Distributed Bandwidth Routing for Data Streaming across Multiple Routes for Improving Call Blocking Rate and Channel Utilization in Wireles Ad Hoc Networks

C. Richard LIN and *Yi-Xiang HUANG*

Abstract: In this paper, we propose a bandwidth-constrained routing algorithm to aggregate the bandwidth of multiple wireless links by splitting a data flow across multiple paths at the network layer. That is, it allows the packet flow of a source-destination pair to be delivered over multiple bandwidth routes with enough overall resources to satisfy a certain bandwidth requirement. Our algorithm considers not only the QoS requirement, but also the cost optimality of the routing paths to improve the overall network performance. We have analyzed the performance characteristics of the aggregation scheme and demonstrated significant gain when the links being aggregated have similar bandwidth and latency. Extensive simulations show that high call-admission ratio and resource utilization are achieved with modest routing overheads. This algorithm can also tolerate the node moving, joining, and leaving.

Keywords: multi-path routing, quality-of-service (QoS) routing, ad hoc network

Correspondent/Affiliation:

Yi-Xiang Huang
Department of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung 804, TAIWAN
Email: yxhuang@wamis0.cse.nsysu.edu.tw
TEL: +886-7-525-4339
FAX: +886-7-525-4301

Distributed Bandwidth Routing for Data Streaming across Multiple Routes for Improving Call Blocking Rate and Channel Utilization in Wireless Ad Hoc Networks

Chunhung Richard Lin and *Yi-Xiang Huang*
Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, TAIWAN
Email: lin@mail.nsysu.edu.tw; yxhuang@wamis0.cse.nsysu.edu.tw

Abstract: *In wireless networks, the bandwidth is likely to remain a scarce resource. We foresee scenarios wherein mobile hosts will require simultaneous data transfer across multiple links to obtain higher overall bandwidth. A call request of a connection in a wireless network is blocked if there exits no bandwidth route. This blocking does not mean that the total system bandwidth capacity is less than the request, but that there is no path in which each link has enough residual unused bandwidth to satisfy the requirement. Like the routing in a datagram network, if packets of a virtual circuit can stream across multiple paths, we can select multiple bandwidth routes such that the total bandwidth can meet the requirement of a source-destination pair. Therefore, even though there is no feasible single path for a bandwidth-constrained connection, we may still have a chance to accept this one if we can find multiple bandwidth routes to meet the bandwidth constraint. The cost we must pay is the overheads on the packet switching in each node, and the packet reordering at the destination side. But we can improve the bandwidth utilization and the call blocking rate. In this paper, we propose a bandwidth-constrained routing algorithm to aggregate the bandwidth of multiple wireless links by splitting a data flow across multiple paths at the network layer. That is, it allows the packet flow of a source-destination pair to be delivered over multiple bandwidth routes with enough overall resources to satisfy a certain bandwidth requirement. Our algorithm considers not only the QoS requirement, but also the cost optimality of the routing paths to improve the overall network performance. We have analyzed the performance characteristics of the aggregation scheme and demonstrated significant gain when the links being aggregated have similar bandwidth and latency. Extensive simulations show that high call-admission ratio and resource utilization are achieved with modest routing overheads. This algorithm can also tolerate the node moving, joining, and leaving.*
Keywords: multi-path routing, quality-of-service (QoS) routing, ad hoc network

1. INTRODUCTION

In more recent years, the interest in wireless ad hoc networks is grown along with wireless communication devices, such as laptop computers, PDAs, and Bluetooth devices. A wireless ad hoc network can be a collection of wireless mobile hosts forming a temporary network without infrastructure. Each mobile host acts as a router, forwarding data packets for other nodes. Because of the dynamic topology, it is difficult to design a routing protocol. The routing protocols in wireless ad hoc networks can be categorized into two types. One is table-driven (e.q., DSDV[25]) and the other is on-demand (e.q., DSR[15], AODV[26]). However, all of the above are only deal with the best-effort traffic. Connections that need quality-of-service (QoS) requirement are not supported.

In wireless ad hoc networks, due to its dynamic nature and no centralized admission control, it is more difficult to establish a QoS connection than in cellular networks. [20] proposed a QoS routing protocol based on table-driven routing protocol that adds bandwidth information into routing tables. Like table-driven protocol, periodically exchange routing information, and while a connection request arrives, looks up the routing table to decide the next hop. There is a difference that it establishes a route, which satisfies the QoS requirement. Similarly, [19] is based on on-demand routing protocol that adds in bandwidth information into Route_Request (RREQ) messages, and broadcasts the messages initialized by source node to its neighbor nodes. Finally, RREQ messages reach destination node. Then one that is satisfied QoS requirement is picked and Route_Response (RREP) message is sent along the picked route to source node. Both two protocols are simple and available, but they also inherit the drawbacks that the overhead of maintaining routing table in table-driven routing protocol even though without any connection in network and broadcasting RREQ messages in on-demand routing protocol.

[6] proposed a ticket-based probe (TBP) scheme to search a QoS route. A ticket is the permission to search one path. The source node issues a number of tickets base on the QoS requirement. Probes are sent from source toward destination to search for a low cost path that satisfies the QoS requirement. At the intermediate node, a probe with more than one ticket is allowed to be split into multiple ones, each search a different downstream sub-path. An intermediate node splits the tickets according to the local state and the

end-to-end information updated periodically by a distance-vector protocol. If all probes arrive destination, it will selec a single path that is the lowest cost and satisfies the QoS requirement. The goal of designing tickets is to bound the number of messages to be sent. However, [6], [19], and [20] are single path routing.

