intermediate node creates a route to the destination. When the source receives the RREP, it records the forward route to the destination and begins sending data. If multiple RREPs are received by the source, the route with the shortest hop count is chosen.

In case a link break is detected, a RERR message is sent to the source of the data in a hop-by-hop fashion. As the RERR propagates towards the source, each intermediate node invalidates route to an unreachable destinations. When the source of the data receives the RERR, it invalidates the route and reinitiates route discovery. Sequence numbers in AODV play a key role in ensuring loop freedom and freshness of the route[8,9].

### *ROUTE MAINTENANCE:*

Once the route is established, a route maintenance protocol is used to provide feedback about the links of the route and to allow the route to be modified in case of any disruption due to movement of one or more nodes along the route. Maintenance of the discovered/established route is necessary for two main advantages, first to achieve stability in the network and secondly to reduce the excessive overhead required in discovering new route[10]. Each time the route is used to forward a data packet, its expiry time is updated to be the current time plus ACTIVE\_ROUTE\_TIMEOUT (ART) and it is set to 3000 milliseconds[5,11]. ART is a constant value that defines as to how long a new discovered route is to be kept in the routing table of a node after the last transmission of a packet on that route. ART is defined for both the source and intermediate nodes in the network. If a route is not used for this predefined period, a node (source or intermediate) cannot be sure whether the route is still valid or not and removes the route from its routing table, this is to ensure no unnecessary packet loss.

In AODV routing movements of nodes affect only the routes passing through the specific node and thus do not have global effects. If the source node moves while having an active session, and loses connectivity with the next hop of the route, it can rebroadcast an RREQ. When either the destination or some intermediate node moves, it initiates an RERR message and broadcasts it to its precursor nodes and marks the entry of the destination in the route table as invalid, by setting its distance to infinity[12]. An active neighbour node list is maintained to keep track of the neighbouring nodes that are using the entry to route data packets. In case link to the next hop is broken these neighbouring nodes are notified with RERR packets. Each such neighbour node, in

turn forwards the RERR to its own list of active neighbours, thus invalidating all the routes using the broken link[13].

#### *ROUTE CACHING:*

Route caching is carried out for two purposes; firstly, a cached route is readily available to the demanding node thus reducing the routing latency significantly. Secondly, route caching avoids route discovery process and in that way reduces the control traffic that is required in searching for a new route[14]. The caching mechanism in AODV allows only one cache entry per destination, therefore, once the initial data packets get a valid cached route, the chances for successful delivery of subsequent packets is almost guaranteed[15]. In AODV routing protocol, a newly discovered route is cached, so that it may be reused the next time when the same route is requested. AODV carries out route caching both at the source node and at intermediate node that has a cached route to the destination and reply to the source with the cached route[14].

Route caching on one hand reduces the route latency but at the same time prolonged caching may results into storing obsolete / invalid routes, which due to frequent movement of the destination or intermediate node(s) in MANETs. Extra traffic overhead and routing delay is incurred when an invalid route is used further it may result in loss of information packets. One approach to minimize the effect of invalid route cache is to purge the cache entry after some Time-to-Live (TTL) interval. If the TTL is set too small, valid routes are likely to be discarded, and large routing delay and traffic overhead may result due to the new route search. On the other hand, if the TTL is set too large, invalid route caches are likely to be used, and additional routing delay and traffic overhead may result before the broken route is discovered[14]. Thus the efficiency of route caching lay between two contradictory conditions, how long the route has to be stored for subsequent use and how often to purge the same in order to avoid invalid routes. The aim in both cases is to avoid overheads and consequently save bandwidth and route latency.

#### *LIMITATIONS OF AODV:*

A large number of control packets are generated while dealing with the route-maintenance in regular AODV routing protocol, which in turn increases the congestion in the route. So, the overhead in the bandwidth increases with the increase in the number of control packets generated. Ultimately, the delay in the transfer of packets increases.

