

Scheduling Optimization in Multiuser detection based MAC design for Ad-Hoc Networks

M. Bouharras, Z. Dziong, F. Gagnon, M. Haider
Department of Electrical Engineering,
Ecole de Technologie Supérieure
Montreal, Quebec, Canada, H3C 1K3

(Mohamed.bouharras.1, Mohamad.Haidar.1)@ens.etsmtl.ca, {Zbigniew.Dziong, Francois.Gagnon}@etsmtl.ca

Abstract— Multiuser detection based Medium Access Control (MAC) can give significant gains in throughput and Quality of Service (QoS) when applied to wireless Ad Hoc networks. To realize these gains, one has to implement a distributed neighborhood scheduling that provides the desired performance objectives. In this paper, we propose an approach for analyzing and comparing optimal or suboptimal distributed neighborhood scheduling schemes with different objectives. Then, we demonstrate the viability of this approach by implementing a scheduling scheme that uses Start Time Fair Queuing (STFQ) algorithm and compares its performance to a published suboptimal distributed scheduling for multiuser detection based MAC. In particular, our numerical results show that the delay performance of the priority voice packets provides improvements over prior work.

Keywords-component; Multiuser detection; Multiuser reception; MAC protocol; scheduling; fairness; and wireless ad hoc networks.

I. INTRODUCTION

Wireless ad hoc networks are expected to support multimedia applications in emergency disaster management and military operations. This class of mission-critical applications demands a certain level of quality of services (QoS) for proper operation. Many MAC protocols have been developed for wireless networks which assume a common channel shared by mobile hosts in the network. These protocols, such as IEEE 802.11 are referred to as protocols based on the single-channel model. Due to relatively high probability of contentions and collisions, the performance of single-channel MAC protocols deteriorates quickly while the number of mobile hosts increases, especially for multi-hop connections [1]. To mitigate this problem, one can consider utilizing multiuser detection (MUD) or multiuser reception (MUR) CDMA technology that allows the reception of multiple CDMA channels at the same time. In [2] and [3] the authors proposed a novel MAC and scheduling algorithms to take advantage of MUD and MUR. One of the key elements of their approach is a distributed neighborhood scheduling mechanism that is based on a protocol that exchanges information between neighbors. They presented numerical results indicating significant gains in the QoS (delay) and network throughput. The objective of their scheduling mechanism was based on a principle that priority was given to the voice packets that waited the longest time in the neighborhood's queues. This

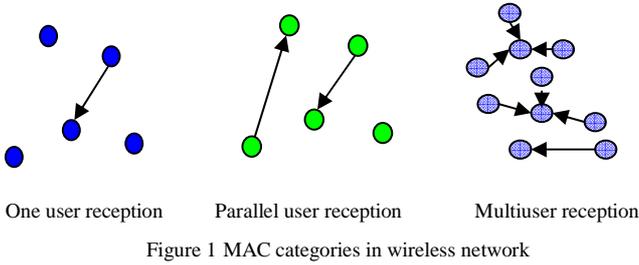
principle is not necessary optimal as shown in the studies of fair queuing mechanisms. Also, the network operation can have multi-objective formulation where the scheduling should balance network throughput maximization objective with traffic class fair access optimization. Therefore, in this paper, we propose an approach for analysing and comparing close to optimal distributed neighbourhood scheduling schemes with different objectives. The approach is based on the notion of a flow contention graph that takes into account the topology of the network. Using this notion, we construct a dependence matrix of flows in each node that constitutes a base for selecting optimal configuration of transmissions in a given slot according to the chosen objective. Then, we demonstrate the viability of this approach by implementing a scheduling scheme that utilizes STFQ (Start Time Fair Queuing) algorithm and by comparing its performance with the suboptimal distributed scheduling presented in [2] and [3]. The numerical results illustrate the total network throughput and the delay performance of the priority voice packets for the two scheduling algorithms.

