# Fair Scheduling of Real-time Traffic over Wireless LANs

Maria Adamou, Sanjeev Khanna, Insup Lee, Insik Shin, Shiyu Zhou
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
email:{adamou, sanjeevk, lee, ishin, shiyu}@saul.cis.upenn.edu

*Abstract*—

**With the advent of the IEEE 802.11 wireless networks that provide high speed connectivity, demand for supporting multiple real-time traffic applications over wireless LANs has been increasing. A natural question is how to provide fair resource allocation to real-time traffic in wireless LANs. Wireless networks are subject to unpredictable location-dependent error bursts, which is different from wired networks. We propose a fair real-time scheduling model whose goal is to minimize the maximum loss rate among all the flows and additionally to maximize the overall throughput. We develop algorithms for this scheduling model that takes the wireless channel error characteristics into consideration.**

## I. Introduction

Wireless communication technology has gained widespread acceptance in recent years. The IEEE 802.11 standard [7] has geared wireless local area networks (LANs) into greater use. With the advent of the IEEE 802.11a[8] and 802.11b[9] supplements that provide high speed connectivity, the demand for supporting multiple time-critical high-bandwidth traffic applications, such as video-on demand and interactive multimedia, in wireless LANs has been increasing. This raises the issue of how to provide fair resource allocation to real-time traffic in wireless LANs.

Many real-time packet scheduling and fair packet scheduling algorithms have been developed for wired networks. However, it it not obvious how well these algorithms work for wireless networks, since wireless channels are subject to unpredictable *location-dependent* and *bursty* errors, which is different from wired networks. In this paper, we consider the problem of fair scheduling of real-time packets with deadline constraints over wireless LANs, taking the characteristics of wireless channel errors into consideration.

There has been previous work on providing QoS guarantee over wireless links [3], [4], [5], and on adapting packet fair scheduling algorithms to a wireless domain taking care of location dependent error bursts [10], [11], [13]. To our knowledge, however, there is no previous work that addresses the issues of deadline-based real-time scheduling considering fairness and wireless channel errors.

This paper is organized as follows: Section II describes our scheduling model and algorithms, and Section III presents the simulation results of the algorithms. Section IV provides some theoretical results that give an insight into the difficulty of finding good schedules in the presence of unpredictable errors, and Section V concludes.

This research was supported in part by NSF CCR-9988409, NSF CISE-9703220, ARO DAAG55-98-1-0393, ARO DAAG55-98-1-0466, and ONR N00014-97-1-0505 (MURI).

## II. Fair Real-time Scheduling over Wireless LANs

### A. Scheduling Model

We consider a packet-switched wireless network that consists of multiple cells. Each cell is assumed to consist of a base station (BS) and multiple mobile hosts (MHs) that have real-time data packets to transmit. The BS performs the scheduling of the packet transmissions by polling the MHs. Each tranmission is either uplink (from a MH to the BS) or downlink (BS to MH) and is modeled as a flow $f_i, 1 \leq i \leq N$. Each flow is associated with a channel which, at any given time $t$, can be in one of two states, namely, *error state* or *error-free state*. A flow being in error state means that either the source or the destination for the associated channel experience an error burst. For simplicity, we assume that the packets are of fixed size $L$ and are generated with a specific rate. Each flow, $f_i$, is also characterized by three parameters $(v_i, d_i, e_i)$, which are made known to the BS:

- $v_i$ is the interval between the arrival time of two successive packets,
- each packet must be scheduled within $d_i$ time units from its generation at the source, and
- $e_i$ is the maximum acceptable packet loss rate.