In bandwidth-constrained routing, bandwidth utilization of single path QoS routing protocol is low, because the residual bandwidth of links can not satisfy the new connection request. It is essential to effectively use resource in wireless ad hoc networks, since it is limited. In this paper, we propose a bandwidth-constrained routing protocol that improves the bandwidth utilization by searching for a multi-path to satisfy QoS requirement. This protocol includes the scheme to search multi-path, splitting algorithms, and management of route.

The rest of the paper is organized as follows. The system models are given in Section 2. The idea of multi-path routing is described in Section 3. The simulation results are presented in Section 4. Finally, Section 5 concludes the paper and feature work.

2. SYSTEM MODEL

One can present a network by a graph $G = (V, E)$, where V is a set of nodes that are interconnected by a set E of full-deplex directed communication links. A link (a, b) means that nodes a considers node b as a valid neighbor for packet forwarding and adjusts transmitting power according to their distance, and vice versa. That is, they are in the transmission range of each other. Assume that the effective transmission distance of every node is uniform. V and E are changing over time where nodes move, join, and leave (or power off). Each node has a unique identifier and has at least one transmitter and one receiver. We assume there exists a neighbor discovering scheme. Any node can know its neighbors by periodically transmitting a BEACON packet identifying itself. Neighboring nodes share a common wireless medium, and packets are transmitted by a local broadcast. There is a MAC protocol to be assumed to resolve the medium contention, support resource reservation, and ensure that, among the neighbors in the local broadcast range, only the intended receiver receives the packet, and the other neighbors discard the packet.

A QoS connection (call) is a connection that has an end-to-end performance requirement such as a delay or bandwidth constraint. The QoS metrics we consider here are the bandwidth only. This is because bandwidth guarantee is one of the most critical requirements for real time applications. We assume every node has the precise information about its local state. A node i keeps the up-to-date local state about all outgoing links. The state information of link (i, j) includes: 1) $BW(i, j)$, the residual (unused) bandwidth of the link; and 2) $cost(i, j)$, the transmission cost, which can be simply one as a hop count or function of the link utilization. The bandwidth and cost of a path $P = i \rightarrow j \rightarrow \cdots \rightarrow k \rightarrow l$ are defined as follows:

$$BW(P) = \min\{BW(i, j), \cdots, BW(k, l)\}$$

$$cost(P) = cost(i, j) + \cdots + cost(k, l).$$

The main objective of the routing algorithms presents here is to set up connections (i.e., virtual circuits) in a network when sessions are initiated, and to maintain them during the lives of the sessions. The set-up process includes a reservation of network resources for the connection, if these resources are not available the set-up fails, or in other words, the connection is blocked. Our aim is to minimize the blocking probability for a connection.

Given a source-destination pair and a bandwidth requirement B, the problem of bandwidth-constrained routing is to find a feasible path P between the source and the destination such that bandwidth(P) $\leq B$. When multiple feasible paths exist, finding the least-cost path is an NP-complete problem. The routing decision influences the blocking probability. For example, if two routing algorithms differ in the length of the selected routes, and thus in the consumption of network resources, usually the probability of successfully finding sufficient resources for more connections is lower for the algorithms that wastes resources. Thus, the quality of the routing decision also effects the blocking probability in the network.

In an ad hoc wireless network, the stationary or slowly moving nodes are likely to exist continuously. Such links are called stationary links. Comparatively, links between fast moving nodes are likely to exist only for a short period of time. Such links are called transient links. An routing path is constructed by a sequence of wireless links. These links should be stationary whenever possible in order to reduce the probability of a path breaking when the network topology changes. That is, the cost of a transient link should be higher than that of a stationary link. Because wireless links in ad hoc network are unpredictable, it is impractical to exactly determine a link to be either stationary or transient. A simple approximation approach discussed in [32] is based on an expirical observation that the links that are just formed are more likely to be broken than the links that have already existed for some time. Thus, whenever a new link is formed, it is set as a transient link. After the link remains unbroken for a time period, it is changed to be a stationary link.

The bandwidth of wireless link is quite limited and unstable. For a network to deliver QoS guarantees, it must reserve and control resources. A virtual circuit (VC) can be accepted if not only it has enough available bandwidth, but also it can not disrupt the existing QoS VCs. Otherwise, it will be blocked.

Sometimes, a network has enough bandwidth capacity to accept the traffic of a new VC. However, the VC may be still blocked because of no existence of a bandwidth route. If packets of a VC traffic are allowed to travel along different paths like the routing for datagram traffic, we can construct multiple routes for a VC. Thus, the bandwidth requirement can be shared among these routes.

Consider the example depicted in Figure 1. The number on each link is the residual bandwidth (say, megabits per second) of the link. Suppose S wants to stream data to T across to the ad hoc networks, and the bandwidth requirement is 10 units. Observe that this call request will be rejected because there is no path to meet this requirement (the residual bandwidth on any outgoing link is less than the requirement). However, if we let the traffic be streamed over multiple outgoing links to enhance the overall bandwidth, then the call can be accepted as shown in Figure 1. Therefore, even though there is no feasible *single* path for a bandwidth-constrained connection, we may still have a chance to accept this one if we can find *multiple* bandwidth routes to aggregate the bandwidth to meet the total bandwidth constraint.

