Pipelined Backoff Scheme for Bandwidth Measurement in QoS Enabled Routing Towards Scalability for MANets

N.Sumathi
S.N.R.Sons College(Autonomous),
Coimbatore, India.
98940 90549
Email:sumi_karivaradan@yahoo.co.in

Dr. Antony Selvadoss Thanamani,
Head, Dept of Computer Science,
N.G.M. College (Autonomous), Pollachi, India.
Email:selvdoss@yahoo.com

ABSTRACT
MANets are very sensitive to control overhead packets due to its limited capacity. If available bandwidth is not accurately estimated, nodes will accept extra QoS requests and network will be overloaded. To improve the available bandwidth and to reduce the control overhead associated with the backoff scheme employed in MAC (Medium Access Control) layer; pipelined concept is applied to backoff procedure. Pipelined process also reduces collision probability. When the medium is busy, remaining nodes start the contention procedure in parallel for the next packet transmission. This reduces the channel waiting time. Bandwidth loss due to collision, pipelined backoff and idle time synchronization are estimated and final available bandwidth is calculated. This bandwidth measurement algorithm is integrated into routing protocol called enhanced QoS AODV (Quality of Service Ad hoc On demand Distance Vector) to find the best route based on bandwidth constraint. Simulation results show that the proposed algorithm improves the performance of the network in terms of throughput, packet delivery ratio, effective bandwidth utilization, energy consumption for different number of nodes.

Categories and Subject Descriptors
C.2.1. [**Computer-communication Networks**]: Network Architecture and Design – *Wireless communication.*

General Terms
Performance, Experimentation.

Keywords
AODV, BEB, Pipelined backoff, QoS Routing**.**

1. INTRODUCTION
Wireless medium has two states: Busy state (transmitting, receiving and carrier sensing medium), idle state [12]. Each node has to constantly monitor when the medium state changes. They dynamically exchange data among themselves when the medium is free. Each node also acts as a router to forward traffic. Each node is free to move over a certain area.

Due to the limited transmission range of wireless nodes, multi hops are usually needed for a node to exchange information with any other node in the network. For this purpose, a routing protocol is needed that quickly adapts to dynamic topology.

Wireless nodes share one common medium and by default the access is managed by Binary Exponential Backoff (BEB) procedure [2]. The access mechanism of IEEE 802.11 DCF (Distributed Coordination Function) is virtual carrier sensing. It involves RTS/CTS/Data/ACK exchange (Request To Send /Clear To Send), in which data packets are transmitted [2] when the medium is sensed idle as shown in fig 1. Medium can be viewed in discrete time slots and all nodes are synchronized in time slots. Virtual carrier sensing reduces the probability of two nodes transmitting simultaneously. It is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism where nodes listen to the medium before transmission [14]. It is used for contention based service.

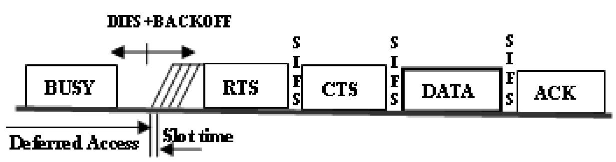

Figure 1. RTS/CTS Access Scheme of IEEE 802.11

The proposed algorithm was motivated by the limited bandwidth available in wireless communications. QoS routing searches for a path that has sufficient bandwidth to meet the bandwidth requirements of the flow. To improve the accuracy of available bandwidth, bandwidth loss due to collision probability, back off procedure, idle period synchronization and utilized bandwidth are taken into account in available bandwidth calculation [4]. Existing work uses BEB to reduce collision and AODV as the routing protocol. BEB follows serial transmission. Also there exists unfair medium access and routing overhead due to collision. To overcome these drawbacks, Implicit Pipelined Backoff Algorithm (IPBA) is proposed. This helps to reduce collision overhead and improves available bandwidth which in turn improves the network performance parameters like throughput, delay, energy consumption and packet delivery ratio. Enhanced QoS AODV is used to find the route from source to destination based on available bandwidth. This paper is organized as follows. Section 2 summarizes related work on backoff algorithms. Section 3 discusses the pipelined backoff algorithm used in available bandwidth measurement and routing protocol. Section 4 shows simulation results and Section 5 concludes the paper.