The most relevant publication to our approach is presented [4] where the authors have developed distributed scheduling approach that achieves weighted fairness while trying to maximize the throughput for IEEE 802.11 based ad hoc network. Their approach is based on the notion of a flow contention graph that takes into account the topology of the network. They have also developed a topology independent model for fair queuing in [5]. In relation to wireless ad hoc networks, little work has been reported for fair packet scheduling. The most relevant work to packet scheduling is presented in [6] where the authors proposed Distributed Fair Scheduling (DFS) for IEEE 802.11 WLAN. Their mechanism was based on Self-Clocked Fair Queuing (SCFQ) scheme which improves fairness in WLAN. However, it is very hard to be directly implemented for wireless ad hoc networks.

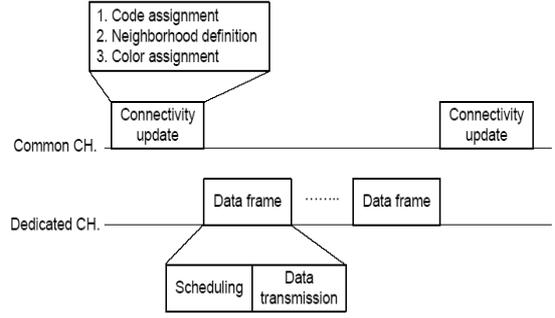
The paper is organized as follows: Section 2 presents the considered MAC environment. Our approach for scheduling is described in Section 3. Section 4 provides the implementation of fair queuing mechanism using the proposed approach. In Section 5, we compare the performance of the fair queuing implementation with the scheduling scheme presented in [2] and [3] by means of simulation. Finally, Section 6 concludes the paper.

II. CONSIDERED MUD BASED MAC ENVIRONMENT

Single-channel MAC mechanisms (like IEEE 802.11) have difficulty with providing required QoS for multimedia services in multi-hop Ad Hoc networks due to the large and variable delays of packet transmissions. One possible direction to accommodate the multimedia services in Ad Hoc networks is to increase the spectrum reuse by using CDMA multi-channel transmissions. Two basic architectures can be considered: parallel user transmissions (PUT) and multiuser reception (MUR). CDMA MUR is widely used in commercial cellular systems [7] while CDMA PUT are studied in full in [8]. The three categories of MAC platforms are illustrated in Figure 1.



The efficiency of MUR can be further increased by applying MUD [9]. In this case, the mutual interference of received signals is mitigated at the expense of increased complexity. Recent technological advances allow integration of a CDMA MUD based receiver on one chip. This development allows considering application of MUD for ad hoc networks in order to take advantage of both: spectrum reuse improvement due to MUR and capacity gain due to MUD. However, applications of MUD and MUR to ad hoc networks are not straightforward. In particular, the issue of MAC scheduling is quite challenging since the number of possible CDMA channel configurations in a node's neighborhood is large. Note that in general, an efficient solution to this problem requires a protocol that exchanges the necessary information among all the neighboring nodes without a large overhead in order to take advantage of CDMA spectrum reuse. A possible solution to this problem was proposed in [2] and [3] where a MAC mechanism based on synchronous frames was proposed as illustrated in Figure 2. In this approach the data frames are divided into a scheduling slot and a data transmission slot. In each signaling slot the information is first exchanged between the neighboring nodes and then the distributed scheduling decision is made based on a principle that priority is given to the voice packets that has the smallest timeout value (the time after which the packet is rejected). Since this principle is not necessary optimal and the operator may want to take into account other factors such as throughput maximization, in the next section we propose an approach for distributed scheduling in the presented MAC environment that can be used for testing and comparing different scheduling mechanisms with different objectives.



III. DISTRIBUTED SCHEDULING APPROACH

In this section we describe a scheduling approach that assumes that each node receives relevant information from all other nodes through the signaling stage prior to this stage (Note that in the scheduling protocol described in [2] and [3] only one hop exchange was taken into consideration.) The approach is comprised of three main components. First, the flow matrix and flow dependence matrix are constructed in each node. Second, the possible configurations (also called cliques) of transmissions in the neighborhood are defined. Third, the best transmission configuration for the forthcoming data transmission slot is selected. In the following subsections we describe each of these components.

