Multipath Routing Protocol with Backtracking for Mobile Ad Hoc Networks

Anuj Kumar¹, Yashwant Singh², Rohit Bajaj³ & Sonika⁴

¹,²,³ CSE Deptt. MDU, Haryana, India

Abstract: A mobile ad hoc network (MANET) is characterized by multi-hop wireless links in the absence of cellular infrastructure and frequent host mobility. In this paper, a new on-demand routing protocol is proposed, named on-demand routing protocol with backtracking (ORB), for multi-hop mobile ad hoc networks. Suppose a source host wants to deliver a message to a destination in a MANET. First, the source host issues a request to search the destination. When a node receives the request packet, it gets a hop count value which can point the way to the source. This node broadcasts the request packet to its neighbors. When a destination receives this request, it issues a reply back to the source. We proposed a scoped flooding approach which is applied to reply information to the source. This approach will find out multi-path from a source to a destination. Every node in the MANET maintains some information for this source-destination pair in its own route table. Each node in the forwarding route uses this information to select a group of nodes, named checkpoint nodes, which may have multi-path maintained to that destination. The checkpoint node characteristic is such that a route can be recovered instantly without the need for extra control packets. When a delivering node is aware that the forwarding route is broken, it transmits an error packet back to the source along the return path. When a checkpoint receives the error packet that is backtracked to the source, it has backup paths to recover the broken route. This new route can be used immediately. The main advantage of our ORB is to reduce the flooding search time and cost when a route has been broken. We show that the proposed scheme outperforms the on-demand routing protocol existing in mobile ad hoc networks.

Keywords: Backup Paths, Mobile Ad hoc Networks Mobile Computing, Routing Protocols, Wireless Networks

1. INTRODUCTION

A mobile ad hoc network is a self-organizing and dynamically reconfigurable wireless network without fixed infrastructure or central management. There are some constraints in this wireless network, such as a smaller radio propagation range, bandwidth limitations and battery power consumption. Routing in MANET is an important issue as it involves sending messages to a destination node in a network. Two nodes are neighbors in the network and can communicate directly when they are within transmission range of each other. Communication between no neighboring nodes requires a multi-hop routing protocol. A packet is sent by the source host that is retransmitted through several intermediate hosts before reaching at the destination host. Communication support design and development for distributed and collaborative applications on MANETs is the theme of this paper. Hosts move arbitrarily in MANET, causing the network topology to change frequently and unpredictably. Many routing protocols have been proposed for ad hoc networks. In general, these routing protocols can be divided into two types, table-driven and on-demand. Table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. A style of routing different from table-driven routing is on-demand routing. In the on-demand routing protocol, each node maintains a route for a source-destination pair, also named a session, without periodic route table update exchanges or full network topological views. On-demand routing protocols build and maintain only the routes needed to reduce the routing maintenance overhead. When a source node requests a route to transmit messages to a destination node, it initiates a route discovery process within the network. This process is completed once the route is found or all possible route permutations have been examined. Once the route has been discovered and established, it is maintained until the destination becomes inaccessible along every path from the source or the route is no longer desired. All of these protocols use a flood search to create routes to destinations.

AODV combines the use of destination sequence numbers in DSDV with the on-demand route discovery technique in DSR to formulate a loop-free, on-demand, single path, distance vector protocol. Unlike DSR, which uses source routing, AODV is based on hop-by-hop routing approach. Ad hoc on-demand multi-path distance vector (AOMDV) is extension to AODV. AOMDV computes “multiple loop-free and link-disjoint paths”. The notion of an “advertised hop count” is used to maintain multiple loop-free paths. A particular property of flooding is used to ensure link-disjointness of the multiple paths computed within a single route discovery. When a destination fails or becomes
unreachable from a network component, the source attempts to use flooding search to obtain other paths to the destination. This method is used in some of today’s on-demand protocols. The flooding-search process produces too many control packets and obstructs the wireless network channel. An efficient on-demand routing protocol, called on-demand routing with backtracking (ORB) is presented in this work for ad hoc networks. Our objective is to develop an on-demand, backup path protocol to avoid using flooding-search to reconstruct routing. When a source attempts to deliver a message to a destination, the source host issues a request to search for all probable paths to that destination. Each node in the MANET maintains a hop-count and neighbors link-state for a session in its own route table. The node hop-count value is the distance from the source to this node and it points the way return to the source. The neighbor link-state determines if the node can forward the message to the destination through the neighbor or not. If a node has two forwarding paths to the destination, it will be a checkpoint node. Possessing backup paths to deliver data messages to destinations is the characteristic of the checkpoint node. A checkpoint node can reconstruct a route if delivery route link failure occurs. Our ORB can effectively reduce the frequency of route reconstruction and the number of control messages required for the source to re-initiate a route.