2. RELATED WORK

MANets (Mobile Adhoc Networks) are very sensitive to control overhead packets as its bandwidth is limited. Backoff algorithm is used to reduce collisions when more than one node tries to access the common medium at a time. In MILD (Multiple Increase and Linear Decrease) [1], CW size is multiplied by 1.5 on collision and decreased by 1 on a successful transmission. It cannot adjust its contention window fast enough because of its linear decrease mechanism. It performs well when the network load is heavy. In BEB [11], even when number of nodes increase to a large value, nodes will use the same CW size. It has the problem of medium capture effect which results in medium domination by successful nodes. In [18], authors discussed to enhance the performance of DCF in the presence of noisy channels. FCR (Fast Collision Resolution) discussed in [8] solves collision more quickly than BEB. Exponential Increase Exponential Decrease (EIED) algorithm discussed in [16] increases or decreases CW exponentially by backoff factors r_I, r_d respectively. Performance is good when r_I=2 and $r_d=2^{1/8}$. EIED outperforms BEB and MILD.

When the number of active nodes changes from high to low, LMILD scheme (Linear/Multiplicative Increase Linear Decrease) [7] out-performs the BEB and MILD algorithms for a wide range of network sizes. GDCF(Gentle DCF) discussed in [5] is flexible for supporting priority access for different traffic types and is very easy to implement it, as it does not require any changes in control message structure and access procedures. Compared to FCR, GDCF achieves better throughput and fairness. P-DCF (Predictive DCF) [13] enables mobile nodes to choose their next backoff times in the collision-free backoff range from the past history of successful transmissions. Mobile nodes calculate the number of idle time slots between priori and current successful transmission. Log based backoff algorithm introduced in [10, 17] uses logarithm of current backoff time to calculate next backoff. The difference between two backoff timers is small. So the chance of losing the medium access is less, which improves the throughput performance. To reduce bandwidth loss due to collision, NBA (Neighborhood Backoff Algorithm) proposed in [9] discussed that optimum value of minimum CW depends on number of contending nodes and traffic. Adaptive BEB++ [6] is designed to consider packet error rate and probability of failed transmission due to noisy mediums.

In DIDD [3] backoff algorithm (Double Increment Double Decrement), CW decreases smoothly after a successful packet transmission. It achieves better performance than BEB. Multi Chain Backoff (MCB) algorithm discussed in [23] allows nodes to adapt to different congestion levels by using more than one backoff chain together with collision events. MCB can achieve a higher throughput by still maintaining fair medium access than the existing backoff algorithms. The k-EC (k-round Elimination Contention) scheme exhibits high efficiency and robustness during the collision resolution [24]. It is insensitive to the number of active nodes. All these algorithms follow serial packet transmission which introduces more medium idle time and collision overhead. Most of the bandwidth is wasted due to collision overhead. Hence to reduce the medium idle time and overhead associated with collision, IPBA is proposed.

3. PIPELINED BACKOFF MECHANISM

3.1 Available Bandwidth Measurement (ABM)

Available bandwidth of a node can be improved by considering medium utilization, idle period synchronization and bandwidth loss due to the collision probability. The steps to measure the available bandwidth (ABM) is given below [4].

1. Evaluate capacity of a node and estimate available bandwidth.

2. Estimate the link's available bandwidth and integrate into ABM. It depends on channel utilization ratio and idle period synchronization.

3. Estimate collision probability and integrate into ABM.

4. Collision leads to retransmission of same frames. When collision occurs, default backoff algorithm supported by DCF is executed. This is an additional overhead which affects the available bandwidth. Bandwidth loss due to this additional overhead is evaluated and integrated into ABM. To reduce control overhead IPBA is proposed which is explained in sec. 3.2

5. Finally estimate the available bandwidth and store in all the nodes with the help of Hello packets.

6. Routing protocol called enhanced QoS AODV finds the route based on this available bandwidth.

In the existing algorithm, default backoff (BEB) mechanism is used to reduce collision probability. In BEB, nodes go through a contention resolution and packet transmission stages sequentially [19, 20] which is shown in fig.2. Contention resolution stage consumes medium bandwidth. Thus time spent on contention resolution is reduced when probability of collision is small. But it is difficult to achieve. Contention resolution cannot be started until the current transmission finishes. Medium idle time and contention overhead is more because of this serial transmission. Also access to a slot is not uniform in BEB. Only the winners repeatedly get the chance to access the medium. This leads to medium capture effect. In a heavily contended network, the collision probability increases which degrades the performance. The main aim is to apply pipelining technique to DCF backoff algorithm to reduce the collision overhead and to improve the available bandwidth.

