QoS Support for Mobile Ad-Hoc Networks Based on a Reservation Pool

Min-Gu Lee and Sunggu Lee, *Member, IEEE*

Electrical and Computer Engineering Division
Pohang Univ. of Science and Technology (POSTECH), Pohang, Korea
{bluehope, slee}@postech.ac.kr

Abstract

Many interesting applications using mobile ad-hoc networks are possible if quality-of-service (QoS) can be effectively supported. Towards that end, this paper proposes a method based on providing a pool of backup paths that can be used if the primary path can no longer support the required level of QoS. Such a capability is especially important for mobile ad-hoc networks because intermediate nodes (within routing paths) can move around in such networks. Route maintenance and switchover should also be performed in an efficient manner. This is accomplished with the help of a method referred to as pseudo-distance routing.

1 Introduction

Many applications for MANETs require timeliness constraints. Such applications include weather forecasting, tracking objects in a battlefield and monitoring blood sugar levels of a diabetic. Let us consider an object tracking application using multiple video sensors that can move around. A user who wants to monitor enemy objects in the battlefield using an ad-hoc network of video sensors should be able to observe smooth and up-to-date video streams from each video sensor. Typically, there are a small number of monitor nodes but many sensor nodes in this type of scenario. The set of video sensors that are currently tracking target objects must be able to transmit their video streams to monitor nodes without being disturbed by other best-effort message streams. Therefore, supporting the required level of quality-of-service (QoS) for each video stream is an important problem in this kind of application.

In order to support QoS in a network, sufficient resources must be reserved in order to satisfy the QoS requirements of a particular application. However, due to node mobility (resulting in network topology changes), unreliable communication channels with high error rates, low bandwidth and high jitter rates, support for QoS in MANETs is a very challenging problem. Recently, several QoS-aware routing algorithms have been proposed to support highly dynamic networks[1, 2, 3]. A QoS-aware routing algorithm is used to find a feasible path that meets the QoS requirements of a data stream, specified in terms of bandwidth, delay, reliability or cost. Then it reserves network resources for each link along the selected path. Using such an algorithm, a source would be able to send one or more data streams, such as real-time video in a multimedia application, to one or more destinations with different levels of service for each data stream.

In a MANET, a QoS-aware routing algorithm must be able to handle topological changes quickly. Each QoS-aware routing algorithm may provide its own mechanism for handling such changes. A typical mechanism involves reserving additional network resources for a backup path. However, such an approach wastes a significant amount of network resources if most data streams do not make use of the backup path. Furthermore, previous algorithms that use this approach do not take link stability into account. Thus, even if the link stability is very high, which implies that the probability of the primary path becoming disconnected is very low, such an algorithm will still need to reserve additional resources for a backup path that may block other best effort packets.

Another approach that can be used to support QoS-aware routing in MANETs is to reconstruct a feasible path when the primary feasible path becomes disconnected. Since this approach does not reserve network resources until they are actually required, waste of network resource is minimized. However, this approach typically results in a relatively long recovery time, which may not be acceptable for certain applications such as multimedia video streaming.

In this paper, as a third alternative, we propose a QoS-aware reservation pool method that reduces the amount of wasted network resources while ensuring fast recovery when the primary path becomes disconnected. This method is used in conjunction with a new method for MANET rout-

Proceedings of the Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing
0-7695-2561-X/06 $20.00 © 2006 IEEE

ing and route maintenance proposed by us in a previous paper. In the next section, related work is discussed. Then, in Section 3, the proposed approach to QoS-aware routing is discussed, and the underlying routing algorithm is presented. Next, Section 4 presents the backup path reservation pool method used to support QoS in MANETs. Finally, conclusions are presented in Section 5.

2 Related Work

Lin and Liu [1] proposed a QoS routing method for MANETs. Lin and Liu's method is based on code division multiplex access(CDMA) over time division multiplex access(TDMA). *Link bandwidth* is defined to be the set of common free slots between two adjacent nodes within a frame and *path bandwidth* is the set of available slots between any two nodes. Paths are found using an underlying routing protocol such as direct sequence distance vector(DSDV) routing[4]. Primary and secondary paths that meet the prespecified bandwidth constraints are reserved in order to support QoS routing. This method has the drawback of wasting network resources since secondary paths will not be used most of the time. Moreover, DSDV is an inefficient routing algorithm that requires periodic update messages, and CDMA over TDMA is difficult to achieve in real applications such as those involving sensor networks.

