# Election based Hybrid Channel Access

Xin Wang[1] and J.J. Garcia-Luna-Aceves[12]

[1] Computer Engineering Department,
University of California, Santa Cruz,
Santa Cruz, CA 95064, USA
[2] Palo Alto Research Center (PARC)
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
{wangxin,jj@soe.ucsc.edu}

**Abstract.** We propose an Election based Hybrid Channel Access (EHCA) protocol for ad hoc network to achieve high throughput and bounded channel access delay at the same time. EHCA reduces the contentions during the channel scheduling formation through fair node elections, which are based on the topology information. Only the elected nodes contend for the channel and broadcast the scheduling result. Numerical analysis and simulation results show that EHCA outperforms alternative designs.

## 1 Introduction

The analysis about the capacity of wireless networks [5] demonstrated that perfect scheduling is the ultimate way to achieve the capacity in the MAC layer. However, in a distributed ad hoc network it is impossible to use perfect channel scheduling, and the random channel access has to be used to some extent. We propose the Election based Hybrid Channel Access (EHCA) protocol to attain both high channel utilization and bounded channel access delay. The former is important for serving data-centric applications, while the latter is critical for voice-related applications. In EHCA, channel access period is divided into four time sections. The first section is used to exchange the neighbor information. After that, all nodes do fair elections to reduce the number of nodes which will contend for the channel access. The nodes which fail in the election will follow the scheduling result of the nodes elected. In the third section, the channel scheduling is distributed in the two-hop range and contention-free transmissions happen in the fourth section. We evaluate the performance of EHCA through analysis and simulation. Compared with existing hybrid channel access scheme

[3] and IEEE 802.11, EHCA can achieve a much higher throughput and smaller channel access delay at the same time.

The rest of the paper is organized as follows. We describe the related work in Section 2. We introduce the details of the proposed approach in Section 3. We analyze the properties of EHCA in Section 4. We evaluate the performance of EHCA and compare it with alternative designs in Section 5. We conclude the paper in Section 6.

## 2 Related Work

Medium access control (MAC) protocols of ad hoc network can be classified into contention-based channel access and contention-free channel access. In contention-based MAC protocols, each node either detects the transmission collision (collision-detection) or tries to avoid the transmission collision through random back-offs (collision-avoidance). Based on its observation of the channel status, each node contends for the channel access in a distributed fashion. Contention-based MAC protocols may experience throughput degradation at high traffic loads and due to their best effort nature, they can not provide Quality-of-Service (Qos) support for real-time applications.

In contention-free MAC protocols, a set of timetables for individual nodes or links is prearranged. Each node/links can only transmit in their assigned time/frequency slots, so that the transmissions from these nodes/links are collision-free within the effective range of the transmissions. Dynamic transmission scheduling protocols can exploit spatial reuse of the wireless channel and have higher channel utilization than static scheduling approaches, e.g. TDMA. Based on whether the schedule scheme needs the topology information, the scheduling-based channel access can be further divided into topology dependent/independent scheduling.

In topology dependent scheduling, global topology information is required to form the correct channel scheduling. Arikan [1] has shown that the problem of establishing an optimal interference-free schedule where the optimal is considered in term of throughput, is NP complete.

Chlamtac [4] first proposed a topology-transparent scheduling algorithm for wireless ad hoc networks. It uses polynomials over a Galois field to assign time slots, which guarantees each node can transmit successfully at least once in a frame. This approach can provide a minimum performance guarantee for each node. It just needs the information of overall number of nodes in the network and the number of neighbors of each node. The frame length is also much smaller than the classic TDMA approach. Konstantinos [8] proposed probabilistic policy to increase the system throughput under various traffic loads. Ju [7] proposed an approach based on code theory to optimize the performance of Chlamtac's algorithm in terms of minimum throughput and maximum delay. However, Carlos [11] has shown that the throughput of topology-transparent scheduling is at most the same with the slotted ALOHA.

Hybrid channel access is proposed to take the advantages of contention-based channel access and topology-dependent scheduling. Nodes first use the contention-based channel access to exchange the neighbor information to build the channel scheduling or reserve time slots in the scheduling-based transmission period. The examples of hybrid channel access protocol are NAMA [3] and CATA [13].

