Initialization Protocols for IEEE 802.11-Based Ad Hoc Networks[*]

Chih-Shun Hsu and Jang-Ping Sheu
Department of Computer Science and Information Engineering
National Central University, Chung-Li, 32054, Taiwan, R.O.C.

Abstract

Leader election and initialization are two fundamental problems in mobile ad hoc networks ($MANETs$). The leader can serve as a coordinator in the $MANETs$ and the initialization protocol can assign each host a unique and short id. As we know that none of the research on initialization for IEEE 802.11-based $MANETs$ has been done. Here, we proposed two contention-based leader election and initialization protocols for IEEE 802.11-based single-hop $MANETs$. Simulation results show that our protocols are efficient.

Keywords initialization, leader election, mobile ad hoc network ($MANET$).

1 Introduction

A *mobile ad hoc network* (MANET) is formed by a cluster of mobile stations and can be quickly deployed without any pre-designed infrastructure or centralized administration. The leader is the coordinator of the network, it can serve as a relay point or it can coordinate its members' actions in $MANET$s. The initialization protocol can provide each host a unique and short id, so that each host can perform id-based algorithms. As our best knowledge, there are only a few leader election and initialization researches [2], [5], [1], [7], [9], [8], [11] in wireless networks.

A simple leader election algorithm for wireless LAN is proposed in [5]. The leader, which is elected by the base station, serves as a reporter to its multicast group members. It will send a feedback to the sender when there is no collision, and thus increase the reliability of the multicast. With a similar idea, a random leader-based reliable multicast protocol is proposed in [1], which overcomes the problem of feedback collision. Both of the two algorithms [5], [1] based on wireless LAN require the help of the base station. Two leader election algorithms based on $TORA$ [10] for $MANET$ are proposed in [7]. One algorithm is for a single topology change, the other tolerates multiple topology changes. Both algorithms work by assigning each host a unique height (6-tuple), which is costly. A uniform leader election protocol for radio networks is proposed in [9]. Randomized leader election and initialization protocols for time-slotted single-hop $MANET$s are proposed in [8]. These protocols (termed as the $Nakano$-$Olariu$ protocols) are efficient but based on an impractical assumption (termed as the $Nakano$-$Olariu$ assumption), that the sender can detect its own transmission status. A leader election algorithm revised from [8] is proposed in [11]. In [11], the leader acts as a coordinator, which initializes the hosts of the same priority by giving each of them a unique id, so that these hosts know when to transmit their frames according to their ids.

To improve previous works, we proposed efficient leader election and initialization protocols for IEEE 802.11-based single-hop $MANET$s with and without the knowledge of the number of hosts. The proposed initialization protocols work as follows. First, elect a leader in the $MANET$, then let the leader serve as a detector, which will tell the sender the status of the transmission. If the transmission is successful, the leader will assign a unique and short id to the sender. Similar to the adaptive round transmission protocol proposed in [4], when the number of hosts is not available, we set the value of contention window (CW) to a predetermined value. After a round, we can estimate the number of hosts in the $MANET$ according to previous round's transmission status and then set the value of CW according to the estimated number of hosts. Simulation results show that our protocols are efficient. When based on the same assumption, our protocols perform better than the $Nakano$-$Olariu$ protocols.

The rest of the paper is organized as follows. Preliminaries are given in section 2. Section 3 presents our leader election and initialization protocols. Simulation results are presented in section 4. Section 5 concludes the paper.

2 Preliminaries

In this paper, we intend to solve the leader election and initialization problems on an IEEE 802.11-based single-

---

[*] This work was supported by the National Science Council of Republic of China under grant NSC 91-2213-E-008-027.

hop $MANET$. We assume that every mobile host in the $MANET$ is synchronized by adjusting its clock according to the beacon frame and can detect the status of its neighboring host's transmission.