A. Construction of the Node Matrix and Dependence Flow Matrix

1) Flow Matrix

Figure 3 displays the topology graph with arrows representing packets (also referred to as flows) selected for potential transmission in next data transmission slot. Each node selects at most one such packet according to the local scheduling. For this graph we can construct the flow matrix, $F = [f_{i,j}]$ for $1 \leq i, j \leq N$, where $f_{i,j}$ is a flow between nodes n_i and n_j , and N is the number of nodes. The entries of flow matrix are defined as follows:

$$f_{i,j} = \begin{cases} 0, & \text{if } i = j \text{ or nodes are outside the transmission range} \\ 1, & \text{if there are packets waiting for transmission from } n_i \text{ to } n_j \\ 2, & \text{if there are no packets waiting for transmission from } n_i \text{ to } n_j \end{cases}$$

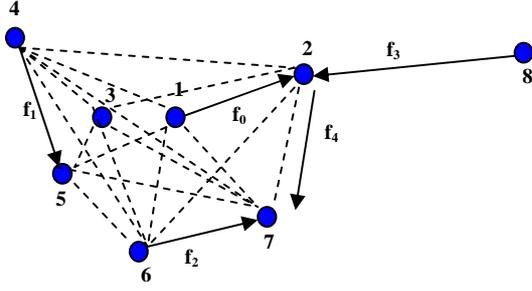


Figure 3 Flow graph

The flow matrix corresponding to the flow graph from Figure 3 is represented in Table 1.

Table 1 Flow matrix

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8
n_1	0	1	1	2	2	2	2	0
n_2	2	0	2	2	2	2	2	2
n_3	2	2	0	2	2	2	2	0
n_4	2	2	2	0	1	2	2	0
n_5	2	2	2	2	0	2	2	0
n_6	2	2	2	2	2	0	1	0
n_7	2	2	2	2	2	2	0	0
n_8	0	1	0	0	0	0	0	0

2) Flow Dependence Matrix

Let us define the flow dependence matrix $D=[d_{xy}]$ for $f_x, f_y \in M$ where M is the set of active flows ($f_x=f_{i,j}=1$) and the entries are defined as follows

$$d_{xy} = \begin{cases} 0, & \text{if } f_x \text{ and } f_y \text{ can be transmitted at the same time} \\ & \text{or } x=y \\ 1, & \text{if } f_x \text{ and } f_y \text{ cannot be transmitted at the same time} \end{cases}$$

Note that in our MAC environment, some pairs of flows cannot be transmitted at the same time due to half duplex operation of transceivers and not due to the channel contention. The flow dependence matrix corresponding to the flow graph from Figure 3 is given in Table 2.

Table 2 Flow dependence matrix

		$f_{1,2}$	$f_{4,5}$	$f_{6,7}$	$f_{8,2}$	$f_{2,7}$
		f_0	f_1	f_2	f_3	f_4
$f_{1,2}$	f_0	0	0	0	0	1
$f_{4,5}$	f_1	0	0	0	0	0
$f_{6,7}$	f_2	0	0	0	0	0
$f_{8,2}$	f_3	0	0	0	0	1
$f_{2,7}$	f_4	1	0	0	1	0

The flow dependence graph corresponding to the flow dependence matrix from Table 2 is depicted in Figure 4.

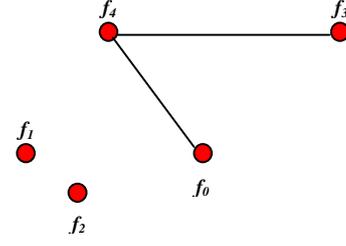


Figure 4 Flow dependence graph

B. Construction of Cliques

Based on its flow dependence matrix each node can create several feasible flow configurations (also called cliques) that can be considered for transmission in the next data transmission slot. We assume that the feasible flow configuration, C_k , can exclude some flows only if these flows are in a conflict with some flows selected for this configuration. Then the algorithm to find the set of all feasible cliques (FC) is defined below:

Let us define the flow set, flow-set = M = {all active flows} and dependence set, dep-set = {flows are in a conflict with some flows} and n is number of active flows. We denote f_x is in conflict with f_y by $f_x \neq f_y$.

