

A TDMA-based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad Hoc Network

Wen-Hwa Liao¹, Yu-Chee Tseng², and Kuei-Ping Shih³

¹Department of Computer Science and Information Engineering
National Central University, Chung-Li, 320, Taiwan

²Department of Computer Science and Information Engineering
National Chiao-Tung University, Hsin-Chu, 300, Taiwan

³Department of Computer Science and Information Engineering
Tamkang University, Tamsui, 251, Taiwan

Abstract—This paper considers the bandwidth reservation problem in a mobile ad hoc network (MANET) to support QoS (quality-of-service) routing. We approach this problem by assuming a common channel shared by all hosts under a TDMA (Time Division Multiple Access) channel model. Existing solutions have addressed this problem by assuming a stronger multi-antenna model [3], [6], where the bandwidth of a link is independent of the transmitting/receiving activities of its neighboring links, or a less stronger CDMA-over-TDMA channel model [7], where the use of a time slot on a link is only dependent of the status of its one-hop neighboring links. While more practical and less costly, using a TDMA model needs to face the challenge of radio interference problems. In this paper, we propose a new protocol that can reserve routes by addressing both the hidden-terminal and exposed-terminal problems. The protocol can conduct accurate bandwidth calculation while performing route discovery. Simulation results are presented to verify how this new protocol performs.

Keywords—mobile ad hoc network (MANET), quality-of-service (QoS), routing, Time Division Multiple Access (TDMA).

I. INTRODUCTION

One research issue that has attracted a lot of attention recently is the design of *mobile ad hoc network (MANET)*. A MANET is one consisting of a set of mobile hosts which can communicate with one another and roam around at their will. Applications of MANETs occur in situations like battlefields and major disaster areas, where networks need to be deployed immediately but base stations or fixed network infrastructures are not available.

Since MANET is characterized by its fast changing topology, extensive research efforts have been devoted to the design of routing protocols for MANETs [2], [5], [8]. These works only concern with shortest-path routing and the availability of multitude routes in the MANET's dynamically changing environment. So only best-effort data traffic is provided. Issues related to quality-of-service (QoS) requirements, such as delay and bandwidth bounds, are less frequently addressed.

This paper considers the problem of searching for a route of a given bandwidth in a MANET. This problem has been addressed by several works in the literature. References [3], [6] have discussed this problem by assuming quite an ideal

model that the bandwidth of a link can be determined independently of its neighboring links. This may be realized by a costly multi-antenna assumption such that a host can send/receive using several antennas simultaneously and independently. A less stronger assumption made in [7] is the CDMA-over-TDMA channel model, where the use of a time slot on a link is only dependent of the status of its one-hop neighboring links.

In this paper, we assume a simpler TDMA model on a single common channel shared by all hosts. So it is inevitable to take the radio interference problems into consideration. We consider the bandwidth reservation problem in such environment. A route discovery protocol is proposed, which is able to find a route with a given bandwidth (represented by number of slots). When making reservation, both the *hidden-terminal* and *exposed-terminal* problems will be taken into consideration. As far as we know, this is the first such result for MANET.

The rest of the paper is organized as follows. Section II contains backgrounds and preliminaries. Our routing protocol is presented in section III. Experimental results are in section IV. Section V concludes this paper.

II. PRELIMINARIES

A. QoS Transmission in MANET

Providing QoS is more difficult for MANET due to at least two reasons. First, unlike wired networks, radios have broadcast nature. Thus, each link's bandwidth will be affected by the transmission/receiving activities of its neighboring links. Second, unlike cellular networks, where only one-hop wireless communication is involved, a MANET needs to guarantee QoS on a multi-hop wireless path. Further, mobile hosts may join, leave, and rejoin at any time and any location; existing links may disappear and new links may be formed on-the-fly. All these raise challenges to QoS routing in a MANET.

Issues related to QoS transmission in MANET have received attention recently. QoS routing is considered in [3], [7]. In [3], a *ticket-based* protocol is proposed to support QoS routing. However, this paper assumes that the bandwidth of a link can be determined independently of its neighboring links. The is quite a strong assumption, because a costly multi-antenna model may be needed. In [7], how to calculate the bandwidth

This work is supported by the National Science Council of the Republic of China under Grant #NSC90-2213-E-009-049 and #NSC90-2213-E-009-154, and the Ministry of Education, the Republic of China, under grant 90-H-FA07-1-4 (Learning Technology).