2.1 IEEE 802.11 MAC protocol

The IEEE 802.11 medium access (MAC) protocol [6] used in $MANET$s is the distributed coordination function (DCF) which is based on the *Carrier Sense Multiple Access with Collision Avoidance* ($CSMA/CA$) mechanism. When a mobile host wants to transmit frames, it first detects the status of the medium. If the medium is busy, the host will defer until the medium is idle for a period of time equal to $DIFS$ (DCF interframe space). After this $DIFS$ idle time, the host will generate a random backoff period, where *backoff time* = $Random() \times ST$. $Random()$ is a random function, which is uniformly distributed between the interval $[0, CW]$ and ST is the length of a backoff time slot. The initial value of the CW is CW_{min}. When a host wants to send data, it first sense the medium. If the medium is idle for a period of time equal to $DIFS$, the backoff procedure will decrease the backoff time, otherwise, it will stop decreasing the backoff time. When the backoff timer expires, the host will transmit the frame. After the sender transmits the frame, if it is a broadcast, the receivers do nothing. Otherwise, if it is a unicast, the receiver will wait for a period of time equals to $SIFS$ (short interframe space, $SIFS < DIFS$) and then reply an Ack to the sender. If the sender does not receive an Ack from the receiver, the sender will double the size of its contention window and repeat the DCF procedure again.

2.2 The $Nakano$-$Olariu$ Protocols

The $Nakano$-$Olariu$ protocols [8] are based on a time-slotted single-hop $MANET$. They assume that the mobile host can detect the status of its own transmission. If the mobile host has the collision detection capability, it can detect three status, namely $NULL$(no transmission), $SINGLE$(exactly one transmission) and $COLLISION$(two or more transmission), of the radio channel at the end of a time slot. However, when the mobile host has no collision detection capability, it can only detect two status, namely $SINGLE$(exactly one transmission) and $NOISE$(collision or no transmission). Under this condition, the mobile hosts in the $MANET$ need to elect a leader to help them to distinguish $NULL$ from $COLLISION$.

When the mobile host has no collision detection capability and the number of mobile hosts in the $MANET$ is unknown in advance, each mobile host contends to be the leader. At first, each mobile host transmits with probability $\frac{1}{2}$. If the status of the channel is $SINGLE$, the mobile host that has transmitted in the previous time slot is declared as the leader. Otherwise, each mobile host continues to transmit with half of previous probability until a host is declared as the leader.

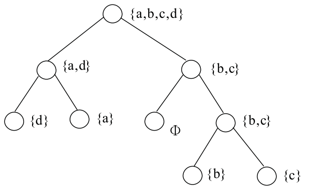

Figure 1. An example of a partition tree.

When the mobile host has the collision detection capability and the number of mobile hosts in the $MANET$ (denoted as n) is known in advance, the mobile hosts need not to elect a leader, they get their ids by contention. At first, each mobile host transmits with probability $\frac{1}{m}$, where m is the number of hosts that have no ids. If the channel status is $SINGLE$, the mobile host that has transmitted in the previous time slot gets $n - m + 1$ as its id. The other hosts that have no ids will follow the same procedure to get their ids until there is no host without id.

If n is unknown in advance, each mobile host will follow the idea of the partition tree to get its own id. Figure 1 is an example of a partition tree. In the beginning, hosts a, b, c and d all transmit on the channel and the channel status is $COLLISION$. Therefore, each host flips a fair coin to decide who can transmit next time. Hosts a and d flip "heads", they can transmit on the channel and the channel status again is $COLLISION$. Therefore, hosts a and d flip fair coins again. In this time slot, only host d flips a "head", so host d transmits on the channel and the channel status is $SINGLE$. Host d sets 1 as its own id. After host d has set its own id, host a transmits on the channel and the channel status is $SINGLE$. Host a sets 2 as its own id. When hosts a and d have set their ids, it is hosts b and c's turn to transmit on the channel. When hosts b and c both transmit on the channel, the channel status is $COLLISION$. Therefore, hosts b and c flip coins. Since none of them flip "heads", none of them transmit on the channel and the channel status is $NULL$. Hosts b and c then flip coins again. In this time slot, host b flips a "head". Therefore, only host b transmit on the channel, host b set 3 as its own id. In the next time slot, host d transmits on the channel. The channel status is $SINGLE$. Therefore, host d set 4 as its own id. Now every hosts have got their ids, the initialization protocol is terminated.

In the $Nakano$-$Olariu$ protocols, no matter the mobile host has the collision detection capability or not, the assumptions are not practical. When the mobile host is transmitting message, it is very hard for itself to detect the channel status. Therefore, we propose more practical leader election and initialization protocols, which are based on the standard of IEEE 802.11 protocol.

