



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY
Volume 11 Issue 3 Version 1.0 March 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Performance Evaluation of Varying Contention Window Size for Bandwidth Constrained Routing in Adhoc Networks

By N.Sumathi, Dr. Antony Selvadoss Thanamani

Abstract- Providing bandwidth efficient routing in ad hoc networks is a challenging task. Available bandwidth of nodes is accurately evaluated before finding route from source to destination. Accuracy of available bandwidth is mainly affected by collision and overhead due to the execution of backoff scheme. Existing bandwidth constrained routing uses binary exponential backoff which follows serial transmission and causes unfair channel access. To overcome these, implicit pipelined backoff procedure is proposed to improve the available bandwidth and reduce the overhead associated with the backoff scheme employed in medium access control layer. In this, when two nodes are sharing the channel, the remaining nodes start the channel contention procedure in parallel to transmit next packet. Thus the channel waiting time is reduced. Each node maintains separate contention window for each phase in pipelined backoff. Proper choice of contention window size has great effect on performance of the network. This proposed algorithm is combined with a reactive link disjoint multipath routing protocol called AOMDV (Adhoc Ondemand Multipath Distance Vector) to find the best path based on bandwidth. Experimental results show that this algorithm outperforms existing approach in terms of QoS metrics such as delay, throughput, packet delivery ratio and energy consumption for different contention window sizes.

Keywords: BEB, Contention Window, Link disjoint AOMDV, Pipelined Backoff, QoS Routing.



Strictly as per the compliance and regulations of:



Performance Evaluation of Varying Contention Window Size for Bandwidth Constrained Routing in Adhoc Networks

March 2011

N.Sumathi, Dr. Antony Selvadoss Thanamani

71

Abstract- Providing bandwidth efficient routing in ad hoc networks is a challenging task. Available bandwidth of nodes is accurately evaluated before finding route from source to destination. Accuracy of available bandwidth is mainly affected by collision and overhead due to the execution of backoff scheme. Existing bandwidth constrained routing uses binary exponential backoff which follows serial transmission and causes unfair channel access. To overcome these, implicit pipelined backoff procedure is proposed to improve the available bandwidth and reduce the overhead associated with the backoff scheme employed in medium access control layer. In this, when two nodes are sharing the channel, the remaining nodes start the channel contention procedure in parallel to transmit next packet. Thus the channel waiting time is reduced. Each node maintains separate contention window for each phase in pipelined backoff. Proper choice of contention window size has great effect on performance of the network. This proposed algorithm is combined with a reactive link disjoint multipath routing protocol called AOMDV (Adhoc Ondemand Multipath Distance Vector) to find the best path based on bandwidth. Experimental results show that this algorithm outperforms existing approach in terms of QoS metrics such as delay, throughput, packet delivery ratio and energy consumption for different contention window sizes.

Keywords: BEB, Contention Window, Link disjoint AOMDV, Pipelined Backoff, QoS Routing.

I. INTRODUCTION

Mobile ad hoc networks (MANET) consist of a collection of wireless mobile nodes which dynamically exchange data among themselves without any base station or infrastructure. Each node acts as a router to forward traffic and is free to move over a certain area. The wireless mobile nodes may enter as well as leave the network dynamically. MANETS are suitable for emergency situations like natural disasters, military conflicts, medical situations etc. MAC layer protocols use a single channel which is shared by many contending nodes. MAC layer standard called

IEEE 802.11 DCF (Distributed Coordinate Function [1] is based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol in which nodes listen to the channel before transmission to avoid collision. DCF does not have a central control and known for its asynchronous data transfer. It is used for contention based service. A mobile node uses the virtual carrier sensing mechanism, which utilizes Request-To-Send (RTS) and Clear-To-Send (CTS) exchanges with a Short InterFrame Space (SIFS) for channel reservation. Channel can be viewed in discrete time slots and all nodes are synchronized in time slots. Virtual carrier sensing reduces the probability when two nodes are trying to transmitting simultaneously.