Let $M_i$ be the total number of packets that flow $f_i$ is supposed to transmit and $M_i^a$ be the number of packets that actually get transmitted successfully. Then the throughput for flow $f_i$, say $t_i$, is defined to be the ratio of $M_i^a$ to $M_i$, and the overall throughput is $\frac{\sum_{i=1}^{N} M_i^a}{\sum_{i=1}^{N} M_i}$. We call $\epsilon_i = 1 - t_i - e_i$ the *loss* rate for flow $f_i$. Let $\epsilon_{max}$ be the maximum $\epsilon_i$ for $1 \leq i \leq N$. Thus each flow transmits at least $\lceil M_i(1 - e_i - \epsilon_{max}) \rceil$ packets. The objective of our scheduling algorithm is to maximize the overall throughput subject to minimizing $\epsilon_{max}$. In other words, our goal is to determine the smallest $\epsilon_{max}$ for which we can ensure that $M_i^a \geq \lceil M_i(1 - e_i - \epsilon_{max}) \rceil$ and maximize $\sum_{i=1}^{N} M_i^a$. We measure the throughput of our scheduling algorithm as the overall throughput and its fairness as $\epsilon_{max}$.

### B. Scheduling Algorithms

Each packet $p_i^k$ of flow $f_i$ is associated with an arrival time $A(p_i^k)$, at which it arrives or is ready to be transmitted and a deadline $D(p_i^k)$, beyond which $p_i^k$ cannot be scheduled. A packet is considered lost (dropped) if its deadline expires and was not trasmitted successfully. We consider the following scheduling algorithms:

- Earliest Deadline First (EDF)

The packet with the earliest deadline is scheduled first. At time $t$, a packet $p_i^k$ is scheduled such that $A(p_i^k) \leq t$ and $D(p_i^k)$ is the minimum among the deadlines of the packets to be scheduled. This policy is known to give the maximum overall throughput when there is no channel error.

- Greatest Loss First (GLF)

The flow that has the greatest loss rate is scheduled first. At time $t$, a packet $p_i^k$ is scheduled such that $A(p_i^k) \leq t$ and $\epsilon_i = \epsilon_{max}$. Intuitively, this is expected to result in minimizing $\epsilon_{max}$.

- Hybrid

To achieve simultaneously both the above goals, we consider a hybrid algorithm. At time $t$, it considers among all the unscheduled packets $p_i^k$ with arrival time $A(p_i^k) \leq t$, those that have deadline $D(p_i^k) = t + 1$. If such packets exist, it schedules the flow $f_i$ with the greatest loss rate among the packets. Otherwise, it chooses according to GLF.

### C. *Wireless Error Handling*

The presence of errors raises the issue of how to handle the situation where a flow is scheduled, but failed its packet transmission due to channel error. It is desirable to re-schedule the packet sometime later so that it will not miss its deadline. Since errors are unpredictable, however, it is not clear how long the re-scheduling of the packet should be delayed. We call this re-scheduling delay *backoff time*.

We define the re-scheduling policy according to the backoff time, $b_i$, for each flow $f_i$, such that it is long enough to escape a potential bursty error.

$$b_i = d_i / 2^{n_i}, \tag{1}$$

where $n_i$ is is the number of failed transmissions, since the last succesful transmission, for flow $f_i$.

Initially $b_i = 0$, that assume that all flows have initially an error-free channel. Using the polling mechanism, the channel condition is probed. If it is bad, a different flow is scheduled. If a packet transmission fails, or is deferred, the backoff mechanism is used to re-schedule the packet before its deadline. A flow $f_i$ is polled at time $t$ only if $b_i < t$, which means that its channel is predicted to be error-free at that time.

## III. Simulation Results

### A. *Simulation Environment*

All the simulations were done in $ns$, which is a discrete event simulator developed by the VINT project at the University of California at Berkeley [6]. The CMU Monarch extensions [2] added wireless and mobility supports to $ns$, including a CSMA/CA medium access mechanism defined in the IEEE 802.11. We added the 802.11 point coordination fuction (PCF) protocol to simulate our algorithm over the PCF protocol.

The capacity of the wireless medium is 11 Mbps following IEEE 802.11b. Each simulation time is 60 seconds.

We used the following error model. For each flow $f_i$, we created blackout period $b_i^k$ where $1 \leq k \leq M$. Any packet transmission for flow $f_i$ that is overlapped with a blackout period $b_i^k$ is considered as a failed transmission. The duration of each $b_i^k$ is uniformly chosen in [2.5, 15] msec. The error duration rate is the ratio of $\sum_{k=1}^{M} b_i^k$ to the total simulation time.