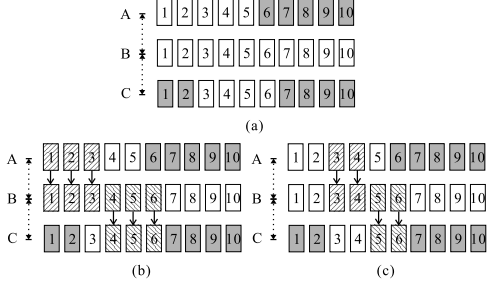


Fig. 1. The bandwidth calculation problem under the TDMA model: (a) free and busy time slots, (b) an assignment with bandwidth = 3 slots from A to C , and (c) an assignment with bandwidth = 2 from A to C .

of a routing path in a MANET is addressed. A CDMA-over-TDMA channel model is assumed. The code used by a host should be different from that used by any of its two-hop neighbors. So a code assignment protocol should be supported (this can be regarded as an independent problem; references can be found in [1], [4]). The bandwidth requirement is realized by reserving time slots on links. Based on such assumption, this paper shows how to allocate time slots on each link of a path such that no two adjacent links share a common time slot.

B. System Model and Challenges

This paper is concerned with QoS routing in a MANET. Different from the above referenced works, we assume a simpler (and perhaps more realistic) TDMA-based channel model. One single common channel is assumed to be shared by all hosts in the MANET. The channel is time-framed.

Because an antenna can not send and receive at the same time, bandwidth calculation in a multi-hop route is a non-trivial problem. Take the path from A to C in Fig. 1(a) as an example, where the white slots associated with each host are free and the gray slots are busy. Matching the free slots between hosts, we obtain five common free time slots $\{1, 2, 3, 4, 5\}$ between A and B and four common free time slots $\{3, 4, 5, 6\}$ between B and C . One may naively think that the path bandwidth from A to C is four ($= \min\{4, 5\}$). Unfortunately, this is not true. As shown in Fig. 1(b), if we reserve slots $\{1, 2, 3\}$ for A to transmit and slots $\{4, 5, 6\}$ for B to transmit, the path bandwidth is only three. In fact, it is not hard to see that it is impossible to further increase the path bandwidth in this example. Even worse, as shown in Fig. 1(c), if one reserves slots $\{3, 4\}$ for A to transmit and slots $\{5, 6\}$ for B to transmit, the path bandwidth will degrade to two, and the situation cannot be improved, unless we change the assignment for A .

The above discussion has already been simplified by purposely ignoring the transmission and reception activities of individual mobile hosts. At this point, we'd like to recall the hidden-terminal and exposed-terminal problems, which are well-known problems in the literature of radio-based communication. Consider the scenario in Fig. 2, where the status of A , B , and C is the same as the above example. Suppose there is another pair, D and E , who are currently using slot 2 to com-

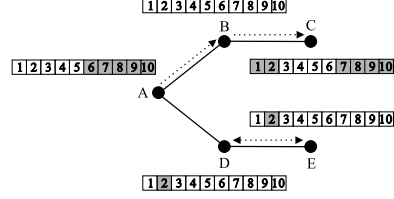


Fig. 2. Example of how bandwidth calculation is interfered by the hidden- and exposed-terminal problems.

municate. Then two cases will occur. If D is a receiver on slot 2, A will not be allowed to send on slot 2 because otherwise collision will occur at D . This is the hidden-terminal problem. So in the example of Fig. 1, the common free time slots between A and B should be reduced to $\{1, 3, 4, 5\}$. Then the case in Fig. 1(b) will not hold anymore, and the path bandwidth from A to C has to downgrade to 2 slots. On the contrary, if D is a sender on slot 2, A will still be allowed to send on slot 2, because this is an exposed-terminal problem. Then the common free time slots between A and B (and thus the path bandwidth) remain the same.

While the above examples already show the complication in the bandwidth reservation problem, we'd like to comment on the data structure used above. From the discussion, we see that simply indicating a time slot as busy or free is insufficient to resolve the hidden- and exposed-terminal problems. For the busy case, we need to tell whether the host is sending or receiving in this slot. This observation motivates our design in the next section.