NAMA uses a hash function, which takes the node identifier and the current time slot number as input to derive a random priority for every neighbor within two hops. If a node has the highest priority, it can access the channel within the corresponding time slot. The advantage of NAMA is that it completely eliminated the communication overhead with regard to building the dynamic channel access schedule, except for collecting the two-hop neighbor information. However, NAMA has the following problems [2]: first, a node may probabilistically derive low priority for a long period of time and never get access to the channel; second, there may be chain effects to the channel access opportunities, in which the priorities of nodes cascade from high priority to low priority across the network. Chain effects will reduce the spatial reuse of the whole system; third, the channel bandwidth may also be wasted when a node does not have data to send in the allocated time slot. Because of the wasted bandwidth causing starvation to the nodes with traffic, NAMA interacts badly with certain applications that are sensitive to the delay, such as TCP congestion control [12] and AODV route update mechanisms [9].

In CATA, the transmission period is composed of a contention period and a group transmission period. During the contention period, nodes contend for the channel access and reserve a space in the group-transmission period. Then during the group-transmission period, one or more nodes can transmit data packets without collisions. The problem of CATA is that when there is a large number of nodes, the length of the contention period may not be long enough for each node to reserve a slot in the group transmission period.

In this paper, we propose a hybrid channel access protocol which reduces the control overhead during the scheduling formation through nodes elections. Compared with NAMA and CATA, it can provide a bounded channel access delay and only nodes with traffic can access the channel. Since it reduces the number of contending nodes through elections, it allows more nodes to successfully reserve the channel through contentions and is more suitable for mobile scenarios.

## 3 Election Based Hybrid Channel Access (EHCA)

We assume that each node is synchronized on slot systems and nodes access the channel based on slotted time boundaries. Each time slot is numbered relative to a consensus starting point. We divide the channel access period into four different sections, as Figure 1 shows:
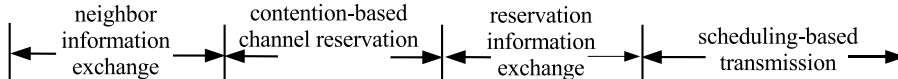
**Fig. 1.** Channel access period

### 3.1 Neighbor information exchange period

The neighbor information exchange period is used to maintain the neighbor information, send the reservation requests and distribute the reservation information in the two-hop range. All nodes adapt the 802.11 Distributed Coordination Function (DCF) to contend for the channel access during the neighbor information exchange period. It can be further divided into two sections: one-hop broadcast period and one-hop re-broadcast period.

Each node will do the following in the one-hop broadcast period:

– If a node does not want to reserve a slot in the scheduling-based transmission period, just send a HELLO packet to maintain the neighbor information.
– Otherwise, a node needs to send a RESERVE_REQUEST packet which indicates the receiver of the transmission (NULL for broadcast packet). The format of the RESERVE_REQUEST is $(src, dest, type)$. The $type$ field indicates the previous scheduling is a failure or not.
– Node $i$ will classify the set of the RESERVE_REQUEST information it has collected during the one-hop broadcast period as one-hop link set $l_i^1$.
– Each node will choose the node with the largest MAC address in its one-hop range as the $leader$ of the network. The $leader$ information is used for the time synchronization across different networks, which will be further discussed in Section 3.5.

Then in the one-hop re-broadcast period, node $i$ will forward the $l_i^1$ to its neighbors, which guarantees the RESERVE_REQUEST will be distributed in the two-hop range. We define the RESERVE_REQUEST information node $i$ has received from node $j$ during the one-hop re-broadcast period as $l_{ij}^2$. The final RESERVE_REQUEST information node $i$ has collected ($l_i$) is:

$$l_i = \{l_i^1 \cup l_{ij}^2 \qquad \forall j \in N_i^1\} \tag{1}$$

where $N_i^1$ is the set of node $i$'s one-hop neighbors.

After one-hop re-broadcast, each node will compare the MAC address of its two-hop neighbors with its current one-hop $leader$, then update the node with the largest MAC address in its two-hop range as the network $leader$. We denote $N_{max1}$ as the maximum number of one-hop neighbors. $N_{max1}$ is a predefined value to control the node density in the network. The length of the neighbor information exchange period $T_{ne}$ needs to be long enough to allow every nodes to broadcast twice. In this paper, we set $T_{ne}$ as $2N_{max1} \times T_b$, where $T_b$ is the maximum time needed to send a broadcast packet using 802.11 DCF, including carrier sensing and exponential back-off.