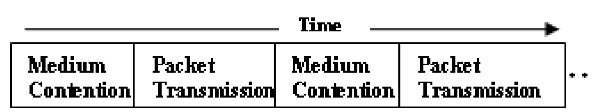

Figure 2. Serial Packet Transmission in BEB

3.2 Implicit Pipelined Backoff Mechanism

The concept of pipelining is to divide the total task into many sub tasks and executing these sub-tasks in parallel [20]. This

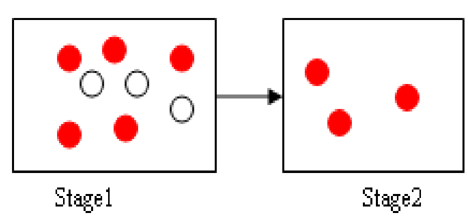

Figure 3. Contending Nodes in Pipelining

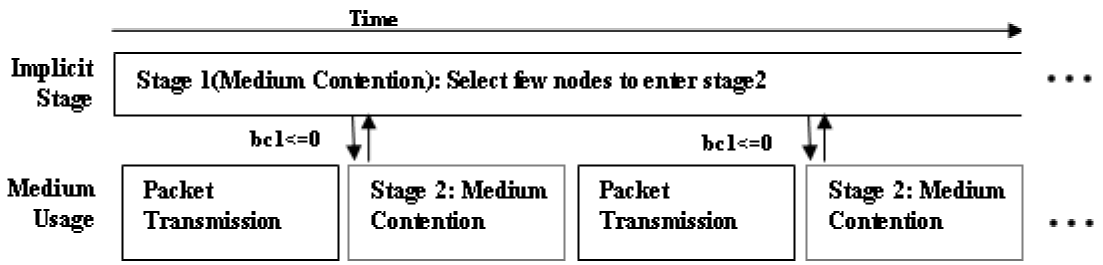

Figure 4. Medium access using implicit pipelining technique

concept is applied to medium contention procedure of MAC protocol to control number of contending nodes [21]. Fig.3 shows that there are more number of contending nodes in stage1 and less number of pipelined nodes in stage2. This pipelined medium contention procedure consumes little bandwidth but improves the performance.

Implicit Pipelined Backoff Algorithm implicitly pipelines the contention resolution stages as stage1 and stage2. It does not consume medium bandwidth. Stage1 functions as a filter to select few nodes to contend for medium in Stage2 [22] as shown in fig. 4. The medium contention can be solved effectively because the number of nodes in stage2 is small. It reduces collision probability and improves medium utilization.

3.2.1 Stage1

Let CW1 be the contention window for stage1, bc1 be the backoff counter value. It has CW1min and CW1max. The initial value of CW1 is CW1min. The value of bc1 is randomly selected from the interval [0, CW1]. If bc1 is less than or equal to zero the node becomes the pipelined node and enters stage2. After a successful packet transmission bc1 value is reduced by F. The value of F depends on number of successfully transmitted packets (tp) heard by contending nodes in stage1 i.e. $F=2^{tp} - 1$. If the value of F is larger, then the probability of becoming a pipelined node is more. Bandwidth is wasted when the medium is idle and no nodes are ready to transmit packets. To avoid this loss, bc1 is also linearly decreased by 1 after each idle slot. When bc1 reaches zero it enters into stage2. If any pipelined node wins the medium in stage2 and transmits its packets successfully then set CW1 to max [CW1 / 2, CW1min+1], tp to 1 and recalculates bc1 from the interval [0, CW1] and return to stage1. If it loses the medium in stage2, its CW1 is set as min [2*CW1+1, CW1max+1], then recalculates bc1 from [0, CW1], resets tp to 1 and returns to stage1.

