Research Paper Enginering

Contention Based Fair Scheduling Protocol for Wireless LAN

*Dr. Vinod Pachghare **Vinod Maid

* Computer Engineering, College of Engineering, Pune, Maharashtra, India – 411 005

**Tech Mahindra Ltd., Phase-III, Hinjewadi, Maharashtra, India – 411 057

ABSTRACT

The IEEE 802.11 standards for Wireless Local Area network (WLANs) has been adopted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as a standard for communication. Collision Avoidance has introduced an overhead that delays should be as small as possible and secondly, the protocol should keep the number of collisions to a minimum, even under the highest possible load. A lot of research has been made and is in progress to mitigate fairness issues arising with CSMA/CA. In this paper, we propose a Contention Based Fair Scheduling for WLAN. This Protocol is based on CSMA/CA and guarantees low delays, reduced idle time, low collision rate and efficient collision resolution. Performance Analysis results shows that the proposed Protocol gives the better results than CSMA/CA.

Keywords : CSMA/CA, WLAN, DCF, DIFS, SIFS, Backoff

1. Introduction

The acronym **CSMA/CD** signifies **carrier-sense multiple access with collision detection** and describes how the Ethernet protocol regulates communication among nodes.

In CSMA/CD the sender senses the medium (eg wire) if the medium is free it starts sending the data if in between any other sender sends data at same time collision occurs and this collision can be detected by all the devices in that network hence the sending device stops sending the data and waits until the medium becomes free. This works well for wired network but in case of wireless network this fails since the collision occurs at receiver side and this collision can't be detected by sender and sender feels that the sent packets are received by sender without collision and continues sending. In this case we use CSMA/CA.

Wireless LAN can not implement CSMA/CD for three reasons [20]:

1. Collision detection implies that the station must be able to send data and receive collision signal at the same time. This implies costly stations and increased bandwidth requirements.
2. Collision may not be detected because of the hidden terminal problem [19, 20].
3. The distance between stations in wireless LANs can be great. Signal folding could prevent a station at one end from hearing a collision at the other end.

Carrier sense multiple access with collision avoidance (CSMA/CA) has been adopted by the IEEE 802.11 standards for wireless local area networks (WLANs). Using a distributed coordination function (DCF), the CSMA/CA protocol reduces collisions and improves the overall throughput [7, 9, 10].

2. CSMA/CA

When PC-D tries to initiate/send a data to PC-A via hub (a half-duplex device) through it's connected channel to hub, if some other data traffic of some other PC in the network is already coming down in the same channel through the hub down to PC-A, then the data hit each other and form a collision. Since, CSMA/CD is a Media Access Control mechanism, whenever any PC wants to send a data, it first sends out a dummy electrical signal into the channel to check whether any incoming data traffic is coming down or the line is free. If the traffic is already coming down, then the dummy signal collides with the incoming traffic and sends back a jam signal to the PC so that, the PC is notified to hold its traffic for a random amount of time before attempting to send the data again. This process will continue until the line/channel becomes free. The moment line is free, the PC immediately initiates/send it's data to the intended destination. This is how the data collision is avoided in half-duplex channels.

The CSMA/CA works as follows:

- Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency

1. The channel uses a persistence strategy with backoff until the channel is idle.
2. After the station is found idle, the station waits for a period of time, called Distributed InterFrame Space (DIFS) [1,2,4,5]; then the station sends a control frame called the request to send RTS.

- After receiving the RTS and waiting a short period of time, called Short InterFrame Space (SIFS), the destination station sends a control frame, called the clear to send CTS, to the source station. This control frame indicated that the destination station is ready to receive data [9].
- The source station sends data after waiting an amount of time equal to SIFS.
- The destination station, after waiting for an amount of time equal to SIFS, sends an acknowledgement to show that the frame has been received [10, 12, 20]. Acknowledgement is needed in this protocol because the station does not have any means to check for the successful arrival of its data at destination. On the other hand, the lack of collision in CSMA/CD is kind of indication to the source that data has arrived.
- When a station sends an RTS frame, it includes the duration of the time that it needs to occupy the channel. The stations that are affected by this transmission create a timer called a Network Allocation Vector (NAV) that shows how much time must pass before these stations are allowed to check the channel for idleness [14,19].

- Each time a station accesses the system and sends an RTS frame; other stations start their NAV [15]. In other words, each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired.