Several questions arise when considering the possible ways to achieve this effect. Should the data stream be split into multiple streams at the application level? In this case, the application would open multiple connections across different network routes and be responsible for splitting the data stream at the server and merging it properly at the client. This approach might be too cumbersome and restrictive since the number of connections might have to be decided in advance in order to figure out how many flows to split the traffic into. Alternatively, should the splitting be done at transport layer or at the network layer? In essence we are considering this general question of striping a connection across multiple paths at the network layer for bandwidth aggregation. The concept is shown in Figure 2. The cost we must pay is the overheads on the packet switching of multiple data streams in each node, and the packets reordering and merging at the destination side. But the significant gain we can get is to enhance the bandwidth utilization. This is especially important for wireless channel because of its much smaller bandwidth than the wired channel. For the issue of the information imprecision for the link state (either local or global), it is not considered in the paper.

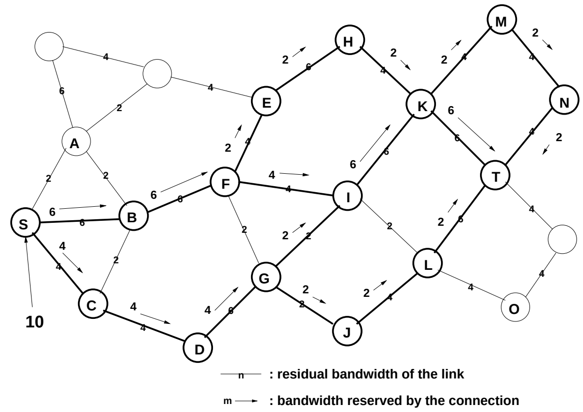

Figure 1: Virtual Circuit

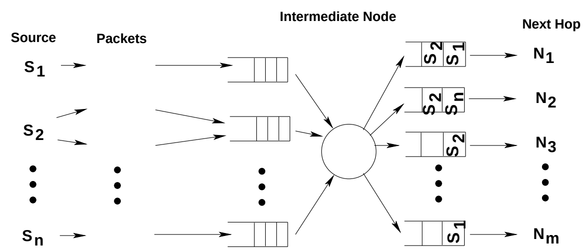

Figure 2: Concept of splitting in transport layer or network layer

3. BANDWIDTH-CONSTRAINED ROUTING

3.1. Look for the Route

We use the request/reject scheme to look for a multipath in an ad hoc network. Initially, the source node sends the request messages to the next hops according to the local state and split algorithm. When an intermediate node receives a request message, the node determines, which can be the next hop. If existence, it sends the request messages to the next hop. Here, the next hop may be not olny one. Else, a reject message will be backed to its former node, until all request messages arrive the destination node. If the intermediate node receives the reject message, it must try to forward the request message to its neighbor nodes unless the former node and the nodes that have be rejected. If it can not forward the request message any more, it will return the reject message to its former node, and so on. Until the source node receives any reject message and cannot forward again, we will declare this connection failure.

Figure 3 illustrates a simplified example of request/reject scheme. Node s wants to establish a QoS connection needed 10 unit of bandwidth toward node t and sends a request with 10 unit of bandwidth toward destination t (denoted by $req_{st}(10)$). When node i receives the $req_{st}(10)$ message and residual units of bandwidth of the links between its neighbor node can not satisfy the QoS requirement but the summary of residual bandwidth of them satisfies the QoS requirement, it splits the request message to $req_{st}(4)$ and $req_{st}(6)$. As node j receives the message $req_{st}(6)$, it originally sends the message to node k. But node k can not continue to forward the request message to its neighbor node and then node k returns the reject message $rej_{st}(6)$ to node j. Hence node j re-forwards the $req_{st}(6)$ message to node t. Until all request messages arrive the destination, the connection establishes successfully.

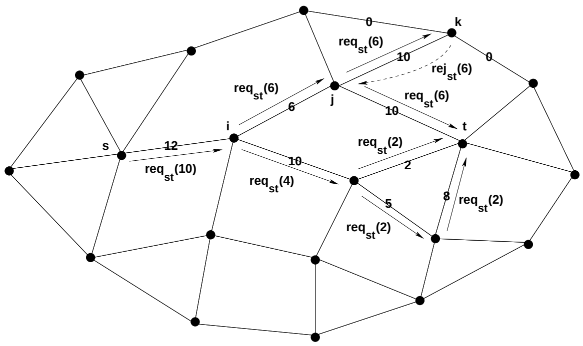

Figure 3: Topology of Ad Hoc Network

In the previous scheme, messages may cycle around loops. Three possible approaches discussed in [6] to avoid cycling infinitely are: 1) at most one message is allowed to be sent to every outgoing link; 2) the number of hops that a message can traverse is bounded. 3) message records the traversed paths that will be discard. We use the third one in our simulation.

3.2. Data Structure

The data structure of request/reject message is shown in Table 1. The "rejected" field is a set of the nodes that have rejected this connection request to avoid forwarding the request message to a node that has rejected this connection request.

Parameters	Description
id	system unique identification for the connection request
type	request or reject message
s	source node
t	destination node
B	initial bandwidth requirement
B'	current bandwidth requirement
path	a linked list of nodes that have traversed so far
rejected	a set of nodes that have be rejected
cost	accumulated cost of the path traversed so far

Table 1: Data Structure

3.3. Split Algorithms

Consider a connection request whose source and destination are s and t, respectively. The request message of the connection, which arrives the intermediate node i is denoted by p_i. Suppose N_i is the set of neighbor nodes of node i and the bandwidth requirement of received message is B_{p_i}. R_p is the linked list of nodes that message p_i has traveled so far. J_p is the set of nodes that message p_i has be rejected so far. Defines that candidate set R_i^p is the nodes, which are the neighbors of node i and the candidates for request message p_i. In another word, $R_i^p = \left\{ j \mid \left\{ N_i - R_p - J_p \right\} \right\}$. We will select the next nodes form the candidate set R_i^p. Lets that $j \in R_i^p$ and link(i, j) is the link between node i and j. The residual bandwidth and cost of link(i, j) are b_{ij} and c_{ij}, respectively.