### 3. PROPOSED PROTOCOL:

In our daily life, we see that people use their office computers or home computers to be accessed in the network. In these systems the position of the computers never changes for a long period of time. So these computers act as the nodes which are static. These static nodes may play an important role in route maintenance of ad-hoc networks. We can easily find the static nodes in the network, by continuously remarking the network for a long period of time. As we know that MANETs and wired networks are both different, with different architectures and with different transmission methods. But, we cannot ignore the static nodes in the realistic mobility model.

The route discovery is done by using the steps of AODV routing algorithm on the basis of ROUTE-REQUEST, ROUTE-REPLY, ROUTE-ACK and ROUTE-ERR messages.

#### *ROUTE DISCOVERY:*

For path discovery source node broadcast a route request (R-REQ) packet to all its neighbor nodes. The header of the RREQ packet contains the Sequence number and ROUTE-ID fields. The neighbor nodes then again re-broadcast the R-REQ packet to their neighbor nodes. So, a reverse path is generated between the source and the neighbor node. The nodes which already get a ROUTE-ID will discard all the next coming ROUTE-IDs in order to prevent the loop formation in the route. So, the R-REQ packets propagate in the network until the destination is found. When the destination is reached by the R-REQ packet reach to the destination, the destination node send a unicast R-REP message towards the source in a hop-by-hop fashion. So, a forward path is generated between the source node and the destination node.

**ALGORITHM FOR ROUTE DISCOVERY:**

1. The source node 'S' broadcast the ROUTE-REQUEST to all its neighbours.
2. After getting the ROUTE-REQUEST the neighbour nodes check the ROUTE-ID, whether the ROUTE-REQUEST has been received before.
3. If the ROUTE-REQUEST packet has been already received by the neighbour node, then it discards the packet.
4. Otherwise, a reverse path is established between the source and the neighbour node.
5. If this node is not the destination or having no path to the destination, then
6. Repeat step 1 and onwards (neighbour node in place of source node)
7. When the ROUTE-REQUEST packet find the destination node or node having path to the destination, the destination node unicast the ROUTE-REPLY towards the source node.
8. When the ROUTE-REPLY packet reach to the source node following the path of intermediate nodes, the route is established in the reverse way i.e. from the destination to the source
9. The route is established, and the data packets can be sent through the established route.

**ROUTE MAINTENANCE:**

The proposed protocol is used for the route-maintenance in the wireless network. After the route discovery process, we traverse the path in order to find the static nodes in the route. These static nodes are consists of buffers to store the necessary information and packets. If the static nodes are found in the route, then a buffer is attached to each static node in order to store the information regarding the static nodes and the packets in case of link failure.

When the link between the two nodes are broken in the network that consist the static nodes, then predecessor node of the broken link send the ROUTE-ERR message towards the source. If there exist a static node in between the source node and the predecessor node then the message has to travel that much path only, i.e. the path between the predecessor node and the static node. Since the static node contains the route information (routing table) to the source, we need not to send the ROUTE-ERR message to the source. If the route is needed then route-caching or route-discovery (based on feasibility) process initiates from the static node not from the source. The buffer attached to the static node is used to store the data packets after the declaration of the broken link in order to prevent the packet loss. After the route-recovery the stored packets in the static node are sent to the destination.

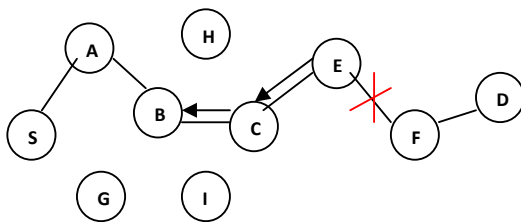


Fig. 1- Transmission of ROUTE-ERR message from Predecessor node to the static node in case of link failure

Here, S is source node, D is destination node, B is static node and E is the predecessor node. Link failure is between E and F