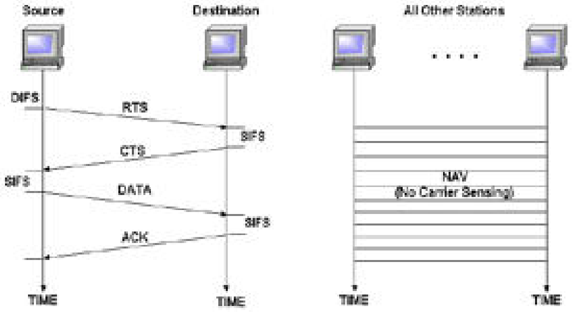

Fig 1: Successful Transmission in CSMA/CA

- Two or more stations may try to send RTS frames at the same time. These control frames may collide. However, because there is no mechanism for collision detection, the sender assumes there has been a collision if it has not received a CTS frame from the receiver. The backoff strategy is employed, and the sender tries again [7, 18, 20].

For calculating backoff interval, binary exponential backoff scheme is used.

3. Preliminaries

In this paper we try to give the fair scheduling Protocol than the IEEE 802.11 DCF by keeping the number of collisions to a minimum, even under the highest possible load and reducing Collision Avoidance delays.

Proposed Approach

We try to get fair scheduling by adopting the following concepts:

1. Using virtual clock concept is used for servicing the packets and virtual clock is updated using appropriate strategy.
2. Initially, the contention window size i.e. CW_{min} is chosen smaller than IEEE 802.11 MAC. Also a CW_{max} is chosen larger than IEEE 802.11 MAC.
3. Initially, to choose the backoff time logarithmic function is used, which will guarantee to have fair unique backoff time for each station.
4. If specific number of idle slots are detected, the backoff time will be decremented exponentially fast which reduces the further idle slots and assures fair channel utilization.
5. When collision occurs, the contention window is increased suitably and the node chooses a backoff interval which is a function of Fibonacci series. It helps to reduce the further probable collisions.

In our Protocol we borrow the concept of virtual clock from Self Clocked Fair Queuing (SCFQ) and appropriately modify it for better results.

4. Proposed Contention Based Fair Scheduling Protocol

Each node i maintains a local virtual clock $v_i(t)$, where $v_i(0) = 0$.

i) Every packet arriving to the queue of a station is tagged with a service tag before it is placed in the queue.

ii) When P_i^k, the k[th] packet of a station i arrives at the queue of the station, the service tag F_i^k is assigned to it as follows:

$$F_i^k = \max\{(v(A_i^k), F_i^{k-1}\} + \frac{L_i^k}{\phi i} \quad (1)$$

Where $v(A_i^k)$ is the virtual time at the time instance of A_i^k, A_i^k is the real time at which packet P_i^k arrives, L_i^k is the size of the packet P_i^k, and ϕi is the weight of flow i.

iii) The virtual time $v(t)$ is updated only for successful packet transmission and is set equal to the service tag of the packet just transmitted successfully.
The virtual time $v(t)$ represents the normalized fair amount of packet transmissions that each station should have performed. When all stations do not have any packet to transmit, the virtual time is reset to zero.

iv) Backoff interval will be chosen such that a packet with smaller service tag will be assigned a smaller backoff interval. This step is performed at time f_i^k, where f_i^k is the real time when packet P_i^k reaches its front of flow.
Node i will choose the backoff interval as follows:

$$Bi = \log(R(1, CW)) \times R(1, CW) \quad (2)$$

Where CW denotes the Contention Window size and $R(1, CW)$ is a random number between 1 and CW.

v) When $[(CW\min + 1) \times 2] - 1$ idle slots are detected, the backoff time will be decremented exponentially fast. i.e.

$$(Bi)new = (Bi)old / 2 \quad (3)$$

If $(Bi)new < 1$, then $(Bi)new = 0$.

vi) Initially a variable *CollisionCounter* is set to zero. If collision occurs, (Let node $i's$ transmission collided with some another node's transmission due to the countdown of their backoff intervals simultaneously), then, node i will choose the new backoff interval by performing following steps:

- Increment the value of *CollisionCounter* by 1
- Choose the new contention window size as follows

$$CW = \min(CW\max, CW \times 2) \quad (4)$$

- Choose the new backoff interval uniformly distributed between

$$[1, fib(CollisionCounter) \times CW] \quad (5)$$

Where $fib(CollisionCounter)$ represents a *CollisionCounter*[th] number in fibonacci series.