Figure 4 is an example that node E forwards a request message with bandwidth requirement 10 to node A. Now, node A determines, which nodes in its neighbor nodes set are the next hop and how many unit of bandwidth to be split. There are three nodes in neighbor nodes set of node A.We will use the following four splitting algorithm to determine that. In the figure, (m, n) is the parameter of link, where m is the residual bandwidth and n is the cost of link. In this example, $R_i^p = \{B, C, D\}$.

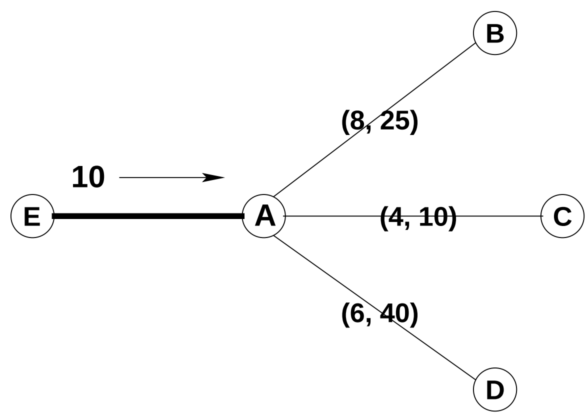

Figure 4: An example of forwarding request message.

Algorithm A

Before splitting the request message, we sort b_{ij} for all $j \in R_i^p$ and obtain a sorted decreasing list of residual bandwidth L_b. Let $L_b = \{b_{i1}, b_{i2}, \ldots, b_{in}\}$, where $n = |R_i^p|$ and $b_{i1} \geq b_{i2} \geq \ldots \geq b_{in}$. In order to limit the number of split, we select the next nodes from the largest residual bandwidth such that $B = \sum_{j=1}^{m} b_j$, where 1≤m≤n. The algorithm is as following:

See Figure 4(a). $L_b = \{8, 6, 4\}$ There are two request messages forwarded. One is forwarded to node B with bandwidth requirement 8 and another is forwarded to node D with bandwidth requirement 2.

```
bandwidth_requirement = B_{p_i};
for (int j=1; bandwidth_requirement>0; j++)
{
    int split_bandwidth = L_b[j]->b_{ij};
    if (split_bandwidth > bandwidth_requirement)
        split_bandwidth = bandwidth_requirement;

    p_j->B' = split_bandwidth;
    bandwidth_requirement -= split_bandwidth;
}
```

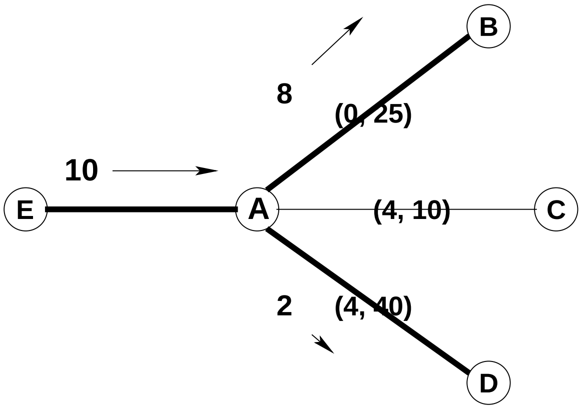

Figure 4(a): An example of forwarding the request message by splitting algorithm A.

Algorithm B

Algorithm B is similar to A. We sort c_{ij} for all $j \in R_i^p$ and obtain a sorted increasing list L_c. Let $L_c = \{c_{i1}, c_{i2}, \ldots, c_{in}\}$ and $c_{i1} \le c_{i2} \le \ldots c_{in}$. In order to obtain the lower cost route, we choose the next hop from the smallest link cost such that $B = \sum_{j=1}^{m} b_j$, where 1≤m≤n. The algorithm is as following:

```
bandwidth_requirement = B_{p_i};
for (int j=1; bandwidth_requirement>0; j++)
{
    int split_bandwidth = L_c[j]->b_{ij};
    if (split_bandwidth > bandwidth_requirement)
        split_bandwidth = bandwidth_requirement;

    p_j->B' = split_bandwidth;
    bandwidth_requirement -= split_bandwidth;
}
```

See Figure 4(b). $L_c = \{10, 25, 40\}$ There are two request messages forwarded. One is forwarded to node C with bandwidth requirement 4 and another is forwarded to node B with bandwidth requirement 6.

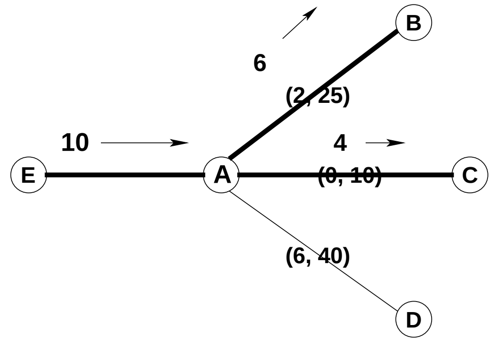

Figure 4(b): An example of forwarding the request message by splitting algorithm B.