4

At the end of the neighbor information exchange period, we use *one-hop link contender election* to reduce the possible contentions in the channel reservation period. Each node $i$ will compare the $l_i^1$ with $l_{ij}^2$ for its every neighbor $j$. If node $i$ finds $l_i = l_i^1 \cup l_{ij}^2$ and $l_i^1 = l_{ij}^2$, it means node $i$ and $j$ have the exactly same set of one-hop links, which constitute the total RESERVE_REQUEST information node $i$ has collected, then node $i$ will compare the MAC address of $i$ and $j$. If $i$ has a smaller MAC address, it will give up doing channel reservation and follow the scheduling result of the node $j$. We define the node with a larger MAC address as *contender*. Consider the case in which a lot of nodes are close to each other, this approach elects the node with the largest MAC address as the *contender* and it will do the scheduling for all the one-hop links, which reduces the possible channel access contentions.

## 3.2 Contention-based channel reservation period

During the contention-based channel reservation period, we extend the 802.11 DCF to form the channel scheduling in a distributed fashion. We define $D_{max}$ as the maximum delay a frame can tolerate, which is dependent on specific application. $N_{mt}$ is the number of maximum transmissions a frame can support in order to satisfy $D_{max}$. The priority of node $i$ ($i_{prio}$) is the overall number of links it has collected in the neighbor information exchange period:

$$i_{prio} = |l_i| \tag{2}$$

The node priority will be broadcasted along with the scheduling results during the reservation information exchange period (section 3.3), then each node compares the node priority it has received. If for node $i$, there are two nodes $j$ and $k$ ($j, k \in N_i^1$) with the same highest priority (larger than $i_{prio}$), $i$ will define the contention link set ($l_{i\_cont}^1$) as follows:

$$l_{i\_cont}^1 = l_i^1 \cap l_{ij}^2 \cap l_{ik}^2$$
$$|l_{i\_cont}^1| = |l_i^1 \cap l_{ij}^2 \cap l_{ik}^2| \tag{3}$$

The contention priority of node $i$ ($i_{cont\_prio}$) is the number of links in the contention link set ($l_{i\_cont}^1$):

$$i_{cont\_prio} = |l_{i\_cont}^1| \tag{4}$$

The length of the back-off time ($T_{backoff}$) is decided by the number of links it has observed and the type of links, as Equation 5 shows:

$$T_{backoff} = \begin{cases} T_{sifs} + Random \times T_s & \text{if } |l_{i\_cont}^1| > 0 \\ T_{sifs} + (2N_{mt} - |l_i|) \times T_s & \text{if } |l_{i\_cont}^1| = 0 \end{cases} \tag{5}$$

Each node keeps carrier sense the channel for a Short Inter Frame Space time ($T_{sifs}$), which is defined in IEEE 802.11 [6]. If the channel is idle and

$|l^1_{i\_cont}| > 0$, the back-off time equals to $T_{sifs} + Random \times T_s$, where $Random$ is a random variable uniformly distributed in $[0, N_{mt}]$, $T_s$ is the minimum time slot length defined in IEEE 802.11 [6]. When $|l^1_{i\_cont}| = 0$, the back-off time equals to $T_{sifs} + (2N_{mt} - |l_i|) \times T_s$. Through this approach, we divide the back-off period into two sections. One section is for the contention links while the other is for the remaining two-hop links, as Figure 2 shows. A node which has observed contention link set always has a shorter back-off period than than a node which has not.
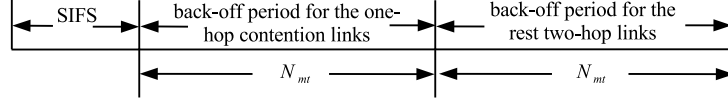


**Fig. 2.** Node back-off scheme

During the back-off period, nodes which have contention links will access the channel and broadcast its scheduling result before the nodes which have not. In other words, contention links are scheduled before the normal links. After the scheduling for the contention links is formed, the node which can observe the largest number of remaining two-hop links will have the shortest $T_{backoff}$. It will first access the channel and build the channel scheduling for the rest links. When a node gets the channel access, it will send an ordered set which indicates the corresponding scheduling as $(source_n, dest_n, type)$. After receiving the scheduling results from neighbor $j$, a node will first compare the $j_{prio}$ and $j_{cont\_prio}$. The scheduling results of contention link set $l^1_{i\_cont}$ is decided by the node with the highest contention priority ($Node_{cont\_prio}$). The scheduling results of rest links ($l_i - l^1_{i\_cont}$) is decided by the node with the highest node priority ($Node_{prio}$). If all the link schedules of a node are already formed by its neighbor with a higher priority, a node will give up its attempt to contend the channel.