Once the node detects that the channel has been free for duration of DCF InterFrame Space (DIFS), it starts a Binary Exponential Backoff (BEB) procedure, i.e., decrementing its back-off counter as long as the channel is idle. If the backoff counter has reduced to zero and the channel is still free, the node begins to transmit. If the channel becomes busy in the middle of the decrement, the node freezes its backoff counter, and resumes the countdown after deferring for a period of time, which is indicated by the Network Allocation Vector (NAV) stored in the transmitted node's packet header [2]. It is possible that two or more nodes begin to transmit at the same time. In such a case, collision occurs. Collisions are detected when there is no CTS or acknowledgement (ACK) from the receiver. After collision, all the involved nodes double their CWs (up to a maximum value-CW_{max}) and compete to gain control of the channel next time. If a node succeeds in channel access, the node resets its CW to CW_{min}.

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 [3]. It is essential to perform routing with maximal throughput and with minimal control overhead. Network layer discovers QoS routes based on MAC layer results. In this paper QoS based routes are traced with bandwidth as a main metric. If the available bandwidth is

About¹- S.N.R.Sons College (Autonomous), Coimbatore.
Email:sumi_karivaradan@yahoo.co.in

About²-Head, Dept of Computer Science, N.G.M. College (Autonomous), Pollachi.

not estimated accurately, nodes will accept the extra QoS requests and network will be overloaded. 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 approach uses BEB as the backoff procedure and AODV as the routing protocol. BEB follows serial transmission. Channel idle time and contention overhead is more because of this serial transmission. Nodes go through a channel contention and packet transmission stages sequentially [5]. Channel contention stage consumes channel bandwidth. Thus time spent on channel contention is reduced when probability of collision is small. But it is difficult to achieve. Because channel contention cannot be started until the current transmission finishes. Also access to a slot is not uniform. Only the winners repeatedly get the chance to access the channel. This leads to channel capture effect. In a heavily contended network, the collision probability increases which degrades the performance. Hence the main objective is to apply pipelining technique to DCF backoff algorithm to reduce the collision overhead and improve the available bandwidth. In case of AODV, link failure causes the execution route discovery procedure to find alternate route for transmission which leads to packet loss and delay. To overcome, enhanced AOMDV is used to find the routes based on available bandwidth from source to destination.

This paper is organized as follows. Section II summarizes related work on backoff algorithms. Section III discusses the pipelined backoff algorithm used in available bandwidth measurement and routing protocol. Section IV shows simulation results and Section V concludes the paper.

II. RELATED WORK

Backoff algorithm is used to reduce collisions when more than one node tries to access the common channel at a time. In MILD (Multiple Increase and Linear Decrease) [6], CW (Contention Window) 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 [7], authors discussed to enhance the performance of DCF in the presence of noisy channels. Exponential Increase Exponential Decrease (EIED) algorithm discussed in [8] increases or decreases CW exponentially by backoff factors r_1 , r_d respectively. Performance is good when $r_1=2$ and $r_d=2^{1/8}$. EIED outperforms BEB and MILD. FCR (Fast Collision Resolution) discussed in [9] solves collision more quickly than BEB. When the number of active nodes

changes from high to low, LMILD scheme (Linear/Multiplicative Increase Linear Decrease) [10] outperforms the BEB and MILD algorithms for a wide range of network sizes. Authors discussed that the CW resetting scheme causes a very large variation of the CW size and degrades the performance of a network when it is heavily loaded since each new packet starts with minimum CW which is too small for heavy network load. GDCF (Gentle DCF) discussed in [11] 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. GDCF with smaller number of retransmissions achieves higher throughput for small n and unfair channel access. Throughput drops dramatically when number of nodes (n) increases. P-DCF (Predictive DCF) [12] enables mobile nodes to choose their next backoff times in the collision-free backoff range from the past history of successful transmissions. In DIDD [13], backoff algorithm (Double Increment Double Decrement) CW decreases smoothly after a successful packet transmission. Its throughput is better for $CW_{min}=32$ but the delay is high. It achieves better performance than BEB. In NBA [14], large bandwidth is wasted due to collisions. Optimum value of minimum CW depends on number of contending nodes and traffic. Optimum value of $CW_{min}=8.5N-5$ ie $\alpha N + \beta$. In BEB [15, 16], even when number of nodes increase to a large value, nodes will use the same CW size. So lot of collisions occurs and throughput is reduced. At the beginning of each slot a node transmits, if its backoff timer has expired. Otherwise depending on the channel state (idle or busy), the node will count down the backoff counter by 1 or will be frozen at a value. Such an algorithm is embedded in IEEE 802.11 DCF. Delay performance of BEB suffers from collisions which causes more number of retransmissions. It has the problem of channel capture effect which results in channel domination by successful nodes. Adaptive BEB++ [16] is designed to consider packet error rate and probability of failed transmission due to noisy channels. After a successful transmission, CW size is set to an optimal value. It adjusts minimum CW size according to active number of nodes. Log based backoff algorithm introduced in [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 channel access is less, which improves the throughput performance.