- The dep-set from flow dependence matrix is defined below:

In the first, dep-set = $\{ \phi \}$,

For $i = 1$ to n

For $j = 1$ to n

if $d_{xy} = 1$ then $f_x \neq f_y$

dep-set = dep-set $\cup \{ (f_x \neq f_y) \}$

e.g. from table 2, dep-set = $\{ (f_0 \neq f_4), (f_3 \neq f_4) \}$

- Create feasible cliques by creating all possible combinations from flow-set and dep-set, i.e. all possible flow sets with the flows are not in a conflict.

For table 2 the above algorithm will create two feasible cliques: FC = $\{ C_1, C_2 \}$, $C_1 = (f_0, f_1, f_2, f_3)$ and $C_2 = (f_1, f_2, f_4)$.

C. Clique Selection

Once the feasible cliques are defined, each node selects the best one, according to the network operator objectives, for scheduling in the next data transmission slot. This selection can be achieved by using a metric that characterizes each clique with respect to the operator objectives. In general, this metric can be formulated as a utility function that can also take into account multi-objective formulations. For example, throughput maximization's objective can use the number of flows in the clique as a metric. On the other hand, priority fairness's

objective can use sophisticated formulas for tag calculations from fair queuing domain and then characterize each clique by the most critical tag among the flows as illustrated in Figure 5. For multi-objective formulations, one can use a heuristics algorithm or apply an approach based on Game Theory where the metric is based on some fairness criteria.

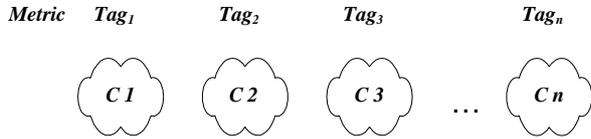


Figure 5 Clique metrics

In summary, the proposed approach allows to test and compare multitude of optimal and suboptimal scheduling algorithms with different objectives including multi-objective formulations.

D. D. Update flow dependence matrix

Whenever node n listens to a new service tag for any flow f_j from its \mathbf{D} , it updates the entry for flow f_j and when node n transmits a head-of-line packet for flow “ f ”, it updates flow “ f ” service tag in \mathbf{D} and piggybacks the service tag in the handshake messages message of control (RTS, CTS, ACK).

IV. FAIR SCHEDULING ALGORITHM IMPLEMENTATION

In this section, we present implementations of our model with Start time Fair Queuing (STFQ) and Timeout priority (TOP) within the approach presented in the previous sections. The STFQ algorithm and its application to our approach are described along with the timeout priority implementation of our model.

A. Start time Fair Queuing (STFQ)

Many studies have been carried out on fair queuing algorithms to achieve a fair allocation of bandwidth on a shared link. Fair queuing algorithms in literature typically attempt to approximate the Generalized Processor Sharing (GPS) discipline [10]. STFQ, which is a GPS discipline is computationally efficient and allocates bandwidth fairly regardless of admission control and server variation. It schedules packets in the increasing order of start tags and uses two tags: a start tag denoted by $S(P_i^k)$, and a finish tag denoted by $F(P_i^k)$. These tags are associated with each packet, where i denotes the flow number and k denotes the round number. Our system assigns start tags using a virtual clock which plays the role of a “flow meter” driven by packet arrivals. According to the flow’s specified average transmission rate, the difference between the Virtual-Clock and the real time clock will show how closely a running flow is following its claimed rate. Virtual Time $v(t)$ is defined as the start tag of the packet in service at time t .

- When a packet P_i^k arrives at time t , it is labeled with the start tag that is calculated as follow:
 - a- In an active period: $S(P_i^k) = F(P_i^{k-1})$

b- In an inactive period: $S(P_i^k) = v(t)$

- Initially the Virtual Time of the server is zero. During an active period, the Virtual Time $v(t)$ at time t is defined to be the start tag of the packet in activity at time t , $v(t) = S(P_i^k)$. At the end of the busy period, virtual time is set to the maximum of finish tag assigned to any packets that have been serviced. Then, $v(t) = \max_i^k F(P_i^k)$.

- The Finish time is computed as follows:

$$F(P_i^k) = S(P_i^k) + L_i / W_i$$

where L_i is the packet size of flow i and W_i is its weight.

