**Q-CSMA: Queue-Length Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks**

**1. Introduction**

For wireless networks with limited resources, efficient resource allocation and optimization play an important role in achieving high network performance and reducing energy consumption. The performance metrics of interest in this paper are throughput and delay. The throughput performance of a scheduling algorithm is often characterized by the largest set of arrival rates under which the algorithm can keep the queues in the network stable. The delay performance of a scheduling algorithm can be characterized by the average delay experienced by the packets transmitted in the network.

Although the recent results on CSMA-type random access algorithms show throughput-optimality, simulation results indicate that the delay performance of these algorithms can be quite bad and much worse than MWS and GMS. Thus, one of our goals in this paper is to design distributed scheduling algorithms that have low complexity, are provably throughput optimal and have good delay performance. Towards this end, we design a discrete-time version of the CSMA random access algorithm. The algorithm generates collision-free data transmission schedules while allowing for collisions during the control phase of the protocol.

**1.1 Network Security**

Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Security policy should keep the malicious users out and also exert control over potential risky users within your organization. In addition, the security policy should dictate a hierarchy of access permissions, grant users access only to what is necessary for the completion of their work.A network security system usually consists of many components. Ideally, all components work together, which minimizes maintenance and improves security.

Once authenticated, a [firewall](http://en.wikipedia.org/wiki/Firewall_(networking)) enforces access policies such as what services are allowed to be accessed by the network users. Though effective to prevent unauthorized access, this component may fail to check potentially harmful content such as worms or [Trojans](http://en.wikipedia.org/wiki/Trojan_horse_(computing)) being transmitted over the network. The content or behavior and other anomalies to protect resources, e.g. from [denial of service](http://en.wikipedia.org/wiki/Denial_of_service) attacks or an employee accessing files at strange times. Individual events occurring on the network may be logged for audit purposes and for later high-level analysis.

**1.2 Wireless Network**

Wireless network refers to any type of [computer network](http://en.wikipedia.org/wiki/Computer_network) that is not connected by cables of any kind. It is a method by which homes,  [telecommunications networks](http://en.wikipedia.org/wiki/Telecommunications_network) and enterprise (business) installations avoid the costly process of introducing cables into a building, or as a connection between various equipment locations. A wireless network links two or more devices using some wireless distribution method and usually providing a connection through an access point to the wider internet. This gives users the mobility to move around within a local coverage area and still be connected to the network.

Wireless Networks are generally composed of two basic elements are access point and other wireless enabled device. Wireless network connectivity without the physical restrictions associated with building wired networks. It generally developed as an extension to an existing wired infrastructure, a wireless network may be stand alone as well.

**1.3 CSMA**

Carrier Sense Multiple Access (CSMA) is a [probabilistic](http://en.wikipedia.org/wiki/Probabilistic) [Media Access Control](http://en.wikipedia.org/wiki/Media_Access_Control) (MAC) protocol in which a node verifies the absence of other [traffic](http://en.wikipedia.org/wiki/Network_traffic) before [transmitting](http://en.wikipedia.org/wiki/Transmission_(telecommunications)) on a shared [transmission medium](http://en.wikipedia.org/wiki/Transmission_medium). "Carrier Sense" describes the fact that a [transmitter](http://en.wikipedia.org/wiki/Transmitter) uses [feedback](http://en.wikipedia.org/wiki/Feedback) from a receiver that detects a [carrier wave](http://en.wikipedia.org/wiki/Carrier_wave) before trying to send. That is, it tries to detect the presence of an encoded [signal](http://en.wikipedia.org/wiki/Signal_(electrical_engineering)) from another station before attempting to transmit. If a carrier is sensed, the station waits for the transmission in progress to finish before initiating its own transmission.