III. OUR QoS ROUTING PROTOCOL

A. Basic Idea

Our routing protocol is an *on-demand* one, so route search is done only when necessary. (The contrary is *proactive*, which is generally regarded to be more costly.) It is based on source routing, and works similar to the DSR protocol [2] on disseminating route-searching packets, but we need to carefully calculate the bandwidth of each route being searched.

A source host S , on requiring a route to a destination D with bandwidth b , will issue through broadcast a QoS route request packet $QREQ(\dots, b, PATH, NH)$ to its neighbors. The field $PATH$ provides the important information to keep track of the partial route and time slots that the $QREQ$ packet has discovered so far. The NH is a list of hosts, each of which may be used as the next hop to extend $PATH$ with one more hop. Any host x listed in NH hearing this $QREQ$ for the first time may rebroadcast this packet, if it has sufficient collision-free time slots (here the route cache design may be raised to reduce the flooding cost; however, we deal this as an independent issue and refer the reader to the literature [8]). In x 's rebroadcast, proper information will be added to $PATH$ and NH .

When D receives the $QREQ$ packet, it can reply a QoS route reply packet $QREP(\dots, PATH)$ destined to the source S . This packet will be routed, through unicast, along the re-

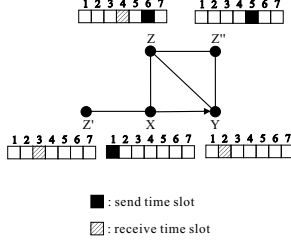


Fig. 3. An example of Lemma 1.

verse direction of $PATH$, and on its way back reserve proper time slots at intermediate hosts according to the content in $PATH$. Our protocol relies on the following lemma to choose time slots in a host.

Lemma 1: A time slot t can be used by a host X to send to another host Y without causing collision if the following conditions are all satisfied:

1. Slot t is not yet scheduled to send or receive in neither X nor Y .
2. For any 1-hop neighbor Z of X , slot t is not scheduled to receive in Z .
3. For any 1-hop neighbor Z of Y , slot t is not scheduled to send in Z .

For example, in Fig. 3, a host X needs a time slot to transmit to a host Y . First, slots 1 and 2 can not be considered because slot 1 is used by X to send and slot 2 is used by Y to receive. Second, slots 3 and 4 can not be considered because they will cause collision at Z and Z' . Third, slots 5 and 6 can not be considered because Z and Z'' are sending on these slots. So we conclude that only slot 7 can be used.

B. Data Structures

We will index hosts by numbers $1, 2, \dots, n$, and time slots by $1, 2, \dots, s$. Each host x will maintain three tables as follows.

- $ST_x[1..n, 1..s]$: The *send table* of x , which records on which time slots a host which is within 2 hops from x will have sending activities. Specifically, $ST_x[i, j] = 1$ if slot j of host i has been reserved for sending; otherwise, $ST_x[i, j] = 0$.
- $RT_x[1..n, 1..s]$: the *receive table* of x , which records on which time slots a host which is within 2 hops from x will have receiving activities. Specifically, $RT_x[i, j] = 1$ if slot j of host i has been reserved for receiving; otherwise, $RT_x[i, j] = 0$.
- $H_x[1..n, 1..n]$: the *hop-count matrix* of x , which is to keep track of the mutual distances between hosts in x 's neighborhood. Specifically, for each host i that is within 1 hop from x , $H_x[i, j] = 1$ if host j is within 1 hop from i ; otherwise, $H_x[i, j] = \infty$.

In Fig. 4, host A is sending to E on the path $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ of bandwidth of 2 slots. Slots $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$, and $\{7, 8\}$ are used by A, B, C, D to send, respectively. They are reflected on each host's ST and RT tables. Note that the rows $ST_x[x, j]$ and $RT_x[x, j]$ indicate the sending and receiving activities of host x itself. In order to create these data

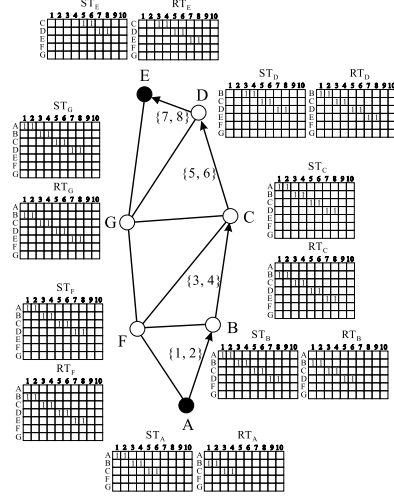


Fig. 4. Example of time slot tables ST and RT .

structures, a host needs to periodically broadcast its own status to its 2-hop neighbors.

