IEEE P802.11
Wireless LANs

|  |
| --- |
| Example Code for Deterministic Backoff |
| Date: 2017-11-06 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Menzo Wentink | Qualcomm | Utrecht, The Netherlands | +31 65 183 6231 | mwentink@qti.qualcomm.com |

Abstract

This document specifies example Matlab code for core Deterministic Backoff routines at a contending node (STA or AP). This code itself is not executable, but it provides routines which need to be inserted at the cited locations in an 802.11 EDCA implementation.

**% Example Matlab code for Deterministic Backoff**

% Menzo Wentink

% Qualcomm

% October 23, 2017

% This document specifies example Matlab code for core Deterministic Backoff routines at a contending node (STA or AP).

% This code itself is not executable, but it provides routines which need to be inserted at the cited locations in

% an 802.11 EDCA implementation.

**% 1. At initialization**

retrycount = 0; % deterministic backoff retry count (separate from src/lrc/ssrc/slrc/qsrc/qlrc/etc.)

interruptioncount = 0; % the number of interruptions during the current backoff

deterministicbackoff = 10; % holds the latest deterministic backoff, initialized at 10 slots

backoffisdeterministic = 0; % tells whether the current backoff is random or deterministic

% The following parameters already exist in an edca implementation, but are repeated here for clarity:

aifsn = 3; % aifsn in slots (deterministic backoff uses best effort parameters)

slottime = 9; % slot time in us for 5 GHz operation

sifstime = 16; % sifs time in us for 5 GHz operation

aifstime = sifstime + aifsn \* slottime; % aifs time in us

**% 2. When the backoff expires and the initial tx of the txop starts**

% Set parameters for the case in which no response is received (which will be overwritten when a response is received)

if retrycount < 7 % if less than 7 consecutive retries occurred

 retrycount = retrycount + 1; % increment the retry count

else

 retrycount = 0; % reset the retry count every 7th retry

end

if backoffisdeterministic == 1 % the past backoff was deterministic

 deterministicbackoff = 10 + interruptioncount; % set the next deterministic backoff

end

if retrycount > 2 % if more than 2 consecutive retries occurred

 backoff = randi(7) - 1; % select a random backoff between 0 and 6 slots

 backoffisdeterministic = 0; % the next backoff is random

else % if 2 or less consecutive retries occurred

 backoff = deterministicbackoff; % set a deterministic backoff

 backoffisdeterministic = 1; % the next backoff is deterministic

end

interruptioncount = 0; % reset the interruption count

backofftime = aifstime + backoff \* slottime; % set the next backoff time (already present in an edca implementation)

**% 3. When a response to the initial tx in the txop is received (e.g. cts, ack or ba)**

retrycount = 0; % reset the retry count

backoff = deterministicbackoff; % set a deterministic backoff

backoffisdeterministic = 1; % the next backoff is deterministic

backofftime = aifstime + backoff \* slottime; % set the next backoff time (already present in an edca implementation)

**% 4. At each cca busy event during a backoff (e.g. when the cca goes from idle to busy)**

if ccaidletime >= aifstime % the idle time leading up to this cca busy event was longer than or equal to aifs

 interruptioncount = interruptioncount + 1; % increment the number of interruptions for the current backoff

end