3.2.2 Stage2

Let CW2 be the contention window of stage2, bc2 be the backoff counter, CW2min be the minimum value of contention window and CW2max be the maximum value of contention window. Initial value of CW2 is CW2min for all nodes entering into stage2. Backoff counter bc2 is calculated from the interval [0, CW2]. As in stage1 bc2 value is also reduced by 1 after each idle slot. Whenever bc2 reaches zero, transmission is allowed. The pipelined node has to wait for the medium to be idle for DIFS duration before backoff procedure starts. If bc2 reaches zero and medium is idle it begins its transmission. Otherwise bc2 is decremented by 1 after each idle slot. Before bc2 reaches zero if a frame is sent by other nodes then this pipelined node looses medium access and returns to stage1. When collision occurs CW2 is doubled and new bc2 is calculated from the interval [0, CW2].

Colliding nodes and pipelined nodes stay in stage 2 and repeat the same procedure. If a pipelined node wins the medium, it transmits the packets successfully, then resets CW1 to max[CW1 / 2, CW1min+1], CW2 to max[CW2 / 2, CW2min+1]their minimum values, tp to 1, calculates bc1 from interval [0, CW1] and go back to stage1. When a pipelined node looses the medium, it doubles CW1 and reset CW2, tp to 1, then calculates bc1 and goes back to stage1. Thus a node has two ways of reducing bc1. One way is through overheard successful transmission and the other way is after each idle slot. The pseudo code for pipelined backoff algorithm is given below.

Implicit Pipelined Backoff Algorithm:

1. For all nodes that want to access medium (Stage1)
   a. Set node_status=‘contending’.
   b. Initialize CW1min, CW1max.
   c. Set CW1 to CW1min, tp = 1.
   d. bc1 = Rand ([0, CW1]).
2. For each contending node, do the following
   a. If it overhears successful transmission then
      tp = tp + 1, F = $2^{tp} – 1$, bc1 = bc1 – F.
   b. If bc1 ≤ 0 then node_status = ‘pipelined’, go to step 3.
      Else
      For each idle slot
      bc1 = bc1 – 1.
      If bc1 = 0 node_status =‘pipelined’, bc2=0,
      initialize CW2min,set CW2=CW2min, go to step 3bii.
3. For each pipelined node do the following (Stage2)
   a. Initialize CW2min,CW2max, set CW2 to CW2min,tp = 1,
      bc2 = random ([0,CW2]).
   b. For each idle slot
      i. bc2=bc2-1.
      ii. If bc2=0 then
         Transmit a packet,
         CW1= max[CW1 / 2, CW1min+1],
         CW2= max[CW2 / 2, CW2min+1], tp=1,
         bc1=rand ([0, CW1]),
         node_status=’contending’, go to step 2.

Else if pipelined node looses the medium

CW1= min[2*CW1+1, CW1max+1],

CW2= CW2min,

bc1=rand ([0, CW1]), tp=1,

node_status='contending', Go to step 2.

Else if collision occurs

CW2= min[2*CW2+1, CW2max+1],

bc2=rand ([0, CW2]).

Contention Window (CW1) size is halved after every successful transmission for a winning node. Due to this medium capture effect is reduced. Stage1 reduces both medium idle time and collision overhead. Stage2 transmits packets and consumes medium bandwidth. Without using any signaling mechanism, IPBA controls the number of pipelined nodes in stage2 effectively.

There is some bandwidth loss due to the additional overhead introduced by backoff mechanism. Hence, the proportion of bandwidth utilized by backoff timer is

$$K = (DIFS + \overline{backofftime})/T(m) \quad (1)$$

where DIFS is DCF Inter Frame Spacing, T(m) is time between two consecutively emitted frames and backofftime is the average number of slots decremented for a frame. The final estimated available bandwidth is calculated as

$$E_{final}(b(s,r)) = (1-K).(1-P).E(b(s,r)) \quad (2)$$

where E(b(s, r)) is the available bandwidth on link (s,r) by monitoring the node and link capacities, P is collision probability and K is bandwidth loss due to backoff scheme. This new available bandwidth is stored in nodes and exchanged with neighbors with the help of Hello messages. Hence this concept is called as pipelined ABM. Then the routing protocol called QoS AODV finds the route based on this available bandwidth.