Multi Chain Backoff (MCB) algorithm discussed in [18] 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 channel access than

the existing backoff algorithms. Each station maintains a transition diagram to determine its current CW with c chains. When a collision is encountered nodes move to a chain with a larger CW_{min} and if no collision is detected. It is moved to a chain with smaller CW_{min}. Conclusion MCB uses multiple backoff chains with minimum CW. Based on collision, it chooses a proper chain. So it offers higher throughput than others. The k-EC (k-round Elimination Contention) scheme exhibits high efficiency and robustness during the collision resolution [19]. It is insensitive to the number of active nodes. All the above algorithms follow serial packet transmission which introduces more channel idle time and collision overhead. Most of the bandwidth is wasted due to collision overhead. Hence to reduce the channel idle time and overhead associated with collision, implicit pipelined backoff algorithm is proposed.

III. PIPELINED BACKOFF MECHANISM

a. Pipelined Available Bandwidth Measurement (Pipelined ABM)

Available bandwidth of a node can be improved by considering channel utilization ratio, idle period synchronization and bandwidth loss due to the collision probability. The existing procedure uses BEB to reduce collision [4].

1. Evaluate the capacity of a node and estimate the available bandwidth.
2. Estimate the link's available bandwidth. It depends on channel utilization ratio and idle period synchronization. Let it be $E(b(s,r))$. It is calculated based on the probability that the channel is free simultaneously at the sender and receiver side.
3. Estimate collision probability P_m for m bits.
4. Collision leads to retransmission of same frames. When collision occurs, implicit pipelined backoff algorithm (IPBA) is executed. This algorithm helps to reduce collisions when more than one node tries to access the common channel. This algorithm is explained in section C.

Usually backoff algorithm leads to additional overhead which affects the available bandwidth. Bandwidth lost due to this additional overhead is evaluated. Let it be K .

$$K = (\text{DIFS} + \text{backoff}) / T(m) \quad (1)$$

where DIFS is DCF Inter Frame Spacing, $T(m)$ is time between two consecutive frames and backoff is the average number of slots decremented for a frame. The above facts are considered and combined to estimate the available bandwidth which is given below.

$$E_{\text{final}}(b(s,r)) = (1-K) \cdot (1-P_m) \cdot E(b(s,r)) \quad (2)$$

where $E_{\text{final}}(b(s,r))$ is the available bandwidth on link by monitoring node and link capacities, P_m is collision

probability and K is bandwidth lost due to pipelined backoff scheme. This new available bandwidth is stored in nodes and exchanged with neighbors with the help of Hello messages. Then the routing protocol called AOMDV (Ad hoc On-demand Multipath Distance Vector) finds the route based on this available bandwidth.

b) 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 concept is applied to channel contention procedure of MAC (Channel Access Control) protocol. When two nodes are sharing the channel, the remaining nodes start the channel contention procedure in parallel for the next packet transmission [21]. Pipelined backoff hides channel idle time and reduces collision probability. It is also used to control number of contending nodes.

