Providing Packet-Level Quality of Services in Multihop Wireless Networks

Jerry Cheng, Gary Zhong, Haiyun Luo and Songwu Lu
UCLA Computer Science Department
Los Angeles, CA 90095-1596

ABSTRACT

Providing packet-level quality of service (QoS) is critical to support both rate-sensitive and delay-sensitive applications in the bandwidth-constrained, shared-channel, multihop wireless networks. This problem is challenging due to the unique issues such as location-dependent contention, inherent conflict between ensuring fairness and maximizing channel utilization, and the distributed nature of packet scheduling in such networks. In order to address these issues, we have taken a new self-organizing approach to QoS solutions for such networks. In this approach, local decision makers self-organize themselves and coordinate among one another, and collectively achieve the desired global property. Some features of our approach include fully localized design, coordination among local decision makers, intentional and optimized information propagation, scaling property and achievable global property. Two key contributions of this work are: (a) a model-referenced self-organizing design methodology for multihop wireless networks; and (b) a table-driven approach and a backoff-based approach to distributed packet scheduling that provides QoS performance bounds in terms of fairness, throughput and delay, maximizes channel spatial reuse, and arbitrates the conflict between fairness and maximal channel utilization. Both proposed designs work within the CSMA/CA MAC framework. We also compare the performance of these two approaches through simulations. Our extensive simulation results have confirmed the effectiveness of the proposed design.

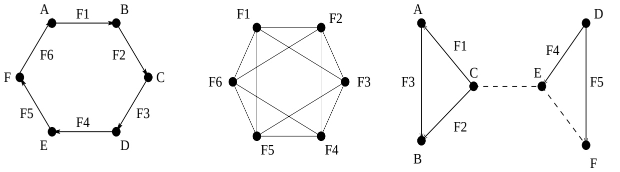
Keywords: quality of service, packet scheduling, multihop wireless networks, self organizing

1. INTRODUCTION

Emerging multihop wireless networking technologies such as MANET [1], bluetooth [2] and sensor networks [3] have offered an appealing paradigm for wireless data networking. In this class of networks, no underlying infrastructure, such as cellular layout and centralized control units (base stations), is available for networking support. A large number of networking devices are allowed to communicate with one another over the shared wireless medium in an ad hoc manner. Besides military applications, future deployment scenarios for such wireless networks include deeply networked conglomerations of embedded devices, emergency rescue operations, "zero conf" meeting setups, smart home, and rapidly reconfigurable metropolitan wireless networks [4]. In these emerging deployment scenarios, the multihop wireless network seeks to support not only a basic toolchest of services such as e-mail, file transfer, etc., but also more advanced applications such as collaborative learning and teleconferencing. This latter class of applications is mission critical and communication intensive, it requires sustained quality of service (QoS) support for effective operation.

Providing QoS in multihop wireless networks is challenging due to the unique issues such as location-dependent contention, fully distributed and localized design, and efficient network resource utilization. State-of-the-art solutions are inadequate to address these issues and meet the applications' QoS requirements. A fundamental problem is that, a multihop wireless network is a large-scale distributed system that may consist of a very large number of wireless networking components (e.g. thousands, even millions in a sensor network), which have limited channel resources. Therefore, any practical QoS solution must provide performance assurance in a fully distributed and localized manner. Network nodes must collectively achieve the desired global QoS requirements.

In this paper, we provide a packet-level QoS solution through packet scheduling, which has been a very effective instrument to support both delay-sensitive and rate-sensitive applications [5-7]. We take a novel self-organizing protocol design approach to packet scheduling in multihop wireless networks. In this approach, local schedulers self-organize themselves and coordinate with their neighbors, and collectively achieve the desired QoS requirements. Some features of our approach include fully localized design, coordination among local decision makers, intentional and optimized information propagation, and scaling property.



(a) Original Topology Flow Contention Graph (b) Conflict between fairness and maximal utilization

Figure 1. Location-dependent contention and spatial reuse

Two key contributions of this work are: (a) a model-referenced self-organizing design methodology for multihop wireless networks; and (b) a table-driven approach and a backoff-based approach to distributed packet scheduling that provides QoS performance bounds in terms of fairness, throughput and delay, maximizes channel spatial reuse, and arbitrates the conflict between fairness and maximal channel utilization. Both proposed designs work within the CSMA/CA MAC framework. We also compare the performance of two approaches through simulations. Our extensive simulation results have confirmed the effectiveness of the proposed design.

The rest of the paper is organized as follows. Section 2 describes key design issues and goals. Section 3 presents our self-organizing design methodology. Section 4 describes two approaches to packet scheduling by applying the self organization principle. Section 5 evaluates the proposed design. Section 6 compares with related work, and Section 7 concludes this paper.