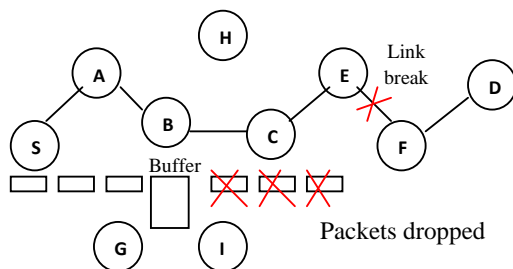


Fig. 2- Buffer attached to the static node, packets storage and packets dropped

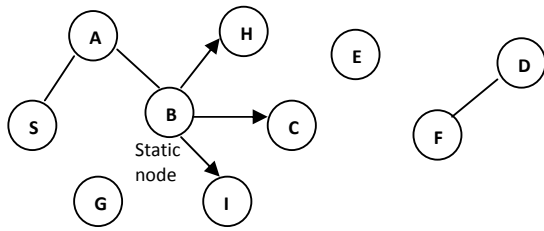


Fig. 3- Re-discovery of route from Static node to the Destination node

Broadcast of ROUTE-REQ packet from static node to its neighbour nodes.

**ALGORITHM FOR ROUTE MAINTENANCE:**

1. Established route is traversed to find the static nodes.
2. If no static nodes are found in the route, then regular AODV route-maintenance is used.
3. If static nodes are found in the established route, then a buffer is attached to every static node.
4. In case of link failure, ROUTE-ERR message is generated at the predecessor node of the broken link.
5. ROUTE-ERR message is sent to the first static node while passing it towards the source (hop-by-hop).
6. After receiving the ROUTE-ERR message, the static node declares the link failure.
7. Packet delivery stops at static node and these packets are then stored in the buffer attached to the static node.
8. Route-caching or route-discovery is applied on the static node instead of source node (based on feasibility).

Route is recovered to transfer the data packets from source to destination again.

**4. EXAMPLE:**

Let us consider a network as:

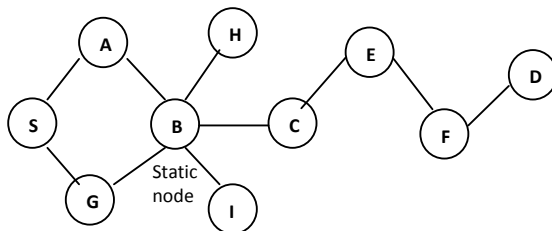


Fig. 4- Example

Node S wants to send data to node D. Now it performs the following operations

**ROUTE DISCOVERY:**

Step 1: node S check its path cache, no information about node D found.

Step 2: node S prepare a ROUTE-REQ message and broadcast it.

Step 3: node A and G receives this ROUTE-REQ, no information about node D is found in their path cache.

Step 4: node A and G broadcast the ROUTE-REQ to their neighbors.

Step 5: Let node B get the ROUTE-REQ from A before the ROUTE-REQ from G arrived. It checks for the ROUTE-ID of ROUTE-REQ sent from G.

Step 6: It found that the ROUTE-ID already exist, so the ROUTE-REQ sent from G is discarded.

Step 7: Similarly, the ROUTE-REQ packet propagate in the network to find the destination

Step 8: Destination D is found by using the reverse path as S-A-B-C-E-F-D.

Step 9: Destination D unicast the ROUTE-REPLY message to the source S (hop-by-hop) to establish the route.

Now the route is established for the data transmission from source to destination.

#### *ROUTE MAINTENANCE:*

Step 1: Suppose the node E send the packet to node F and did not get the ACK, it means the link between E and F is broken.

Step 2: Node E prepares a ROUTE-ERR message and sent it to the static node B in hop-by-hop fashion.

Step 3: Node B on getting the ROUTE-ERR message stop the transfer of data packets.

Step 4: Node B store the data packets in the buffer attached to it.

Step 5: Route caching or route maintenance is applied to the node B.

Step 6: Node B broadcast the ROUTE-REQ to all its neighbors H, C and I.

Step 7: Node H, C and I further broadcast the ROUTE-REQ to their neighbors.

Step 8: Since node E is in the transmission range of the node C. Therefore the next node in the reverse path is E.

Step 9: Similarly D is found following the path B-C-E-F. Since F is having path to D.