3.3 Discussion on Enhanced QoS AODV

Now bandwidth consumed by this pipelined backoff (K) and collision probability (P) is evaluated to find the final available bandwidth. This bandwidth is stored in routing table of nodes [15]. The main objective of QoS routing is to find a route with enough available bandwidth to satisfy a QoS request. Each node in the network consists of two data structures such as routing table and neighbor list. Routing table at each node stores the list of reachable nodes and their bandwidth. It consists of destination_id, next_id and bandwidth. Neighbor list is used to store the information of all the neighboring nodes. The RREQ (Route REQuest) contains source-addr, source-sequence#, dest-addr, dest-sequence#, hop-cnt and application requested bandwidth. The routing algorithm is able to detect the dynamic topology and generate path between nodes and it should also handle route failures. The routing algorithm is performed in three phases. They are:

• Route discovery phase: This phase finds all possible paths from source node to destination node.

• Route maintenance phase: This phase strengthens the path between the nodes.

• QoS Loss recovery: If any node along the source to destination fails or moves away from the network, alternate paths will be generated.

4. SIMULATION RESULTS

Goal of any QoS is to provide guarantees to the application in terms of throughput, bandwidth, delay, PDR (Packet Delivery Ratio) etc. The proposed algorithm is implemented using NS2 simulator tool [25]. The duration of simulation is set to 200s with a grid size of 1000×1000 m. It selects random way point mobility model with CBR (Constant Bit Rate) traffic. This algorithm is tested with varying number of nodes. Contention window sizes depend on physical layer characteristics. The values chosen for CWmin and CWmax are 31 and 1023 for stage1 and for stage2, 15 and 1023 respectively. The parameters used to measure the performance are throughput, average end to end delay, energy consumption and packet delivery ratio. The experiment is run for 25, 50, 75, 100 nodes. Table1 shows simulation parameters.

Table 1. Parameters for Simulation

Parameter	Value
Transmission range	250 m
Carrier Sense Range	550 m
Packet Size	512 bytes
Medium Capacity	2 Mbps
Grid Size	1000×1000 m
Mobility Speed	20 s
Simulation Time	200 s
No. of nodes	25,50,75,100
CW1min-CW1max	31 - 1023
CW2min-CW2max	15 - 1023

Throughput is calculated as dividing the number of bits transmitted by the time used to transmit these bits. The packet delivery ratio is calculated as the ratio of the data packets delivered to the destination to those transmitted by the CBR source. The ability to deliver a high percentage of packets to a destination increases the overall utility of the system. The amount of data transmitted by the traffic sources varies based on the admission decisions during the simulation. Average end-end delay is calculated based on the average time required to transmit packet from the source to destination. This end-end delay includes delays caused by buffering during route discovery latency, queuing, retransmission delays, propagation and transfer times.

The performance metrics are compared with existing ABM-AODV. Fig, 5 shows throughput performance. ABM-AODV has closest performance to pipelined ABM when the number of nodes is less, where ABM obtains 24.3 kbps throughput and pipelined ABM achieves 24.4 kbps throughput. When number of nodes is increased pipelined ABM achieves 23.6 kbps throughput and ABM gets 23.9 kbps throughput. This is because packet drop rate is more in ABM. Fig. 6 shows that PDR for pipelined ABM is better compared to ABM. PDR for ABM is 93.9% while for pipelined ABM is 95.9% for 100 nodes. Pipelined ABM not only improves throughput but also reduces end to end delay and energy consumed. End to end delay of a packet includes the medium waiting time and the time spent on packet retransmission.

