International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

ACHEVING QoS IN WIRELESS NETWORKS USING Q-CSMA/CA ALGORITHM

M. Megala[1], M.usha[2]

[1]*Department of Computer Science and Engineering, PG Student, Velammal Engineering College, Chennai.*
[2]*Department of Computer Science and Engineering, Asst.Prof- I, Velammal Engineering College,Chennai.*

megalaram24111@gmail.com, ushamahalingam@gmail.com

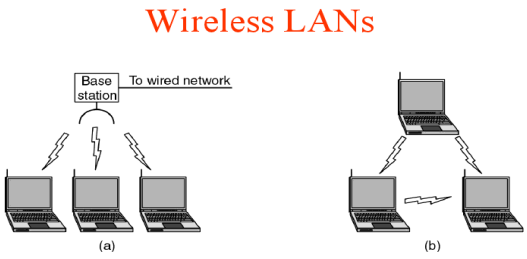
Abstract

A primary challenge in wireless networks is to use available resources efficiently so that the Quality of Service (QoS) is satisfied while maximizing the throughput and reducing the Delay of the network. In a wireless network, a sophisticated algorithm is required to schedule simultaneous wireless transmissions while satisfying interference constraint that *two neighbouring nodes cannot transmit simultaneously*. The scheduling algorithm need to be excellent in performance while being simple and distributed so as to be implementable. Many scheduling algorithms are used to achieve Qos of Wireless networks. Recently Carrier-sense multiple access(CSMA) algorithms used to achieve the maximum possible throughput in ad hoc wireless networks. However these algorithms assume an continuous-time CSMA protocol and Discrete-time CSMA protocol. In continuous-time CSMA protocol where collisions can never occur but simulation results indicate that the delay performance of these algorithms are quite bad. In this paper, we propose a Discrete-time version of the CSMA algorithm. A discrete-time distributed randomized algorithm is based on the glauber dynamics from statistical physics, in a single timeslot where multiple links are allowed to update their states. This algorithm also generates collision free transmission schedules while explicitly taking collision into account during the control phase of the protocol thus relaxing the perfect CSMA assumption. Greedy Maximal Scheduling(GMS) algorithm is a algorithm. It is not throughput optimal in general. We motivated to design a distributed scheduling algorithm called D-GMS for achieving low delay. We evaluate the performance of different scheduling algorithms via simulations. Q-CSMA algorithm can also eliminate the hidden and exposed terminal problem instead of exchanging the RTS(Request-To-Send)/CTS(Clear-To-Send) message RTD(Request-To-Decide)/CTD(Clear-To-Decide) message are exchanged in a synchronized manner.

Keywords: **CSMA/CA algorithm, Throughput, Delay.**

I. INTRODUCTION

Wireless network is becoming more and more popular in nowadays. Comparing to the traditional wired network, wireless network set up the connections through wireless channel. Generally there are two kinds of wireless networks shown in [FIG:1]. One has a wired backbone network in which the base stations are the boundary nodes, and the extended connections between mobile users and the base station are wireless channels. This one-hop wireless network is very popular currently, i.e., the cellular networks and WLANs. The other is wireless adhoc network, which has more than one hop wireless channels in the connection. This kind of topology is not widely implemented yet, but it is useful sometimes, especially in military applications and sensor networks.



a) Wireless networking with a base station. (b) Ad hoc networking.

FIG:1 Wireless LAN

Primary challenge in wireless networks is to use available resources efficiently so that the Quality of Service (QoS) is satisfied while maximizing the throughput of the net-work.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Among different resource allocation strategies, power and spectrum allocations have long been regarded as efficient tools to mitigate interference and improve the throughput of the network. Also, achieving a low transmission delay is an important QoS requirement in buffer-limited networks, particularly for users with real-time ser-vices. For these networks, too much delay results in dropping some packets. Therefore, the main challenge in networks with real-time services is to utilize an efficient power allocation scheme so that the delay is minimized while achieving a high throughput.