2. PRELIMINARIES

The backtracking routing concept is introduced in this section. The main idea in ORB is to recover a route when an error message backtracks to the source. This method is designed primarily for highly dynamic ad hoc networks in which link failures and route breaks occur frequently. The ORB protocol contains a method that establishes multi-path to the destination. The nodes in the main route use this information to select a group of nodes, called checkpoint nodes, which have backup routes to the destination. When a node in the forwarding route becomes aware that the next hop in the forwarding route cannot be made, it sends an error message to the source. When a checkpoint node in the return path receives this error message backtracked to the source, it immediately delivers a data message to a backup route. A backup route from the checkpoint node to the destination is then substituted for the broken route. The proposed ORB method can effectively reduce the route discovery frequency and discovery cost. We assumed that S is the source and D is the destination for a session in this paper. In Figure 1 S has a forwarding route (path 1) with two checkpoint nodes, CP1 (node A) and CP2 (node B), to D. In MANET, each mobile host is free to move, causing the network topology may be changed frequently. The topology of the local network between nodes B and D has changed. The forwarding route (path 1) has therefore failed. When a node in the forwarding route detects link failure and cannot deliver a message to the destination, it sends an error message back to the source. This error message is delivered back to the source along the reverse route path. When an intermediate node, which is a checkpoint node in return route, receives this error message, it can recover the route and retransmit the data message to D.

The characteristic of a checkpoint node is that it has backup paths maintained to the destination node and can immediately recover the broken routes. Therefore, the source can reduce the number of rediscover routes and the number of discovery packets in MANET. Every node is assumed to have a buffer space for storing data messages for a period of time. B (CP2) has a backup path and reconstructs route path 2 to D. When B received the error message, it retransmits the data message to D along the backup path, S would then use route path 2 to transmit data messages.

3. ON-DEMAND ROUTING PROTOCOL WITH BACKTRACKING

Three phases are used in our ORB protocol. The first phase is the route discovery phase. The first phase has a route request and route reply process. After the source im-plements the first phase, all probable routes are built from the source to the destination. The message delivery phase is the second phase. After the route has been established, the source delivers data messages to the destination. The third phase is the path maintenance phase. In the proposed ORB protocol, a group of nodes, called checkpoint nodes, are used to recover the broken forwarding routes. The checkpoint node selection is based on the node has multi-path or not. When a node detects its neighbors have changed, it deletes the related information from route table in link pruning phase.

3.1. Route Discovery Phase

3.1.1. Route Request

We assumed that N is a node in MANET, N’s neighbor Ni is recorded in N’s route table, where 1 ≤ i ≤ k, k is the number of N’s neighbors. HCN is the hop-count value for N and represents the hop distance from the source to N. HCN is the minimum distance from the node N to the source. N1 N
LS is defined the link-state between in node $N$ and $N$'s neighbor $Ni$. $LSN$ is the highest priority value from $Ni N LS$ $1 \leq i \leq k$. $N$ computes $Ni N LS$ based on three different values recorded in $N$'s route table, $HCN$, $Ni HC$ and $LSNi$. Therefore $N$ obtains $k$ link-states ($Ni LSN$ $1 \leq i \leq k$) due to $k$ neighbors and $N$ sets the link-state $LSN$ value from $Ni LSN$ $1 \leq i \leq k$ with the highest priority value. $<S1, S2, \ldots, Sk>$ represent the route sequence from $S1$ to $Sk$.

When a route is requested but no information to $D$ is known, $S$ floods route request packet ($RREQ$) to discover a route. We assumed that a node $F$ forwards the $RREQ$ to its neighbors. The $RREQ$ format is $RREQ$ ($SIP$, $DIP$, $FIP$, $HCF$), where $SIP$ is the IP address of $S$. $DIP$ is the IP address of $D$. $FIP$ is the IP address of $F$. $HCF$ is the hop-count for $F$. This value is the hop distance from $F$ to $S$. When $S$ initially broadcasts $RREQ$, $HCF$ is set to 0. In the proposed ORB protocol, each node $N$ builds a route table $RTN$ (Node, Hop-Count, Link-State) for a session. The “Node” field is recorded neighbor $Ni$. The “Hop-Count” and the “Link-State” fields are recorded $HCNi$ and $Ni LSN$, respectively. If the route table entry is not refreshed within a given period of time, it is deleted.