**Carrier Sense Multiple Access Collision Avoidance** a [network](http://www.webopedia.com/TERM/N/network.html) [contention](http://www.webopedia.com/TERM/C/contention.html) [protocol](http://www.webopedia.com/TERM/P/protocol.html) that listens to a network in order to avoid collisions. CSMA/CA contributes to network traffic because, before any real data is transmitted, it has to broadcast a signal onto the network in order to listen for collision scenarios and to tell other devices not to broadcast.

The most important different between the wireless LAN and MAC protocol of model wired networking applications is the impossibility to detect collisions. With the receiving and sending antennas own. As a result, the complete packet will be sent before the incorrect checksum reveals that a collisions has happened. It is therefore of utmost importance that the number of collisions be limited to the absolute minimum.

**2. WORK DONE IN PHASE ONE**

An analytical model to evaluate the delay performance of the CSMA/CA protocol under unsaturated conditions. Our delay analysis is unique in that we consider the end-to-end packet delay, which is the duration from the epoch that a packet enters the queue at the MAC layer of the transmitter side to the epoch that the packet is successfully received at the receiver side. The analytical results give excellent agreement with the simulation results, which represents the accuracy of our analytical model. The results also provide important guideline on how to set the parameters of the burst assembly policy. Based on these results, we further develop an efficient adaptive policy so as to optimize the throughput and delay performance of the CSMA/CA protocol.

**3. SYSTEM ORGANIZATION**

No

Packet Generation

Sending Packets

User Polling

Networking Monitoring

Ack <=ts

Proceed Transmission

Time Mode transformation

Yes

No

Add packet to queue

Ack <=Backof time

Yes

Remove packet from queue

Resend the packet

**3.2 System Architecture:**

Queue

Queue

Channel

**4. CLASSIFICATION & RETRIEVAL**

**4.1 Existing System**

It is applicable for general topology itself. GMS may only achieve a fraction of the capacity region. Moreover, while the computational complexity of GMS is low, the signaling and time overhead of decentralization of GMS can increase with the size of the network Process done at a single time slot itself. A scheduling algorithm is a procedure to decide which schedule to be used in every time slot for data transmission. We focus on the MAC layer so we only consider single-hop traffic.

**Demerits**

* + Hidden nodes causing collisions not to be detected every time, resulting in data loss.
  + The exposed nodes causing a machine to think the network is busy, while it is not. This is solved by the use of CSMA/CA.

**4.2 Proposed System**

Q- CSMA since the activation probability of a link is determined by its queue length to achieve maximum throughput, and collisions of data packets are avoided via carrier sensing and the exchange of control messages. The distributed queue length based CSMA/CA protocol that leads to collision-free data transmission schedules. The protocol is provably throughput-optimal. The discrete time formulation allows us to incorporate mechanisms to dramatically reduce the delay without affecting the theoretical throughput optimality property.

**Merits**

* Scheduling for large links
* Collusion free data transmission
* High throughput
* Low delay

**4.3 Throughput Guarantees Through Maximal Scheduling in Wireless Networks**

The simple distributed scheduling strategy, maximal scheduling, and prove that it attains a guaranteed fraction of the maximum throughput region in arbitrary wireless networks. The guaranteed fraction depends on “interference degree” of the network which is the maximum number of sessions that interfere with any given session in the network and do not interfere with each other. Depending on the nature of communication, the transmission powers and the propagation models, the guaranteed fraction can be lower bounded by the maximum link degrees in the underlying topology, or even by constants that are independent of the topology. The guarantees also hold in networks with multicast communication and an arbitrary number of frequencies.

Maximizing the network throughput by appropriately scheduling sessions is a key design goal in wireless networks characterized the maximum attainable throughput region and also provided a scheduling strategy that attains this throughput region in any given wireless network. In the node-exclusive spectrum sharing model, the only scheduling constraint is that a node cannot communicate with multiple nodes simultaneously. This specific interference model holds only when every node has a unique frequency in its two-hop neighborhood.

The interference regions of nodes involved in transmissions may vary widely depending on the signal propagation conditions, and may be different for different transmitter-receiver pairs. A basic question that remains open is whether a distributed scheduling strategy can attain a guaranteed fraction of the maximum achievable throughput region for arbitrary interference models. The network still has a single frequency and all communications use the same power.