The *ticket based probing (TBP) protocol*[2] is a QoS-aware routing algorithm proposed by Chen and Nahrstedt. A ticket gives the holder permission to search one path. There are two types of tickets: yellow and green. A yellow ticket is used to maximize the probability of including a feasible path because it prefers paths with smaller delays. A green ticket is used to maximize the probability of finding a low-cost path because it prefers paths with low cost. By controlling the number of probing messages using tickets, this method controls the routing overhead required. Each intermediate node forwards a probing message to its selected neighbor nodes based on the color of the ticket associated with that message. TBP provides a localized path repairing scheme. TBP provides three levels of path redundancies. The first level of redundancy involves establishing multiple routing paths for the same stream. Every data packet is sent along each path independently. This obviously implies a lot of network overhead. The second level of redundancy involves establishing multiple routing paths but use only one primary path regularly. Although multiple resources are reserved for each data stream, only resources along the primary path are used until it becomes disconnected. However, resources reserved on the backup paths still have an adverse effect on other best-effort and real-time message streams. The third level of redundancy involves establishing a secondary path only when the primary path fails. However, establishment of this secondary path may take a long time, and this secondary path may not able to meet the specified QoS requirements at the time when the primary path fails.

Stateless wireless ad hoc networks(SWAN)[3] provide stateless QoS routing for MANETs. "Stateless" as used here means that each node does not keep any per-flow or aggregate state information, but uses feedback information from the network. A source sends a probing message toward the destination. Then an intermediate node marks its bandwidth in the packet header when its own available bandwidth is detected as the bottleneck bandwidth. When the destination node receives a probing message, it replies to the source node with the bottleneck bandwidth. Admission control is performed at the source node when it receives a response packet from its destination. If the reply message indicates a lack of bandwidth, the source initiates a rerouting procedure. When overloaded states are detected in a reserved path, feedback messages are sent to the source node. The source then initiates another rerouting procedure. The underlying routing protocol must be able to provide an alternate path when a given path fails the admission test at the source. This algorithm does not provide secondary paths. Thus, when a path is broken, a feedback message is sent to the source node, after which the source node reinitiates the route discovery phase, which involves a long delay and a large amount of overhead. Finally, this algorithm suffers from the false admission problem because it does not use a forward reservation scheme.

3 QoS-Aware Routing

In QoS-aware routing, when a source node wishes to establish a connection with a certain level of service, it must specify its requirements (bandwidth, reliability, cost, level of tolerance to delays, etc.) to the system. The system must then determine whether such a connection can be established to the requested destination node. If successful, then the connection is termed to be *feasible* and the connection request is granted; otherwise, the connection request is denied. With a fickle network such as a MANET, however, even if such a connection is initially established, the connection may quickly deteriorate to the point where the initial QoS requirements are no longer met (for simplicity, we can consider this as a disconnection). In this case, if possible, an alternate (backup) path must quickly be established.

Towards that end, this paper proposes the establishment of a *reservation pool* of backup paths that can be used in the event that the primary path becomes disconnected. Such a reservation pool consists of a set of network resources, which can form backup paths, reserved for use by a set of data streams. Additions can be made to the reservation pool when a data stream makes a backup path reservation re-

0-7695-2561-X/06 $20.00 © 2006 IEEE

quest. Several data streams can share a single reservation pool of network resources (overbooking can occur). Therefore, when a backup path becomes a primary path, the resources used by that path are marked as "used" in the reservation pool and alternative resources are added to the reservation pool (for a new backup path) if necessary. If there are insufficient resources for the maintenance of an effective reservation pool, then feedback messages to that effect should be sent to source nodes that initiate backup path requests.

First, let us consider the problem of finding and maintaining feasible paths in a MANET. Feasibility testing, which is another important problem for QoS, is left as future work and will not be considered in this paper. For now, let us consider a link connection to be "broken" if the connection has deteriorated to such a point that the required level of QoS can no longer be supported. In a MANET, where the nodes are mobile and wireless connection quality can fluctuate, such an event can occur fairly frequently. In order to support QoS-aware routing, the underlying routing algorithm should be able to provide short-distance paths, loop free paths and multiple redundant paths. In addition, it would be desirable for the routing algorithm to be executable in a fully distributed manner.