When there are few contending nodes in the network, a smaller CW will reduce the channel idle time and channel bandwidth is utilized better. When number of contending nodes is more, CW size is increased to reduce the collision probability and to achieve better throughput. The collision cost is much higher in wireless networks because a node cannot detect collision immediately [22]. This pipelined channel contention procedure consumes little bandwidth but improves the performance. Thus Implicit Pipelined Backoff Algorithm (IPBA) is proposed.

c) Implicit Pipelined Backoff Mechanism

In implicit pipelining there is no separate control channel. It implicitly pipelines the contention resolution stages as phase1 and phase2. Phase1 functions as a filter to select few nodes to contend for channel in Phase2 [22] as shown in fig. 3.1. The channel contention can be solved effectively because the number of nodes in phase2 is small. This reduces collision probability and improves channel utilization.

1) Phase1

Let FCW be the contention window (First Contention Window) for phase1, bt_1 be the backoff timer value. It has FCW_{min} and FCW_{max}. The initial value of FCW is FCW_{min}. The value of bt_1 is randomly selected from the interval $[0, FCW]$. If bt_1 is less than or equal to zero the node becomes the pipelined node and enters phase2. After a successful packet transmission bt_1 value is reduced by F . The value of F depends on number of successfully transmitted packets (tp) heard by contending nodes in phase1 i.e. $F = 2^{tp} - 1$. If the value of F is larger, then the probability of node becoming a pipelined is more. Bandwidth is wasted when the channel is idle and no nodes are ready to transmit packets. To avoid this loss, bt_1 is also linearly decreased by 1 for each idle slot. When bt_1 reaches zero it enters into phase2. If any pipelined node wins the channel in

phase2, it transmits its packets successfully then set FCW to $\max[FCW / 2, FCW_{min}+1]$, tp to 1 and recalculates bt1 from the interval $[0, FCW]$ and return to phase1. If it loses the channel in phase2, its FCW is set as $\min[2*FCW+1, FCW_{max}+1]$, then recalculates bt1 from $[0, FCW]$, resets tp to 1 and returns to phase1.

2) Phase2

Let SCW be the contention window (Second Contention Window) of phase2, bt2 be the backoff timer, SCWmin be the minimum value of contention window and SCWmax be the maximum value of contention window. Initial value of SCW is SCWmin for all nodes entering into phase2. Backoff timer bt2 is calculated from the interval $[0, SCW]$. As in phase1 bt2 value is also reduced by 1 after each idle slot. Whenever bt2 reaches zero, transmission is allowed. The pipelined node has to wait for the channel to be idle for DIFS duration before backoff procedure starts. If bt2 reaches zero and channel is idle then it begins its transmission. Otherwise bt2 is decremented by 1 after each idle slot. Before bt2 reaches zero if a frame is sent by other nodes then this pipelined node loses channel access and returns to phase1. When collision occurs SCW is doubled and new bt2 is calculated from the interval $[0, SCW]$. Colliding nodes and pipelined nodes stay in phase2 and repeat the same procedure. If a pipelined node wins the

channel, it transmits the packets successfully, then resets FCW to $\max[FCW / 2, FCW_{min}+1]$, SCW to $\max[SCW / 2, SCW_{min}+1]$, tp to 1, calculates bt1 from interval $[0, FCW]$ and go back to phase1. When a pipelined node loses the channel, it doubles FCW and resets SCW, tp to 1, then calculates bt1 and go back to phase1. Thus a node has two ways of reducing bt1. One way is through overheard successful transmission and the other way is after each channel idle slot. The pseudo code for implicit pipelined backoff algorithm is given below.

In this algorithm, Contention Window (FCW) size is halved after every successful transmission for a winning node. Due to this channel capture effect is reduced. CW sizes are doubled on collision. Phase1 reduces both channel idle time and collision overhead. Phase2 transmits packets and consumes channel bandwidth. IPBA controls the number of pipelined nodes in phase2 effectively. Now bandwidth loss due this pipelined backoff (K) is estimated and the final available bandwidth is calculated using the formula (2). This method is called as pipelined ABM. This bandwidth is stored in nodes with the help of *hello* messages. Then the routing protocol finds the route based on bandwidth stored in each node.