We cannot therefore draw conclusions about the performance of maximal scheduling for arbitrary interference constraints from the results in a few representative scenarios. Also, given that large numbers of interference models arise, case by case investigations may not be feasible. We therefore proceed to design a framework for characterizing the throughput region of maximal scheduling in arbitrary wireless networks. The regions of maximal scheduling are significantly different for the bidirectional and unidirectional power models.

The performance of a simple distributed scheduling policy, maximal scheduling, which had earlier been investigated in context of node-exclusive spectrum sharing model and input-queued switches. The characterizations demonstrate that the performance bounds depend heavily on the nature of communication and interference models. We prove that maximal scheduling is guaranteed to attain a constant fraction of the maximum throughput region for certain communication and interference models, while it is also guaranteed to not attain a constant fraction in the worst case for some other models.

**4.4 A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks**

In multi hop wireless networks, designing distributed scheduling algorithms to achieve the maximal through put is a challenging problem because of the complex interference constraints among different links. Traditional maximal weight scheduling (MWS), although through put optimal, is difficult to implement in distributed networks. CSMA scheduling algorithm that can achieve them axial through put distributive. Some of the major advantages of the algorithm are that it applies to a very general interference model and that it is simple, distributed and asynchronous. Furthermore, the algorithm is combined with end-to-end congestion control to achieve the optimal utility and fairness of competing flows. Simulations verify the effectiveness of the algorithm. Also, the adaptive CSMA scheduling is a modular MAC- layer algorithm that can be combined with various protocols in the transport layer and network layer.

The transport layer needs to inject the right amount of traffic into the network based on the congestion level and the MAC layer needs to serve the traffic efficiently to achieve high throughput. It is well known that maximal weight scheduling (MWS) is throughput optimal. That is, that scheduling can support any incoming rates within the capacity region. The extensions to more general interference models, as discussed and involves extra challenges. The algorithm may not be directly comparable to the through put optimal algorithms mentioned above since it utilizes the carrier sensing capability.

The few distinct features:

* Each node only uses its local information. No explicit control messages are required among the nodes.
* It is based on CSMA random access, which is similar to the protocol and is easy to implement.
* Time is not divided into synchronous slots. Thus no synchronization of transmissions is needed.

The combine proposed scheduling algorithm with end-to-end congestion control using a novel technique, to achieve fairness among competing flows as well as maximal through put. We show that the proposed CSMA scheduling is a modular MAC-layer algorithm and demonstrate its combination with optimal routing, any cast and multicast. In the idealized CSMA model we used, the sensing time is zero and there is no collision.

This allows us to focus on the scheduling problem without worrying about the contention resolution problem. The resulting performance can serve as a benchmark. However in practice, since the sensing time is not zero, the back off time is usually chosen to be a multiple of mini slots where each mini slot cannot be arbitrarily small. Therefore collisions occur given the discrete distribution of back off times.

The distributed CSMA scheduling algorithm, and showed that, under the idealized CSMA, it is through put optimal in wireless networks with a general interference model . We have utilized the product form stationary distribution of CSMA net works in order to obtain the distributed algorithm and the maximal through put. Furthermore, we have combined that algorithm with congestion control to approach the maximal utility, and showed the connection with back pressure scheduling.

The adaptive CSMA algorithm is a modular MAC- layer component that can work with other algorithms in the transport layer and network layer.

**4.5 Multi hop Local Pooling for Distributed Throughput Maximization in Wireless Networks**

Efficient operation of wireless networks requires distributed routing and scheduling algorithms that take into account interference constraints. Local Pooling focused mostly on obtaining abstract conditions and on networks with single-hop interference or single-hop traffic. The Local Pooling conditions, thereby enabling the use of such graphs in network design algorithms. Then, we study the multi hop implications of Local Pooling. Regarding multi hop traffic, we show that if the network satisfies only the single-hop Local Pooling conditions, distributed joint routing and scheduling algorithms are not guaranteed to achieve maximum throughput.