2. DESIGN ISSUES AND GOALS

In this section, we explore the issues to be addressed, and identify a number of design goals.

We consider a packet-switched, infrastructureless, and multihop wireless network [8-12] in which the sheer number of nodes will not be constrained, and it may reach thousands or even millions. The wireless medium is shared among multiple contending nodes, i.e., a single physical channel with capacity C is available for wireless transmissions. Each node is capable of forwarding data packets to its neighbors over a shared wireless channel. Wireless transmissions are locally broadcast and only receivers within the transmission range of a sender can receive its packets. A collision occurs when a receiver is in the reception range of two simultaneously transmitting nodes, thus unable to cleanly receive signal from either of them; we ignore capture effect in this work.

2.1. Design Issues

In order to provide packet-level QoS through packet scheduling, we need to address the following three unique issues in multihop wireless networks:

Location-dependent contention in the shared wireless medium and channel spatial reuse The locality of wireless transmissions implies that collisions, and hence contention for the shared medium, are location dependent. The location-specific nature of contention, coupled with the multi-hop nature of the network, allows for channel spatial reuse. Any two flows that are not interfering with each other can potentially transmit data packets over the physical channel simultaneously. The selection of simultaneous transmitters thus determines the aggregate channel utilization. Hence, resource management algorithms need to perform a judicious selection of such simultaneous transmissions in order to maximize channel spatial reuse.

Consider the example shown in Figure 1.a, which shows a six-node network graph and each arrow-line denotes a packet flow from the sender to the receiver. Flow F_1 contends with flows F_2, F_3, F_5, F_6, since these four flows are within the transmission range of either the sender or the receiver of flow F_1. Therefore, these four flows should restrain from transmissions when F_1 transmits. Similarly, Flow F_2 contends with flows F_1, F_3, F_4, F_6. Hence, each flow has a different contending flow set depending on its location. For spatial reuse, consider flows F_1 and F_4. Since they are not contending with each other, and they can transmit concurrently, thus enabling channel spatial reuse.

Fully decentralized control Each network component should not perform centralized computation, and must perform localized computation only. Ideally, the network should also have local information propagation only, no network node should flood its information to the entire network. That is, information propagation must be controlled in a local scale, no global-scale information propagation should be permitted in general. In case that limited global information propagation is inevitable, it must be done in a controlled manner. That is, global information propagation should be rare and happen only during transient states.

Consider Figure 1.a again, each of the six senders $A - F$ does not know the packet-level flow information at the other nodes. This illustrates that packet scheduling in multihop wireless networks is a distributed computation problem by its nature.

Conflicting design goals: fairness versus maximal utilization Network system design typically involves multiple objectives, and these objectives may be potentially in conflict. For example, it is well known that fair network resource allocation does not generally lead to maximizing resource utilization in a network. In shared-channel infrastructureless wireless networks, fair channel allocation and channel spatial reuse may be conflicting in a generic network topology.

Consider the five-flow example shown in Figure 1.b. The system capacity will be $2C$ if we let F_3 and F_5 transmit all the time. However, it is easy to verify that the total effective capacity will be less than $2C$ if all five flows have to transmit and get a fair share. This example illustrates the fundamental conflict between achieving flow fairness and maximizing overall system throughput.

Because of these unique challenges, state-of-the-art solutions [9-12] are inadequate to meet all design goals. In fact, many problems such as fair packet scheduling for shared-channel multihop wireless networks have remained largely unaddressed.

2.2. Design Goals

Our ultimate goal is to provide sustained QoS support in terms of fairness, throughput, delay, packet loss and power consumption. In a large-scale multihop wireless network, we assert that a QoS solution with the following characteristics is required:

- *The solution must be efficient.* The state-of-the-art wireless communication technology offers a capacity much lower, even by an order of magnitude, than its wired counterpart. This makes a case for efficient resource management in the scarce and shared wireless medium. Because of channel spatial reuse, the selection of simultaneous transmitters determines the aggregate channel utilization. Hence, the packet scheduling discipline needs to perform a judicious selection of such simultaneous transmissions in order to increase spatial reuse, while taking into account fairness considerations across flows.
- *The design must be coordinated among interacting nodes.* The nature of location-specific contention implies that, any scheduling decision made at a node may have global impact and incur domino effects in the entire connected network graph. As a result, packet scheduling in such a network has to be coordinated among neighbors that have contending flows, and this coordination should be conducted in both the time domain and the spatial domain.
- *The solution must be scalable.* The number of nodes in the multihop wireless network can be large and the target can be a dense network, the solution should scale well under these condition. Besides, the solution should equally scale well in the presence of frequent node mobility and failures.
- *The solution must be fully distributed, and it involves only local computations by using local information only.*
- *The solution must exhibit desired global behavior, e.g., fairness property.*