Routing algorithms that are developed for ad-hoc networks can be divided into two categories: proactive and reactive. Proactive routing algorithms attempt to maintain consistent and up-to-date routing information from each node to every other node in the network. In order to maintain consistent and up-to-date routing information, proactive algorithms have to frequently exchange routing information. Therefore, proactive algorithms tend to have large control message overheads. On the other hand, reactive algorithms create routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery procedure within the network. After routes are found, each node updates its own route information to adapt to changes in the network topologies.

For QoS-aware routing, the major shortcomings of TORA[5], link reversal[6] and other reactive algorithms are the possibility of significant delay during the route discovery and maintenance phases. Since route discovery is based on flooding of query messages in most reactive algorithms, it is very costly. In time-constrained applications, excessive delays during route discovery may result in deadline misses. However, a hybrid proactive and reactive algorithm can be used to reduce both route discovery time and route distance. Thus, in this paper, a hybrid routing algorithm referred to as pseudo-distance routing (PDR) [7] is used as the underlying routing algorithm.

3.1 Pseudo-Distance Routing

PDR assumes that all links are bidirectional and reliable by building on top of the internet MANET encapsulation protocol (IMEP) [8]. Furthermore, it also assumes that detection of neighboring nodes is performed by an underlying protocol layer like IMEP. PDR assumes a broadcasting network, which is a widely adopted assumption. Finally, PDR also assumes that each node has its own unique identifier (like a MAC address).

3.1.1 Pseudo-Distance

PDR adopts the destination-oriented direct acyclic graph (DAG) presented in [6]. To transform a given network graph $G = (V, E)$ into a destination-oriented DAG, each node should have a "height" value. Unlike TORA and link reversal algorithms, a height in PDR is not a value representing temporal order but a *pseudo-distance* to the destination. The height of a node v_i, relative to a node v_j, is written as $H_{i,j} =< \lambda, -\alpha, -\beta >$. A pseudo-distance λ is a distance metric between a node v_i and v_j. α is the number of neighbors that have lower λ values than v_i and β is the number of neighbors that have the same λ value as v_i.

α represents the number of neighbors that are expected to be closer than the current node to a destination node and β represents the number of neighbors that are expected to be the same distance as the current node to a destination node. PDR tries to forward packets to a neighbor with greater α and β values in an attempt to follow paths with more redundancies. Note that in the height metric, α and β are represented as negative values (in order to be able to facilitate comparison of height values). When routing a message on a destination oriented DAG, each node selects a neighbor with minimum height as its next hop.

Two types of links are identified using the pseudo-distance concept. Primary links are mainly used to route packets along shortest-distance paths. Auxiliary links are used when all primary links are broken. Suppose that the heights of two adjacent nodes v_i and v_k, relative to destination node v_j, are $H_{i,j}$ and $H_{k,j}$ with $H_{i,j} > H_{k,j}$. Then, v_i sets $e_{i,k}$ as the primary outgoing link only if $\lambda_{i,j} > \lambda_{k,j}$. Otherwise, it sets $e_{i,k}$ as the auxiliary outgoing link.

As described in [6], in order to build a destination-oriented DAG, PDR only requires each node to collect information from its neighbors. Suppose that a node v_j is the destination. Then another node v_i only needs to collect $H_{k,j}$ information for $v_k \in N_i$ to properly set the directions for all of its links $e_{i,k}$.

Figure 1 shows an example of a destination-oriented DAG, oriented toward destination node v_6, using the pseudo-distance concept. Numbers in the vertices represent unique identifiers for each node and numbers beside the vertices are the heights $H_{i,6}$ of each node. Solid arrows represent primary links and dashed arrows represent auxiliary links. Note that the pseudo-distance between two nodes is a multiple of δ, which is the default difference, in units of λ, between adjacent nodes.

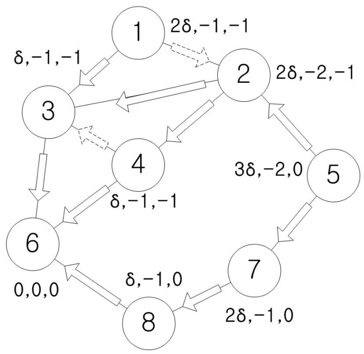

Figure 1. An example of a destination-oriented DAG, using pseudo-distance, for destination node v_6.