To search for a QoS route, we mainly use the packet $QREQ(S, D, id, b, x, PATH, NH)$, whose parameters are defined as follows.

- S : the source host.
- D : the destination host.
- id : an identity which is unique to each route-searching request issued by S . So the triplet (S, D, id) can be used to detect duplicate $QREQ$ to avoid endless looping.
- b : the bandwidth requirement, represented by an integer number of slots.
- x : the host currently relaying the $QREQ$ packet.
- $PATH$: the partial path, together with the available time slots, that has been discovered so far. It has the format $((h_1, l_1), (h_2, l_2), \dots, (h_k, l_k))$. Each $h_i, i = 1..k$, is a host identity, so the sequence h_1, h_2, \dots, h_k, x represents the current partial path. Each l_i contains a total of b time slots that are found to be available for h_i to transmit to h_{i+1} , with the exception that h_k 's intending receiver is host x .
- NH : a list next-hop hosts of the format $((h'_1, l'_1), (h'_2, l'_2), \dots)$. Each host h'_i has potential to serve as the next hop of host x to extend the current partial path (so the new path will be $h_1, h_2, \dots, h_k, x, h'_i$). However, this will depend on whether h'_i has sufficient time slots or not (this will become clear in the protocol). The corresponding parameter l'_i contains b time slots that can be used by x to transmit to h'_i without collision.

When a route is found, we need to initiate from the destination D a packet $QREP(S, D, id, PATH)$ to the source S . This packet will travel on the reverse direction of $PATH$ and reserve time slots, as discovered, on the path. These parameters carry the same meanings as above.

C. Protocol Details

Now suppose a host y receiving a broadcasting packet $QREQ(S, D, id, b, x, PATH, NH)$ initiated by a neigh-

boring host x . If the same route request (uniquely identified by (S, D, id)) has not be heard by y before, it will perform the following steps:

A1.

if (y is not a host listed in NH) **then**
 exit this procedure.

else

Let (h'_i, l'_i) be the entry in NH such that $h'_i = y$.
 Construct a list $PATH_temp = PATH \mid (x, l'_i)$,
 where \mid means list concatenation.

end if.

A2. Construct two temporary tables, $ST_temp[1..n, 1..s]$ and $RT_temp[1..n, 1..s]$, as follows.

i. Copy all entries in $ST_y[1..n, 1..s]$ into $ST_temp[1..n, 1..s]$, and similarly copy all entries in $RT_y[1..n, 1..s]$ into $RT_temp[1..n, 1..s]$.

ii. Let $PATH = ((h_1, l_1), (h_2, l_2), \dots, (h_k, l_k))$. For each $i = 1..k - 1$, assign $ST_temp[h_i, t] = 1$ and assign $RT_temp[h_{i+1}, t] = 1$ for every time slot t in the list l_i . Assign $ST_temp[h_k, t] = 1$ and assign $RT_temp[x, t] = 1$ for every time slot t in the list l_k .

iii. Recall l'_i (the slots for x to send to y). Let $ST_temp[x, t] = 1$ and $RT_temp[y, t] = 1$ for every time slot t in the list l'_i .

These temporary tables, ST_temp and RT_temp , are obtained from ST , RT , $PATH$, and NH . This is because we are in the probing stage, but ST and RT only contain slot status already confirmed. The information in $PATH$ and NH has to be introduced into these temporary tables.

A3.

Let $NH_temp = \emptyset$ (i.e., an empty list).

for each 1-hop neighbor z of y **do**

$L = \text{select_slot}(y, z, b, ST_temp, RT_temp)$

if $L \neq \emptyset$ **then**

$NH_temp = NH_temp \mid (z, L)$

end if

end for

The above step calls for a procedure $\text{select_slot}()$, which will return, if possible, b available slots that can be used by y to send to z (the details will be shown later). If the above loop can find at least one host to extend the current path, the $QREQ$ will be rebroadcast, as shown below.