3. A SELF-ORGANIZING ALGORITHM DESIGN APPROACH

3.1. Basic Concept

We take a self organization design approach to packet scheduling in this work. A self-organizing algorithm or protocol is a class of fully distributed design, which has the following features:

- Fully localized design: It must be based on local computation only.
- Self organization and coordination among local decision makers: Local decision-makers self-organize themselves and coordinate among one another to achieve desired global properties.
- Intentional and optimized information propagation: Each node should propagate only the best and correct information to its appropriate neighbors only. It should not propagate its local information to the entire network. In our packet scheduling design, each node only propagates the flow's information that it serves as the sender and thus has the correct information on these flows to its neighbors. In certain scenarios where limited global information propagation is inevitable, we take two approaches to minimize this effort: (1) Controlled or aggregate global information is propagated only during transient states, not during steady states. For example, in order to achieve global topology-independent fair queueing (in proportional to flow weight), aggregate flow-level weight information will be propagated to each node only if a new flow joins or an existing flow leaves the network. No packet-level information is propagated when nodes perform per-packet scheduling decisions during steady states. (2) Using packets to carry limited and necessary information. For example, we piggyback flow information in the MAC layer information exchange. Through both ways, we seek to propagate information locally and control global information propagation when inevitable.
- Desired global property: It is the collective behavior of individual local designs via local interactions.

3.2. A model-referenced design approach to self-organizing protocol and algorithm

In this section, we describe a "model-referenced" approach to self-organizing algorithm and protocol design for multihop wireless networks. Given a set of global properties or expected system behavior, the goal is to design localized algorithms that collectively achieve the global properties and exhibit the desired system behavior. While we focus on packet scheduling to apply these ideas, we believe the proposed method is equally applicable in other contexts. Our proposed approach consists of three main steps:

Step 1: Design of the ideal localized model or algorithm Given a set of desired global property, our goal for this step is to devise localized models/algorithms that collectively achieve the global property. Toward this end, we first ignore the practical issues such as information propagation, neighborhood discovery, interaction with other protocols (e.g., interaction of packet scheduling with the underlying MAC protocol), and resource dynamics such as channel errors and node mobility. Instead, we temporarily assume perfect knowledge on these types of information. Our goal is to design ideal localized model/algorithm that is fully decentralized by using local computation only and typically local information only. Our model is idealized by ignoring the practical issues and resource dynamics.

We proposed two approaches for localized model design:

- **Approach 1: Desired global property → centralized model/algorithm → localized algorithm by approximating the centralized model**

  In this approach, we first devise the ideal centralized model/algorithm that possesses the desired global property. Based on the centralized algorithm, we develop localized algorithms that approximate the centralized algorithm. This has been a popular approach to distributed algorithm/protocol design. In order to design self-organizing packet scheduling algorithm that arbitrates the conflicts between global fairness and maximal channel utilization, we first design a centralized algorithm that consists of a fair queueing phase in the basic channel and a maximum independent set phase to maximize spatial reuse. We then approximate the centralized model in a local backoff-based algorithm. A potential disadvantage of this approach is that it is generally very hard to precisely characterize the error in approximation between the centralized model and its localized approximation through analysis or simulations.

- **Approach 2: Desired global property → mapping to local property → localized model that realizes the local property**

  In this approach, we do not devise any centralized model/algorithm. Instead, we seek to directly map the desired global properties to appropriate local properties, which can be readily implemented via local models/algorithms. Toward this end, the local property set that each node possesses is typically a superset of the global properties. Therefore, the global property set is a subset of the local property set, and global properties are satisfied if local properties are satisfied, though not verse versa. We take this approach to solve the problem of distributed fair queueing in multihop wireless networks. According to fair queueing design principle, the flow that has the global minimum service tag (in the virtual time domain), should be scheduled with precedence. However, identifying the global minimum-service-tag flow requires global computation and is not feasible in a large-scale infrastructureless wireless network. In our design, we map this global property to a local property: we require each node to schedule the flow with a local minimum service tag in its neighborhood with precedence. Since a global minimum point must be a local minimum point but not verse versa, then we can develop local packet scheduling algorithms that schedule flows with local minimum service tags with high priority. This way, by selecting all non-interfering flows with local minimum service tags simultaneously, we can should analytically that the resulting scheduler is guaranteed to schedule the flow with global minimum service tag in the entire wireless network.

Step 2: Addressing practical issues Having devised an ideal localized model/algorithm, we now address the practical implementation issues such as information propagation in local neighborhood, and interactions with other protocols, etc. An example for this step in packet scheduling is how to exchange information with neighboring nodes and how to interact with the CSMA/CA MAC protocol.