3 Our Leader Election and Initialization Protocols

Two efficient leader election and initialization protocols for IEEE 802.11-based single-hop $MANET$s are proposed herein, one is for a $MANET$ whose number of hosts is known in advance, the other one is for a $MANET$ without the knowledge of the number of hosts. In the following, we assume that the number of hosts in the $MANET$ is known in advance.

3.1 The Leader Election Protocol

Before initializing a $MANET$, we need to elect a leader to serve as a coordinator in the network. Every host in the network has an equal chance to become a leader. Without loss of generality, we assume that there are n hosts, H_1, H_2, H_3, $\ldots$, H_n, in the $MANET$. In the beginning, every host basically follows a DCF procedure, mentioned in Section 2.1 to contend as a leader in the $MANET$. However, the value of CW is set according to the number of hosts in the $MANET$ and each host will not contend again until an election round is over (CW is set as $m-1$ and an election round is said to be over if the m-th backoff time slot has expired). When host H_i's ($i = 1 \ldots n$) backoff timer has expired and there is no other host successfully claimed itself as the leader, host H_i claims itself as the leader by broadcasting its MAC address. Assume that host H_a successfully broadcasts its MAC address. Since there is no collision, the claim can be heard by all hosts except H_a. Once the claim is successful, the other hosts, whose backoff timers have not expired, will wait until their backoff timers have expired and then send an acknowledgement to H_a. They broadcast H_a's MAC address to inform H_a that it is the new leader of the $MANET$. Again, if there is only one host, say H_b, broadcasting H_a's MAC address, the acknowledgement can be heard by all hosts except H_b, the acknowledgement is said to be successful and all the hosts except H_b know that a new leader has been elected. After a short period of time equals to $SIFS$, host H_a can then announce itself as the new leader and the leader election process is successful and completed.

According to the above description, a successful leader election process required at least two successful broadcasts from different hosts (A broadcast is said to be successful if there is only one host broadcast the message.) It first requires a successful claim, and then a successful acknowledgement from another host. If an election round is over and no any host is elected as a leader, every host follows the same procedure to contend as the leader in the next round until a new leader has been elected. The size of CW is set to $m-1$ in each election round. If there is a host, say H_a, whose claim is successful but the acknowledgements sent by other hosts are all failed after an election round, all the hosts except H_a in the next election round will inform host H_a until the acknowledgement is successful. Note that H_a will still broadcast a claim message because H_a does not know that it has a successful claim.

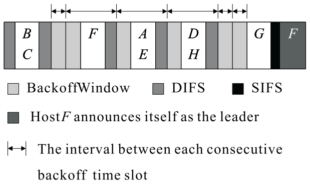

Figure 2. An example of a successful leader election.

For example, assume that there are 8 hosts A, B, C, D, E, F, G, and H, in the $MANET$ and the length of a backoff time slot is denoted as ST. Each host follows the DCF procedure to claim itself as a leader of the $MANET$. The size of CW is set as 8. As Figure 2 shows, the backoff timers of hosts B and C are set as 0, host F is set as $2 \times ST$, hosts A and E are set as $3 \times ST$, hosts D and H are set as $4 \times ST$, and host G is set as $6 \times ST$, respectively. After a $DIFS$, hosts B and C claim themselves as the leaders of the $MANET$ and a collision occurs, so this claim is not successful. After host F's backoff timer has expired, host F claims itself as the leader. Since there is only one host claim itself as the leader, the claim is successful. Therefore, hosts A, D, E, H, and G stop claiming themselves as the leaders, they all try to send an acknowledgement to host F by broadcasting its MAC address when their backoff timers expired. Hosts A and E send their acknowledgements simultaneously and a collision occurs. The same thing happens as the backoff timers of hosts D and H have expired. Finally, host G's backoff timer expired, host G send an acknowledgement and the acknowledgement is successful. After receiving the acknowledgement, the new leader, host F, waits for an $SIFS$ and announces itself as the new leader of the $MANET$.