When $N$ receives a non-duplicate $RREQ$ broadcast from its neighbor $Ni$, it builds a route entry for the <Source, Destination> pair and records the previous nodes $Ni IP$ and $HCNi$ to this route entry. $N$ then computes $HCN$ on the basis of $HCNi$ and records it into the route table. The $HCN$ value is equal to that for the previous node plus 1, i.e., $HCN = HCNi + 1$. This previous node information is needed later to reply to the route reply packet ($RREP$). The $N$'s IP and $HCN$ replaced with the original $FIP$ and $HCF$ of $RREQ$.

$N$ then broadcasts the $RREQ$ to its neighbor nodes. If $N$ receives a duplicate $RREQ$, it records the $FIP$ and $HCNi$ of the $RREQ$ into the route entry for the <SIP, DIP> pair and then discards the $RREQ$. If $HCNi +1$ is lower than the Original $HCN$, $HCNi +1$ is replaced by $HCN$ and recorded in the route table.

After $RREQ$ flooded, every node $N$ obtains a $HCN$ for $S$ and knows its neighbor’s hop-count in the route table as shown in Figure 2(a). The distance from node $A$ to node $S$ is one hop, i.e., $HCA=1$. After the $RREQ$ is flooded, every node records its own and its neighbor’s hop-count values in the hop-count entry of the route table. When a node $N$ wants to reply a packet to the source, it sends the packet to its previous node $Np$. The previous node is select from $N$'s table and $HCN<HCN$. Figure 2(b) is shown the reply path for every node.

### 3.1.2. Route Reply

In the ORB, the link-state value indicates if the source can deliver the data messages to the destination via $N$. Every node sets a link-state value for a neighbor. A node can determine the forwarding path to destination using the link-state. Every node $N$ sets $HCN$ in the discovery request phase. $Node$ $N$ then sets $LSN$ on the basis of $HCN$, the hop-count value and the link-state for its neighbors. The link-state value is set in the route discovery reply phase. The link-state $Ni N LS$ has four respective state values, defined below with the different priorities 3, 2, 1, and 0 from high to low.

- 0: indicates the initial state or unknown state.
- 1: the farther this path is from $N$ to the destination via $Ni$. $N$ forwards the message to $Ni$ that has greater hop-count.
- 2: the farther path is from $N$ to destination via $Ni$. $N$ forwards the message to $Ni$ that has the smaller hop-count.
- 3: the shortest path is from $N$ to the destination via $Ni$.

When $D$ receives the $RREQ$, it begins to execute the route reply process. $D$ replies with a route reply packet ($RREP$) back to $S$ via all probable paths. We assumed that a node $F$ forwards the $RREP$ to its neighbors. The $RREP$ format is $RREP$ ($SIP$, $DIP$, $FIP$, $HCF$, $LSF$), where $SIP$ is the IP address of $S$. $DIP$ is the IP address of $D$. $FIP$ is the IP address of $F$. $HCF$ is the hop-count of $F$. $LSF$ is the link-state of $F$. $LSF$ is the highest priority value from $Fi F LS$ $1 \leq i \leq k$.

When a node $N$ broadcasts a $RREP$, all neighbor nodes $Ni$ should receive this packet compute $Ni N LS$. We propose a state transition function (STF) to compute the link-state. The STF is shown in Figure 3. The STF defines how node $N$ computes $Ni N LS$ . We assumed that node $R$ receives the $RREP$ packet broadcast from neighbor $F$. A node $R$ receives $RREP$ and records $F$ as the neighbor node in the “Node” field, $HCF$ in the “Hop-Count” field and $LSF$ in the “Link-State” field. This information is needed later to deliver the data messages to the destination and compute $F R LS$ . If $R F LSR > LS$, then $F R R LS = LS$ . If $LSR>0$ and $R$ never broadcast this $RREP$ before, the IP of $R$, $HCR$ and $LSR$ are replaced the original $FIP$, $HCF$ and $LSF$ for the $RREP$. $R$ then broadcasts $RREP$ to its neighbors. If $LSR=0$, $R$ stops to...
broadcast RREP to its neighbors. RREP is a scope flood from the destination to the source. This process is reiterated until S receives the RREP. Figure 4(a) is shown an example of the reply phase. D broadcasts a RREP to its neighbors. Nodes A and E receive the RREP, and they record HCD in their route table and compute link-state D = 3 A LS and D = 2 E LS, respective. Because D = 3 A LS and D = 2 E LS, nodes A and E broadcast RREP to their neighbors. If node N sets = 0 N LS or has forwarded the RREP before, it stops to broadcast the RREP. This process is reiterated until S receives the RREP.