Step 3: Online adaptation to channel dynamics, node mobility and node failures Wireless channels are prone to interference, fading and other types of channel errors. In addition, nodes in a multihop wireless network may be mobile or fail due to power depletion. These types of network dynamics cannot be known a priori. Therefore, the last step in our self-organizing design to use online adaptation to adapt to these time-varying network conditions. For example, we have developed well-documented adaptation techniques for wireless packet scheduling in the presence of channel errors [13-17].

4. PACKET SCHEDULING ALGORITHMS

In this section, we use packet scheduling to illustrate how to apply the self-organizing design approach proposed in Section 3. The ultimate goal for the self-organizing packet scheduling design is to devise algorithms that achieve packet-level QoS performance bounds, which include fairness, throughput and packet delay bounds, and channel spatial reuse. Fairness is critical to ensure that well-behaved users are not penalized because of excessive demands of aggressive users. Maximizing spatial reuse and providing throughput and delay bounds are critical to effectively support communication-intensive applications, which can easily stress the bandwidth-constrained wireless channel.

For simplicity of presentation, we first convert packet flows in a generic network topology into a flow contention graph, which characterizes the space-time contention relationship among transmitting flows. In a flow contention graph, each vertex represents a backlogged flow, and an edge between two vertices denotes that these two flows are contending. Vertices that are not connected denote flows that can transmit simultaneously. Thus, an independent set in the flow contention graph denotes a set of non-conflicting transmissions, which can scheduled simultaneously. Figure 1.a illustrates the generation of the flow contention graph from a given network topology.

In the following, we describe two specific approaches to packet scheduling in multihop wireless networks by applying the design principles proposed in Section 3. We use a backoff based mechanism to resolve conflicts between fairness and maximal channel utilization, and a table based mechanism to achieve distributed packet scheduling. Both approaches result in fully distributed designs, which satisfy different QoS requirements.

4.1. A backoff based approach to resolving conflicts between fairness and maximal spatial reuse

The goal of this algorithm is to address the tradeoffs between achieving fairness and maximizing channel utilization. We achieve this goal by enforcing a basic notion of fairness that ensures that each flow receives a minimum fair channel allocation, and maximizing aggregate channel utilization subject to this constraint. In the following, the algorithms that we propose can achieve both local and global fairness models, and we evaluate the fairness and utilization tradeoffs for these two algorithms. We take the following approach to self-organizing algorithm design: we first propose centralized algorithms, then approximate them via novel distributed and localized algorithms, and finally address practical implementation issues.

Step 1: Designing Ideal Centralized Model Our approach is to first achieve the fairness model by selecting flows for transmission in a fair queueing phase, and then maximize channel utilization by selecting additional flows for transmission in a maximum independent set phase, subject to the selection of the flows in the fair queueing phase. The precise details of the algorithm in the two phases decide whether the fairness model is global or local.

(a) Fair queueing phase: Achieving a minimum fair share through fair queueing As discussed above, we have two options for fairness model:

- Topology-independent global fairness: In the global fairness model, each backlogged flow receives allocation in the basic channel in proportional to its flow weight with respect to all backlogged flows in the network. This fairness property is identical to the one approximated by the wireline packetized fair queueing algorithm. Thus, we use any standard fair queueing algorithm to provide a "basic" allocation, and subject to this allocation, we seek to maximize the aggregate channel reuse according to the following algorithm:
    1. Select a flow i for transmission according to the packetized fair queueing algorithm.
    2. Select the maximum independent set S_i in $G - N(i)$, where $N(i)$ denotes the closed neighborhood of node i in the flow contention graph.
    3. Schedule packets for transmission in i and S_i. Increment the start and finish tags for flow i, but not for any of the flows in S_i. The fact that the tags are not incremented for the flows in S_i enables the scheduler to achieve the maximum possible additional channel reuse given the allocation for i "for free", i.e. the flows that receive additional channel allocation are not charged for it by increasing their tags.
- Topology-dependent local fairness: In the local fairness model, a backlogged flow receives allocation in the basic channel in proportional to its flow weight, only with respect to backlogged flows in its neighborhood. The packetized fair queueing algorithm is more involved, and it needs recursive deletion of flows in appropriate contending flow sets.

(b) Approximating the maximum independent set: In the algorithm described above, our idealized scheduling algorithms uses a maximum independent set generation algorithm in order to maximize channel utilization subject to minimum fairness constraints. While this is a well-known NP-complete problem, we use a minimum-degree greedy algorithm to approximate the maximum independent set [18]. It is shown in [18] that it achieves a performance ratio of $(\Delta + 2)/3$ for approximating independent sets in graphs with degree bounded by Δ.