The leader election algorithm is shown as follows:

```
Algorithm 1: Leader-Election($n$,$m$)
$n$: number of hosts in the $MANET$
$m-1$: the value of $CW$
$H_a$: the first host broadcast its claim successfully
Initial: $Claim = false$, and every host randomly set its backoff timer as $R \times ST$, where $R \in N$ and $0 \leq R \leq m-1$.
while (no host announces that it is the leader) do

    if the $m$-th backoff time slot has expired then every
```

```
        host randomly set its backoff timer as $R \times ST$.
        When any host $H_i$'s backoff timer expires $\Rightarrow$
            if ($Claim = false$) then host $H_i$ broadcasts its own $MAC$ address.
            else host $H_i$ broadcasts host $H_a$'s $MAC$ address.
            endif
            if a host can hear a successful broadcast from $H_i$ then
                if (the received $MAC$ address = my $MAC$ address) then the host waits for a period of time equal to $SIFS$ and then announces itself as the leader.
                else
                    $Claim = true.$
                    $H_a = H_i$
                endif
            endif
endwhile
```

3.2 The Initialization Protocol

After the leader is being elected, we can initialize the $MANET$ with the help of the leader. Every host (except the leader) sets the value of CW (set as $m-1$) according to the number of hosts and basically follows the DCF procedure to send a request id message by broadcasting its own MAC address. If the leader can receive the request id message without any collision, it will assign an id to the host by broadcasting the host's id after receiving the request id message for a period of time equals to $SIFS$. When the m-th backoff time slot has expired and all the hosts (except the leader) have broadcast their request id messages, an initialization round is over. Assume that before the initialization round begins, there are $r1$ hosts without being assigned ids by the leader and after the initialization round is over, there are still $r2$ hosts without being assigned ids by the leader. In the next initialization round, the hosts with no ids will reset the value of CW as $m-1$, where $m = \lceil m \times r2/r1 \rceil$. The initialization procedure will repeatedly be executed until all the hosts have got their ids.

For example, assume that there are 4 hosts in the $MANET$. Host A is the elected leader in the MANET with $id = 1$. First, the other three hosts (B, C, D) set the initial value of CW as 2, and then follow the DCF procedure to send their request id messages. As Figure 3 shows, host C sets its backoff timer as 0, hosts B and D set their backoff timers as $2 \times ST$. So host A (the leader) can receive host C's request id message without any collision. After an $SIFS$, host A broadcasts host C's id(=2). When hosts B and D's backoff timers expire, they broadcast their request id messages simultaneously and a collision occurs, so host A cannot receive the request id message successfully. When the first initialization round is over, hosts B and D both change the value of CW to 1 and follow the DCF procedure to send their request id messages. In the second initialization round, host B sets its backoff timer as ST and host D sets its backoff timer as 0. This time, host A can receive the request id messages of both hosts D and B successfully, so host A assigns 3 and 4 to hosts D and B, respectively. Finally, every host in the $MANET$ has its own id and complete the initialization procedure.

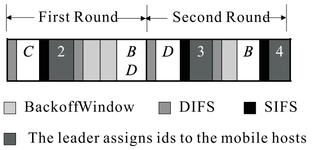

Figure 3. An example of the initialization protocol

The initialization algorithm is shown as follows:

Algorithm 2: Initialization(n, m)
n: number of hosts in the $MANET$
$m-1$: the value of CW
$r1$: number of hosts that have not got their ids before an initialization round
$r2$: number of hosts that have not got their ids after an initialization round
Initial: $id = 1$, $r1 = r2 = n-1$ and every host (except the leader) randomly set its backoff timer as $R \times ST$, where $R \in N, 0 \le R \le m-1$.

```
while ($id \neq n$) do
    if the $m$-th backoff time slot has expired then
        $m = \lceil m \times r2/r1 \rceil, r1 = r2$
        The hosts that have not obtained their $id$s randomly set their backoff timers as $R \times ST$.
    When any host $H$'s backoff timer expires $\Rightarrow$
        Host $H$ broadcast a request $id$ message.
        if the leader detects that there is no collision then
            $id = id + 1$
            The leader waits for a period of time equal to $SIFS$ and then assigns $id$ to host $H$ by broadcasting $H$'s $id$.
            The hosts with no $id$s set $r2 = r2 - 1$.
        endif
endwhile
```

3.3 Our Protocols Without the Knowledge of the Number of Hosts

When the number of hosts is unknown in advance, we follow the same protocol described in Section 3.1 to elect the leader, except the value of CW. First we set the value of CW to a predetermined value. If the leader cannot be elected in an election round, in the next election round, the size of contention window will be doubled until m = 256.