- Packets are serviced in the increasing order of the start tags and the ties are broken randomly

B. STFQ Implementation

In [2] and [3], the priority is given to a packet with the smallest values of timeout after which the packet is destroyed (the timeout value is decremented by one after each frame). In the remainder of this paper, we compare the performance of this approach with a more sophisticated mechanism adapted from the STFQ algorithm. This mechanism can be divided into two parts: tagging and scheduling. Tagging maintains a track of lead and lag in the amount of service each flow receives according to the algorithm presented in the previous subsection. Then, in each node the distributed scheduling mechanism, described in section 3, selects a clique that contains the lowest-tag flow in order to preserve fair scheduling across all flows. The selected clique defines the function of the node (receiver or transmitter) and the packets (flows) to be sent in the next data transmission slot.

C. Time out priority Implementation with our model

The same metric as prescribed in [2] and [3] is used in our timeout priority (TOP) implementation. The distributed scheduling mechanism in each node selects a clique that contains the smallest values of timeout after which the packet is destroyed in order to preserve fair scheduling across all flows.

V. ANALYSIS AND RESULTS

In this section, we compare the performance of the scheduling based on STFQ and TOP presented in Sections 4 and the scheduling based on TOP proposed in [2] and [3]. In both cases, the same MUR and MUD based MAC are applied. Throughput and QoS characteristics are analysed for MUD and MUR cases. In both cases, all applied MAC and scheduling algorithms are the same except for the interference and power requirement calculations.

The numerical results are obtained by means of a discrete event simulation that models an ad hoc network with parameters presented in Table 3. Initially the nodes are randomly distributed in the modeled circle area. Then a

mobility model is used which mimics human and vehicle movement behavior [11]. The voice packet arrival rate represents 20% of total arrival rate.

Table 2 Simulation parameters

<i>Spreading gain</i>	<i>128</i>
<i>Modeled circle area</i>	<i>1000 m</i>
<i>The speed limit</i>	<i>50km/h</i>
<i>Maximum trans. power</i>	<i>7w</i>
<i>Types of traffic</i>	<i>voice and data</i>
<i>The number of nodes</i>	<i>60</i>
<i>Simulation time</i>	<i>100000 frames</i>

Average voice packet delay and total throughput (voice + data packets) are two performance criteria selected for comparison. Throughput is defined as the number of packets transmitted during the simulation. Average voice packet delay is defined as average queuing delay in each node and is expressed in frame duration units (10ms). Figures 6, 7, 8 and 9 depict the performance characteristics as a function of MUD capacity that limits the number of simultaneously received CDMA signals and, therefore, also limits the number of transmission neighbors of each node.

Figures 6 and 7 provide a comparison of voice packet delay between our clique model (CM) with STFQ and TOP metrics and TOP implementation presented in [2] and [3]. MUR-STFQ-CM and MUD-STFQ-CM cases provide a significant gain in the delay performance when compared with MUR-TOP and MUD-TOP cases. Also MUR-TOP-CM and MUD-TOP-CM cases give a tangible although smaller gain. In particular the delay in MUR-STFQ-CM algorithm is improved by around 10 % whereas the gain in MUR-TOP-CM case is around 2.6 %. Moreover, in the MUD scenario, the MUD-STFQ-CM algorithm improvement reached 11 % while in MUD-TOP-CM case it reached 3 %. The results indicate also a very good performance of the MUD approach when compared with the MUR case. This confirms the previously stated claim that MUD based MAC can significantly improve QoS for multi-media applications.

Figures 8 and 9 present a comparison of total throughput between our clique model (CM) with STFQ and TOP metrics and TOP implementation presented in [2] and [3]. The total throughput performance indicates that the gain in voice packet average delay for STFQ is achieved at the price of slightly reduced throughput. This result shows that there is a tradeoff between priority fairness and throughput in the considered system. On the other hand, the difference between the MUD and MUR throughputs results from the reduced interference in case of MUD, since both schemes have the same spectrum reuse mechanism. The throughput reduction in MUR-STFQ-CM algorithm is about 5 % and in MUD-STFQ-CM is about 4.4 % when compared with TOP cases, respectively.