A major challenge in the design and operation of wireless networks is to jointly route packets and schedule transmissions to efficiently share the common spectrum among links in the same area. The policy of applies to a multi hop wireless network with a stochastic packet arrival process and is guaranteed to stabilize the network whenever the arrival rates are within the stability region. The results have been extended to various settings of wireless networks and input-queued switches. However, throughput optimal algorithms based on require the repeated solution of a global optimization problem, taking into account the queue backlog information for every link in the network.

These conditions are referred to as Local Pooling (LoP) and are related to the properties of all maximal independent sets in the conflict graph. The LoP conditions were recently generalized in to provide conditions under which a greedy maximal weight matching algorithm obtains some guaranteed fractional throughput. In order to allow the development of algorithms that take advantage of LoP, we focus on identifying topologies of interference and network graphs that satisfy the LoP conditions, and on studying the effect of multi hop interference on these topologies.

These observations increase the number of graphs that are known to satisfy LoP by a few orders of magnitude. We also show that all odd rings with at least 9 nodes and all even rings with at least 6 nodes do not satisfy LoP. Using the latter observation, we show that all bipartite graphs that are not chordal bipartite do not satisfy LoP. Networking environments in which the traffic is inherently single-hop and where packets must depart the system upon transmission across a link are rare.

This results from the fact that many connections are necessarily multi hop connections, due to geographical and physical constraints on user connectivity. To the best of our knowledge this is the first attempt to study the multi hop implications of Local Pooling. The obtained results can serve as a basis for the development of Local Pooling based algorithms.

The consideration of Local Pooling has the potential to enable efficient distributed operation of wireless networks. However, since previously LoP was studied mostly under the assumptions of single-hop traffic and primary interference, in this paper we focused on its multi hop implications. We identified several graph subclasses of the OLoP Satisfying class and increased the number of known graphs that satisfy LoP by a few orders of magnitude.

**4.6 Throughput of Random Access without Message Passing**

We develop distributed scheduling schemes that are based on simple random access algorithms and that have no message passing. In spite of their simplicity, these schemes are shown to provide high throughput performance. The quest for throughput optimal (distributed) scheduling schemes in wireless networks has attracted a lot of attention during the past ten years. We can categorize such distributed scheduling schemes into two broad categories. The idea there is that each transmitter tunes its transmission probability depending on information in its local neighborhood.

This is in sharp contrast to the collision-free algorithms or random access algorithm requiring queue-length information exchange mentioned above, making it the most practical MAC protocol. Actually in many practical scenarios, e.g. in case of networks with hidden terminals, it proves difficult to exchange information among neighbors. The extends model to propose simple schemes whose throughput performances are no less than maximal scheduling. Packets are assumed to have a fixed size, so that the packet transmission on any link has a fixed duration that we take equal to 1.

The model of our example, there might be collisions when two interfering links start transmitting simultaneously, where the collision durations are set to be equal to those of successful packet transmissions. In the following sections, we will consider the case where the collisions last a single slot only. Then, we expect even higher throughput efficiency.

The simple random access algorithms by considering the following two types of systems:

1) Synchronous systems. In these systems, time is divided into frames. At the beginning of each frame, each transmitter with non-empty buffer will attempt to use the channel with fixed probability at each slot until it can actually start transmitting, or until it senses activity of its neighbors succeeding to start transmitting, in which case it will wait for the next frame for further transmission attempts.

2) Asynchronous systems. In these systems, transmitters always sense the channel, and when their buffers are not empty, they start transmission of data just after an idle slot. As a consequence, the channel busy periods in different areas of the network may be not synchronized.

In asynchronous systems, obtaining the similar result to synchronous systems is less obvious, since identifying a schedule with maximum size requires the underlying loss network to be close to its stationary regime. The specific networks, the throughput performance of the Maximum Size scheduling schemes that can be realized using random access algorithms for large transmission probabilities.