Figure 4: An Example of Routing Reply Phase

3.2. Message Delivery Phase

When S receives the RREP, the routing has established. S can deliver the data message to D along forwarding path. We assumed that node N is a node in the forwarding route. Node NN is the next node of N in the forwarding route. Node NP is the next node of N in the return route. MP is a message packet. RREP is an error packet. The MP and ERRP formats are MP (SIP, DIP, RIP, FIP, Data) and ERRP (SIP, DIP, RIP, FIP), where node F broadcasts the packet to its neighbors. SIP is the IP address of S. DIP is the IP address of D. RIP is the IP address of R that receives the packet. FIP is the IP address of F that forwards the packet. Data is the payload data. We assumed that S begins to deliver a MP to D along the forwarding route. When S delivers a MP, it is regarded as node N. N selects a node NN that has the highest priority value from N LS 1 ≤ i ≤ k in route table. The message always is forwarded using shortest path to the destination. N first detects whether NN is in its radio range or not. This detecting action is RTS-CTS in the MAC layer. If NN is in N's radio range, N delivers MP to NN. After NN receives MP delivered from N, it determines if it is D or not. If NN is D for the <SIP, DIP> pair, the delivery process is completed. If NN is not D, NN would act as N and reiterate the message delivery process. If NN is not in N's radio range, N first determines if it was a checkpoint or not. The checkpoint selection is described in Section 3.3.1. If it is, it can find a backup path to D. If it is not a checkpoint, it replies with an ERRP to NP. The nodes in the return route reiterate this process until ERRP reaches the checkpoint node or S. If S receives ERRP and it is not a checkpoint node, S rediscovers the destination node using the route discovery phase.

3.3. Path Maintenance Phase

3.3.1. Checkpoint Selection

The essential checkpoint feature is that it can deliver a message along backup paths to the destination node when forwarding route is broken. When a node in the forwarding route detects a broken route, it replies with an error message that backtracks to the checkpoint node. The checkpoint node then uses another path that can recover the route to the destination. The checkpoints are selected from the forwarding path route. If a node has two or more then two forwarding routes, it is a checkpoint node. In Figure 4(a), node A is a checkpoint node. Node A is A LS and D = 3 A LS. Node A has two forwarding path, so it determines it is a checkpoint node. Node A can deliver a message along a backup path to D. The backup path is <A, B, C, E, D>.

3.3.2. Link Pruning Phase

When the local connectivity of a mobile node is involved, each mobile node can detect the other nodes in its neighborhood via a local broadcast technique using a beacon packet. The route tables of the nodes within the neighborhood are organized to optimize the response time to local movements and provide a quick response time for requests for the establishment of new routes. In the ORB protocol, every node periodically broadcasts a beacon packet to detect its neighbor nodes. When a neighbor node receives a beacon packet from N, it refreshes the N’s route table entry. If the route table entry is not refreshed within a given period of time, it is deleted. If a node N detects its neighbor not in its radio range. N broadcasts a RPRU packet to prune break path. The PRUP format is RPRU (SIP, DIP), where SIP is the IP address of S. DIP is the IP address of D. When a next node NP in forwarding route receives a RPRU packet from N, NN deletes the route table entry of N. If NN has other return paths in its route table, it stops to broadcast the RPRU to its neighbor. If N*N has any return path, it deletes all routing entries for <SIP, DIP> pair and broadcasts RPRU to its neighbors. When a previous node NP in the return path receives a RPRU packet from N, NP deletes the route table entry of N. If NP has other forwarding paths in its route table, it stops to broadcast the RPRU to its neighbor. If NP has not any forwarding path, it deletes all routing entries for <SIP, DIP> pair and broadcasts RPRU to its neighbors. An example of link pruning phase is shown in Figure 5. When node B cannot detect node C, B broadcasts a RPRU to its neighbors. B has no any forwarding path, so it deletes all routing entries for <S, D> pair. When node A receives this RPRU, it deletes the routing entry of node B. Node A stops
to broadcast the RPRU, because it has another forwarding path in its route table. When node C cannot detect node B, it broadcasts a RPRU to its neighbors. C has no any return path, so it deletes all routing entries for <S, D> pair. When node E receives this RPRU, it deletes the routing entry of node C. Node E also has no any return path, it broadcasts a RPRU to its neighbors and deletes all routing entries for <S, D> pair. When node D receives the RPRU, it deletes the routing entry of node E.