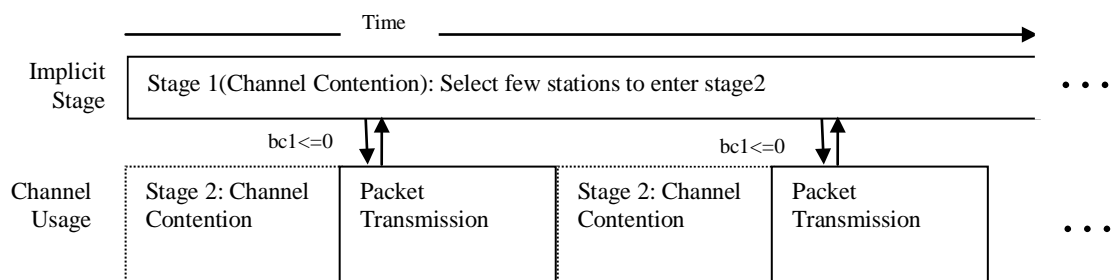


Fig 3.1. Implicit Pipelined Backoff Process

Implicit Pipelined Backoff Algorithm: Pseudo code

Phase 1:

- For all nodes that want to transmit data packets, do
 - Initialize FCW sizes(min,max) and number of packets transmitted (tp).
 - Calculate bt1 (backoff timer) randomly from $[0, FCW]$
- When a node overhears a successful transmission, increase tp by 1 and decrease bt1 by $2^{tp}-1$.
- If bt1 is less than 0, node becomes pipelined and enters into Phase2.
- Else bt1 is reduced by 1 for each idle slot. If it is 0 go to Phase 2.

Phase 2:

- For each pipelined node do the following

- Initialize SCW sizes (min ,max), calculate bt2 (backoff timer) randomly from $[0, SCW]$.
- For each idle slot
 - bt2 is decremented by 1,
 - if bt2 is 0 , transmit a packet, halve FCW, reset tp, go to phase1.
- If pipelined node loses the channel, double FCW upto its maximum, reset SCW, calculate bt1 and initialize tp, go to phase 1.
- If collision occurs SCW is doubled upto its maximum and calculate bt2.

d) Discussion on Multipath QoS routing protocol

Adding QoS to routing protocols help to optimize the performance of traffic on the network. This pipelined backoff algorithm is integrated into a routing protocol called enhanced link disjoint multipath AODV (AOMDV) [23] to find the routes from given source to destination based on available bandwidth. Usually

frequent route breaks make intermediate nodes to drop packets when there is no immediate alternate path available to the destination. This affects throughput, packet delivery ratio and increases delay. But multipath routing protocol finds multiple paths for a destination during single route discovery process. In case of link failure, source may switch to an alternate path instead of initiating another route discovery. Enhanced AOMDV modifies the base AODV protocol's route discovery mechanism, to enable discovery of multiple link-disjoint paths for a particular source node. To discover link-disjoint paths [24], each node forwards only one route request towards the destination during the route discovery process; however, it maintains a queue of the previous hop nodes for each RREQ received from a unique neighbor of the source. When a link failure occurs, the node upstream of the link detects the failure, invalidates its routing table entry for that destination and unicasts an RERR message towards the source. Once the source node receives the RERR, it switches its primary path to the next best alternate link-disjoint path. It is designed mainly for highly dynamic ad hoc networks when route breaks and link failures occur frequently. This protocol introduces extra control traffic to monitor alternate paths.

IV. SIMULATION RESULTS

Performance analysis of pipelined ABM is carried out by varying contention window sizes in phase1. Goal of any QoS is to provide guarantees to the application in terms of throughput, bandwidth, delay, packet delivery ratio (PDR) 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 50 nodes. Simulation results present throughput, delay, energy, PDR for different values of CW parameters. Graph helps us to decide the optimal value of CW as [31-1023].

The parameters used to measure the performance are throughput, average end to end delay, energy consumption and packet delivery ratio. To achieve optimum result, system parameters must be selected according to traffic condition. Throughput is calculated as dividing the number of bits transmitted by the time used to transmit these bits. Transmission time includes queuing time, transmission time, interframe space times (SIFS, DIFS etc), control message overhead time and retransmission time. The packet delivery ratio is calculated as the ratio of the data packets delivered to the destination to those transmitted by the CBR traffic. The ability to deliver a high percentage of packets to a destination increases the overall utility of the system. Because the amount of traffic 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. Table I shows simulation parameters. The performance metrics are compared by varying contention window sizes in phase1. Contention window size for phase2 is 15-1023.