**4.7 Improved Bounds on the Throughput Efficiency of Greedy Maximal Scheduling in Wireless Networks**

The low complexity, Greedy Maximal Scheduling (GMS), also known as Longest Queue First (LQF), has been studied extensively for wireless networks. GMS achieves 100% throughput in all networks with eight no des or less, under the two-hop interference model. Further, we obtain performance bounds that improve up on previous results for larger networks up to a certain size. We also provide a simple proof to show that GMS can b e implemented using only local neighborhood information in networks of any size.

In wireless communication networks with limited resources, efficient resource allocation plays an important role in achieving high performance and providing good quality of service. The performance metric of interest is through put and we restrict our attention to MAC layer throughput as opposed to end-to-end throughput.

In the sense that it can stabilize the queues in the network for all traffic rates in the capacity region of the network. These drawbacks greatly limit the deployment of MWS in real networks. Even in small net-works, MWS can require quite a lot of operations because its complexity is tied to the number of maximal schedules of the network.

Thus, we focus our attention obtaining performance bounds for small to moderate size networks although our results on information complexity apply to general networks. The Greedy Maximal Scheduling (GMS) algorithm, also known as the Longest-Queue-First (LQF) algorithm, has low complexity and hence can b e deployed in practical systems. Its performance has been observed to be close to optimal for a variety of network scenarios in simulations and experiments. Indeed, most wireless ad ho c networks that one may encounter in reality are small. More precisely, the maximum size of network under which GMS is guaranteed to achieve full capacity.

GMS is a low complexity scheduling algorithm which has b een observed to achieve near optimal throughput performance in a variety of wireless network simulations. However, theoretical bounds to date on the performance of GMS only show that it can achieve a fraction of the capacity region. The networks of small to moderate size, which is the case in many practical wireless ad-hoc and mesh networks.

**5. IMPLEMENTATION AND RESULTS**

**Modules**

* + - Packet Generation
    - User Polling
    - Network Monitoring
    - Packet Scheduling

**5.1 Packet Generation**

A packet generation model is a traffic generation model of the [packet flows](http://en.wikipedia.org/wiki/Packet_flow) or data sources in a [packet-switched network](http://en.wikipedia.org/wiki/Packet-switched_network). These models are useful during the development of telecommunication technologies, in view to analysis the performance and capacity of various protocols. The network performance can be analyzed by [network traffic measurement](http://en.wikipedia.org/wiki/Network_traffic_measurement) in a [test bed](http://en.wikipedia.org/wiki/Testbed) network, using a network traffic generator such as [iperf](http://en.wikipedia.org/wiki/Iperf), [bwping](http://en.wikipedia.org/wiki/Bwping) and [Mausezahn](http://en.wikipedia.org/wiki/Mausezahn). The traffic generator sends dummy packets, often with a unique packet identifier, making it possible to keep track of the packet delivery in the network. Packet filters act by inspecting the "packets" which transfer between computers on the Internet. If a packet matches the packet filter's set of rules, the packet filter will drop the packet, or reject.

Packet Generation

Network Traffic

Packet Matches

Not Match

Packet Identifier

**5.2 User Polling**

User polling is a variation of the traditional [polling](http://en.wikipedia.org/wiki/Polling_(computer_science)) technique and allows emulation of an information push from a server to a client. With user polling, the client requests information from the server in a similar way to a normal poll. The server does not have any information available for the client, instead of sending an empty response; the server holds the request and waits for some information to be available.

Once the information becomes available, a complete response is sent to the client. The client will normally then immediately re-request information from the server, so that the server will almost always have an available waiting request that it can use to deliver data in response to an event. User polling is itself not a true push, but can be used under circumstances where a real push is not possible, and offers many of the same benefits in terms of rapid delivery.