In order to send packets, each node v_i simply selects a node $v_k \in N_i$ that has the smallest value of $H_{k,6}$. In Fig. 1, v_1 selects v_3 as its next hop since $H_{3,6} < H_{2,6}$. In this case, $e_{1,3}$ is the only primary outgoing link for v_1. Suppose that v_5 wishes to send a packet to destination v_6. In this example, the pseudo-distances of both v_2 and v_7 are the same, and both are primary links. However, v_5 should be able to select v_2 as the next node to send a packet to because v_2 has more shortest-distance paths to the destination node. Since $H_{2,6} < H_{7,6}$, v_5 selects v_2 as its next hop. Suppose instead that v_5 selects v_7 as the next hop. In this case, if links $e_{7,8}$ or $e_{8,6}$ are broken, then the packet sent from v_5 will fail to be delivered to destination v_6. However, since v_5 selects v_2 as its next node, some failure(s) of links can be tolerated (such as $e_{2,4}, e_{4,6}$ and others).

3.1.2 Route Discovery and Maintenance

When a node wishes to send packets to a destination, it broadcasts a QRY packet. Note that every node except the destination initially sets its own height to null, while the destination node sets its height to zero. An intermediate node with a null height that receives the QRY rebroadcasts it. When a node with a non-null height receives a QRY, it broadcasts a REP with its height. An intermediate node that receives a REP updates its neighbor height table and its own height, and then broadcasts a REP with its height.

A route maintenance phase is triggered by v_k when it loses all of its primary outgoing links. v_k updates its pseudo-distance and transmit an UPD packet to its neighbors to convert its incoming or auxiliary links to outgoing links. When v_l receives an UPD from its neighbors, it updates its neighbor's heights and checks whether it still has outgoing links or not. If it loses its last outgoing link, then it also updates its own height and transmits an UPD to its neighbors.

3.2 Notation

A node v_i is assumed to maintain up-to-date local state for all of its links. The state information of $e_{i,j}$ includes $\mathcal{D}(e_{i,j})$, representing the channel delay of link $e_{i,j}$, and $\mathcal{ABW}(e_{i,j})$, representing the current available bandwidth on that link. $e_{i,j}$ may have $nStrm_{i,j}$ data streams, referred to as $S_k^{i,j}$, where $0 < k \leq nStrm_{i,j}$. The capacity (maximum bandwidth) of $e_{i,j}$ is referred to as $\mathcal{C}(e_{i,j})$. $\mathcal{BW}(S_k^{i,j})$ is the bandwidth reserved for stream S_k on $e_{i,j}$. The current residual bandwidth of $e_{i,j}$ can be calculated as $\mathcal{ABW}(e_{i,j}) = \mathcal{C}(e_{i,j}) - \sum_{k=1}^{nStrm_{i,j}} \mathcal{BW}(S_k^{i,j})$. For a given path $P^{i,j}$, $\mathcal{D}(P^{i,j})$ is the summation of the total delay of all links in $P^{i,j}$, and $\mathcal{ABW}(P^{i,j})$ is the minimum bandwidth among all links in $P^{i,j}$.

3.3 Finding a Feasible Path

In order to find a feasible path from a source to a destination, the source node sends probing messages to the destination based on multiple paths discovered by PDR. There are three control messages used to find and reserve a feasible path - DFP(Discovery/Feasible-path Probing), DNP(Discovery/Nack of Probing) and DRP(Discovery/Response of Probing). DFP is a probe message that is triggered by the source node when it wishes to establish a connection to a destination. Each node forwards a received DFP towards one of its outgoing links. DNP is a NACK response of discovery probing message indicating that there is no feasible path via that link. DNP is triggered when a node that receives a DFP does not have valid outgoing links. DRP is a response to a probe message that converts resources in the reservation pool to actual reservation of those resources (DRP is used to notify the source node of successful connection establishment). When the destination node receives DFP, it sends a DRP packet if the path meets the desired QoS requirements.

There are three methods that can be used to check for the feasibility of paths - complete search, single probing and multiple probing. Complete search involves traversing all possible paths from the source to the destination to check for feasibility of QoS requirements. Although it requires an excessive amount of overhead, complete search can be used if the best path *must* be found.