Step 2: Approximating the centralized model in localized models After developing the ideal centralized algorithms, we need to design distributed and localized models to approximate them. Our approximation consists of two components:

(a) Approximating the fair queueing algorithm in the basic channel: For the global topology-independent fairness model, we approximate the fair queueing algorithm by a weighted round robin (WRR) with spreading algorithm. The WRR with spreading is essentially an approximation of the WFQ algorithm by assuming that each flow were always backlogged. Its worst-case performance bound, in terms of throughput, delay and fairness, is the same as the WFQ algorithm. However, if certain flows become idle, then the above algorithm will deviate from the WFQ algorithm. Specifically, extra bandwidth (due to idle flows) will not be allocated to backlogged flows that are waiting to be served in the basic channel; instead, we will give spatial reuse higher priority.

(b) Realizing the minimum-degree greedy algorithm via a novel backoff based approach: In our implementation, we take a backoff-based approach to the minimum-degree greedy approximation of the maximum independence set problem. The backoff based mechanism works within the CSMA/CA MAC protocol as follows: for each packet transmission, each flow sets a backoff timer and waits for a number of mini-slots, before transmitting a RTS request to its neighboring flows. Upon hearing a RTS request, every flow (in its neighbors) will disable its backoff timer and restrain from transmission until the transmitting flow finishes its current packet transmission. We set the backoff value to be equal to its flow degree. Therefore, flows with smaller flow degree will always transmit before the flows with larger degree if there are no transmissions going on in its neighborhood. This effectively approximates the minimum-degree greedy algorithm for spatial reuse.

Step 3: Addressing practical issues We implement our localized models within the framework of the CSMA/CA MAC paradigm. We address two important practical issues:

(a) The underlying MAC-layer support: In our MAC-layer design, a sequence of RTS-CTS-DATA-ACK handshake is initiated for each data packet transmission, and this message exchange is preceded by a backoff of certain number of minislot times. When a node has a packet to transmit, it will also wait for an appropriate number of mini-slots. For flows with minimum scheduling order in the basic channel, its backoff value is zero. For flows in concurrent transmissions due to spatial reuse, its backoff is set to be the flow degree.

(b) Information propagation via the conflict-free shared tree: When a new flow comes in or an existing flow terminates its transmission, if we adopt a global fairness model, this flow information should be known by all senders in the graph. This is inevitable in order to achieve the global topology-independent fairness model [4]. How to minimize global information propagation becomes an issue. To this end, the initiating flow will propagate this information to a pre-specified core node in the specific graph, and the core node will multicast an aggregate information (i.e., the sum of flow weights for all flows) to each sender in the network topology. Our second design goal is to propagate this information, in minimum time, from the core node to the rest of nodes. This is equivalent to constructing a conflict-free minimum-height spanning tree. We seek to build up a core-based shared tree that provides minimum time transmissions from the core node to all other nodes and ensures conflict-free concurrent delivery for sibling nodes at the same height of the tree. The detailed algorithm is described in [4].

Note that information propagation along the shared tree is only performed during flow-level transient state, i.e., when new flows come or existing flows leave. We do not propagate any packet-level or flow-level information when each flow performs packet-level scheduling decisions during steady states. Besides, if the local topology-dependent fairness model is chosen, there is no need to build the tree and propagate information globally.

4.2. A table based approach to distributed fair queueing

We now take the second approach to designing a self-organizing algorithm that realizes distributed fair queueing, in which each packet flow receives service in proportional to its flow weight. Toward this end, we first identify desired global properties, then map them to local properties, and devise localized algorithms to realize the local properties.

Step 1: Desired Global Properties We would like the distributed fair queueing algorithm to possess the following global properties:

(a) Fair share of bandwidth: Each flow is still served in proportional to its flow weight in its local scheduler. To provide a fair share for each flow in the connected flow graph, the flow that receives the minimum normalized service (normalized according to its flow weight r_f) in the network must transmit first. Equivalently, the flow with the global minimum service tag at t should always be transmitted with precedence. This is what we called ”maximizing global minimum” property.

(b) Increasing channel spatial reuse: Since wireless channel is bandwidth constrained, the proposed model should increase channel spatial reuse as much as possible and improve the aggregate channel utilization.

(c) Bounding unfair spatial reuse if needed: While spatial reuse increases network efficiency, it may cause certain flows' services unbounded in some topological scenarios. In certain application-specific scenarios, we may still want to limit the unfairness bound caused by spatial reuse.