A real-time traffic flow is modeled to have a deadline of 10 msec, a packet interval of 15 or 20 msec, and a packet length of 1460 bytes. Simulations have been performed with 12 flows to compare EDF, GLF, and Hybrid algorithms with varying error duration rates.

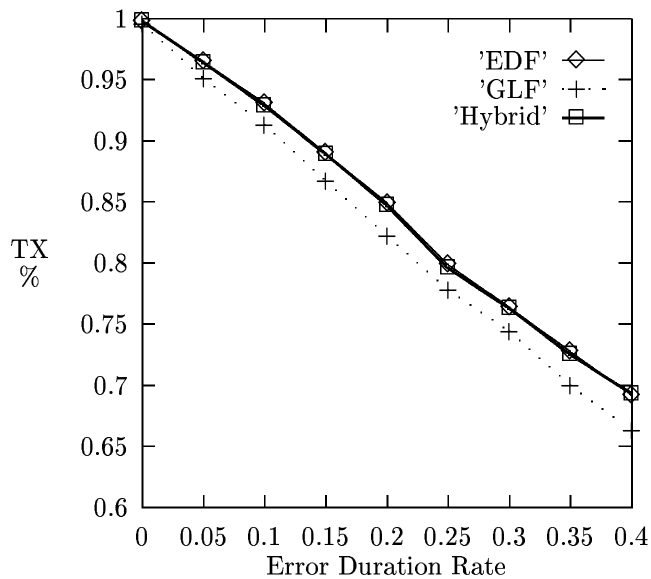


Fig. 1. Overall Throughput ($\frac{\sum_{i=1}^{N} M_i^a}{\sum_{i=1}^{N} M_i}$)

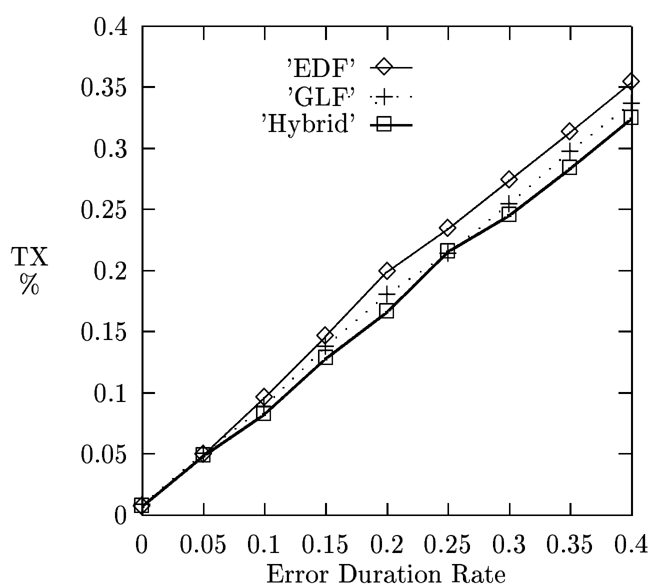


Fig. 2. Maximum Loss Rate ($\epsilon_{max}$)

### B. *Simulation Results*

Figure 1 plots the overall throughputs of the various algorithms. The firgure shows that EDF provides a better throughput