A scheduling algorithm, or policy, is employed, which at each time slot chooses the queues that are scheduled for transmission. The nature of the policy typically has a significant effect on the resulting network performance, measured in terms of throughput and delay. In addition, a policy carries out certain computations at each time step, and practical considerations call for low computational requirements.

Many scheduling algorithms are used to achieve the Maximum Throughput and Low Delay. Some of the algorithms are:

1. Maximum Weight algorithm
2. Greedy Maximal algorithm(Longest Queue First)
2. CSMA algorithm
3. CSMA/CA algorithm

1.1 MAMIMAL WEIGHT SCHEDULING(MWS):

Maximal-weight scheduling (MWS) is *throughput-optimal*. That is, that scheduling can support any incoming rates within the capacity region. In MWS, time is assumed to be slotted. In each slot, a set of non-conflicting links (called an "independent set," or "IS") that have the maximal weight are scheduled, where the "weight" of a set of links is the summation of their queue lengths. (This algorithm has also been applied to achieve 100% throughput in input-queued switches.) However, finding such a maximal-weighted is NP-complete in general and is hard even for centralized algorithms. Therefore, its distributed implementation is not trivial in wireless networks.

1.2 LONGEST-QUEUE-FIRST(LQF) ALGORITHM:

In LQF scheduling, the schedule is computed by queue length based priorities, i.e., links are added according to their queue lengths, from the longest to the shortest, and a link with non-empty queue is added to the schedule whenever there is no conflict.

The Longest Queue First (LQF) algorithm takes into account the queue lengths of the nonempty queues.

It shows good throughput performance in simulations. In fact, LQF is proven to be throughput-optimal if the network topology satisfies a "local pooling" condition or if the network is small. In general topologies, however, LQF is not throughput-optimal, and the achievable fraction of the capacity region networks. Proutiere et al. developed asynchronous random-access-based scheduling algorithms that can achieve throughput performance similar to that of the Maximum Size scheduling algorithm. In general topologies, however, LQF is not throughput-optimal, it is suitable to small networks .It will not give optimal throughput in large networks.

1.3 CSMA ALGORITHM:

Distributed adaptive carrier sense multiple access (CSMA) algorithm for a general interference model. It is inspired by CSMA, but may be applied to more general resource sharing problems (i.e., not limited to wireless networks).If the packet collisions are ignored the algorithm can achieve maximal throughput. When more than one station attempts to transmit a frame at the same time, a *collision* occurs, and subsequently all frames get corrupted. The standard mechanism for contention resolution in computer networks is called *carrier-sense multiple access (CSMA)*. CSMA algorithms attempt to break symmetries of failing transmissions being restarted at almost the same time by using randomized binary exponential back off procedures. While wired devices can listen during their own transmissions and employ CSMA with collision detection (CSMA/CD), stations in wireless networks usually cannot listen to their own transmissions, and consequently colliding transmissions can only be detected after they have been completed. Thus wireless devices use CSMA with collision avoidance (CSMA/CA or CSMA-CA). The CSMA-CA algorithm works as follows: When a station or device or node wants to send some information, the algorithm works as follows.

1.3.1 MAIN PROCEDURE OF CSMA ALGORITHM:

I. Is my frame ready for transmission? If yes, it goes on to the next point.

II. Is medium idle? If not, wait until it becomes ready

III. Start transmitting.

IV. Did a collision occur? If so, go to collision detected procedure.

V. Reset retransmission counters and end frame

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

1.3.2 CSMA/CA ALGORITHM:

Carrier sense multiple access with collision avoidance is a network multiple access method in which carrier sensing is used, but nodes attempt to avoid collision by transmitting only when the cannel is sensed to be "Idle". Flow chart for CSMA/CA in