Step 2: Mapping global properties to local properties We now map the three desired global properties to local properties that can be readily achieved by localized models:

(a) Maximizing local minimum: Because identifying the global minimum involves global search that cannot avoid global computation, we identify all flows with local minimum service tags, and schedule all such flows for transmission. This is what we called maximizing local minimum policy. As far as service tag is concerned, since the global minima must be a local minima (but not vice versa), we know that the flow with the global minimum tag must be among these transmitted flows that have local minimum tags and it is guaranteed to be transmitted first. Hence, "maximizing local minimum" policy is a superset of the "maximizing global minimum" policy, but not verse versa.

(b) Transmitting noninterfering flows simultaneously to increase spatial reuse: Flows that are not interfering with those having local minimum service tags should also transmit simultaneously, in order to increase spatial reuse and the aggregate channel utilization.

(c) Bounding flow unfairness at each node: Since unfairness is bounded among all correlating flows in each neighborhood in the connected network graph, if we bound the maximum allowed flow unfairness at each node and its neighbors, then the global unfairness is also bounded.

Step 3: Designing ideal localized model that possesses the local properties In our proposed local model, each node is responsible for assigning service tags and scheduling flows that it serves as the sender; it still uses standard fair queueing algorithms to assign start tags and finish tags for the flows that this node acts as their sender. In addition, each node maintains a local table. Each table records current flow information for all flows that the node serves as the sender or receiver and all flows in their closed one-hop neighborhood in the flow contention graph. Specifically, each table entry records the following flow service tag information: [flow id, flow tag], where flow tag is the most recent service tag that the node hears for flow flow id.

Our proposed local model uses three mechanisms to realize the three local properties identified above:

(a) Maximizing local minimum by transmitting flows with local minimum service tags: A node immediately transmits a flow only if this flow has the minimum service tag flow tag among all backlogged flows in its table.

(b) Using a backoff mechanism to increase spatial reuse: If a flow does not have the local minimum service tag flow in its sender's table, the sender sets a backoff time for this flow in order to increase channel spatial reuse. A flow transmits once its backoff period expires and no other flow is transmitting in the channel. We set the backoff timer for flow f to be the number of flows in the table whose current service tags are less than flow f. This way, we tailor the flow's backoff value to both the flow's local fairness (i.e., its service compared to its neighbors) and its local contention degree (i.e., the number of contending flows in its neighborhood) in the flow graph.

(c) Using sliding windows to limit the unfairness spatial reuse bound: each node maintain an upper bound for flow unfairness; whenever any flow's service tag reaches beyond that allowed by the window size, the flow is restrained from transmissions temporarily.

Step 4: Addressing Practical Issues in Implementation We further address the following practical issues in our implementation within the framework of the CSMA/CA MAC paradigm:

(a) Exchanging table information at a flow's sender and its receiver: In the local model described above, each node needs to maintain local information for flows within its one-hop neighborhood in the flow contention graph. However, one-hop neighborhood in a flow graph will translate to the two-hop neighborhood (i.e., the neighborhood of both the sender and the receiver) in the real node graph in practice.

(b) Exchanging each flow's updated virtual time: In our model, the table of each node needs to record the most recent virtual time for each neighboring flow in the flow graph. Whenever a flow transmits, all the senders of its neighboring flows should update the new virtual time for this flow. A naive approach is to include the new virtual time either in the control messages of RTS and CTS, or packets DATA or ACK. However, spatial reuse may prevent certain flows always hear collisions and never get their relevant table entries updated. Consider Figure 1 again, when F1 and F4 are transmitting simultaneously, F and C nodes will always hear collisions and will not be able to hear the new virtual times of F1 and F4.

A brief overview of our self-organizing implementation that addresses the above issues is the following:

(a) Basic Message Exchange Sequence: In our protocol, each data transmission follows a basic sequence of RTS-CTS-DS-beason-DATA-ACK-beason handshake, and this message exchange is preceded by a backoff of certain number of minislot times. (b) Exchanging table information between the sender and the receiver: Remember for each flow f in concurrent transmissions during spatial reuse, its backoff is set to be the number of flows in the table whose service tags are less than flow f. According to this policy, we should set the backoff value for a flow, by taking into account both tables at the sender and the receiver. Therefore, this motivates us to have developed a two-step procedure to set the correct backoff value for each flow.

(c) Propagating a flow's updated virtual time locally: We use the beacons that follow after DS and ACK to propagate a flow's updated virtual time to its one-hop neighbors in the flow graph. Because a flow's sender always has correct and updated information on its service tag, the sender is always responsible to propagate this accurate information to its neighbors. Hence, only accurate information will be propagated in each local neighborhood of the graph in our design.

Further details are described in [19].