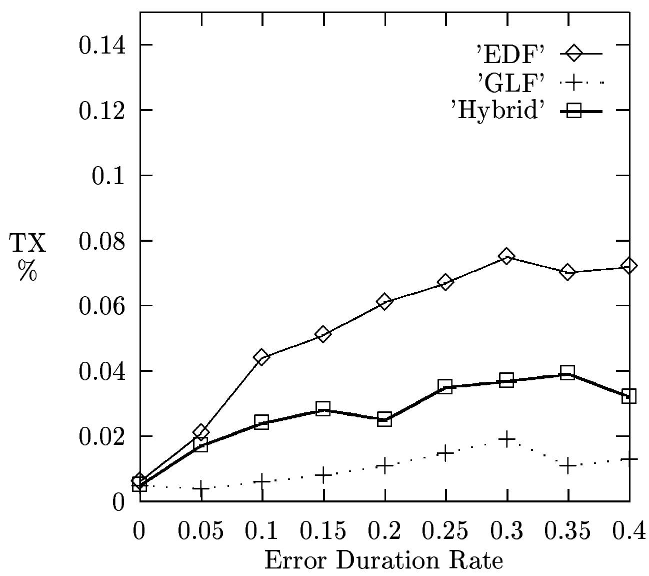


Fig. 3. Maximum of $|\epsilon_i - \epsilon_j|$

than GLF. It is interesting to see that Hybrid generates a comparable throughput with EDF.

Figure 2 plots the maximum loss rate among all flows, $\epsilon_{max}$. While it is expected that GLF would provide a smaller maximum loss rate than EDF, it is surprising that Hybrid provides a smaller $\epsilon_{max}$ than GLF. Our possible explanation for this phenomenon is the higher throughput of the Hybrid algorithm.

Figure 3 shows the maximum difference among the loss rates of all flows, $max(|\epsilon_i - \epsilon_j|)$. According to the figure, GLF provides the smallest value of $max(|\epsilon_i - \epsilon_j|)$, EDF provides the largest value, and Hybrid provides medium values.

## IV. Theoretical Results

We now establish some theoretical results that give an insight into the dificulty of finding good schedules in presence of errors. We start with some simple results concerning the worst-case behaviour of online algorithms for our problem. We measure the performance of any online algorithm using the competitive analysis framework of Sleator and Tarjan [12]. We say that an online algorithm is $c$-competitive if on any input sequence, it is guaranteed to produce a solution that is at least $1/c$ times as good as an optimal solution. Thus a 1-competive algorithm gives essentially an optimal solution itself. The following proposition can be easily proven.

*Proposition 1.* For any $\epsilon > 0$, there does not exist a $(2-\epsilon)$-competitive online algorithm for throughput maximization even when the system contains only two hosts.

In fact, the lower bound above is tight for throughput maximization.

*Proposition 2.* There exists a 2-competitive online algorithm for throughput maximization.

Given the dificulty of the online case even for the simpler goal of throughput maximization, a natural question to ask is if the problem remains intractable when the errors are all known in advance (i.e., an offline setting). We next show that this case is essentially equivalent to the maximum flow problem, well-known to be solvable efficiently in polynomial time (see [1], for instance).

*Theorem 1:* There is a polynomial time offline algorithm that determines an optimal fair schedule with maximum throughput.

*Proof.* We will reduce our problem to the maximum flow problem. For any $\epsilon > 0$, let $M_i^\epsilon = \lceil M_i(1 - e_i - \epsilon) \rceil$ where $1 \leq i \leq N$. Roughly speaking a schedule is said to be an $(\epsilon, \alpha)$-*schedule* if (i) it schedules at least $M_i^\epsilon$ packets for each flow $i$, and (ii) it sends at least $\sum_{i=1}^{N} M_i^\epsilon + \alpha$ packets overall. Our goal is to determine a pair $(\epsilon^*, \alpha^*)$ such that (i) for any $\epsilon > \epsilon^*$ there does not exist an $(\epsilon, 0)$-schedule, and (ii) for any $\alpha > \alpha^*$, there does not exist an $(\epsilon^*, \alpha)$-schedule. We will construct an instance of the maximum flow problem for every candidate pair $(\epsilon, \alpha)$ and output the solution corresponding to the best such pair found. Let $p$ denote $\sum_{i=1}^{N} M_i$, the total number of packets. It is easy to verify that there are only $O(\log p)$ candidate pairs need to be considered. We first use binary search over the set $\{1/p, 2/p, ..., 1\}$ to identify the smallest $\epsilon$ for which an $(\epsilon, 0)$-schedule exists. Then another binary search on the set $\{1, 2, ..., p\}$ determines the largest $\alpha$ for which this schedule stays feasible. Thus from here on, we assume without loss of generality that we know the optimal values of $\epsilon$ and $\alpha$.