vii) For stations which are in deferring state(waiting for medium access), the station will increase its contention window size and pick a new random backoff time as follows:

- Choose the new contention window size as follows

$$CW = \min(CW\max, CW \times 2) \quad (6)$$

- If the value of *CollisionCounter* is zero, choose the new backoff interval in uniformly distributed between $[1, CW]$, Otherwise,

$$[1, fib(CollisionCounter) \times CW] \quad (7)$$

Where $fib(CollisionCounter)$ represents a *CollisionCounter*[th] number in fibonacci series.

viii) For every successful transmission of a packet, its contention window size will be reduced to CW min. And the backoff time will be calculated as

$$Bi = \log(R(1, CW)) \times R(1, CW) \quad (8)$$

5. Performance Evaluation

The Contention Based Fair Scheduling (CBFS) Protocol proposed in this paper works better than IEEE 802.11 MAC Protocol.

- In CBFS, the station which successfully transmitted has the minimum contention window and smaller backoff time; hence it has the higher probability to gain access of the medium. At the same time other stations have relatively larger contention window size and larger backoff timers. Also, after a number of successful transmissions for one station, another station may win a contention and this new station will then have higher probability for gaining medium access for a period of time.

- To reduce the idle slots, CBFS reduces the backoff times of stations exponentially fast when the medium is sensed to be idle for particular period of time.
  For example, if a station has a backoff timer 2047, which will be decreased by a slot time at each idle slot until it reaches to 2040 (because here we assume that CWmin = 3, hence $[(CW\min + 1)\times 2] - 1 = 7)$. After that, the idle slots continue; then the backoff timer will be decreased by half. i.e. $(B)new = (B)old/2$, at each additional idle slot until it either reaches to zero or a non-idle slot is detected. Thus, after 7 idle slots we will have $B = 1020$, after 8th idle slot we will have $B = 510$, after 9th idle slot we will have $B = 255$ and after 17th idle slot we will have $B = 0$. Therefore, the wasted idle backoff time is guaranteed to be less than or equal to 18 slots for above scenario.
- If collision occurs, in CBFS, the colliding station increases its contention window and recalculates backoff time as a function of fibonacci series, which fairly reduces the probability of future collisions.

For example, TABLE I shows the IEEE 802.11 MAC operations with the contention window size $CW = 2^{(n+3)} - 1$ with *CW*min = 7 and *CW*max = 1023. In this example there are 10 active stations contending for the use of medium based on IEEE 802.11 MAC. When the contention begins (i.e. the medium is determined to be idle for DIFS period by carrier sensing mechanism), each station performs the backoff procedure with its random backoff time determined from the initial contention window range [0, 7]. When the station detects the current slot idle, it will decrement its backoff time by a slot time (i.e. by one unit).

After one idle slot, the backoff timers of stations 7 and 8 reach zero, thus in the following slot, both station 7 and station 8 will transmit their packets at the same time and a collision will occur. The backoff procedures of all deferring stations are suspended and will resume after the medium is determined to be idle for DIFS period (i.e., next contention period). After stations 7 and 8 notice that their packet transmissions fail, their contention window sizes will be increased to 15 and their backoff timers will be chosen in the range of [0, 15] randomly. When a new DIFS period is detected, stations 1 and 6 transmit packets after one idle slot and a collision occurs. Stations 2 and 4 transmit packets and a collision occurs in the following contention period. After that, when the next DIFS period is detected, station 0 has a successful packet transmission. In the whole contention cycle (the time period starting with the end of a successful packet transmission and ending with the start of the next successful packet transmission), there have been three consecutive collisions before one successful packet transmission. We observe in Table I that most contention window sizes chosen for the backoffs are not big enough to avoid future packet collisions. Since the IEEE 802.11 MAC cannot provide the proper contention window size as the number of active stations increases, collisions are not resolved quickly, which leads to poor throughput performance.