Figure 5: An Example of Link Pruning Phase.

4. EXPERIMENTAL RESULTS

4.1. Simulation Environment

A simulator was designed and implemented as an experimental platform. The main part of this simulator is a discrete event-driven engine designed to simulate systems that can be modeled using processes communicating through signals. We built a simulation platform to perform the routing operations in a MANET. Three routing protocols, AODV, AOMDV and our ORB were implemented. Our simulation modeled a network of 100 mobile hosts placed randomly within a 1000 meter × 1000 meter area. The radio propagation range of each node was 200 meters and the channel capacity was 2Mb/s. Each run was executed for 500 seconds of simulation time. There were thirty data sessions, each with a traffic rate of four to eight packets per second. The size of data payload was 512 bytes and 8 packets per second. The random waypoint model [1] was used to model mobility. Here, each node started its journey from a random location to a random destination at a randomly chosen speed (uniformly distributed between 0 and max. speed). Once the destination was reached, another destination was targeted instantly. The maximum speed was set from 10 to 90 Km/hr.

4.2. Performance Metrics

We evaluate three key performance metrics:

(i) Packet delivery ratio: ratio of data packets delivered to the destination to those generated by the source; or a related metric received throughput in Kb/sec received at destination. (ii) Route discovery frequency: the total number of route discoveries initiated per second.

(iii) Normalized routing load: the total number of routing packets transmitted for each delivered data packet. Each hop-wise transmission of these packets is counted as one transmission.

(iv) Control packet overhead: the control packets are needed to establish route from the source to the destination. The control packets include RREQ, RREP and ERRP.

4.3. Results and Analysis

The average packet delivery ratio is shown in Figure 6, our ORB has higher packet delivery ratio then other two protocols at any model mobility, especially exhibited higher performance at higher speed. Because ORB has multi-path to recover broken path and can backtrack to search other backup path, ORB lost fewer packets than AODV and AOMDV. Our ORB provides efficient fault tolerance in the sense of faster and efficient recovery from route failures in dynamic networks.

Figure 6: Packet Delivery Ratio.

Figure 7 shows the discovery frequency performance. The ORB needed less discovery times to maintain routes for sessions. The AODV is a simple path routing protocol, so the source must broadcasts a lot of discovery packet to recover the broken path. The AOMDV is a multi-path routing protocol, but the AOMDV has fewer multi-paths then our ORB. The AOMDV cannot backtrack to search other backup path, so the AOMDV also has more discovery time then our ORB. Figure 8 shows the normalized route load performance. The ORB was better than other protocols. The ORB used fewer control packets to maintain routes, and had better packet delivery ratio. From the simulation results our ORB can efficient to reduce the traffic load to system. Figure 9 shows the amount of all control packets. The ORB has fewer discovery times, so it needs fewer RREQ then others.
Our ORB uses a scoped flooding approach, so the amount of RREP fewer than the AOMDV. The AODV only needs fewer RREP to reply, because it is a simple routing protocol. The AOMDV needs much RREP to find out possible path, because RREP is flooded to MANET. Our ORB is using ERRP to backtrack and recover the broken path, so it needs more ERRP than the AOMDV.

Figure 7: Route Discovery Frequency.

Figure 8: Normalized Routing Load.

5. CONCLUSIONS

An on-demand, multi-path, backtrack routing protocol was proposed in this paper. A scoped routing reply phase flooding approach is proposed in our ORB protocol. All possible paths and checkpoint nodes are found in the reply phase. The checkpoint has backup paths to the destination, and each node in the forwarding route can reply to an error packet backtrack route to find a checkpoint when the route is broken. The ORB effectively decreases the number of rediscovery times and cost. The number of flooded control packets is a main issue that obstructs the network. The proposed ORB protocol could search for more routes to the destination, and reduce the number of rediscovery times and control the number of packets flooded into the MANET.

Figure 9: Control Packet Overhead.

REFERENCES