In single probing, the source sends only one probe message to the destination along the shortest path P_s. Each intermediate node v_j in P_s checks whether its available bandwidth $\mathcal{ABW}(e_{i,j})$ and $\mathcal{D}(e_{P_s^{i,j}})$ are sufficient for the required level of QoS. If v_j meets all these constraints, it forwards the probe message to its minimum height neighbor v_k with current path information $P_s^{i,j}$, $\mathcal{ABW}(P_s^{i,j})$ and $\mathcal{D}(P_s^{i,j})$. However, if v_k fails to meet its delay or bandwidth requirements, it informs its previous hop node v_j of this fact. Then v_j can send a probe message to another outgoing neighbor v_l to traverse another path. If v_l also replies that it can not meet the QoS constraints and v_j does not have other outgoing links, then v_j informs its previous hop node v_i of this fact. Then v_i can send a probe message to another of its another outgoing links. The procedure for finding a feasible path terminates with one of two results - it finds a feasible path to the destination or the source node has no more outgoing links to probe. However, this method may require an excessive amount of search time because it traverses possible paths one at a time.

Multiple probing is a method in which the source initiates several probe messages toward its destination. In order to traverse paths in parallel, the source node can increase the number of probe messages to np. Figure 2 shows an example of multiple probing. Thick lines such as $e_{2,3}$ and $e_{4,5}$ represent highly congested links. In (a) of Figure 2, v_8 sends two DFP probe messages (solid and gray arrows) to its two smallest-height neighbors. In (b), v_2 forwards the DFP via $e_{1,2}$ because $e_{2,3}$ is highly congested. v_5 also forwards the DFP via $e_{5,6}$ which is the next-to-minimum-height neighbor. In (c), v_6 replies with a DNP to v_5 because it does not have any available outgoing links toward the destination. v_1 forwards the DFP to its minimum-height neighbor via $e_{1,3}$. In (d), destination v_0 receives the DFP. Note that v_5 also replies with a DNP to v_8 because it does not have any available outgoing links toward destination. In (e), destination v_0 sends an DRP to v_3. v_8 sends a DFP to its third-minimum-height neighbor because it receives a DNP from its next-to-minimum-height neighbor. (f) shows a primary path from v_8 to v_0 that meets QoS constraints. Note that v_7 replies with a DNP to v_8 eventually because it does not have any feasible outgoing links toward the destination v_0. When v_8 receives the DNP from v_7, v_8 can discard it.

3.4 Feasible Path Reservation

When each node forwards a DFP message to its neighbor, it makes forward reservation of network resources. Forward reservation uses network resources only in the reservation pool. If a link does not find sufficient resources in the reservation pool, then it attempts to reserve additional resources, which are then added to the reservation pool (as many resources as required by the current DFP message).

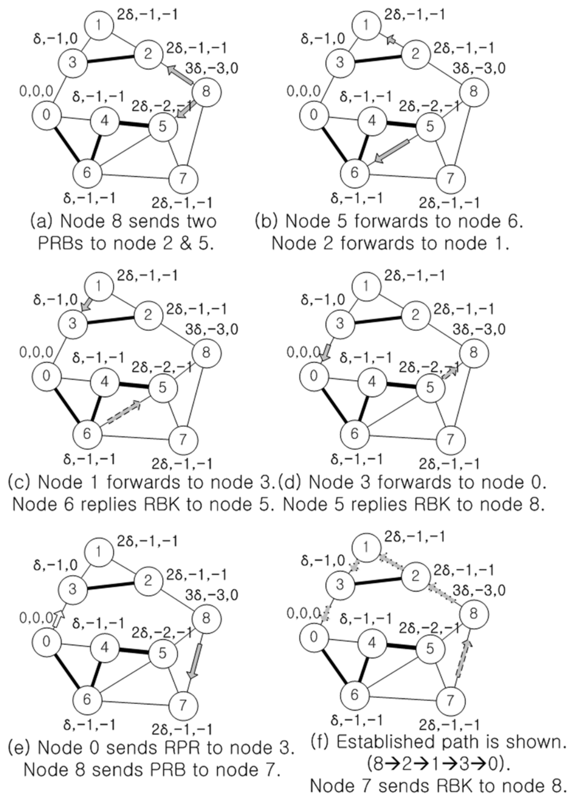

Figure 2. Example of multiple probing.

When each intermediate node receives a response to an DRP probe message, it reserves network resources from the reservation pool and forwards that information to the source node.