Algorithm C

Define $r_{ij} = \dfrac{b_{ij}}{c_{ij}}$. We sort r_{ij} for all $j \in R_i^p$ and obtain a sorted decreasing list of the ratio L_r. Let $L_r = \{r_{i1}, r_{i2}, \ldots, r_{in}\}$, where $n = |R_i^p|$ and $r_{i1} \ge r_{i2} \ge \ldots \ge r_{in}$. In order to limit the number of split and obtain a lower route, we select the next node from the largest ratio such that $B = \sum_{j=1}^{m} b_j$, where 1≤m≤n. The algorithm is as following:

```
bandwidth_requirement = B_{p_i};
for (int j=1; bandwidth_requirement>0; j++)
{
    int split_bandwidth = L_r[j]->b_{ij};
    if (split_bandwidth > bandwidth_requirement)
        split_bandwidth = bandwidth_requirement;

    p_j->B' = split_bandwidth;
    bandwidth_requirement -= split_bandwidth;
}
```

In Figure 4(c), (m, n q) is the parameter of link, where m is residual bandwidth, n is link cost and q is the bandwidth-cost ratio. $L_r = \{0.4, 0.32, 0.15\}$ There are two request messages forwarded: one is forwarded to node C with bandwidth requirement 4 and another is forwarded to node B with bandwidth requirement 6.

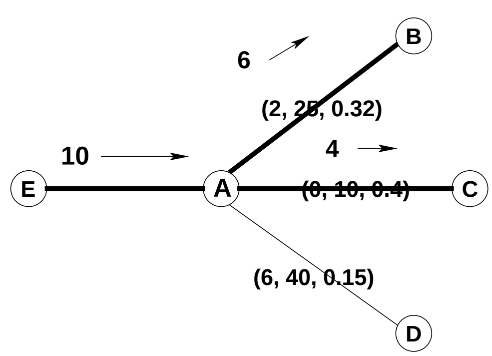

Figure 4(c): An example of forwarding the request message by splitting algorithm C.

Algorithm D

Supposes that $j \in R_i^p$ and defines L_b as the same as in Algorithm A. The bandwidth requirement that node i forwards to node j is $\left\lceil \frac{b_{ij}}{\sum_{k \in R_i^p} b_{ik}} \times B_{p_i} \right\rceil$. The algorithm is as following:

bandwidth_requirement = B_{p_i};
for (int j=1; bandwidth_requirement>0; j++)
{
    int split_bandwidth = $\left\lceil \frac{L_b[j]->b_{ij}}{\sum_{k \in R_i^p} L_b[k].b_{ik}} \times B_{p_i} \right\rceil$;
    if (split_bandwidth > bandwidth_requirement)
        split_bandwidth = bandwidth_requirement;

    p_j->B' = split_bandwidth
    bandwidth_requirement -= split_bandwidth;
}

See Figure 4(d). $L_b = \{8, 6, 4\}$ There are three request messages forwarded. One is forwarded to node B with bandwidth requirement 5, one is forwarded to node C with bandwidth 1 and another is forwarded to node D with bandwidth requirement 4.

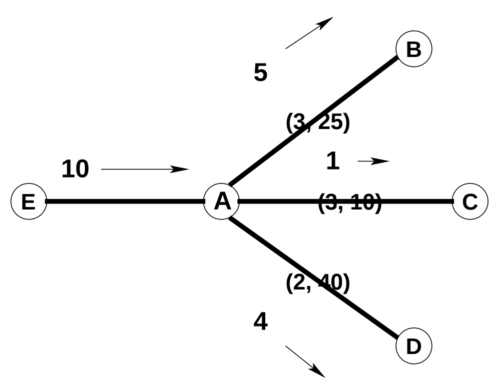

Figure 4(d): An example of forwarding the request message by splitting algorithm D.

3.4. Rerouting

Each node in ad hoc network is mobile. Once the link becomes disconnected due to that the packets can not be forwarded to the next hop. We should reconstruct the broken paths as soon as possible. During the reconstructing time, the QoS guarantee is disrupted. There are three approaches to reconstructing the broken paths: rerouting, path redundancy, and path repair. Rerouting is the common approach to deal with the problem of path breaking in ad hoc network.

Once an intermediate nose detects the broken link, it blocks the packet that wants to forward and then starts to reroute. The scheme of rerouting is as same as the routing, using request and reject message. The node detecting broken link forward the request message by splitting algorithms toward the destination. Until the new routing is established, the node starts to forward the packets in the buffer again. During establishing the new route, the buffer of this session may be full and the packets start to lost.

3.5. Soft State

Once the request message passes a node, the resource has been reserved. In the duration of searching a route, the reserved resource can be released by the reject message. But, after routing process, as the network topology changes dynamically, routing paths may be broken into pieces, and the network may even be partitioned. There is a problem how to release the reserved resource.

There are two way to solve this problem. One is that, as an intermediate node detects that a previous node of connection is not its neighbors any more, it sends the release message to free the reserved resource along with the downstream sub-paths toward destination. The other is that using soft states helps to release the unused resource automatically. In our simulation, we adopt the second method to avoid forwarding too many messages.

Every node in network maintain a connection table, which has an entry for every connection passing the node, containing the incoming and outgoing links, source and destination node ID, connection ID, reserved resource, and Time-to-Live (TTL). The description of every entry in connection table is shown in Table 2. If the data packets pass the node, TTL of entry will be refreshed, and if not refreshed, the entry will be deleted.