We give a simple example to show how the channel scheduling is formed, as Figure 3 shows. We assume node $D$ is the node with the highest priority 4, which can observe links $AB$, $BD$, $DE$ and $EG$. Then node $D$ will first access the channel and broadcast the scheduling result of those four links to nodes $\{B, C, E, F\}$, which will distribute the scheduling information to nodes $A$ and $G$ during the reservation information exchange period (Section 3.3).

We also consider the case that two nodes observe the same number of links, their node priorities ($Node_{prio}$) are the same and their channel accesses will experience a collision. For example, we assume the nodes priorities and the contention link sets in Figure 3 are:

- $A_{prio} = D_{prio} = G_{prio} = 6$
- $B_{prio} = C_{prio} = E_{prio} = F_{prio} = 2$
- $l^1_{B\_cont} = l^1_{C\_cont} = \{AB, DB\}$
- $l^1_{E\_cont} = l^1_{F\_cont} = \{DE, GE\}$

6
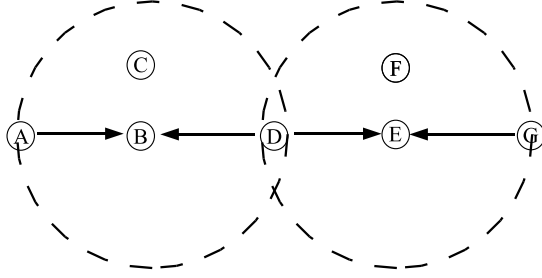
**Fig. 3.** Channel scheduling formation example

In the first round of scheduling, node $A$ and $D$ will have the same $T_{backoff}$ and collide at hidden terminals $B$ and $C$, node $D$ and $G$ will collide at hidden terminals $E$ and $F$.

We use *failed-link contender election* to solve this problem. as Algorithm 1 shows. We define $l^1_{i\_fail}$ as the set of links which are collected by node $i$ during one-hop broadcast period with type *failure*. $l^2_{ij\_fail}$ is the set of links node $i$ has received from node $j$ during one-hop re-broadcast period with type *failure*. Then at the end of neighbor information exchange period, we compare the $l^1_{i\_fail}$ and $l^2_{ij\_fail}$ for each neighbor $j$. If $l^1_{i\_fail} \subset l^2_{ij\_fail}$, node $i$ will not schedule the failed links, just follow the schedule result of node $j$. If $l^1_{i\_fail} = l^2_{ij\_fail}$, we compare the $i_{prio}$ and $j_{prio}$, the one with a higher priority will be elected as the *contender*. If $i_{prio} = j_{prio}$, we further compare the MAC address to break the tie.

Now when we revisit the previous example, node $B$ and $C$ will elect one node as *contender*. According to the backoff scheme we have introduced, this *contender* will access the channel before nodes $A$ and $D$ to build the scheduling for contention link set $\{AB, DB\}$. Then nodes $A$ and $D$ just need to schedule the rest four links. It is the same case for nodes $\{D, E, F, G\}$.

The length of the contention-based channel reservation period ($T_{cr}$) needs to be long enough to for nodes with the same highest contention priority to send their reservation packets, which is the worst case for the contention-based channel reservation. In this paper, based on the simulation experiment, we set $T_{cr}$ as $6 \times (T_{max\_backoff} + T_r)$, where $T_{max\_backoff}$ is the maximum back-off time and $T_r$ is the needed to send a reservation packet.

### 3.3 Reservation information exchange period

During the reservation information exchange period, nodes broadcast the scheduling results they have received and the related $Node_{prio}$ to the neighbors, thus the scheduling results are distributed in the two-hop range. If a node receives a different channel scheduling with the same priority, it will mark the *type* of corresponding link as *failure*.