The initialization protocol proposed herein is also similar to the one described in Section 3.2, except the value of CW. We first set the value of CW to a predetermined value. After each initialization round, the leader will tell hosts without ids that how many collisions are in the initialization round and how many hosts obtain their ids, so that these hosts can use an interpolation method [3] to estimate how many hosts are in the $MANET$ and how many hosts remains without id. In the next initialization round, each host will set the size of contention window according to the estimated number of hosts that remain without ids. The procedure will continue until there is no collision in the initialization round and the initialization procedure is considered to be over. When the initialization procedure is over, the leader will realize the current number of mobile hosts in the $MANET$.

4 Simulation Results

We have evaluated the performance of our protocols in [3]. With the evaluation results, we can decide how to set the size of contention window so that our protocols will take less time to elect a leader and initialize a $MANET$. To compare the performance of our protocols and $Nakano$-$Olariu$ protocols, we develop a simulator using C. Our MAC protocols are following the IEEE 802.11 standard [6], however, the $Nakano$-$Olariu$ protocols are simulated in a $TDMA$-based $MANET$ with collision detection capability. In our simulations, the transmission rate is 2M bits/sec, $SIFS$ is set as 10 μs, $DIFS$ is set as 50 μs, ST is set as 20 μs, and $SLOT$ is set as 200 μs, respectively. The number of hosts in the $MANET$ is tuned from 20 to 100.

Three leader election protocols are simulated here: our leader election protocol with the knowledge of the number of hosts (denoted as $HSLE(K)$ and the value of CW is set according to evaluation results in [3]), without the knowledge of the number of hosts (denoted as $HSLE(U)$ and the value of CW is set as 127), and the $Nakano$-$Olariu$'s leader election protocol without the knowledge of the number of hosts (denoted as $NOLE(U)$). When the leader cannot be elected in the first few time slots, there is a great possibility that the $NOLE(U)$ protocol will never elect a leader, because the probability that each host contends to be a leader becomes smaller after each failed election time slot. Therefore, we only consider the case that the $NOLE(U)$ protocol can elect a leader successfully. Figure 4 shows the average time required for the $Nakano$-$Olariu$'s and our leader election protocols. The $Nakano$-$Olariu$'s leader election protocol performs better than our leader election protocols only when the number of hosts is smaller than 40.

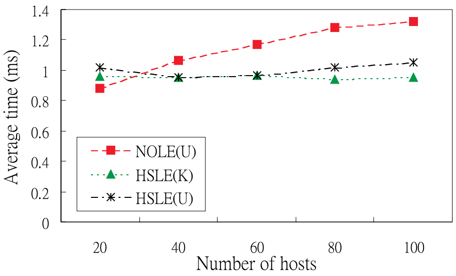

Figure 4. Average time required for the $Nakano$-$Olariu$'s and our leader election protocols

Four different initialization protocols are simulated here: our initialization protocols with the knowledge of the number of hosts (denoted as $HSIN(K)$ and the value of CW is set according to the evaluation results in [3]), without the knowledge of the number of hosts (denoted as $HSIN(U)$ and the value of CW is set as 127), the $Nakano$-$Olariu$'s initialization protocols with the knowledge of the number of hosts (denoted as $NOIN(K)$), and without the knowledge of the number of hosts (denoted as $NOIN(U)$). The performance of the $Nakano$-$Olariu$'s and our initialization protocols are presented in Figure 5. The $Nakano$-$Olariu$'s initialization protocols perform better than our initialization protocols, because the mobile host can detect its own transmission status in the $Nakano$-$Olariu$ assumption. Therefore, it requires only one successful broadcast to get a unique id. However, in the IEEE 802.11-based $MANET$s, the mobile host requires other hosts to tell its transmission status. Therefore, the $HSIN(K)$ and $HSIN(U)$ protocols require two successful broadcasts to get a host's unique id. That is why our initialization protocols will take longer time to finish the job. The performance of our protocols is slightly worse than that of $Nakano$-$Olariu$ protocols, but our protocols are more practical than the $Nakano$-$Olariu$ protocols. The performances of our protocols with and without the knowledge of the number of hosts are quite close to each other, which indicates that we have made a good estimation of the number of hosts and set a proper size of the contention window.