Table I. Parameters for Simulation

Parameter	Value
Transmission range	250 m
Carrier Sensing range	550 m
Packet Size	512 bytes
Channel Capacity	2 Mbps
Grid Size	1000×1000 m
No. of nodes	50
FCWmin - FCWmax sizes	31-1023,127-1023, 511-4095
SCWmin-SCWmax	15-1023
Mobility Speed	20 s
Simulation Time	200 s

This algorithm is tested by varying contention window sizes such as 31-1023,127-1023 and 511-4093 and keeping window size for phase2 as 15-1023. This is because number of contending nodes is more in phase1 than number of pipelined nodes in phase2. The performance metrics with respect to number of nodes, speed and pause time are shown in Fig 4.1 – 4.5 for three sets of contention window sizes. Fig 4.1 shows that the throughput for 31-1023 window size is better. When load increases, 511- 4093 window size has closer to previous one.

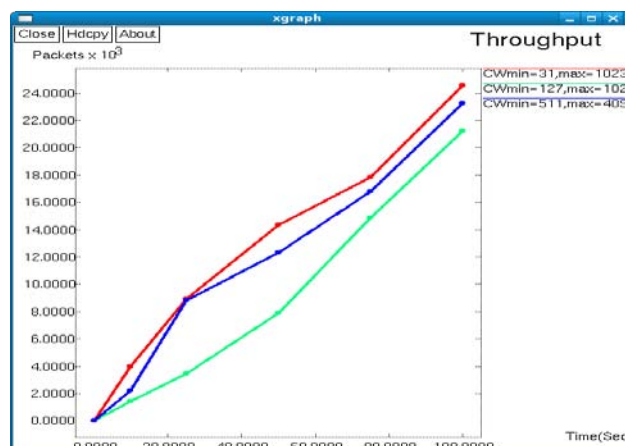


Fig. 4.1 Throughput

Average end to end delay for FCWmin =31 is almost 35% higher than other two window sizes which is shown in fig 4.2.

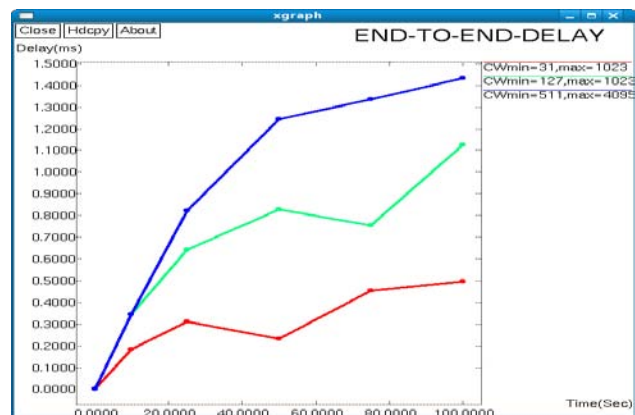


Fig.4.2 End to end delay

Number of packets delivered by the first category is more than the other two as shown in fig 4.3. Energy consumed by the nodes is less when the traffic is low for FCWmin=31. When traffic increases, energy consumption increases than FCWmin=511 as shown in fig 4.4.