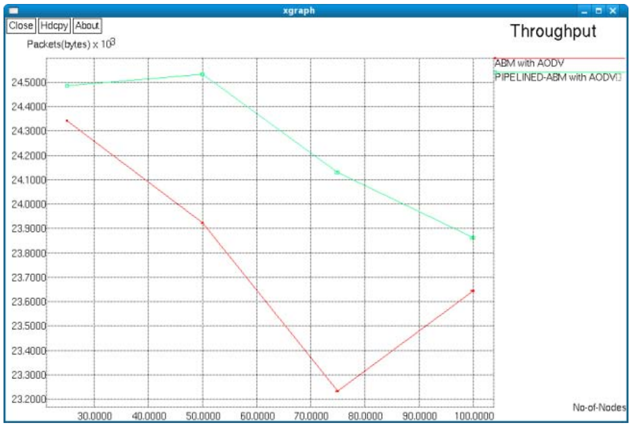

Figure 5. Throughput

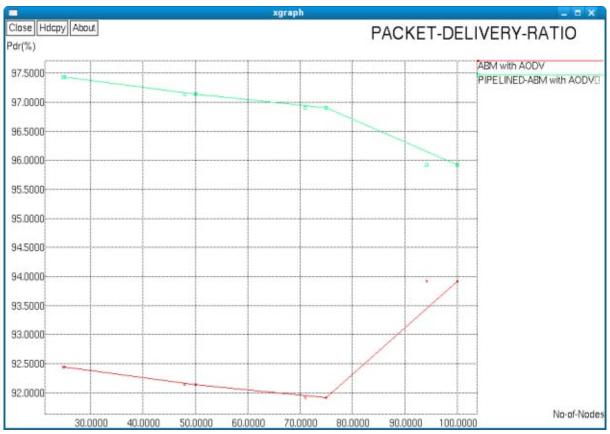

Figure 6. Packet Delivery Ratio

In pipelined ABM more number of retransmissions is avoided due to reduced collision. Also medium waiting time is decreased. It has less end to end delay for large number of nodes. When n=100, delay for pipelined ABM is 1.69ms while ABM has 2.1ms which is shown in fig. 7.

Figure 7. Average End – End Delay

As number of retransmissions is reduced, pipelined ABM consumes less energy. Initially, 1000 joules is assigned to every node. Pipelined ABM consumes 218 joules/kbps of energy but ABM consumes more energy which is 237joules/Kbps as shown in fig. 8. Number of packets dropped in pipelined ABM is less compared to ABM. This is because of reduced collision probability. Fig. 9 shows the bandwidth consumed is less to achieve maximum throughput for pipelined ABM.

Figure 8. Energy Consumption

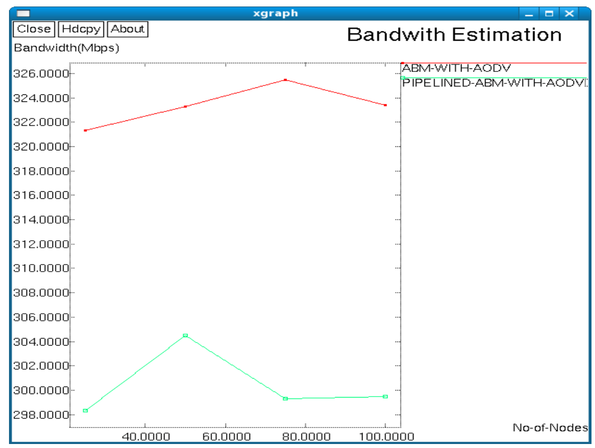

Figure 9. Bandwidth Consumption

5. CONCLUSION

This paper introduces Implicit Pipelined Backoff Algorithm to reduce the overhead associated with collision and to improve the accuracy of available bandwidth. After estimating the available bandwidth, this algorithm uses enhanced QoS AODV protocol to find the best path between source and destination with bandwidth as an additional constraint. Pipelined backoff stages consume less bandwidth which is negligible. Performance of QoS routing is evaluated based on the bandwidth information obtained from the MAC layer. It makes the utilization of resources more efficient by minimizing the unnecessary control messages and stopping the traffic that cannot meet the given QoS requirements. Results show that there is a better performance for this proposed algorithm.

6. REFERENCES

[1] Bharghavan, A. Demers, S. Shenker, and L. Zhang. 1994. MACAW: A Media Access Control Protocol for Wireless LANs. *ACM SIGCOMM Conference.*.

[2] Bianchi.G. 2000.Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected areas in Communication.*,(March 2000), Vol. 18. No. 3.

[3] Chatzimisios.P, A.C. Boucouvalas, V. Vitsas, A. Vafiadis, A. Oikonomidis, and P. Huang. 2005. A simple and effective backoff scheme for the IEEE 802.11 MAC protocol. *In CITSA*, Orlando, Florida, USA.

[4] Cheikh Sarr, Claude Chaudet, Guillaume Chelius, and Isabelle Gue´ rin Lassous. 2008. Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks. *IEEE TRANS ON MOBILE COMPUTING* (OCT 2008), VOL 7, NO. 10.

[5] Chonggang Wang, Bo Li, Lemin Li. 2004. A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. *IEEE Transactions on Vehicular Technology*, Vol 53, Issue 4 (July 2004) Page(s): 1235 – 1246.