5. SIMULATIONS

We have evaluated the backoff-based approach of Section 4.1 and the table-based approach of Section 4.3 through both analysis and extensive simulations. A brief summary of the results is given as follows:

(a) Throughput and delay bounds: Both algorithms are able to provide performance bounds in terms of throughput and fairness.

(b) Fairness: The service received by each flow in both algorithms satisfies the designated fairness model.

(c) Spatial reuse and aggregate channel utilization: Spatial reuse tends to be better for the algorithm of Section 4.1 in a generic network topology, and aggregate channel utilization is the largest if we adopt the global topology independent fairness model. However, this is achieved at a smaller fair share in the basic channel.

(d) Implementation complexity: Both the backoff-based and the table-based implementations are localized, thus satisfying our design criteria. The table-based implementation is slightly more complex within the CSMA/CA MAC framework, but its nice feature is to provide better fairness in the presence of idle flows. The backoff-based implementation in Section 4.1 needs the core based tree to distribute flow information.

In the following, we use two examples to briefly illustrate the comparison. We compare both algorithms described in Section 4 with FIFO scheduling plus IEEE802.11 MAC. Our simulator is written in ns-2. We modified the IEEE MAC 802.11 module of the simulator. The radio model has the transmission range of 250 meters and channel capacity of 2Mbits/sec. The packet size is 512 bytes. Each simulation is run for 50 seconds. We use equal flow weight throughout the network.

Simulation Scenario 1 We simulate five flows in the network topology illustrated in Figures 2 and 3. The simulation results are shown in Table 1. From the table, we observe that the backoff-based design achieves maximum throughput, and the table-based approach achieves perfect inter-flow fairness. Both approaches provide minimum fair share for each flow. However, the FIFO scheduling together with IEEE802.11 MAC cannot provide minimum fair share, flow F_1 is almost starved.

Simulation Scenario 2 We simulate fifteen flows in the topology shown in Figures 4 and 5, and the results are given in Table 2. From the table, we see that the backoff algorithm again achieves maximum throughput, and the table-based design provides perfect fair share. However, in 802.11 MAC, three flows F_1, F_4 and F_5 are almost starved. This again illustrates that both our proposed approaches are able to provide QoS assurances in terms of throughput, delay and minimum fair share.

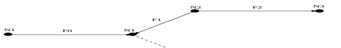


Figure 2. Ex. 1: Node graph

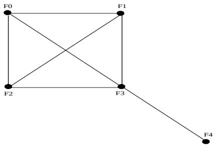


Figure 3. Ex. 1: Flow graph

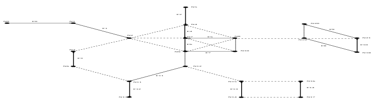


Figure 4. Ex. 2: Node graph

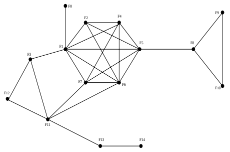


Figure 5. Ex. 2: Flow graph

Flow	802.11 MAC	BACKOFF	TABLE
0	402	4369	1023
1	4821	2185	1023
2	5472	2186	1023
3	2015	2185	1023
4	2466	8738	1022
Total	15176(100%)	19663(129.6%)	5114(33.7%)

Table 1. Ex. 1: Performance Comparisons

Flow	802.11 MAC	BACKOFF	TABLE
0	10334	8346	540
1	82	597	537
2	2325	6558	539
3	2060	597	540
4	25	597	537
5	40	597	537
6	177	597	537
7	163	597	538
8	4199	597	542
9	2765	597	544
10	5296	7750	543
11	409	597	541
12	11408	7750	543
13	165	597	543
14	850	8346	544
Total	40298(100%)	44720(111%)	8105(20.1%)

Table 2. Ex. 2: Performance Comparisons

6. RELATED WORK

Wireline and wireless Packet scheduling Packet scheduling has been the subject of intensive study in the networking literature and numerous algorithms have been proposed, among which are WFQ [5, 6], and Start-time Fair Queueing [7], etc. In recent years, there are several research efforts on adapting fair packet scheduling to wireless cellular networks, notably IWFQ [13], CIF-Q [15], SBFA [16], CBQ-CSDPS [14] and WFS [17]. The goal of these wireless fair scheduling algorithms has been to hide short bursts of location-dependent channel errors from well-behaved flows by dynamically swapping channel allocations between backlogged flows (that perceive channel errors) and backlogged flows (that do not), with the intention of reclaiming the channel access for the former when it perceives a clean channel. Therefore, lagging flows (that lag behind their error-free reference service due to channel errors) receive compensation from leading flows. In multihop wireless networks, providing minimum throughput bounds and bounded delay access has been studied at the MAC layer [9-12]. A popular approach has been to establish transmission schedules and allocate stations to different time slots of a TDMA cycle in a way that no collisions occur. The goal is to design conflict-free packet scheduling schemes that seeks to maximize spatial reuse and remain immune to topological changes in a mobile networking environment. Another study [16] also investigates the fair link activation problem in such a network. However, all these previous studies seek to provide throughput bounds or weighted fairness for wireless links, not for packet flows; hence, they do not address the problem of packet scheduling for packet flows. Besides, these algorithms tend to work with a fixed TDMA cycle, and do not have the dynamic scheduling feature. Furthermore, the focus of these MAC-layer studies has been on the mechanisms of channel access by assuming that the packet scheduling algorithm has been worked out, rather than the other way around. Finally, these works do not consider the problem of arbitrating fairness and maximal channel utilization.