User Polling

Inform-ation

Empty Response

Normal Poll

Wait for Server

Complete Response

**5.3 Network Monitoring**

The term network monitoring describes the use of a system that constantly monitors a [computer network](http://en.wikipedia.org/wiki/Computer_network) for slow or failing components and that notifies the [network administrator](http://en.wikipedia.org/wiki/Network_administrator). A network monitoring system monitors an internal network for problems. It can find and help resolve snail-paced webpage downloads, lost-in-space e-mail, questionable user activity and file delivery caused by overloaded, crashed servers, dicey network connections or other devices.

Monitoring an active communications network in order to diagnose problems and gather statistics for administration and fine tuning. Network monitoring can be achieved using various software or a combination of plug-and-play hardware and software appliance solutions. Virtually any kind of network can be monitored. It doesn't matter whether it's wireless or wired, a corporate LAN, VPN or service provider WAN.

Networks

Packet Driver

Queue

Ack Verifier

Data

ACK

**5.4 Packet Scheduling**

IP-packets sent from a station or AP is first converted from bit streams to radio signal, which is modulated to a frequency for a channel and after that transmitted into the air medium. All network nodes within the BSS network are using the same channel to communicate and all stations within the LAN have equal opportunity to access to the medium using DCF protocol, including AP.

AP, which is the bridging node, has no privileges to transmit more on its downlink when sharing resources, but all flows from AP are treated equally and there is no contention among these. There is only one outgoing queue, which is a First in First out (FIFO), where aggregate flows are queued in an arrival manner. With N flows on the downlink, each flow receives 1/N of the available bandwidth capacity fairly.

Packet scheduler

ACK verifying

Process Further

If less

Mode Transmission

Delete from queue

If less

Repeat process

**5.5 Q-CSMA Algorithm**

1. Link i selects a random (integer) back off time Ti uniformly in [0 ,W − 1] and waits for Ti control mini-slots.

2. IF link i hears an INTENT message from a link in C (i) before the (Ti +1)-th control mini-slot, i will not be included in m (t) and will not transmit an INTENT message anymore. Link i will set xi (t)= xi ( t − 1).

3. IF link i does not hear an INTENT message from any link in C (i) before the ( Ti +1) -th control mini-slot, it will send (broadcast) an INTENT message to all links in C (i) at the beginning of the (Ti +1)-th control mini-slot.

* If there is a collision (i.e., if there is another link in C (i) transmitting an INTENT message in the same mini-slot), link i will not be included in m (t) and will set xi (t)= xi ( t − 1).
* If there is no collision, link i will be included in m (t) and decide its state as follows:

if no links in C (i) were active in the previous data slot

xi (t)=1 with probability pi , 0 <pi < 1 ;

xi (t)=0 with probability ¯pi =1 − pi .

else

xi (t)=0 .

4. IF xi (t)=1 , link i will transmit a packet in the data s lot.

**5.5 D-GMS Algorithm**

1. Link i selects a random backoff time

Ti = W ×└B − log b ( qi (t)+1)┘+ + Uniform[0 ,W − 1]

and waits for Ti control mini-slots.

1. IF link i hears an RESV message (e.g., an RTS/CTS pair) from a link in C (i) before the (Ti +1)-th control mini-slot, i will not be included in x(t) and will not transmit an RESV message. Link i will set xi (t)=0 .
   * + If there is a collision, link i will set xi (t)=0 .
     + If there is no collision, link i will set xi (t)=1 .

3. IF xi (t)=1 , link i will transmit a packet in the data s lot.

**5.6 Hybrid Q-CSMA Algorithm**

IF wi (t) >w0 ( Q-CSMA Procedure)

1.1 Link i selects a random backoff time Ti = Uniform [0 ,W0 − 1].

1.2 If link i hears an INTENT message from a link in C (i) before the (Ti+1)-th control mini-slot, then it will set xi (t)= xi ( t −1) and go to Step 1.4

1.3 If link i does not hear an INTENT message from any link in C (i) before the (Ti +1) -th control mini-slot, it will send an INTENT message to all links in C (i) at the beginning of the (Ti +1)-th control mini-slot.