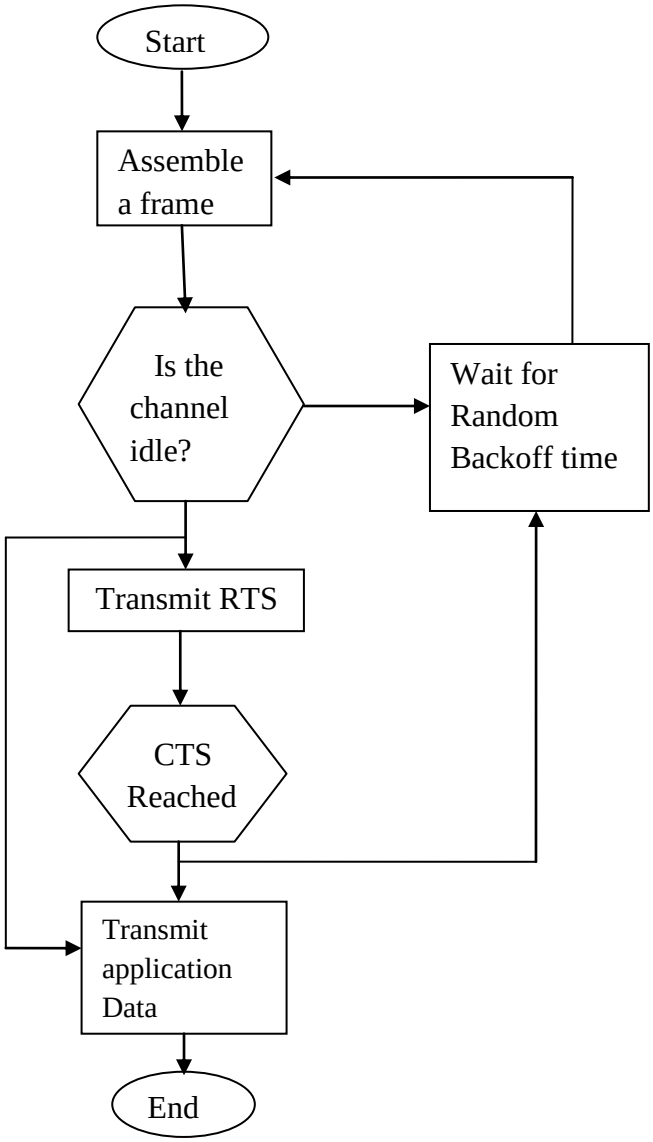


Fig:2 CSMA/CA Algorithm

1.3.3 CONTINUOUS-TIME CSMA/CA ALGORITHM:

Station continuously sensing a channel and transmits packet, when the channel is "Idle". Diagram representation for continuous shown in FIG:3

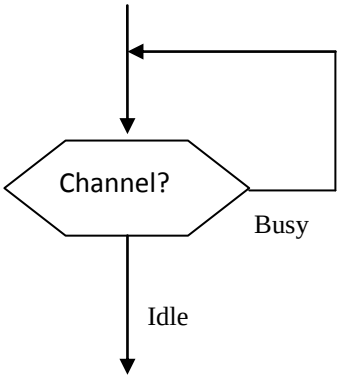


FIG:3Continuous-time CSMA/CA algorithm

1.3.4 DISCRETE-TIME CSMA/CA ALGORITHM:

Station sensing channel a randomly and transmits packet when the channel is "Idle". Represented in FIG:4.