**Algorithm 1** Failed-link contender election algorithm

/* First step, get the contend link sets */
1: **for each** node $j, k \in N_i^1$ **do**
2:    **if** $j\_prio == k\_prio == max(N_{i\_prio}^1)$
3:          && $max(N_{i\_prio}^1) > i\_prio$ **then**
4:   ▷ /* node $j$ and node $k$ are the highest among all node $i$'s one-hop neighbors */;
5:       $l_{i\_cont}^1 = l_i^1 \cap l_{ij}^2 \cap l_{ik}^2$;
6:       $|l_{i\_cont}^1| = |l_i^1 \cap l_{ij}^2 \cap l_{ik}^2|$;
7:    **end if**
8: **end for**
/* Second step, elect the contender for the failed links */
9: **for each** node $j \in N_i^1$ **do**
10:    **if** $l_{i\_fail}^1 \subset l_{ij\_fail}^2$ **then**
11:       $contender = j$;
12:    **end if**
13:    **if** $l_{i\_fail}^1 == l_{ij\_fail}^2$ **then**
14:       **if** $i_{prio}! = j_{prio}$ **then**
15:          $contender = max\_prio(i, j)$;
16:       **else**
17:          $contender = max\_mac\_address(i, j)$;
18:       **end if**
19:    **end if**
20: **end for**

The length of the reservation information exchange period ($T_{re}$) needs to be long enough to allow every node to broadcast once, which is half of the neighbor information exchange period ($T_{re} = N_{max1} \times T_b$).

### 3.4 Schedule-based transmission period

In schedule-based transmission period, each node will follow the channel scheduling with the highest priority to send its own packets. The length of the schedule-based transmission period ($T_{st}$) equals to $N_{mt} \times T_{data}$, where $T_{data}$ is the time needed to send a data packet with the maximum payload length. When a node experiences a collision during the schedule-based transmission, it will mark the *type* of corresponding link as *failure*.

### 3.5 Network Merge Consideration

Under mobile scenarios, when two networks which are in different time sections merge into one, they need to synchronize on one time section to form the correct scheduling. We address this problem by *leader election*. We get the network *leader* information during the neighbor information exchange period (section 3.1). Then we add the *leader*, $Node_{priority}$ and current time section information in the header of each frame sent after the neighbor information exchange period. The basic principle is to detect the network merge by identifying

the *leader* of the network, then letting the network with the fewer transmissions to synchronize on the time section of the network with more transmissions. If two network have the same number of transmissions, all nodes will follow the network *leader* with a larger MAC address.

If two networks merge before neighbor information exchange period and both of them have not formed the *leader* and *priority* information, they can finally form a new network, although it may not have all the transmission information. If one network is already in the time section after the neighbor information exchange period while the other is not, the latter will synchronize on the time section of the former.

## 4 Performance Analysis

Through fair node election, each node in EHCA can reserve $\lfloor \frac{1}{N_{max2}+1} \rfloor$ of the schedule-based transmission period and access the channel in up to two time frames, which are at the order of $\Theta(N_{max1})$ slots. We compare the per-node throughput and maximum channel access delay $(d_{max})$ of EHCA with other channel access schemes through numerical analysis, as Table 1 shows, where $N_{max2}$ is the maximum number of nodes in the two-hop range. $N$ is the overall number of nodes in the network. It demonstrates that EHCA achieves a good balance between throughput and delay.

**Table 1.** Comparison with existing channel access schemes

| Protocol | Per-node throughput | $d_{max}$ (unit:slot) |
|---|---|---|
| EHCA | $\Theta(\frac{1}{N_{max2}})$ | $\Theta(N_{max1})$ |
| NAMA [3] | $O(\frac{1}{N_{max2}})$ | $\infty$ |
| TDMA | $\Theta(\frac{1}{N})$ | $\Theta(N)$ |
| Topology transparent [4] | $O(\frac{log^2(NN_{max2})}{N_{max2}^2 log^2 N})$ | $O(\frac{N_{max2}^2 log^2 N}{log^2 N_{max2}})$ |

## 5 Performance Evaluation

We have implemented the EHCA and NAMA under Qualnet [10]. We use the simulation setting that 50 nodes are uniformly distributed across a $400 \times 400$ square meters area. Each node uses 802.11a as the physical layer and the transmission rate is 54 Mbps. Packets are served in First-In First-Out (FIFO) order. The duration of the simulation is 90 seconds. The transmit power is set to 16 dBm, receive sensitivity to -69 dBm. We set $N_{max1}$ equal to 20 and $N_{mt}$ equal to 400. The simulations are repeated with ten different seeds to average the results for each scenario.