We construct a directed graph $G = (V, E)$ as follows. The vertex set of $G$ contains two special vertices, namely a source $s$ and a sink $t$, vertices $s_1, ..., s_N$ for each of the $N$ flows, a vertex $w$, vertices $u_1, ..., u_p$ corresponding to the packets generated by the various flows, and vertices $v_1, ....., v_T$ corresponding to the various time slots that are available for scheduling packets. There is a directed edge from a vertex $u_i$ to a vertex $v_j$ if the packet corresponding to $u_i$ could be scheduled at the $j$th time slot (i.e., the corrsponding flow is in good state and the deadline of the packet has not yet expired). There is an edge from any vertex $s_i$ to a vertex $u_j$ if $u_j$ corresponds to the packet generated by the $i$th flow. There are edges from each vertex $v_j$ to the vertex $t$. All edges described thus far have a capacity of 1 on each edge. Finally, there are edges from the source vertex $s$ to each of the $s_i$'s as well as to the vertex $w$. The vertex $w$ is connected to each $u_i$'s by an edge of capacity 1. The directed edge $(s, s_i)$ has capacity equal to $M_i^\epsilon$ and the edge $(s, w)$ has capacity equal to $\alpha$.

It is now an easy consequence of our construction that there exists an $s$-$t$ flow of value $\sum_{i=1}^{N} M_i^\epsilon + \alpha$ if and only if there exists an $(\epsilon, \alpha)$-schedule. □

## V. Conclusion

In this paper, we propose a fair real-time scheduling model whose goal is to maximize the overall throughput subject to minimizing the maximum loss rate among all the flows and develop algorithms for the scheduling model that take wireless channel error characteristics. We are currently refining and evaluating these algorithms.

## References


[1] R.K.Ahuja, T.L. Magnanti, and J.B.Oslin, *Network Flows*, Prentice-Hall, Englewood Cliffs,1993

[2] *The CMU Monarch Project's Wireless and Mobility Exntensions*

*to $ns$*, The CMU Monarch Project. August 1999. Available from http://www.monarch.cs.cmu.edu/.

[3] J. Capone and I. Stavrakakis, *Delivering Diverse Delay/Dropping QoS Requirements in a TDMA Environment,* ACM/IEEE MOBICOM Conference, September 1997.

[4] C. Chang, K. Chen, M. You, and J. Chang, *Guaranteed Quality-of-Service Wireless Access to ATM Networks,* IEEE Journal of Selected Areas in Communications, Vol. 15, No. 1, January 1997.

[5] S. Choi and K. G. Shin, *A Unified Wireless LAN Architecture for Real-time and Non-real-time Communication Services,* ACM Transaction on Networking, Febrary 2000.

[6] Kevin Fall and Kannan Varadhan, editors. *$ns$ Notes and Documentation*, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997. Available from http://www-mash.cs.berkeley.edu/ns/.

[7] IEEE, *IEEE Std 802.11 - Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications,* 1997.

[8] IEEE, *IEEE Std 802.11a - Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications: High Speed Physical Layer(PHY) in the 5 GHZz Band,* 1999.

[9] IEEE, *IEEE Std 802.11b - Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications: High Speed Physical Layer(PHY) in the 2.4 GHZz Band,* 1999.

[10] S. Lu, V. Bharghavan, and R. Srikant, *Fair Scheduling in Wireless Packet Networks,* IEEE/ACM Transactions on Networking, Vol. 7, No. 4, August 1999.

[11] S. Lu, T. Nandagopal, and V. Bharghavan, *A Wireless Fair Service Algorithm For Packet Celluar Networks,* ACM/IEEE MOBICOM Conference, October 1998.

[12] D.D. Sleator and R.E.Tarjan, *Amortized efficiency of list update and paging rules* Communications of the ACM, 28(2): 202-208, 1985

[13] T. Ng, I. Stoica, and H. Zhang, *Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Error,* ACM/IEEE MOBICOM Conference, October 1998.