0	1	2	3	4	5	6	7	8	9	Station Number
4(7)	2(7)	3(7)	6(7)	3(7)	6(7)	2(7)	1(7)	1(7)	7(7)	Contention Begins
3(7)	1(7)	2(7)	5(7)	2(7)	5(7)	1(7)	0(7) 8(15)	0(7) 14(15)	6(7)	Collision on station 7 & 8
2(7)	0(7) 9(15)	1(7)	4(7)	1(7)	4(7)	0(7) 4(15)	7(15)	13(15)	5(7)	Collision on station 1 & 6
1(7)	8(15)	0(7) 5(15)	3(7)	0(7) 10(15)	3(7)	3(15)	6(15)	12(15)	4(7)	Collision on station 2 & 4
0(7) 3(7)	7(15)	4(15)	2(7)	9(15)	2(7)	2(15)	5(15)	11(15)	3(7)	Successful Packet Transmission on station 0

Each item indicates "Backoff Time(contention window size)"

TABLE I: IEEE 802.11 MAC operations results

Figure 2 shows an example for the CBFS Protocol with the contention window size $CW = 2^{(n+2)} - 1$ with *CW*min = 3 and *CW*max = 2047.

Initially the value of CWmin is 3 and so all the contending stations will choose the backoff using (2). In the first contention period stations 2 & 7 will collide and will select their CW as 6 by using (6) and will choose the backoff using (7). The deferring stations will select their CW using (4) and will select the backoff using (5). In second contention stations 0 and 8 will collide and will accordingly choose the new CW and backoff. The deferring stations also choose their new CW and backoff accordingly. In the next contention, node one wins and get the medium access.

0	1	2	3	4	5	6	7	8	9	Station Numbers
2(3) 1(6)	2(3) 3(6)	0(3) (3(6))	2(3) 2(6)	1(3) 4(6)	3(3) 6(6)	2(3) 6(6)	0(3) 5(6)	1(3) 1(6)	3(3) 4(6)	Initially CW=3 [illegible] Collision on Stations 2 & 7 CS: CW=6, DS: CW=6
0(6) 3(12)	1(12)	4(12)	5(12)	6(12)	10(12)	11(12)	12(12)	0(6) 9(12)	2(12)	Collision on Stations 0 & 8 CS: CW=12, DS: CW=12
4(24)	0(12) 1(3)	7(24)	10(24)	15(24)	11(24)	8(24)	8(24)	5(24)	3(24)	Station 1 transmits successfully ST: CW=3, DS: CW=24
7(48)	0(3) 3(3)	30(48)	40(48)	8(48)	12(48)	22(48)	5(48)	1(48)	42(48)	Station 1 transmits successfully ST: CW=3, DS: CW=48
80(96)	91(96)	75(96)	10(96)	15(96)	25(96)	52(96)	7(96)	0(48) 1(3)	3(96)	Station 8 transmits successfully ST: CW=3, DS: CW=96

Each item indicates "Backoff Time(contention window size)"
CW: Contention Window, CS: Colliding Stations, DS: Deferring Stations, ST: station which Successfully Transmitted

TABLE II: CBFS results are better

We can see from figure 4 that the contention window sizes and backoff for the colliding stations and stations in deferring stations are increasing rapidly which resolves the collisions efficiently than IEEE 802.11 MAC. Also if specific numbers of idle slots are observed, the contention window size and accordingly backoff get decreased rapidly and thus assures high throughput than IEEE 802.11 MAC by reducing number of idle slots efficiently.

Figure 3 shows the result analysis comparing IEEE 802.11 MAC protocol results and CBFS Protocol results indicating the fairness and efficiency of CBFS Protocol over IEEE 802.11 MAC Protocol.

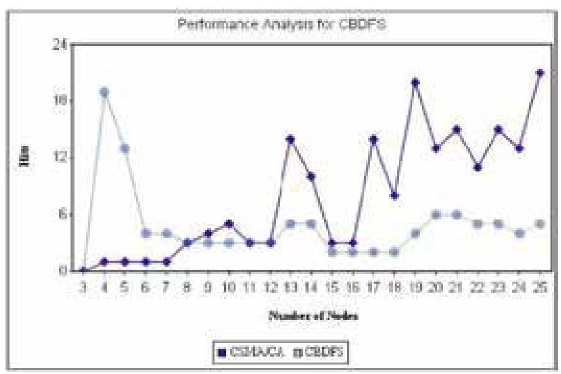

Fig 2: Result Analysis

6. Conclusion

The Contention Based Fair Scheduling Protocol (CBFS) presented in this paper assures maximum throughput than IEEE 802.11 MAC protocol. In CBFS, at every contention period, the backoff time is calculated for each station, which efficiently reduces the future collisions in the network. It handles and resolves the collisions efficiently by rapidly increasing the contention window size and backoff time if collision happens. Also it reduces the idle time of the medium very efficiently than IEEE 802.11 by decreasing contention window size and backoff time exponentially fast if specific numbers of idle slots are observed.