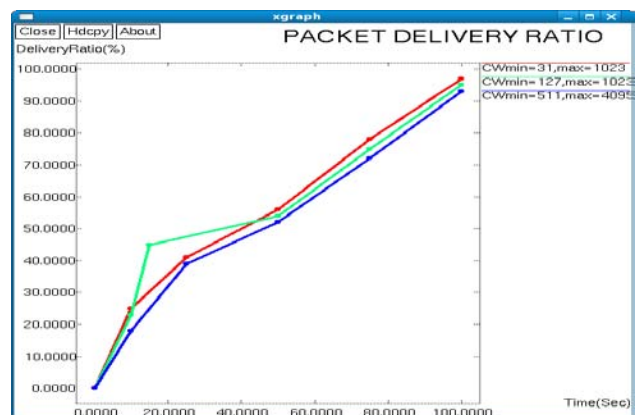


Fig. 4.3 Packet Delivery Ratio

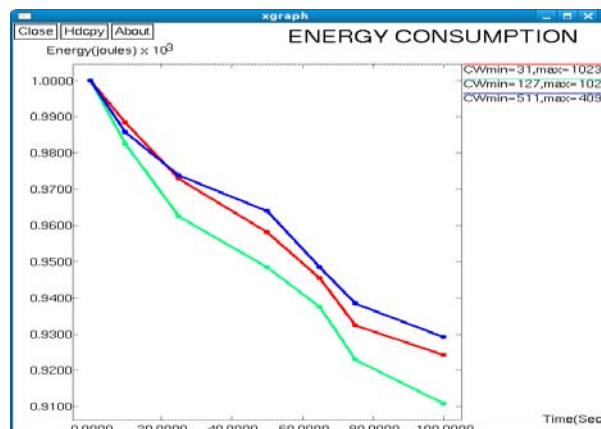


Fig. 4.4 Energy Consumption

When the number of contending nodes are less, small FCWmin will not waste channel bandwidth whereas FCWmin=511 occupies more channel bandwidth as shown in fig. 4.5. On the average, CW size 31-1023 shows better performance.

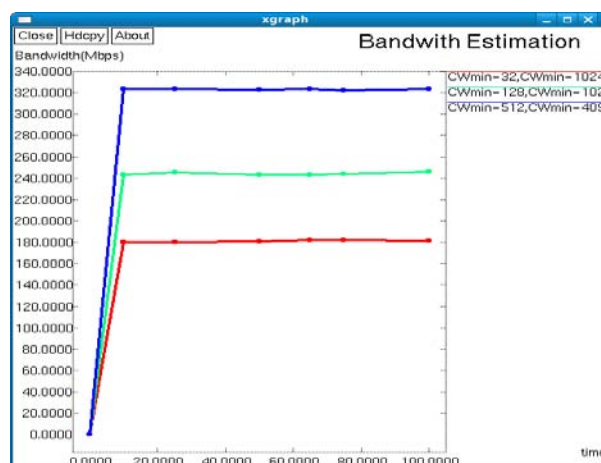


Fig. 4.5 Bandwidth Estimation

V. CONCLUSION

Performance of QoS routing is evaluated based on the bandwidth information obtained from the MAC layer. This paper discusses 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 AOMDV to find the best path between source and destination with bandwidth as an additional constraint. Pipelined backoff stages consume less bandwidth which is negligible. Pipelined ABM makes the utilization of resources more efficient by minimizing the unnecessary control messages and stopping the transmissions that cannot meet the given QoS

requirements. When the number of contending nodes is less with large value of CWmin but no one is sending the data wastes a lot of bandwidth. The choice of CWmax parameter on the network performance is little. But it cannot be reduced to a low value. Simulation results present throughput, delay, energy, PDR for different values of CW parameters. Results show that there is a better performance for FCWmin=31 and FCWmax=1023.