Figures 6 and 7 show that when based on the $Nakano$-$Olariu$ assumption, the performance of our leader election and initialization protocols is better than that of the

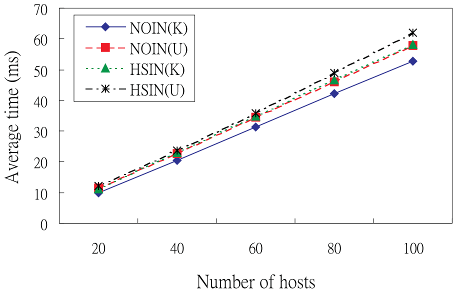

Figure 5. Average time required for the $Nakano$-$Olariu$'s and our initialization protocols

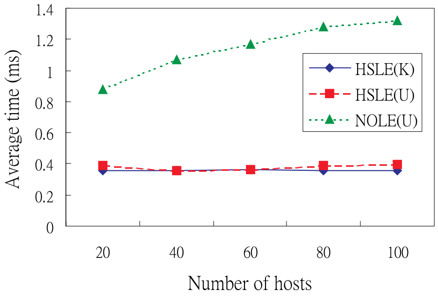

Figure 6. Average time required for the $Nakano$-$Olariu$'s and our leader election protocols based on the $Nakano$-$Olariu$ assumption

$Nakano$-$Olariu$ protocols. Since the contention window in our protocols is set properly, our protocols can avoid unnecessary collisions and transmissions, and thus perform better than the $Nakano$-$Olariu$ protocols.

5 Conclusions

In this paper, we have proposed two leader election protocols and initialization protocols for IEEE 802.11-based single-hop $MANETs$. As we know that no initialization protocol for IEEE 802.11-based single-hop $MANETs$ has been proposed before. Simulation results show that our protocols are practical and efficient. When based on the same assumption, our protocols are more efficient than the $Nakano$-$Olariu$ protocols. With a little modification, our protocols can be easily implemented in the IEEE 802.11-based $WaveLAN$ cards.

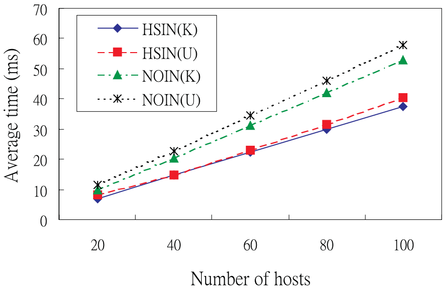

Figure 7. Average time required for the $Nakano$-$Olariu$'s and our initialization protocols based on the $Nakano$-$Olariu$ assumption

References

[1] H. C. Chao, S. W. Chang, and J. L. Chen. Throughput improvements using the random leader technique for the reliable multicast wireless lans. *ICN*, 2001.

[2] K. P. Hatzis, G. P. Pentaris, P. G. Spirakis, V. T. Tampakas, and R. B. Tan. Fundamental control algorithms in mobile networks. *Proc. 11th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 251–160, 1999.

[3] C. S. Hsu and J. P. Sheu. Design and performance evaluation of the leader election and initialization protocols. *Technical Report of HSCC Lab. , http://axp1.csie.ncu.edu.tw/∼cshsu/evaluation.htm*, 2002.

[4] R. Kalinowski, M. Latteux, and D. Simplot. An adaptive anti-collision protocol for smart labels. *Technical Report LIFL University Lille*, 2001.

[5] J. Kuri and S. K. Kasera. Reliable multicast in multi-access wireless lans. *IEEE INFOCOM*, pages 760–767, 1999.

[6] LAN MAN Standards Committee of the IEEE Computer Society. IEEE Std 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE*, 1999.

[7] N. Malpani, J. L. Welch, and N. Vaidya. Leader election algorithms for mobile ad hoc netwroks. *ACM DIAL M Workshop*, pages 96–103, 2000.

[8] K. Nakano and S. Olariu. Randomized initialization protocols for ad hoc network. *IEEE Transactions on Parallel and Distributed Systems*, 11:749–759, July 2000.

[9] K. Nakano and S. Olariu. Uniform leader election protocols for radio networks. *IEEE International Conference on Parallel Processing*, pages 240–248, 2001.

[10] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *IEEE INFOCOM*, pages 7–11, 1997.

[11] J. P. Sheu, C. H. Liu, S. L. Wu, and Y. C. Tseng. A priority mac protocol to support real-time multimedia traffic in ad hoc networks. *European Wireless*, pages 149–155, Feb. 2002.