[6] Chunyu Hu, H.Kim, J.C. Hou. 2005. An Analysis of the Binary Exponential Backoff Algorithm in Distributed MAC Protocols.*UILU-ENG-2005-1794*.

[7] Deng.J, P.K. Varshney, and Z.J. Haas.2004. A new backoff algorithm for the IEEE 802.11 distributed coordination function. *Proc. CNDS 2004*, Sandiego, CA, (Jan. 2004).

[8] Kwon.Y, Y. Fang, and H. Latchman. 2003. Novel Medium Access Control Protocol with Fast Collision Resolution for wireless LANs. *Proc. Infocom Conf.*

[9] Mahmoud Taifour, Farid Na®_t-Abdesselam and David Simplot-Ryl.2005. Neighbourhood Backoff Algorithm for Optimizing Bandwidth in Single Hop Wireless Ad-Hoc Networks. *LIFL/IRCiCA Laboratory - INRIA POPS Project.*.

[10] Manaseer.S and M. Ould-kauoa. 2005. A New Backoff Algorithm for MAC Protocol in MANETs. *21st Annual UK Performance Engg. Workshop*, pp 159-164.

[11] Marek Natkaniec Andrzej, R. Pach.2000. An analysis of the Back-Off Mechanism Used in IEEE 802.11 Networks. *Proc the Fifth IEEE Symposium on ISCC*.

[12] Mohammed Saghir, Tat-Chee Wan, Rahmat Budiarto. 2006. QoS Multicast Routing Based on Bandwidth Estimation in Mobile Ad Hoc Networks. *Proceedings of the Int. Conf. on Computer and Communication Engineering, ICCCE'06* Vol. I, 9-11(May 2006), Kuala Lumpur, Malaysia.

[13] Nakjung Choi, Yongho Seok, Yanghee Choi, Gowoon Lee, Sungmann Kim and Hanwook Jung. 2005. P-DCF: Enhanced Backoff Scheme for the IEEE 802.11 DCF. *IEEE VTC 2005-Spring*, Stockholm, Sweden.

[14] Shan Gong. 2006. Quality of Service Aware Routing Protocols for Mobile Ad Hoc Networks. *MS theses*, (August 2006).

[15] Stéphane Lohier, Sidi Mohammed Senouci, Yacine Ghamri-Doudane and Guy Pujolle. 2003. A Reactive QoS Routing Protocol for Ad Hoc Networks(October, 2003). *LNCS-Springer*.

[16] Song.N, B.Kwak, J. Song, and L.E. Miller, 2003. Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm. *Proc. IEEE VTC 2003-Spring*.

[17] Sumathi.N, Antony Selvadoss Thanamani. 2009. An improved QoS based routing in available bandwidth measurement using logarithmic backoff algorithm in MANets. *ICADIWT '09*, London.

[18] Wu.H and S. Cheng. 2002. IEEE 802.11 Distributed Coordination Function (DCF): Analysis and Enhancement. *Proc. IEEE ICC*, (Apr 2002) vol. 1, pp. 605-609.

[19] Yang.X and N.H. Vaidya. 2002. Pipelined Packet Scheduling in Wireless LANs. *Technical report,* Coordinated Science Laboratory, Univ. of Illinois at Urbana-Champaign.

[20] Yang.X and N.H. Vaidya. 2003. Explicit and Implicit Pipelining for Wireless Medium Access Control. *Proc IEEE Seminar Vehicular Technology Conf.*

[21] Yang.X N.H.Vaidya.2002. DSCR – A Wireless MAC Protocol Using Implicit Pipelining. *Technical report*, Coordinated Science Laboratory, University of Illinois.

[22] Yang.X, and N.H. Vaidya. 2006. A Wireless MAC Protocol Using Implicit Pipelining. *IEEE TRANS ON MOBILE COMPUTING*,(Mar 2006) VOL. 5, NO. 3.

[23] Ye.R and Y.-C. Tseng. 2006. A Multichain Backoff Mechanism for IEEE 802.11 WLANs. *IEEE Transactions on Vehicular Technology* (Sep. 2006) vol. 55, Issue 5.

[24] Zhou, A. Marshall, and T Lee. 2007. A k-round Elimination Contention Scheme for WLANs. *IEEE Transactions On Mobile Computing*.

[25] NS manual http://www.isi.edu/nsnam/ns/ns-documentation.html.