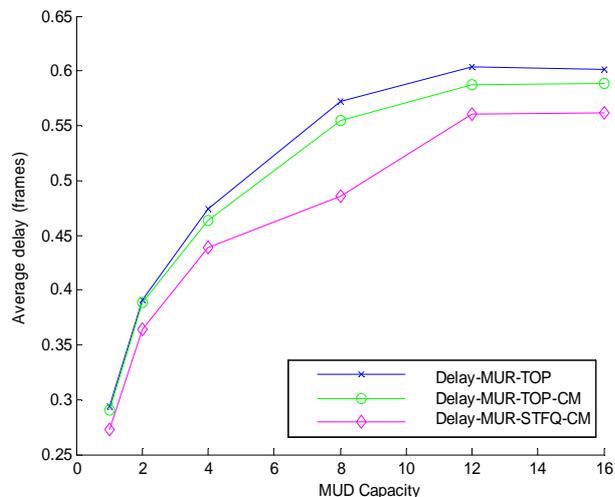


Figure 6 Average transmission delay in MUR

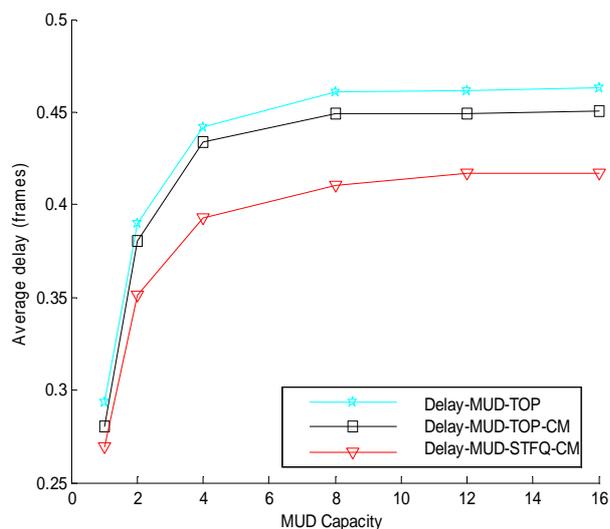


Figure 7 Average transmission delay in MUD

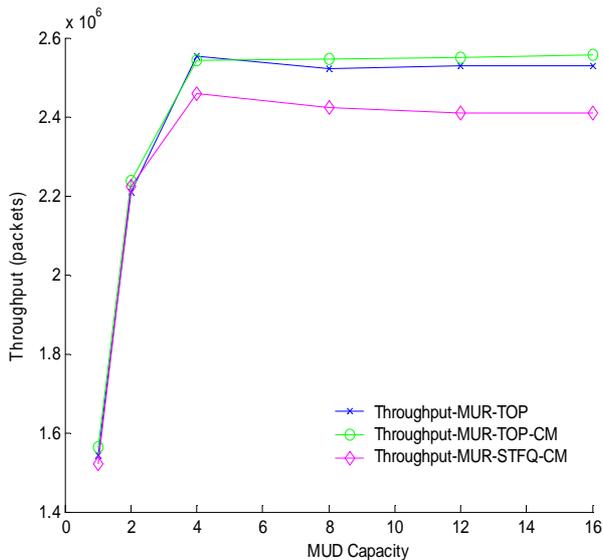


Figure 8 Total packet throughputs in MUR

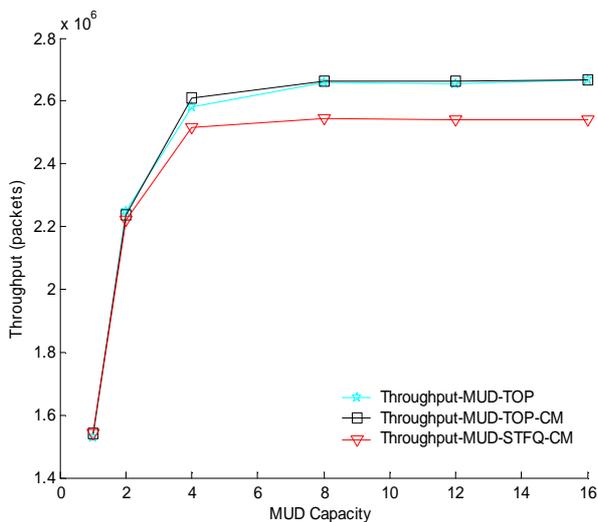


Figure 9 Total packet throughputs in MUD

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we addressed the issue of scheduling optimization for wireless ad hoc networks with MAC based on MUD. MUD can give significant gains in throughput and QoS performance. Nevertheless, achieving these gains requires optimization of distributed neighborhood scheduling that provides the desired performance objectives. Therefore, an approach for analyzing and comparing optimal or suboptimal distributed neighborhood scheduling schemes with different objectives was proposed. The approach is based on the flow and flow dependence matrices that are used to create a set of possible scheduling configurations. Then, the selection of configurations used

for transmissions is based on the chosen scheduling objective. We demonstrated the viability of this approach by implementing a scheduling scheme that utilizes STFQ algorithm and by comparing its performance to a scheduling algorithm based on timeout priorities. The numerical results showed that the STFQ implementation can improve the voice packets' average delay at the expense of some reduction in the total network throughput.

Future work is undergoing in several directions. The proposed approach assumes that each node receives scheduling information from all other nodes, which is difficult to achieve in each frame. Therefore, we plan to augment the one hop signaling protocol developed in [2] and [3] to a two-hop protocol that should be sufficient for the proposed approach. At the same time, reduction of the scheduling signaling loads by exchanging information with larger cycle than one frame and only when needed is being considered. Another important direction that is being addressed is multi-objective scheduling. Here we plan to implement game theory approach to efficiently balance the delay, throughput, and fairness objectives.