In this example we see that it eliminates many problems which occur in the regular AODV routing protocol. The first is that this protocol reduces the path of the ROUTE-ERR message which reduces the packet transfer delay. Also we see that the number of control packets generated is less as compared with the regular AODV routing protocol. So, the overhead in the bandwidth is also reduced. The use of buffer at the static nodes is responsible for the less packet drop as compared with the regular AODV.

#### 5. CONCLUSION:

This paper has presented a protocol which improves the performance of AODV especially in the route maintenance procedure in the realistic mobility model. This protocol reduces the delay experienced in the packet transfer (hop-by-hop). The control packets generated are less; therefore the congestion is less in the route and the overhead in the bandwidth is also less. The packets drop is less in the proposed protocol. The efficiency of the proposed protocol increases with the increase in number of static nodes and when the number of nodes in the network is in the range of hundreds or thousands.

#### REFERENCES:

- [1] Morteza Romoozi, S. M. Vahidipour & Hamideh Babaei, "Improvement of Connectivity in Mobile Ad-hoc Networks by Adding Static Nodes Based on a Realistic Mobility Model", IEEE 2009 Second International Conference on Machine Vision, pages 138-142.
- [2] C.E. Perkins & P. Bhagwat, "Highly Dynamic Destination Sequence-Vector Routing (DSDV) for Mobile Computers", Computer Communication Review, 24(4), 1994, 234-244.

- [3] David B. Johnson and David A. Maltz. "Dynamic source routing in ad hoc wireless networks", Mobile Computing, Kluwer Academic Publishers, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181, 1996.
- [4] Mario Cagalj: Performance Evaluation of AODV Routing Protocol: RealLife Measurements, LCA, EPFL. Alexander Zurkinden, SCC June, 2003 <http://icawww.epfl.ch/Publications/Cagalj/ZurkindenCH03.pdf>
- [5] Satyadeva P. Konduru and Rajendra V. Boppana: On Reducing Packet Latencies in Ad Hoc Networks, The University of Texas, San Antonio, USA. <http://www.cs.utsa.edu/faculty/boppana/papers/Wcnc00.pdf>
- [6] S. R Das, C. E Parkins and E. M Royer: Performance Comparison of Two on Demand Routing Protocols for ADHOC Networks, in Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, 2630 Mar 2000.
- [7] Dan Yu and Hui Li: A Model for Performance Analysis of Mobile Ad hoc Networks, Mobile Network and Applications, Vol. 6, No. 3/June 2001.
- [8] Michael Sirivianos and Athanasios Leontaris: Comparative Evaluation of Ad hoc Routing Protocols in Highly Dynamic Environment, University of California, Irvine & San Diego. <http://www.ics.uci.edu/~msirivia/publications/adhochighmobility.pdf>
- [9] Luke KleinBerndt: A Quick Guide to AODV Routing, National Institute of Standards and Technology, US Department of Commerce, USA. [http://www.antd.nist.gov/wctg/aodv\\_kernel/aodv\\_guide.pdf](http://www.antd.nist.gov/wctg/aodv_kernel/aodv_guide.pdf)
- [10] Damla Turgut, S. K. Das and Mainak Chatterjee: Longevity of Routes in mobile Ad hoc Networks, Proceedings of IEEE VTC (Spring), Rhodes, Greece, 69 May 2001.
- [11] Network Working Group RFC 3561 Jul 2003. <http://www.ietf.org/rfc/rfc3561.txt>
- [12] C. E. Parkins and E. M. Royer: Adhoc Ondemand Distance Vector Routing, In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana, USA, Feb 1999.
- [13] Rajendra V. Boppana: An Adoptive Distance Vector Routing Algorithm for Mobile, Ad Hoc Networks, The University of Texas at San Antonio, San Antonio, TX 782490667.
- [14] Ben Liang and Zygmunt J. Haas: Optimizing Route Cache Lifetime in Ad Hoc Net Works, in Proceedings of IEEE INFOCOM '03, San Francisco, USA, 30 Mar – 3 Apr 2003. [http://www.ieeeinfocom.org/2003/papers/07\\_04.PDF](http://www.ieeeinfocom.org/2003/papers/07_04.PDF)
- [15] Shao-Cheng Wang and Ahmed Helmy: Effects of Small Transfers and Traffic Patterns on Performance and Cache Efficacy of Ad Hoc Routing, MOBICOM 2003, San Diego, California. USA, September 2003.