## 5.1 Static Topology

We use twenty Constant Bit Rate (CBR) flows with varying inter-packet times to evaluate the performance for real time applications. The packet length of the CBR flow is 512 bytes. The senders and destinations are more than two-hops away from each other. This ensures that the metrics measured are reflective of multi-hop traffic. The simulation results are shown in Figure 4.
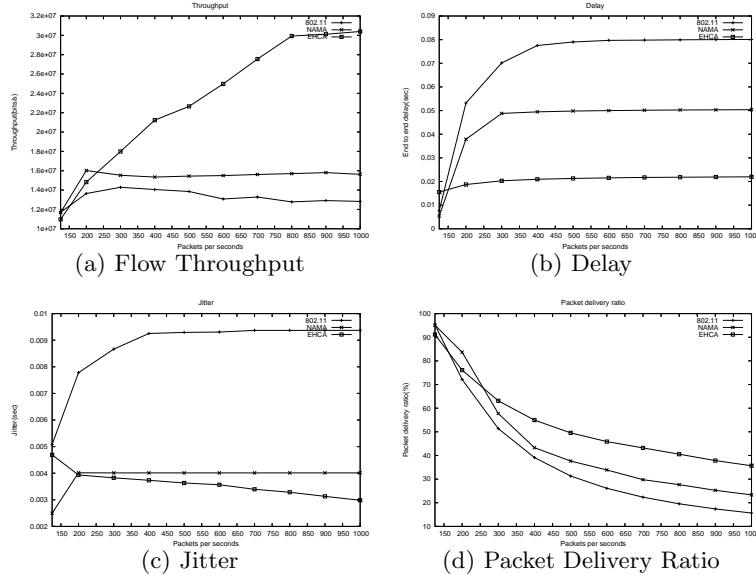


(a) Flow Throughput      (b) Delay

(c) Jitter      (d) Packet Delivery Ratio

**Fig. 4.** Static Topology

## 5.2 Dynamic Topology

We use the Random-WayPoint as the mobility model and the way point speeds randomly varying from 1 to 10 meters/second. The pause time is 10 seconds. DSR is used as the routing protocol. The results are shown is the Figure 5.

Through the comparisons of Figure 4(a) - 4(d) and Figure 5(a) - 5(d). We can see that under light traffic loads, EHCA performs similar to IEEE 802.11 and NAMA, but with the increase of the traffic load, the contention-based protocol begins to perform badly. EHCA also outperforms NAMA because NAMA does not have enough spatial reuse. The end to end delay of EHCA remains almost constant.
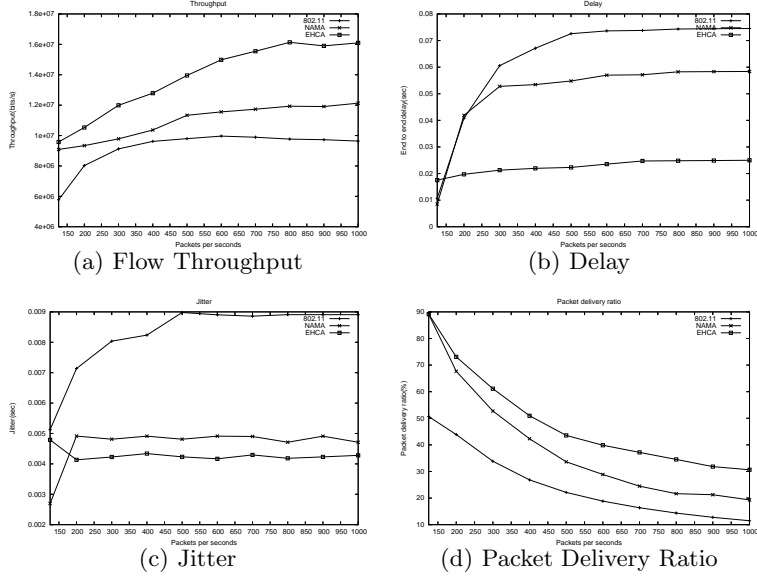
(a) Flow Throughput      (b) Delay