A4.

if $NH_temp \neq \emptyset$ **then**

broadcast $QREQ(S, D, id, b, y, PATH_temp,$
 $NH_temp)$

end if

The source host S will initiate the $QREQ$. It can be regarded as a special case of intermediate hosts, and can perform similarly to the above steps by replacing host y with S . We only summarize the modifications required for S . First, S has not $PATH$ and NH . So in S1, the checking of NH is unnecessary. We can simply set $PATH_temp = \emptyset$. Also, step A2 can be simplified to only executing step i. The other steps

remain the same.

When the destination D receives packet $QREQ(S, D, id, b, x, PATH, NH)$, a satisfactory path has been formed. D can accept the first $QREQ$ received, or choose based on other policy. Then following steps will be executed.

B1. Let (h'_i, l'_i) be the entry in NH such that $h'_i = D$.

B2. $PATH_temp = PATH \mid (x, l'_i)$.

B3. Send $QREP(S, D, id, PATH_temp)$ to S .

Note that the $QREP$ packet will travel in the reverse direction of $PATH$ through unicast. Each intermediate host should relay this packet. In addition, proper sending and receiving activities should be recorded in their sending and receiving tables. Specifically, let the whole path be $PATH = ((h_1, l_1), (h_2, l_2), \dots, (h_k, l_k))$. For each intermediate host $x = h_i$, the following steps should be conducted.

C1.

for $j = i - 2$ **to** $i + 2$ **do**

Let $ST_x[h_j, t] = 1$ for each time slot t in l_j .

end for

C2.

for $j = i - 2$ **to** $i + 2$ **do**

Let $RT_x[h_j, t] = 1$ for each time slot t in l_{j-1} .

end for

D. Time Slot Selection

The procedure $\text{select_slot}(y, z, b, ST_temp, RT_temp)$ is for host y to choose b free time slots to send to z . It mainly relies on Lemma 1 to do the selection. Specifically, for each time slot i , $1 \leq i \leq s$, we check the following conditions D1, D2, and D3. If all conditions hold, slot i is a free slot that can be used by y to send to z .

D1. $(ST_temp[y, i] = 0) \wedge (RT_temp[y, i] = 0) \wedge (ST_temp[z, i] = 0) \wedge (RT_temp[z, i] = 0)$.

D2. $\forall w : (H_{\downarrow}^y[y, w] = 1) \implies RT_temp[w, i] = 0$.

D3. $\forall w : (H_{\downarrow}^y[z, w] = 1) \implies ST_temp[w, i] = 0$.

To respond the procedure call in A3, if there are at least b time slots satisfying the above conditions, we should return a list of b free slots to the caller; otherwise, an empty list \emptyset should be returned.

When there are more than b time slots available, we can further choose slots based on some priority. The basic idea is to increase channel reuse (which is generally favorable in almost all kinds of wireless communications). Those slots which have the exposed-terminal problem can be chosen with higher priority. To reflect this, we can give a legal time slot i a higher priority such that $ST_temp[w, i] = 1$ for some neighbor w of x .

IV. EXPERIMENTAL RESULTS

We have developed a simulator to evaluate the performance of the proposed bandwidth reservation scheme. A MANET in a $1000m \times 1000m$ area with 30 mobile hosts was simulated. Each mobile host had the same transmission range of 300 meters. Hosts roamed around with speed $0 \sim 20$ m/s in a ran-

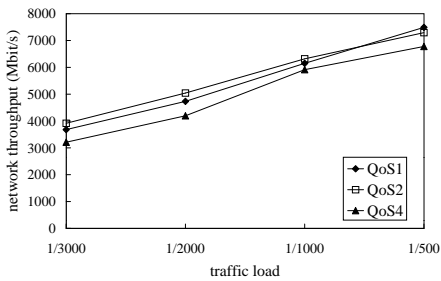


Fig. 5. Network throughput vs. traffic load (= QoS requirement times call arrival rate), where number of hosts=30 and mobility=4m/s.

domly chosen direction and changed direction every second. A data transmission rate of 11 Mbit/s was used. Each time frame had 16 time slots, with 5 ms for each time slot.