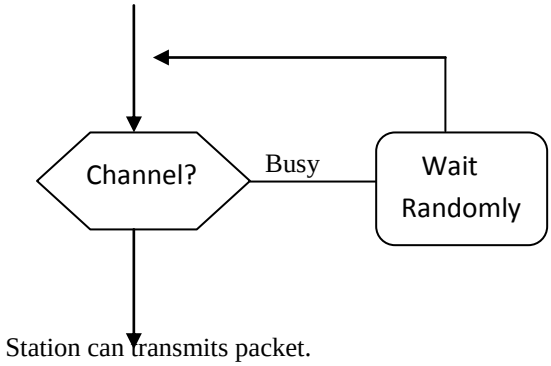


FIG:4 Discrete-time CSMA/CA algorithm

II. PREVIOUS WORK

A particular policy that applies to a broad class of constrained network scheduling problems, and which has received much attention, is the maximum-weight policy introduced by Tassiulas and Ephremides [10]. This policy assigns a weight to each candidate schedule, equal to the sum of the sizes of the queues that are selected for service by that schedule, and chooses a schedule with the largest weight. This policy is known to achieve maximal throughput [10]. Furthermore, it results in low delay under "friendly" arrival traffic.

However, this policy needs to find a maximum weight schedule at each time step, which can be computationally burdensome when the constrains involved are of a combinatorial nature.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

For example, in a wireless network with interference constraints specified in terms of independent set constraints, the problem of finding a maximum weight schedule amounts to solving a maximum weight independent set problem, which is NP-hard.

The above discussion suggests the need for alternative, low-complexity and high-throughput policies. Towards this purpose, Tassiulas [9] proposed a simple throughput-optimal randomized policy for switch scheduling that requires only a polynomial amount of computation at each time slot. This policy is easily extended to networks with general combinatorial scheduling constraints including the independent set constraint. It can even be implemented in a distributed manner, using a gossip mechanism [7]. However, the best known bound on the resulting average delay grows exponentially with the number of nodes.

However, it is not known whether a low complexity (polynomial per time step), low-delay, and throughput-optimal policy is possible for networks with general combinatorial scheduling constraints.

A recent interest in wireless network has led to proposal of distributed scheduling algorithms under matching constraints [10]–[13]. Most of these algorithms, based on finding maximal matching, guarantee only a constant fraction of throughput. Recently, Modiano, Shah and Zussman [14] exhibited a throughput optimal distributed scheduling algorithm with matching constraints. This algorithm, as discussed in [1], easily extends to provide throughput optimal algorithm for resource allocation and scheduling problem under matching constraints.

Apart from matching constraints, other scheduling constraints have received limited attention primarily due to inherent hardness of the other constraints. For example, Sharma, Mazumdar and Shroff [15] identify that max-weight scheduling with K−hop matching constraint becomes an instance of computationally hard combinatorial optimization problem.

They provide a centralized throughput optimal algorithm for unit disk graphs based on work by Hunt et. al. [16]. However, hardness of the max. wt. problem does not imply non-existence of throughput optimal algorithm.

The Greedy Maximal Scheduling (GMS) algorithm, also known as the Longest-Queue-First (LQF) algorithm [11], has low complexity and hence can be deployed in practical systems.

Its performance has been observed to be close to optimal for a variety of network scenarios in simulations and experiments (e.g., [10]).

Hence it is intriguing and important to analyze/understand the performance of GMS for networks with common topologies, e.g., providing sufficient conditions for GMS to be throughput optimal, and calculating/bounding the efficiency ratio of GMS when it is no throughput optimal.

To identify sufficient conditions for GMS to be throughput optimal, the concept of *local pooling* (will be defined formally later) was introduced in [3]. The authors showed that if the network satisfies the local pooling condition then GMS is throughput optimal. In particular, if the interference graph of the network is a tree, then the local pooling condition is satisfied and GMS is throughput optimal. In addition to tree (interference) graphs, [1, 15] identified several classes of graphs (e.g., trees of cliques, perfect graphs, chordal graphs) which also satisfy the local pooling condition. Independently [9] and [15] showed that the local pooling condition is satisfied for tree networks (note that the interference graph of a tree network may not be a tree) under the *k*-hop interference model.