Entry	Description
id	system unique identification for the connection request
s	source node
t	destination node
incoming	a set of links which the node receive packets from
outgoing	a set of links which the node forward packets to
resource	the reserved resource for this connection
TTL	Time-to-Live

Table 2: Connection Table

4. SIMULATION AND RESULTS

The network topology used in our simulation is randomly generated. Thirty nodes are randomly placed within a $10 \times 10\ m^2$ area. The radius of service range of each node is 3.5 m. If the distance between two nodes is smaller than the service range, a link is added between them. The source node, destination nodes and bandwidth requirement of each connection are randomly generated. The bandwidth requirement is uniformly distributed in the range of [5, 15]. The cost of each link is uniformly distributed in [1, 200].

Five algorithms are simulated: a routing algorithm with single path and four multi-path routing algorithms that we propose. The routing scheme of single path algorithm is the same as request/reject scheme that multi-path routing algorithms use, but does not split bandwidth request, if there is not any link satisfying bandwidth requirement, that is the connection will be declared failure. The four routing algorithm with multi-path is described in Section 3.2.

4.1. Bandwidth Utilization

The interval time of connection generated has exponential distribution with mean 3 clocks. Once the connection is permitted by administrator control, the source node continuously sends packets to the destination node. AS the number of connections become later, the system is going to a full state, meaning that no connection can be permitted. The bandwidth utilization is defined as follows:

$$bandwidth\ utilization = \frac{number\ of\ link\ bandwidth\ in\ use}{total\ number\ of\ initial\ link\ bandwidth}$$

In Figure 5, every curve limits to a value of link utilization, respectively. As expected, the algorithms of multi-path have higher bandwidth utilization than algorithm of single path. This is because the algorithms with multi-path can allow the connection, which bandwidth requirement is smaller than residual bandwidth provided by a single link, but is larger than total number of residual bandwidth provided by all links between its neighbor nodes.

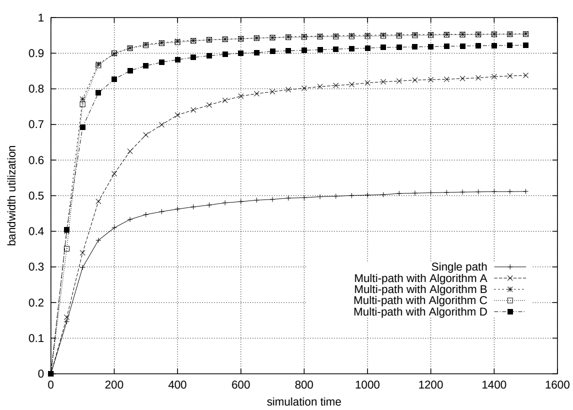

Figure 5: Bandwidth utilization.

4.2. View of the Connection

In Figure 6, the control message including request and reject message is counted. The figure shows that the number of control messages are sent to establish a connection. Because the better paths are used by former connections, the later connections need more control messages to establish route.

In the figure, the simulation environment can permit 15 connections that use multi-path algorithm but only permit 13 connections that use single path algorithm at most. Because the bandwidth is used by former connections, the later established connection needs more message to be sent. The number of virtual circuit that is established by algorithm D is larger than the other, so the number of messages is larger than the other. The single path algorithm establishes only one virtual circuit, so it needs the least number of messages.

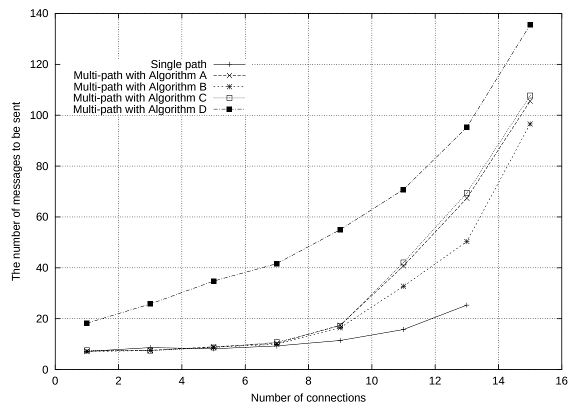

Figure 6: The number of message to be sent.

The bandwidth requirement of Figure 7 and 8 is exponential distribution. Figure 7 compares the percentage of intermediate nodes (includes source node) having more than one child in all intermediate nodes. Figure 8 compares the number of children that each splitting intermediate node (includes source node) has. In Figure 7, as the bandwidth requirement is larger, the splitting nodes is increasing, except the single path algorithm, which does not split any child. The condition of splitting with algorithm D is most serious because of loading balance and the other algorithms are so close because they are so simular.

In Figure 8, we want to show the condition of splitting, so do not simulate the single path algorithm. The number of children with multi-path algorithm A, B, C and D is about 2.2, but one with multi-path algorithm D is 3.3. That is because the algorithm D is designed to balance load, so that the data flow is split to more children.

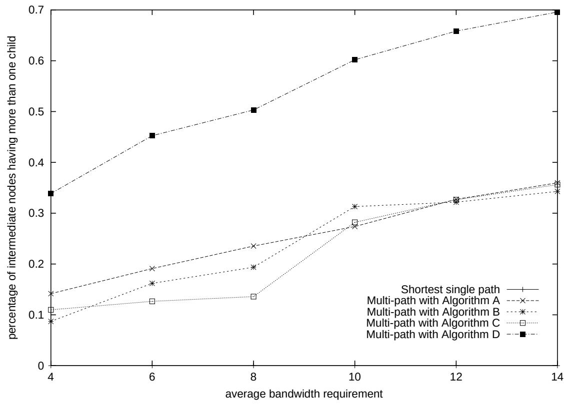

Figure 7: The number of nodes that have more than one child.