4 Backup Path Maintenance for QoS-Aware Routing

4.1 Maintaining Backup Paths

In order to provide high availability service, a QoS-aware routing algorithm should provide alternate feasible paths toward the destination node. One method for accomplishing this is to reserve a backup path B_k for each stream S_k that is edge-disjoint with the primary path P_k, as in the second level redundancy technique used in the TBP protocol [2]. However, maintaining and reserving an up-to-date feasible backup path in a MANET is very complex and wasteful of resources. Furthermore, an intermediate node that detects a broken link must immediately notify the source node by sending it a feedback message. Until the source receives the feedback message, it will fail to

0-7695-2561-X/06 $20.00 © 2006 IEEE

properly service data streams. To address this problem, we propose the use of localized backup path maintenance of backup paths using the local maintenance functionality of PDR.

Four control messages are defined to support the maintenance of backup paths. MBP(Maintanance/Backup-path Probing) is a backup path probe message used to find a backup path when a node detects that a link is going to be broken. MBP is used to find a local detour to the destination in order to bypass any weak links. MNP(Maintenance/Nack of Probing) is a NACK of backup-path probing message that states that there are no feasible paths via the requested link. MNP is the response to a MBP. MRP(Maintenance/Response of Probing) is a response to a backup path probe message sent by the destination. MRP is used to inform the node that originated the MBP that a detour has been found. REL is a release message that releases the resources reserved for a data stream. The source sends a REL message toward the destination to release all resources reserved by a data stream.

Let us consider an example. Figure 3 shows a portion of a path that satisfies the QoS requirements of a data stream S. In (a) of Figure 3, the original primary path P_{ori} is (v_1, v_2, v_5, v_4). In (b), v_5 moves away from v_2. When the link stability of $e_{2,5}$ is detected as progressively getting worse, v_2 initiates the maintenance procedure. As a result of maintenance, the stream S reserves bandwidth on links $e_{2,3}$ and $e_{3,4}$ in the reservation pool. In (c), even though v_2 loses its connection to v_5, it can continue to forward packets from v_1 to v_4 via $e_{2,3}$ without any additional operations. Therefore, S can be serviced continuously.

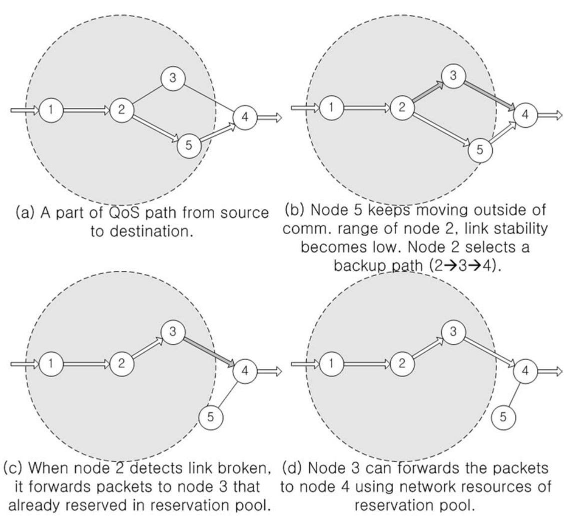

Figure 3. On-demand maintenance of a data stream.

In order to find backup paths in an on-demand manner, a QoS-aware routing algorithm should able to estimate link stability. For this purpose, the advanced signal strength based link stability estimation model(ASBM)[9] or pilot signal based link stability estimation model(PBM)[10] can be used. If the underlying radio device provides signal strength information, then ASBM may perform better than PDM. However, if the underlying radio device does not provides signal strength information, then only PBM can be used.

Figure 4 shows an example of backup path maintenance. There are three sources denoted with 'S' marks and a destination denoted as 'D'. White arrows represent data streams from each of these sources, and gray arrows represent the reservation pool for each link. The initial status is shown in