REFERENCES RÉFÉRENCES REFERENCIAS

- 1) G.Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", IEEE Journal on Selected areas in Communications, Vol. 18. No. 3, March 2000.
- 2) Shan Gong, "Quality of Service Aware Routing Protocols for Mobile Ad Hoc Networks", MS theses, August 2006.
- 3) Stéphane Lohier, Sidi Mohammed Senouci, Yacine Ghamri-Doudane and Guy Pujolle, "A Reactive QoS Routing Protocol for Ad Hoc Networks", October, 2003.
- 4) Cheikh Sarr, Claude Chaudet, Guillaume Chelius, and Isabelle Gue'rin Lassous, "Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks", IEEE TRANS ON MOBILE COMPUTING, VOL 7, NO. 10, OCT 2008.
- 5) X. Yang and N.H. Vaidya, "Pipelined Packet Scheduling in Wireless LANs", technical report, Coordinated Science Laboratory, Univ. of Illinois at Urbana-Champaign, Aug. 2002.
- 6) Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Control Protocol for Wireless LANs," Proc. SIGCOMM '94 Conf. ACM, pp. 212-225, 1994.
- 7) H. Wu and S. Cheng, "IEEE 802.11 Distributed Coordination Function (DCF): Analysis and Enhancement", Proc. IEEE ICC, vol. 1, pp. 605-609, Apr. 2002.
- 8) N. Song, B. Kwak, J. Song, and L.E. Miller, "Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm", Proc. IEEE VTC 2003-Spring, 2003.
- 9) Y. Kwon, Y. Fang, and H. Latchman, "A Novel Channel Access Control Protocol with Fast collision Resolution for wireless LANs," Proc. Infocom Conf., 2003.
- 10) [10] J. Deng, P.K. Varshney, and Z.J. Haas, "A new backoff algorithm for the IEEE 802.11 distributed coordination function", Proc. CNDS 2004, San Diego, CA, Jan. 2004.
- 11) [11] Chonggang Wang; Bo Li; Lemin Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF", IEEE Transactions on Vehicular Technology, Volume 53, Issue 4, July 2004 Page(s): 1235 – 1246.
- 12) [12] Nakjung Choi, Yongho Seok, Yanghee Choi, Gwooon Lee, Sungmann Kim and Hanwook Jung, "P-DCF: Enhanced Backoff Scheme for the IEEE 802.11 DCF", IEEE VTC 2005-Spring, Stockholm, Sweden, May, 2005.
- 13) P. Chatzimisios, A.C. Boucouvalas, V. Vitsas, A. Vafiadis, A. Oikonomidis, and P. Huang. "A simple and effective backoff scheme for the IEEE 802.11 MAC protocol". In CITSA, Orlando, Florida, USA, July 2005.
- 14) Mahmoud Taifour, Farid Na@_t-Abdesselam and David Simplot-Ryl, "Neighbourhood Backoff Algorithm for Optimizing Bandwidth in Single Hop Wireless Ad-Hoc Networks", LIFL/IRCiCA Laboratory - INRIA POPS Project, 2005.
- 15) Marek Natkaniec Andrzej, R. Pach "An analysis of the Back-Off Mechanism Used in IEEE 802.11 Networks", Proc the Fifth IEEE Symposium on ISCC 2000.
- 16) Chunyu Hu, H.Kim, J.C. Hou, "An Analysis of the Binary Exponential Backoff Algorithm in Distributed MAC Protocols", July 2005.
- 17) S. Manaseer and M. Ould-kauoa, "A New Backoff Algorithm for MAC Protocol in MANETs", 21st Annual UK Performance Engineering Workshop, pp 159-164, 2005.
- 18) R. Ye and Y.-C. Tseng, "A Multichain Backoff Mechanism for IEEE 802.11 WLANs", IEEE Transactions on Vehicular Technology, vol. 55, Issue 5, Sept. 2006.
- 19) Zhou, A. Marshall, and T Lee, "A k-round Elimination Contention Scheme for WLANs", IEEE Transactions On Mobile Computing, 2007.
- 20) X. Yang and N.H. Vaidya, "Explicit and Implicit Pipelining for Wireless Channel Access Control", Proc IEEE Seminar Vehicular Technology Conf., Fall 2003.
- 21) Xue Yang N.H.Vaidya, "DSCR – A Wireless MAC Protocol Using Implicit Pipelining", Technical report, Coordinated Science Laboratory, University of Illinois, December 2002.
- 22) Xue Yang, Member, IEEE, and N.H. Vaidya, "A Wireless MAC Protocol Using Implicit Pipelining", Senior Member, IEEE TRANS ON MOBILE COMPUTING, VOL. 5, NO. 3, MAR 2006.
- 23) Mahesh K. Marina Samir R. Das, "On-demand Multipath Distance Vector Routing in Ad Hoc Networks", Proceedings of the 9th International IEEE Conference on Network Protocols (ICNP), November 2001.

24) Perumal Sambasivam, Ashwin Murthy, E. M. Belding-Royer, "Dynamically Adaptive Multipath Routing based on AODV", Proc. 3rd Annual MAHN, 2004.

25) NS manual <http://www.isi.edu/nsnam/ns/ns-documentation.html>.