* If there is a collision, link i will set xi (t)= xi (t − 1).
* If there is no collision, link i will decide its state as follows:

if no links in C (i) were active due to the Q-CSMA procedure in the previous data slot, i.e., NAi =0,

xi (t)=1 with probability pi , 0 <pi < 1 ;

xi (t)=0 with probability ¯pi =1 − pi .

else

xi ( t)=0 .

1.4 If xi ( t)=1 , link i will send an RESV message to all links in C (i) at the beginning of the (W0 +1)-th control mini-slot. It will set NAi =0 and transmit a packet in the data s lot. If xi ( t)=0 and link i hears an RESV message from any link in C ( i) in the ( W0 +1)-th control mini-slot, it will set NAi =1 ;otherwise, it will set NAi =0 .

IF wi ( t) ≤ w0 ( D-GMS Procedure)

2.1 If link i hears an RESV message from any link in C (i) in the (W0 +1) -th control mini-slot, it will set NAi =1 and xi (t)=0 and keep silent in this time s lot. Otherwise, link i will set NAi =0 and s elect a random backoff time

Ti =( W0 +1)+ W1 ×└B − log b ( qi ( t)+1)┘ + + Uniform [0 ,W1 − 1] and wait for Ti control mini-slots.

2.2 If link i hearsanRESVmessagefromalinkin C (i) before the (Ti +1)-th control mini-slot, it will set xi ( t)=0 and keep silent in this time slot.

2.3 If link i does not hear an RESV message from any link in C (i) before the (Ti +1)-th control mini-slot, it will send an RESV message to all links in C (i) at the beginning of the (Ti +1)-th control mini-slot.

* If there is a collision, link i will set xi ( t)=0 .
* If there is no collision, link i will set xi ( t)=1 .

2.4 If xi ( t)=1 , link i will transmit a packet in the data s lot.

**5.7 Result:**

The final result shows how efficiently utilize the bandwidth and reducing retransmission of packets and time efficiency for packet transformation.

**System requirements**

**Software Requirements**

* OS : Windows Xp
* Language : Java
* IDE : NetBeans 6.9.1

**Hardware Requirements**

* System : Pentium IV2.4GHz.
* Hard Disk : 250 GB.
* Monitor : 15 VGA Color
* Mouse : Logitech.
* Ram : 1GB.

**Conclusion**

A slotted distributed queue-length based CSMA/CA protocol that leads to collision-free data transmission schedules. The protocol is provably throughput-optimal. 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.

**Future Enhancement:**

Our future design focuses on routing that also balances the energy consumption, and forward the packet toward the sink through dense energy areas so as to protect the nodes with relatively low residual energy. However, with only this energy density field, the routing algorithm is not practical since it would suffer from the serious problem of routing loops. The depth potential field will play an important role in eliminating the routing loops or to protect low energy nodes.

**Reference**

1. P. Chaporkar, K. Kar, and S. Sarkar. “**Throughput guarantees through maximal scheduling in wireless networks”**. In Proceedings of 43rd An-nual Allerton Conference on Communication, Control and Computing , 2005
2. L. Jiang and J. Walrand. **“A distributed CSMA algorithm for throughput and utility maximization in wireless networks“**. In Proceedings 46th An-nual Allerton Conference on Communication, Control and Computing , September 2008.
3. G. Zussman, A. Brzezinski, and E. Modiano. **“Multi hop local pooling for distributed throughput maximization in wireless networks”**. In Proceedings of IEEE INFOCOM , April 2008.
4. Proutiere, Y. Yi, and M. Chiang. **“Throughput of random access without message passing”** . In Proceedings of CISS , March 2008.
5. M. Leconte, J. Ni, and R. Srikant. **“Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks”** . In Proceedings of ACM MOBIHOC , May 2009.