REFERENCES

- [1] S. Kumar Sarkar, T.G. Basavaraju, C. Puttamadappa. "Ad hoc Mobile Wireless Networks, Principles, protocols, and application," Auerbach Publications edition, 2008.
- [2] J. Zhang, Z. Dziong, F. Gagnon, M.Kadoch, "Multiuser Detection Based MAC Design for Ad Hoc Networks," accepted in IEEE Transactions on Wireless Communications (May 5, 2008).
- [3] J. Zhang, Z. Dziong, M. Kadoch, F. Gagnon, "Performance Evaluation of A Multiuser Detection Based MAC Design for Ad Hoc Networks," Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th Volume, Issue, Sept. 30 2007-Oct. 3 2007 Page(s):294 – 298.
- [4] L. H, Medvedev. P, Cheng, J, Lu. C,"A Self-Coordinating Localized Fair Queueing in Ad Hoc Wireless Networks," IEEE Transactions on Mobile Computing, vol., vol. 3, NO. 1, January/March 2005.
- [5] L. H, P, LU. S, "A topology-Independent Wireless Fair Queueing in Ad Hoc Wireless Networks," IEEE Journal on Selected AREAS In communications, Vol. 23. No 3, March 2005.
- [6] N. Vaidya, A. Dugar, S. Gupta, and P. Bahl,"Distributed Fair Scheduling in a Wireless LAN," IEEE Transactions on Mobile Computing, vol. 4, NO. 6, November/December 2005.
- [7] J. Zhang, J.M.Mark, and X. Shen, "An adaptive handoff priority scheme for wireless MC-CDMA cellular networks supporting realtime multimedia applications," Computer Communications, vol. 28, pp. 1240–1250, 2005.
- [8] K. T. Jin and D. H. Cho, "Multi-code MAC for multi-hop wireless Ad Hoc networks," in Vehic. Tech. Conf. IEEE, vol. 2, pp. 1100–1104, 2002.
- [9] Z. Xie, R. T. Short, and C. K. Rushforth, "A family of sub optimum detectors for coherent multi-user communications," IEEE Journal on Selected Areas in Communications, vol. 8, pp. 683–690, May 1990.
- [10] A. K. Parekh, R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," IEEE/ACM Transactions on Networking (TON), vol. 1, no. 3, pp. 344–357, 1993.
- [11] J. Zhang and J. M. Mark and X. Shen,"An adaptive resource reservation strategy for handoff in wireless cellular CDMA networks", Can. J. Electr.Comput. Eng., vol. 29, no. 1/2, pp. 77–83, Jan./Apr. 2004.