In [1, 15], the local pooling condition was tested for small interference graphs via exhaustive numerical search. They found that local pooling is satisfied (GMS is throughput optimal) for networks with up to 5 links under the 1-hop interference model and for networks with up to 7 links under the *k*-hop ($k \geq 2$) interference model. In [8, 9], the notion of local pooling was generalized to ¾-local pooling and the concept of *local-pooling factor* was introduced. They showed that the efficiency ratio of GMS is equal to the local-pooling factor of the network and proposed a recursive procedure to estimate/bound the local-pooling factor of a network. Our results primarily build upon the results in [3] and [8]. Compared with GMS, the Local GMS (LGMS) algorithm [12] has an even lower complexity (its computational complexity is linear in the number of links in the network), and is amenable to distributed implementation [6].

Another class of scheduling algorithms are Aloha or *carrier sense multiple access* (CSMA) type random access algorithms. Under CSMA, a node (transmitter of a link) will sense whether the channel is busy before it transmits a packet. When the node detects the channel is busy, it will wait for a random back off time. Since CSMA-type algorithms can be easily implemented in a distributed manner, they are widely used in practice (e.g., the IEEE 802.11 MAC protocol). In [4], the authors derived an analytical model to calculate the throughput of a CSMA type algorithm in multi hop wireless networks.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

They showed that the Markov chain describing the evolution of schedules has a product-form stationary distribution under an idealized continuous-time CSMA protocol (which assumes zero propagation/sensing delay and no hidden terminals) where collisions can never occur. Then, the authors proposed a heuristic algorithm to select the CSMA parameters so that the link service rates are equal to the link arrival rates (which were assumed to be known). No proof was given for the convergence of this algorithm. This model was used in [29] to study throughput and fairness issues in wireless ad hoc networks. The insensitivity properties of such a CSMA algorithm have been recently studied in [18].

III. PROPOSED WORK

We design a discrete-time version of the CSMA-type random access algorithm that achieves the same product-form distribution over schedules as in [4]. Our algorithm generates collision-free data transmission schedules while allowing for collisions during the control phase of the protocol (as in the 802.11 MAC protocol), thus relaxing the perfect CSMA assumption of the algorithms in [4], [13], and [25]. Our approach to modelling collisions is different from the approaches in [14] and [20]. In [20], the authors pointed out that as the transmission probabilities are made small and the transmission lengths are made large, their discrete-time model approximates the continuous-time model with Poisson clocks.

In this paper, we focus on the MAC layer so we only consider single-hop traffic. The *capacity region* of the network is the set of all arrival rates for which there exists a scheduling algorithm that can stabilize the queues, i.e., the queues are bounded in some appropriate the purposes of this paper, we will assume that if the arrival process is stochastic, then the resulting queue length process admits a Markovian description, in which case stability refers to the positive recurrence of this Markov chain. It is known (e.g., [27]) that the capacity region is given by

$$\wedge=\{\lambda|\mu \in Co(M); \lambda<\mu\}, \qquad (1)$$

Where Co(M) is the *convex hull* of the set of feasible schedules in M. When dealing with vectors, inequalities are interpreted component wise. We say that a scheduling algorithm is *throughput-optimal*, or achieves the *maximum throughput*, if it can keep the network stable for all arrival rates in $\wedge$.

A conflict graph and corresponding CSMA Markov chain. (a) Conflict graph. (b) CSMA Markov chain as follows in FIG:5:

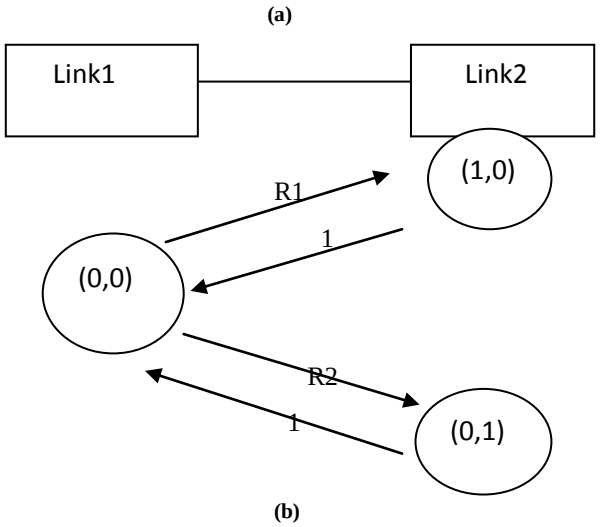


FIG:5(a) Conflict graph. (b) CSMA Markov chain

We divide each timeslot into a *control* slot and a *data* slot. (Later, we will further divide the control slot into control mini slots.) The purpose of the control slot is to generate a collision-free *transmission schedule* used for data transmission in the data slot. To achieve this, the network first selects a set of links that do not conflict with each other. Note that these links also form a feasible schedule, but it is not the schedule used for data transmission. We call the *decision schedule* in timeslot t .

We present a distributed implementation of the basic scheduling algorithm. The key idea is to develop a distributed randomized procedure to select a (feasible) decision schedule in the control slot. To achieve this, we further divide the control slot into control mini-slots. Note that once a link knows that it is included in the decision schedule, it can determine its state in the data slot based on its carrier-sensing information (i.e., whether its conflicting links were active in the previous data slot) and activation probability. We call this implementation *Q-CSMA* since the activation probability of a link is determined by its queue length, and collisions of data packets are avoided via carrier sensing and the exchange of control messages.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

IV. HIDDEN AND EXPOSED TERMINAL PROBLEM

4.1.HIDDEN TERMINAL PROBLEM:

In wireless networking, the **hidden node problem** or **hidden terminal problem** occurs when a node is visible from a wireless access point (AP), but not from other nodes communicating with said AP. This leads to difficulties in media access control.

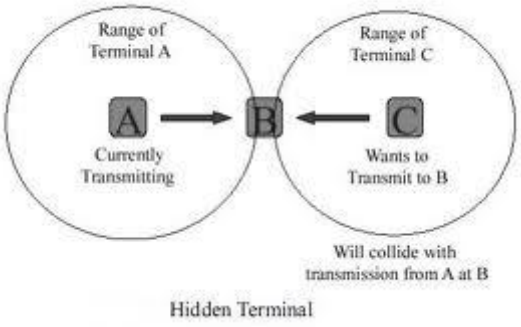


FIG:6 Hidden terminal problem

4.2.EXPOSED TERMINAL PROBLEM:

In wireless networks, the **exposed node problem** occurs when a node is prevented from sending packets to other nodes due to a neighbouring transmitter. Consider an example of 4 nodes labelled R1, S1, S2, and R2, where the two receivers are out of range of each other, yet the two transmitters in the middle are in range of each other. Here, if a transmission between S1 and R1 is taking place, node S2 is prevented from transmitting to R2 as it concludes after carrier sense that it will interfere with the transmission by its neighbour S1. However note that R2 could still receive the transmission of S2 without interference because it is out of range of S1.

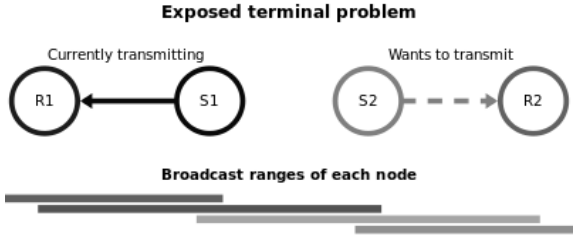


FIG:7 Exposed terminal problem

To overcome these two problems RTS/CTS signals are used. But there also chance for occurring collision. The following diagram represents the collision occurs in transmitting RTS/CTS messages[FIG:8].