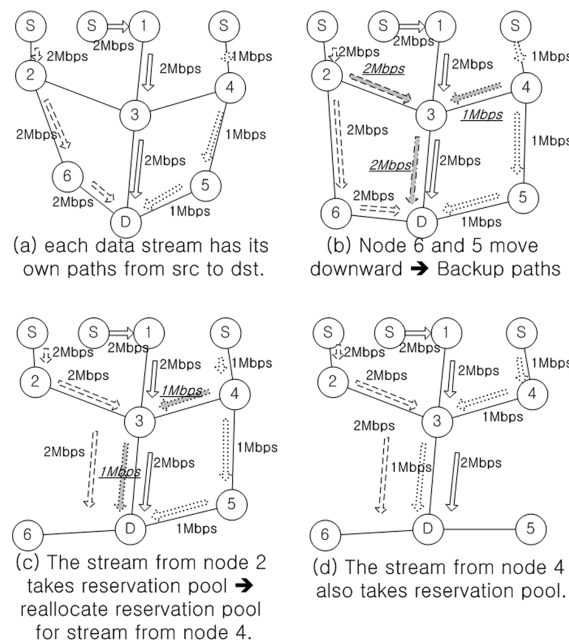

Figure 4. Example of backup path maintenance.

(a) of Figure 4. At this time, because v_6 and v_5 are moving, link stabilities of both $e_{2,6}$ and $e_{4,5}$ are getting weaker. However, since the link stabilities are still greater than the threshold, they do not need to initiate backup path maintenance. As time goes by, v_2 detects that the link stability of $e_{2,6}$ is less than the threshold in (b) of Figure 4. v_4 also detects that the link stability of $e_{4,5}$ is less than the threshold. Therefore v_2 and v_4 initiate backup path maintenance procedures. Because v_2 has only one outgoing link excluding $e_{2,6}$, which is a link used in the current primary path, v_2 sends a backup path probe message MBP via $e_{2,3}$.

Similarly, v_4 also sends a MBP via $e_{3,4}$. Because sufficient bandwidth for $e_{2,3}$ is not available in the reservation pool, v_2 adds 2Mbps bandwidth for $e_{2,3}$ into the reservation pool. Assume that v_3 can accept the backup path request. Suppose that v_3 receives the MBP from v_2 before it receives the MBP from v_4. v_3 forwards the MBP with a 2Mbps forward backup path reservation request. Then destination v_D also initiates a reservation for $e_{3,D}$ in the reservation pool and replies that the path has been reserved. Next, when v_3 receives the MBP from v_4, it simply forwards the MBP via $e_{3,D}$ because 2Mbps bandwidth for $e_{3,D}$ is already in the reservation pool, and this is sufficient for the 1Mbps request.

Suppose that v_3 receives the MBP from v_3 before it receives the MBP from v_4. In this case, 1Mbps bandwidth for $e_{3,D}$ is entered into the reservation pool. However, after processing the request from v_4, that resource in the reservation pool is changed to 2Mbps in order to support the request from v_2. In (c) of Figure 4, the stream from v_2 can be transmitted without disruption of service by using resources in the reservation pool for its primary path. Note that, because the stream from v_2 uses $e_{3,D}$'s bandwidth in the reservation pool, there is no remaining resource for $e_{3,D}$ in the reservation pool. Therefore, v_3 has to reinitialize 1Mbps in the reservation pool for the stream from v_4 and v_D also has to initialize 1Mbps in the reservation pool. In (d), the reinitialized reservation pool becomes a primary path, and there are no further entries in the reservation pool because no backup paths are required and no forward reservation is performed. The following shows a pseudocode description of the backup path maintenance procedure. Figure 5 shows reactions of a node v_k that receives MBP message, figure 6 shows reactions of a node v_k that receives MNP message and figure 7 shows reactions of a node v_k that receives MRP message.

```
recvMBP(packet p) {
    if(isReceived(p)) {
        drop(p);
        return;
    }
    if(p.DST = v_k) {
        sendMRP(v_prev, p);
        return;
    } else if (v_k has a feasible path to p.DST) {
        sendMRP(v_prev, p);
        return;
    }
    while((v_nxt = selectNextHop()) ≠ NULL) {
        if(e_{k,v_nxt} meets constraints of p) {
            forwardReservationInResvPool(p);
            updateConstraints(p);
            storeNextHop();
            sendMBP(v_nxt, p);
            return;
        }
    }
    sendMNP(v_prev, p);
}
```

Figure 5. Pseudo code for recvMBP.

4.2 Release of Reserved Resources

Normally, a source sends a REL message to the destination to release all reserved resources. However, some resources should be released before the source closes the established connection because a path may become disconnected. In such a case, some resources that can no longer be used will remain reserved. Therefore, each node must check the status of the streams, and if it does not receives packets for N^1 times, then it should release those resources and merge them into the reservation pool. Afterwards, if it still does not receive data streams for N^2 times, then it should completely remove those resources from the reservation pool.