7. Future Scope

The numbers of hits required for first successful transmission using CBFS Protocol are more than the CSMA/CA and can be reduced with some adjustment factor.

REFERENCES

[1] N.H. Vaidya, A. Dugar, Seema Gupta, Paramvir Bahl, “Distributed Fair Scheduling in Wireless LAN”, IEEE Transactions on Mobile Computing, November/December 2005 (Vol. 4, No. 6) pp. 616-629. | [2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A Media Access Protocol for Wireless LANs,” Proc. ACM SIGCOMM Conf., pp. 212-225, Aug. 1994. | [3] P. Goyal, H.M. Vin, and H. Cheng, “Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks,” IEEE/ACM Trans. Networking, vol. 5, pp. 690-704, Oct. 1997. | [4] X. Wu, C. Yuen, Y. Gao, H. Wu, B. Li, “Fair Scheduling with Bottleneck Consideration in Wireless Ad-Hoc Networks,” Proc. IEEE Int'l Conf. Computer Comm. and Networks (ICCCN), pp. 568-572, Oct. 2001. | [5] Younggoo Kwon, Yuguang Fang and Haniph Latchman, “A novel MAC protocol with fast collision resolution for wireless LANs”, INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, April 2003 pp. 853- 862 vol.2. | [6] Anderson, J.H. Block, A. Srinivasan, A., “Quick-release fair scheduling”, Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE, Dec 2003, pp 130- 141. | [7] Chuan Heng Foh and Moshe Zukerman, “Performance Analysis of the IEEE 802.11 MAC Protocol”, EW2002 Conference Proceedings, March 2002. pp 217-234 vol 3. | [8] Garces, R. Garcia-Luna-Aceves, J.J., “A near-optimum channel access protocol based on incrementalcollision resolution and distributed transmission queues”, INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, April 1998, pp 158-165 vol.1. | [9] Huei-Jiun Ju Rubin, I. Yen-Chang Kuan, “An adaptive RTS/CTS control mechanism for IEEE 802.11 MAC protocol”, Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual, April 2003, pp 1469- 1473 vol.2. | [10] Pablo Brenner, “A Technical Tutorial on IEEE 802.11 Protocol”, BreezeCOM 1997. | [11] Yaser Fallah, Hussein Alnuweiri, “A controlled-access scheduling mechanism for QoS provisioning in IEEE 802.11e wireless LANs”, Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, 2005, pp 122 – 129. | [12] Lin, P. Benssou, B. Ding, Q.L. Chua, K.C., “CS-WFQ: a wireless fair scheduling algorithm for error-prone wireless channels”, Ninth International Conference on Computer Communications and Networks, 2000. Proceedings., 2000, pp 276-281. | [13] Songwu Lu, Vaduvur Bharghavan, R. Srikant, “Fair Scheduling in Wireless Packet Networks”, IEEE/ACM Transactions on Networking, Vol. 7, No. 4, August 1999. | [14] “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, ANSI/IEEE Std 802.11, 1999 Edition. | [15] Songwu Lu, Thyagarajan Nandagopal, Vaduvur Bhargavan, “A wireless fair service algorithm for packet cellular networks”, Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, 1998, pp 10 – 20. | [16] Chunyu Hu, Hwangnam Kim, and Jennifer C. Hou, “An Analysis of the Binary Exponential Backoff Algorithm in Distributed MAC Protocols”, Report No. UIUCDCS-R-2005-2599 UILU-ENG-2005-1794, July 2005. | [17] S. J. Golestani, “A self-clocked fair queueing scheme for broadband applications”, in IEEE INFOCOM, 1994. pp 636-646 vol 2. | [18] V. Bharghvan, “Performance evaluation of algorithms for wireless medium access”, IEEE International Computer Performance and Dependability Symposium IPDS'98, pp.142-149, 1998. | [19] Jochen Schiller, “Mobile Communications”, 2nd Edition, Addison Wesley. | [20] Forouzan, “Data Communication and Networking”, Third Edition, Tata McGraw Hill Publications