(c) Jitter      (d) Packet Delivery Ratio

**Fig. 5.** Mobile Topology


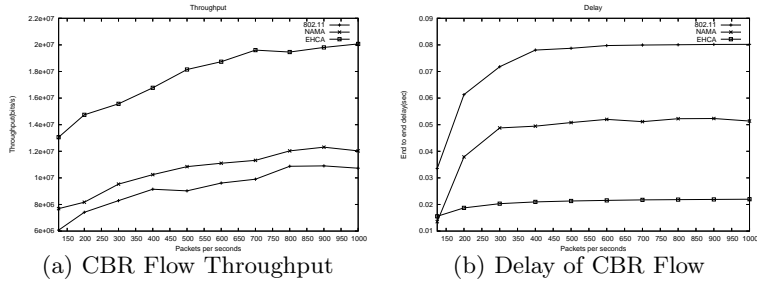
(a) CBR Flow Throughput      (b) Delay of CBR Flow

**Fig. 6.** Interaction with TCP

### 5.3 Interaction With TCP

We generate a traffic scenario which integrates the CBR traffic and TCP traffic to evaluate the interaction between EHCA and TCP. Twenty FTP flows are pumped into the network along with twenty CBR flows. The sources and destinations of the FTP flows are randomly chosen such that they are more than 2 hops away from each other. The results are shown in Figure 6(a) - 6(b), which indicate that EHCA performs well with TCP traffic.

# 6 Conclusion

This paper introduced an election based hybrid channel access (EHCA) protocol. The advantage of EHCA is that reduces the number of contention nodes through node elections, thus reducing the additional control overheads during the channel scheduling formation. EHCA can achieve high system throughput and bounded channel access delay at the same time. It is particularly suited for multi-hop ad hoc networks over which both voice and data services must be provided. We have shown through analysis and simulation that EHCA outperforms TDMA, IEEE 802.11 and existing hybrid channel access scheme.

# References

1. E. Arikan. Some Complexity Results about Packet Radio Networks. *IEEE Transactions on Information Theory*, 30(4):681–685, Jul 1984.
2. L. Bao. MALS: Multiple Access Scheduling Based on Latin Squares. In *IEEE MILCOM 2004*, October 31-November 3, 2004.
3. Lichun Bao and J. J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *ACM Seventh Annual International Conference on Mobile Computing and networking(Mobicom)*, 2001.
4. I. Chlamtac and A. Farago. Making Transmission Schedules Immune to Topology Changes in Multi-hop Packet Radio Networks. *IEEE/ACM Transactions on Networking*, 2(1):23–29, February 1994.
5. P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Trans. on Inf. Theory*, 46:388–404, March 2000.
6. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Nov 1997.
7. Ji-Her Ju and Victor O. K. Li. An Optimal Topology-Transparent Scheduling Method in Multi-hop Packet Radio Networks. *IEEE/ACM Transactions on Networking*, 6(3):298–306, June 1998.
8. Konstantinos Oikonomou and Ioannis Stavrakakis. Analysis of a Probabilistic Topology-Unaware TDMA MAC Policy for Ad-Hoc Networks. *IEEE JSAC Special Issue on Quality-of-Service Delivery in Variable Topology Networks*, 22(7):1286–1300, September 2004.
9. C. Perkins, E. Belding-Royer, and S. Das. RFC 3561- Ad hoc On-Demand Distance Vector (AODV) Routing, Jul 2003.
10. Qualnet Simulator. Scalable Network Technologies, http://www.scalable-networks.com/.
11. Carlos H. Rentel. *Network Time Synchronization and Code-based Scheduling for Wireless Ad Hoc Networks*. Ph.D. Thesis, Carleton University, January 2006.
12. T. Socolofsky and C. Kale. RFC 1180 - A TCP/IP Tutorial, Jan 1991.
13. Z. Tang and J.J. Garcia-Luna-Aceves. A Protocol for Topology-Dependent Transmission Scheduling. In *Proceedings of IEEE Wireless Communications and Networking Conference(WCNC)*, September 21-24, 1999.