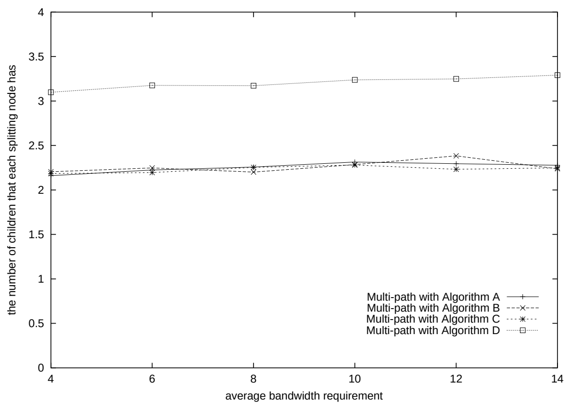

Figure 8: Average of children that each splitting nodes has.

4.3. Mobility Test

In our mobility model, every node stays at its current location for a period, called pause time. Then it randomly selects the new location and moves to it. When the node detects broken link on the route, it would block the data in the buffer and repair the route. The buffer size of every node is 100. If the buffer is full, the data starts to lose. If the route can not be repaired, this connection is forced terminated by administrator control. The forced terminate ratio is defined as follows:

$$forced\ terminate\ ratio = \frac{the\ number\ of\ forced\ terminated\ connections}{the\ total\ number\ of\ established\ connections}$$

In Figure 9, we adjust the mean of mobility rate, that is exponential distribution, and observe the forced terminated ratio. The forced terminate means that the connection, which administration control has permitted to join to the system is be terminated because of mobility. The pause time is uniformly distributed in the range of [1, 20 clocks]. Here, we only compare with single path and multi-path algorithm, which we choice algorithm C to represent. Figure 9 shows that the forced terminate ratio increase as mobility and the forced terminate ratio with multi-path algorithm is larger than one with single path algorithm because route established by multi-path algorithm passes more links, and increases the probability of link broken.

In Figure 10, we adjust the mean of pause time, that is exponential distribution, and observe the ratio of data lost. The data lost ratio is defined as follows:

$$data\ lost\ ratio = \frac{the\ number\ of\ data\ lost}{the\ total\ number\ of\ data\ be\ sent}$$

In the condition with high mobility, the data lost ratio is serious. The condition of splitting affects the ratio of data lost, too. The Algorithm D has the highest data lost ratio.

Finally, we know that the number of splitting bandwidth requirement is affected by mobility. If the algorithm is designed to split more bandwidth requirement, it will affected by mobility seriously. Under the mobile environment, multi-path algorithm is not a best ideal, but if we want to have higher utilization, multi-path algorithm is an approach to achieve it.

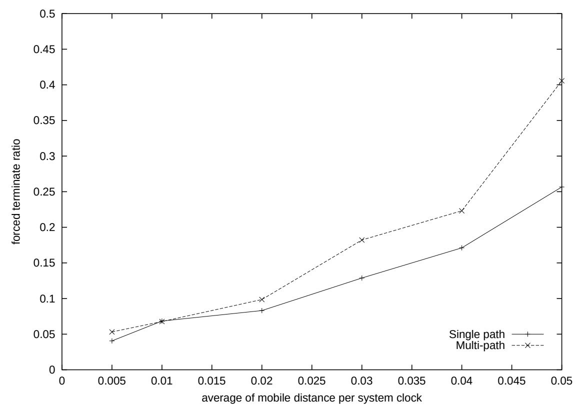

Figure 9: Force terminate ratio

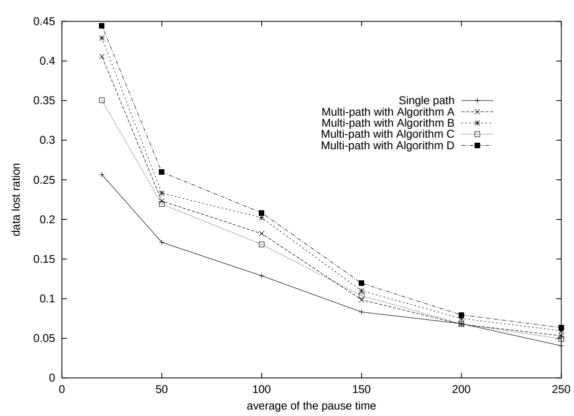

Figure 10: Data lost ratio

5. CONCLUSION AND FEATURE WORK

In this paper, the scheme of establishing route is simular to depth-first search algorithm, so it takes a lot of time to look up a route. In the future, we hope to discover good scheme to look up multi-path that satisfies QoS requirement.

The algorithm to split the bandwidth requirement does not have highest performance, but in some condition, we prefer to choice it. For example, considers the power consumption. In the future, We will apply this approach to solve the problem about power consumption.

REFERENCES

[1] H. Adiseshu and G. Parulkar and G. Vaghese, ''A reliable and scalable striping protocol'', *ACM SIGCOMM,* 1996, pp. 131-141.

[2] D. Bertsekas and R. Gallager, *Data Networks,* Pretice-Hall, Inc., 1987.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, ''A performance comparison of multi-hop wireless ad-hoc network routing protocols'', *Proceedings of IEEE ACM 1998*

[4] J. -H. Chang and L. Tassiulas, ''Energy Conserving Routing in Wireless Ad-hoc Networks'', *Proceedings of IEEE INFOCOM 2000*

[5] T. Chen, M. Gerla, and J. T. Tsai, ''QoS routing performance in a multi-hop, wireless networks'', *Proceedings of IEEE ICUPC 1997.*

[6] S. Chen, and K. Nahrstedt, ''Distributed quality-of-service routing in ad hoc network'', *IEEE Journal on Selected Areas on Communications,* August 1999, pp. 1488-1505.