Resources reserved by forward reservation are released when RBK or MNP messages are received. Note that resources reserved for backup paths will often not be used. In order to release such unused resources, each node maintains a timer that expires at specific times. After reservation of a backup path, the timer is started. Then, if the timer expires, those backup path resources can be released in the reservation pool. Each node should send periodic update messages to nodes on the backup path before the timer expires in order to prevent the premature release of backup path resources. Note that all reserved resources can be released individually.

5 Conclusion

In this paper, we propose a QoS-aware routing method for MANETs based on the use of a reservation pool. A backup path scheme in which a node that detects a broken primary path initiates a search for a new backup path may not be satisfactory because of the excessive delay incurred. On the other hand, reservation of a fixed backup path for each primary path may be wasteful of precious network resources. However, the proposed QoS-aware routing method can recover quickly from broken links and reduce the amount of wasted network resources by using a reservation pool. Furthermore, by incorporating pseudo-distance routing and its localized reconstruction algorithm, the proposed method can bypass the time-consuming primary path

```
recvMNP(packet p) {
   if(isReceived(p)) {
      drop(p);
      return;
   }
   if(v_k = predecessor of the initator) {
      setInitiator(v_k, p);
   }
   releaseResvPool(p);
   while((v_nxt = selectNextHopExceptPreviously-
Selected()) ≠ NULL) {
      if(e_{k,v_nxt} meets constraints of p) {
         fowardReservationInResvPool(p);
         updateConstraints(p);
         sendMBP(v_nxt, p);
         return;
      }
   }
   sendMNP(v_prev, p);
}
```

Figure 6. Pseudo code for recvMNP.

breakage feedback and backup path switchover processes. This paper has presented preliminary ideas on the effective support of QoS on MANETs. Feasibility testing for QoS on MANETs will be addressed as part of future work.

6 Ackowledgement

This work was supported by Korea Research Foundation Grant (KRF-2004-041-D00586).

```
recvMRP(packet p) {
   if(isReceived(p)) {
      drop(p);
      return;
   }
   reserveResources(p);
   if(v_k ≠ initiator of maintenance phase) {
      sendMRP(v_prev, p);
      return;
   }
}
```

Figure 7. Pseudo code for recvMRP.

References

[1] C.R. Lin and J. Liu, "QoS Routing in Ad Hoc Wireless Networks", *IEEE J. on Selected Areas in Comm.*, Vol. 17, No. 8, pp. 1426–1438, Aug. 1999.

[2] S. Chen and L. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks", *IEEE J. on Selected Areas in Comm.*, Vol. 17, No. 8, pp. 1488–1505, Aug. 1999.

[3] G-S.Ahn, A. T. Campbell, A. Veres and L-H. Sun, "Supporting Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN)", *IEEE Trans. Mobile Computing*, Vol. 1, No. 3, pp. 192–207, July-Sept. 2002.

[4] C.P.P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers", *Proc. ACM SIGCOMM '94*, pp. 234–244, Sept. 1994.

[5] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", *Proc. IEEE INFOCOM '97*, 1997.

[6] E. Gafni, D. Bertsekas, "Distributed Algorithms for Generating Loop-Free Routes in Networks with Frequently Changing Topology", *IEEE Trans. Comm*, Vol. COM-29, No. 1, pp. 11-18, Jan. 1981.

[7] M.-G. Lee and S. Lee, "Pseudo-Distance Routing(PDR) Algorithm for Mobile Ad-hoc Networks", *Proc. ITC-CSCC2005*, Vol. 2, pp. 797–798, July 2005.

[8] M.S. Corson and V. Park, "An Internet MANET Encapsulation Protocol (IMEP)", http://tools.ietf.org/html/draft-ietf-manet-imep-spec-00.txt, Internet draft, Aug. 1998.

[9] G. Lim, K. Shin, S. Lee, H. Yoon and J. S. Ma, "Link Stability and Route Lifetime in Ad-hoc Wireless Networks", *Proc. Int'l Conf. on Parallel Processing Workshop*, pp. 116–123, Aug. 2002.

[10] C-K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks", *Wireless Personal Communications*, Vol. 4, No. 2, pp. 1–36, Mar. 1997.

IEEE COMPUTER SOCIETY