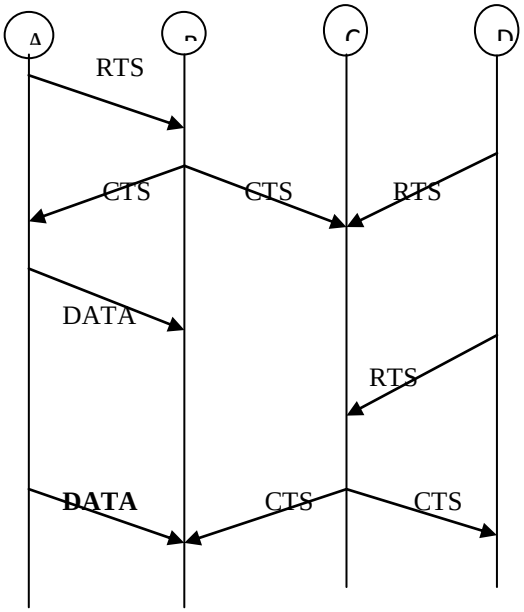


FIG:8Collision in RTS/CTS signals

As depicted in Figure 8, let us assume that there are four sensor nodes A, B, C, D as shown and links between nodes denote their communication ranges. Let us assume that A wants to communicate to B. So, it sends RTS to B and in turn, B broadcasts CTS. At the same time, let us suppose that D transmits RTS to C resulting in a RTSCTS collision at C.

Node D timeouts and sends its RTS after the timeout interval. Once C gets RTS from D, it broadcasts CTS. This results in a collision with data from A at B. This occurs due to lack of proper decoding of CTS at C. Thus, RTS/CTS cannot avoid collisions in such a scenario and leads to low system throughput and increased average packet latency.

V. BEST SOLUTION FOR HIDDEN AND TERMINAL PROBLEM

In IEEE 802.11 DCF, the RTS/CTS mechanism is used to reduce the *Hidden Terminal Problem*. However, even if RTS/CTS is used, the hidden terminal problem can still occur, as illustrated in FIG.,8, where we use Si and to Ri denote the sender and receiver (in the sense of data transmission) of link i . Under 802.11, it is possible that is S2 sending an RTS to R2 while R1 is returning a CTS to S1 at the same time.R1 cannot detect the RTS from S2 since it is transmitting. Likewise, cannot detect the CTS from R2 . Therefore, both links 1 and 2 will be scheduled, which will cause a collision at R1 .

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Our Q-CSMA algorithm can eliminate the hidden terminal problem due to the fact that the RTD/CTD messages are exchanged in a synchronized manner. Suppose Ti is the back off expiration time of node Si.

T1<T2: During the (T1+1)th mini-slot, sends an RTD in the first sub-mini-slot, then in the second sub-mini-slot, R1 returns a CTD, and link 1 will be included in the decision schedule. Since this CTD is not intended for S2, S2 disables its role as a sender of a link in the decision schedule, thus link 2 will not be included in the decision schedule.

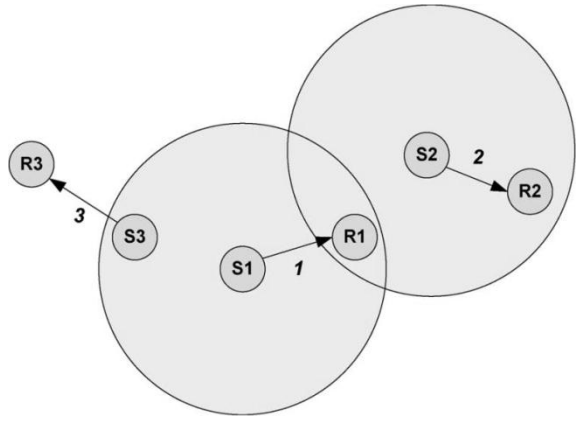


FIG:9 Hidden and exposed terminal problem

T1=T2: During the (T1+1)th mini-slot, both S1 and S2 send an RTD in the first sub-mini-slot. In this case, R1 senses a collision of RTD and will not return a CTD. Thus, link 1 will not be included in the decision schedule, while link 2 will be included in the decision schedule.

T1>T2: Similarly, in this case link 2 but not link 1 will be included in the decision schedule.