[7] I. Cidon, R. Rom, and Y. Shavim, ''Ayalysis of multi-path routing'', *IEEE/ACM Transactions on Networking,* pp. 885-896, Dec. 1999.

[8] I. Cidon, R. Rom, and Y. Shavitt, ''Multi-path routing combined with resource reservation'', *Proceedings of IEEE INFOCOM '97,* pp. 92-100.

[9] E. S. Elmallah, H. S. Hassanein, and H. M. AboElFotoh, ''Supporting QoS routing in mobile ad hoc networks using probabilistic locality and load balancing'', *Proceedings of IEEE GLOBECOM 2001* pp. 2901-2906

[10] M. Gerla, X. Hong, and G. Pei, ''Landmark routing for large ad hoc wireless networks'', *Proceedings of IEEE GLOBECOM 2000,* pp. 1702-1706

[11] M. Gerla and S.-J. Lee, ''Split multipath routing with maximally disjoint paths in ad hoc networks'', *Proceedings of ICC '01, Helsinki, Finland,* June 2001.

[12] R. Guerin and A. Orda, ''QoS-based routing in networks with inaccurate information: Theory and alrotithms'', *Proceedings of IEEE INFOCOM 1997,* pp. 75-83

[13] C. Hou, ''Routing virtual circuits with timing requirements in virtual path based ATM networks'', *Proceedings of IEEE INFOCOM 1996,* pp. 320-328.

[14] D. B. Johnson, ''Routing in ad hoc networks of mobile host'', *Proceedings of IEEE WMCSA 1994,* Dec. 1994.

[15] D. Johnson and D. Maltz, ''Dynamic source routing in ad hoc wireless networks,'' *Mobile Computing,* Norwell, MA: Kluwer, 1996.

[16] S.-J. Lee and M. Gerla, ''Split multipath routing with maximally disjoint paths in ad hoc networks'', *Proceedings of IEEE ICC 2001* pp. 3201-3205

[17] W.-H. Liao et. al., ''A Multi-path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network,'' *Proceedings of IEEE ICN '00, CREF, Colmar, France,* July 2001.

[18] C. R. Lin, ''Admission Control in Time-Slotted Multihop Mobile Networks'', *IEEE Journal on Selected Areas in Communications,* October 2001

[19] C. R. Lin, ''On-Demand QoS Routing in Ad Hoc Wireless Networks'', *IEICE Transactions on Communications,* July 2000

[20] C. R. Lin and J. -S. Liu, ''QoS Routing in Ad Hoc Wireless Networks'', *IEEE Journal on Selected Areas in Communications,* August 1999.

[21] A. Nasipuri and S. R. Das, ''On-demand multipath routing for ad-hoc networks'', *Proceedings of the IEEE ICCCN '99, Boston,* Oct. 1999, pp. 64-70.

[22] R. Ogier, V. Ruenburg, and N. Shacham, ''Distributed algorithms for computing shortest pairs of adsjoint paths'', *IEEE Transactions on Information Theory,* vol. 39, no. 3, pp. 443-455, Mar. 1993.

[23] V. D. Park and M. S. Corson,''A highly adaptive distributed routing algorithm for mobile wireless networks'', *Proceedings of IEEE INFOCOM 1997,* pp. 1405-1413.

[24] M. R. Pearlman, Z. J. Haas, P. Scholander, and S. S. Tabrizi, ''On the impact of alternate path routing for load balancing in mobile ad hoc networks'', *Proceedings of ACM MobiHOC '00, Boston,* August 2000, pp.119-128.

[25] C. Perkins and P. Bhagwat, ''Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers'', *Proceedings of ACM SIGCOMM '94,* pp. 234-244.

[26] C. E. Perkins and E. M. Royer ''Ad-hoc on-demand distance vector routing'', *Proceedings of IEEE WMCSA 1999* pp. 90-100.

[27] G. Pei, M. Gerla, and T. Chen, ''Fisheye state routing: a routing scheme for ad hoc wireless networks'', *Proceedings of IEEE ICC 2000,* pp. 70-74

[28] D. S. Phatak and T. Goff, ''A novel mechanism for data streaming across multiple IP links for improving throughput and reliablility in mobile environments'', *Proceedings of IEEE INFOCOM 2002.*

[29] j. Raju and J. J. Garcia-Luna-Aceves, ''A new approach to on-demand multipath routing'', *Proceedings of IEEE ICCCN '99, Boston,* Oct. 1999, pp. 522-527

[30] A. Snoeren, ''Adaptive Inverse Multiplexing for Wide-Area Wireless Networks'', *IEEE Globecom,* 1999, pp. 1665-1672.

[31] N. Taft-Plotkin, B. Bellur, and R. Ogier, ''Quality of

service routing using maximally disjoint paths'', *Proceedings of the IEEE IWQoS '99, London UK* , June 1999, pp. 119-128.

[32]
C.-K. Toh, ''Associativity-based routing for ad-hoc mobile networks'', *Wireless Personal Commun.* 1997.

[33]
Z. Wang and J. Crowcroft, ''QoS routing for supporting resource reservation'', *IEEE Journal on Selected Areas on Communications,* Sept. 1996.

[34]
L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, ''RSVP: A new resource reservation protocol,'' *IEEE Network,* Sept. 1993, pp. 8-18.