There is a recent work that also addresses fairness issues in multihop wireless networks [21]. In [21], the authors have studied the problem of distributed fair queueing in a wireless LAN. However, the focus of [21] is to ensure fairness by adapting fair queueing to such a network, and it does not make explicit efforts to maximize spatial reuse subject to fairness constraints.

7. CONCLUSION

In this paper, we have proposed a self-organizing approach to distributed packet scheduling in multihop wireless networks, in order to provide packet-level QoS support for rate-sensitive flows. Our proposed approach seeks to devise scalable and efficient solutions to distributed packet scheduling, and these solutions provide fairness and increase spatial reuse. Our solutions only rely on local information and local computations, and multiple localized schedulers coordinate their interactions and collectively achieve desired global properties such as fairness, scaling and efficiency. We demonstrate the effectiveness of our proposed design through both simulations and analysis.

REFERENCES

1. The mobile ad-hoc networks (MANET) working group, *http://www.ietf.org/html.charters/manet-chater.html.*
2. The bluetooth special interest group, *http://www.bluetooth.com/.*
3. E. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," *ACM MOBICOM'99*, 1999.
4. H. Luo, S. Lu and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," *ACM MOBICOM'00*, August 2000.
5. A. Demers, S. Keshav and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *ACM SIGCOMM'89*, August 1989.
6. A. Parekh, "A generalized processor sharing approach to flow control in integrated services networks," *PhD Thesis*, MIT Laboratory for Information and Decision Systems, Technical Report LIDS-TR-2089, 1992.
7. P. Goyal, H.M. Vin and H. Chen, "Start-time fair queueing: A scheduling algorithm for integrated service access," *ACM SIGCOMM'96.* August 1996.
8. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A medium access protocol for wireless LANs," *ACM SIGCOMM'94*, 1994.
9. I. Chlamtac and A. Lerner, "Fair algorithms for maximal link activation in multihop radio networks," *IEEE Trans. Communications,* 35(7), July 1987.
10. J. Ju and V.O.K. Li, "An optimal topology-transparent scheduling method in multihop packet radio networks," *IEEE/ACM Trans. Networking,* 6(3), June 1998.
11. I. Chlamtac and A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Trans. Networking,*, 2(1), February 1994.
12. Z. Tang and J.J. Garacia-Luna-Aceves, "A protocol for topology-dependent transmission scheduling in wireless networks," *WCNC'99*, September 1999.
13. S. Lu, V. Bharghavan and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. Networking*, August 1999.
14. M. Srivastava, C. Fragouli, and V. Sivaraman, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queueing with Channel-State-Dependent Packet Scheduling," *IEEE INFOCOM'98*, March 1998.
15. T.S. Ng, I. Stoica and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors," *IEEE INFOCOM'98*, March 1998.
16. P. Ramanathan and P. Agrawal, "Adapting packet fair queueing algorithms to wireless networks," *ACM MOBICOM'98*, October 1998.
17. S. Lu, T. Nandagopal, and V. Bharghavan, "Fair scheduling in wireless packet networks," *ACM MOBICOM'98*, October 1998.
18. M. M. Halldorsson and J. Radhakrishnan, "Greed is good: approximating independent sets in sparse and bounded-degree graphs," *ACM STOC'94*, 1994.
19. H. Luo and S. Lu, "A self-coordinating approach to distributed fair queueing in wireless ad hoc networks," *INFOCOM'01*, 2000.
20. N. H. Vaidya and P. Bahl, "Fair scheduling in broadcast environments," *Microsoft Research Tech. Rep. MSR-TR-99-61.*
21. N. H. Vaidya, P. Bahl and S. Gupta, "Distributed fair scheduling in a wireless LAN," *ACM MOBICOM'00*, August 2000.
22. H. Luo and S. Lu, "A topology-independent fair queueing model in ad hoc wireless networks," *IEEE ICNP'00*, Nov. 2000.