FLOW-CHART:

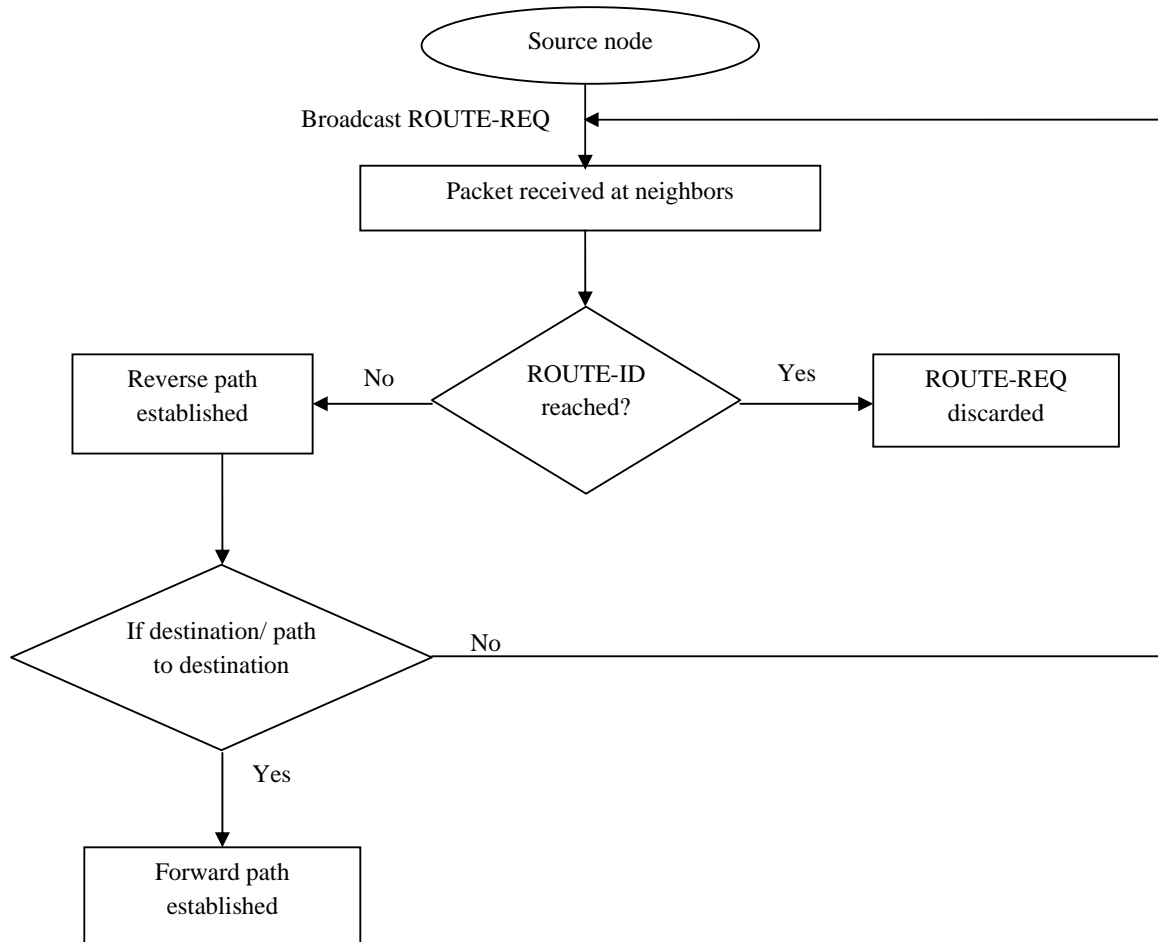


Fig. 5- Route discovery process