Traffic was generated with random source-destination pairs with bandwidth requirement of 1, 2, or 4 slots (denoted as QoS1, QoS2, and QoS4, respectively). New calls arrived with an exponential distribution of mean rate $1/12000 \sim 1/500$ per ms. Each call had duration of 180 sec. Since our goal was to observe multi-hop communication, each source-destination pair must be distanced by at least two hops. The total simulation time was 1000 sec.

We make observations from several aspects.

A) Network Throughput: When calculating throughput, we only count packets that successfully arrive at their destinations. In Fig. 5, we show the network throughput under various loads, where load is defined to be the bandwidth requirement (which are 1, 2, and 4 for QoS1, QoS2, and QoS4, respectively) times the corresponding call arrival rate. Among the simulated ranges, the throughputs all increase linearly with respect to loads for all QoS types. This indicates that QoS routing can be supported quite well by MANET based on our protocol. As comparing different bandwidth requirements, QoS4 performs slightly worse than QoS1 and QoS2. The reason will be elaborated below.

To understand the above scenarios, we further investigate the call success rate (the probability to accept a new call) under the same inputs. The results are in Fig. 6. When the traffic load increases, the success rates decrease for all QoS types. The success rate of QoS1 is the largest, which is followed by QoS2, and then QoS4. This is reasonable because larger bandwidth requirements are more difficult to satisfy.

B) Effect of Mobility: In Fig. 7, we show the throughput under various host mobility. We see that throughput is very sensitive to mobility in all QoS types. In our simulation, whenever a route is broken, an error message will be sent to the source host. Before the source host knows this fact, all packets already sent will still consume time slots without contributing to the real throughput. Furthermore, before a new route is discovered, some time slots will be idle. This is why we see significant drop on throughput as mobility increases, which also indicates a challenging problem deserving further research.

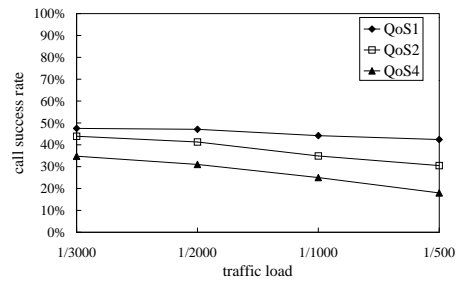


Fig. 6. Call success rate vs. traffic load, where number of hosts=30 and mobility=4m/s.

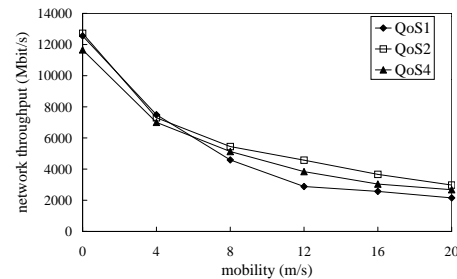


Fig. 7. Network throughput v.s. mobility, where number of hosts=30 and traffic load=1/500.

V. CONCLUSIONS

In this paper, we have proposed a TDMA-based bandwidth reservation protocol for QoS routing in a MANET. One interesting point is that our protocol can take the difficult hidden-terminal and exposed-terminal problems into account when establishing a route. So more accurate route bandwidth can be calculated and the precious wireless bandwidth can be better utilized.

REFERENCES

- [1] A. A. Bertossi and M. A. Bonuccelli, "Code Assignment for Hidden Terminal Interference Avoidance in Multihop Radio Networks," *IEEE/ACM Trans. on Networks*, vol. 3, no. 4, pp. 441–449, August 1995.
- [2] J. Broch, D. B. Johnson and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *Internet draft*, December 1998.
- [3] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1488–1505, August 1999.
- [4] J. J. Garcia-Luna-Aceves and J. Raju, "Distributed Assignment of Codes for Multihop Packet-Radio Networks," *Proc. of IEEE MILCOM '97*, 1997.
- [5] W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," *Telecommunication Systems*, vol. 18, no. 1, pp. 37–60, 2001.
- [6] W.-H. Liao, Y.-C. Tseng, S.-L. Wang and J.-P. Sheu, "A Multi-Path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network," *IEEE International Conference on Networking*, vol. 2, pp. 158–167, 2001.
- [7] C. R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1426–1438, August 1999.
- [8] C. Perkins and E. M. Royer, "Ad Hoc On Demand Distance Vector (AODV) Routing," *Internet draft*, August 1998.