Therefore, under synchronized RTD/CTD, the decision schedule is collision-free, which implies that the transmission schedule is collision-free if we start with a collision-free transmission schedule Hence, the hidden terminal problem is eliminated.

Another problem, known as the *Exposed Terminal Problem*, may also occur under 802.11. In FIG:8, if S1 sends an RTS to R1,S3 will receive this RTS and will be silenced under 802.11, which is unnecessary because the potential transmission of link 3 will not interfere with link 1. On the other hand, under Q-CSMA, if S1 sends an RTD to R1,S3 , will ignore this RTD and can still send an RTD to R3 . Therefore, both links 1 and 3 can be included in the decision schedule and the transmission schedule, thus avoiding the exposed terminal problem.

Note that the presence of hidden and exposed terminals not only leads to loss of efficiency, but also poses mathematical difficulties.

In general, it is impossible to define a set of schedules that are consistent with both the definition of a feasible schedule. Our RTD/CTD mechanism eliminates such problems.

VI. CONCLUSION

In this paper, we have proposed a distributed CSMA scheduling algorithm, and showed that, under the model of perfect CSMA, it is throughput-optimal in wireless networks with a general interference model. We considered the design of efficient CSMA algorithms called Queue-length based discrete time CSMA/CA algorithm that are throughput optimal and have a good delay performance. The algorithm is essentially a Glauber Dynamics with, potentially, multiple-site updates at each time-slot. Access probabilities depend on links weights, where the weight of each link is chosen to be an appropriate function of its queue-length. The discrete-time formulation allows us to incorporate mechanisms to dramatically reduce the delay without affecting the theoretical throughput-optimality property. In particular, combining CSMA with distributed GMS leads to very good delay performance. RTD/CTD messages are used to eliminate the Hidden and Exposed terminal problems.

REFERENCES

[1] Journal paper**:** P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *Proc. 43rd Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2005, pp. 28–30. 836 IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 20, NO. 3, JUNE 2012

[2] Conference paper: A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Adv. Appl. Probab.*, vol. 38, no. 2, pp. 505–521, 2006.

[3] Conference paper: L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–149, 2006. [11] J.Ghaderi and R. Srikant, "On the design of efficient CSMA algorithms for wireless networks,"Mar. 2010 [Online]. Available: http://arxiv.org/ abs/1003.1364

[4] Conference paper: L. Jiang and J.Walrand, "Convergence analysis of a distributed CSMA algorithm for maximal throughput in a general class of networks," UC Berkeley, Berkeley, CA, Tech. Rep., Dec. 2008. [13] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proc.46th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2008, pp.1511–1519.

[5] Journal paper: L. Jiang and J. Walrand, "Approaching throughput-optimality in a distributed CSMA algorithm with contention resolution," UC Berkeley, Berkeley, CA, Tech. Rep., March 2009.[15] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1132–1145, Aug.2009.

[6] Journal paper: S. C. Liew, C. Kai, J. Leung, and B. Wong, "Back-of-the-envelope computation of throughput distributions in CSMA wireless networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 9, pp. 1319–1331, Sep. 2010.

[7] Journal paper: P. Marbach, A. Eryilmaz, and A. Ozdaglar, "Achievable rate region of CSMA schedulers in wireless networks with primary interference constraints," in *Proc. IEEE CDC*, Dec. 2007, pp. 1156–1161.

[8] Conference paper: E. Vigoda, "A note on the Glauber dynamics for sampling independent sets," *J. Combin.*, vol. 8, pp. 1–8, 2001.

[9] Journal paper: X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proc. IEEE INFOCOM*, Mar. 2005, vol. 1, pp. 23–34.

[10] Conference paper: X. Wu, R. Srikant, and J. R. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 595–605, Jun. 2007.

[11] Journal paper:G. Zussman, A. Brzezinski, and E. Modiano, "Multihop local pooling for distributed throughput maximization in wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1139–1147.