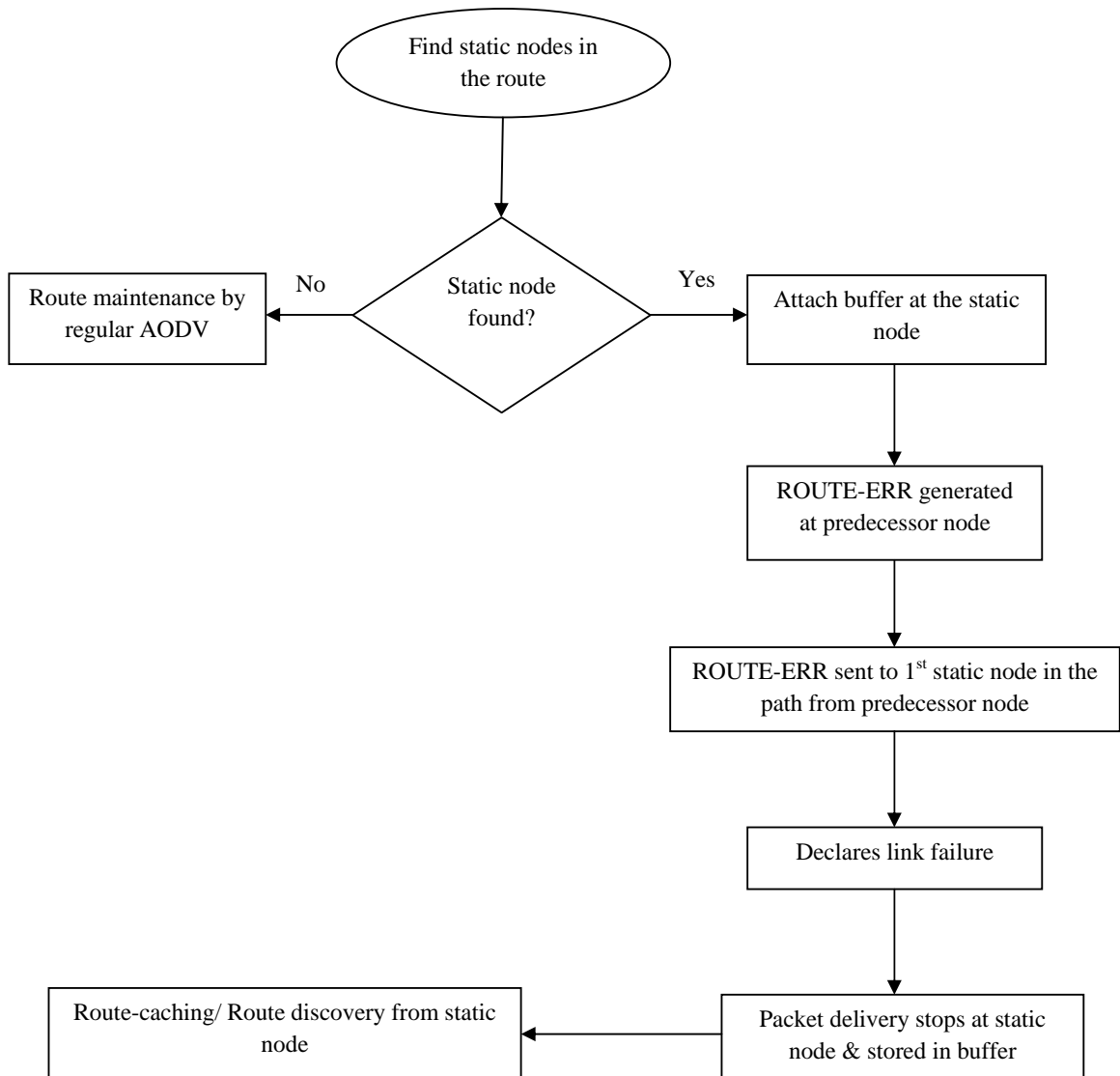


Fig. 6- Route Maintenance Process