Integrating DFS into ath9k/mac80211

Introducing Neratec's Design Approach

Zefir Kurtisi   Wojciech Dubowik

Neratec Solutions AG, CH-8608 Bubikon
`http://www.neratec.com`

Linux Wireless summit @ LinuxCon North America 2011
August 15-16, Vancouver, Canada

About us

Who we are

The Neratec Solutions AG is

- a small R&D company located in Switzerland
- working in the area of Industrial Wireless
- strongly customer orientated

Open Source Development Strategy

Our system platform so far is

- OpenWRT based
- using Atheros' WiFi cards
- using proprietary WLAN drivers for
  - efficiency and stability
  - regulatory compliance

Motivation to join linuxwireless

Remaining step: move fully to Open Source by migrating to ath9k

Fact: Industrial Wireless demands for DFS
$\Rightarrow$ we need DFS supported by ath9k/mac80211, including

- certified DFS compliance (for ETSI)
- high performance and reliable operation in DFS channels

Outline

Duration

- approx. 30 min.

DFS Basics

Purpose

DFS (Dynamic Frequency Selection) allows WLAN devices to share the 5 GHz spectrum with radar devices.

DFS Channels

		DFS frequency ranges		
Frequency [MHz]	5160 - 5240	5250 - 5350	5470 - 5725	5745 - 5805
Channel IDs	36 - 48	52 - 64	100 - 140	149 - 161

Core Requirement

WLAN devices must not operate in channels that are used by radar devices.

Functional Requirements

To achieve this core requirements, master devices has to support

1. detecting radars
2. manage spectrum based on detections

Regulatory Domains

As of today, there are three regulatory domains

- FCC for North America
- JP for Japan (based on FCC)
- ETSI for Europe

They share the basic principles but differ slightly in the parameter space.

Note

In the remainder, ETSI is used as regulatory domain.

Detection Requirements

Master devices must be able to detect defined types of radar patterns.

Radar Test Signals

Radar patterns are specified as radar pulse sequences parametrized with

- Pulse Width W $[\mu s]$
- ETSI: Pulse Repetition Frequency PRF $[Hz]$
  FCC: Pulse Repetition Interval $PRI = PRF^{-1}$ $[\mu s]$
- Number of different PRFs
- Pulses Per Burst PPB

Example: single PRF ETSI radar test signals

Test Signal	Pulse Width	PRF	PPB
reference	1	700	18
1	$0.8 \dots 5$	$200 \dots 1000$	10
2	$0.8 \dots 15$	$200 \dots 1600$	15
3	$0.8 \dots 15$	$2300 \dots 4000$	25
4	$20 \dots 30$	$2000 \dots 4000$	20

Detection Requirements

Minimum Detection Probability

During certification, master devices must detect specified radar bursts

- under well defined conditions
- with a given minimum detection probability P_d
  ETSI: $P_d = 60\%$

Management Requirements

Channel State Handling

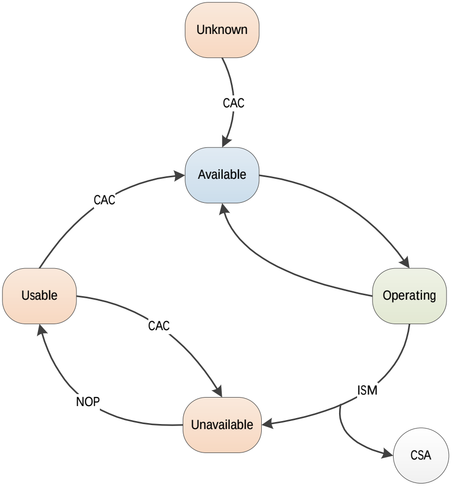
Master devices must keep track of DFS spectrum and channel states

Abbreviations

CAC	Channel Availability Check
ISM	In-Service Monitoring
NOP	Non-Occupancy Period
CSA	Channel Switch Announcement

Relevant ETSI Values

CAC Time	60 s
NOP	30 m
Channel Move Time	10 s
Channel Closing Transmission Time	1 s

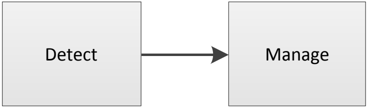


Design Concept

Architectural Considerations

DFS Architecture is given by regulatory requirements:

- common management part
- HW / driver specific radar detection



Managing Component

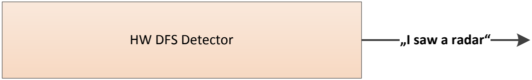
- announce DFS capabilities
- handle channel states (aka NOL handling)
- assure CAC
- on radar detection
  - perform CSA
  - switch to available channel

Detector Design

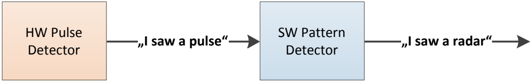
DFS Detector Types

Support for both types of existing devices

- radar detection fully supported by HW



- radar pulses detected in HW, pattern matching in SW (like Atheros)



Two-tier Approach

- common interface to management component
- split DFS detection into
  - chip or family specific pulse detector
  - (ideally) HW independent pattern detector

Pulse Detection

Functional Components

- set-up pulse detection
- evaluate reported pulses (post filtering)
- report pulse events to pattern detector

Challenges

- requires in-deep chipset knowledge
- documentation not publicly available
- requires tight co-operation with chip-designers for optimal performance

$\Rightarrow$ ideally to be done my manufacturer

Pulse Detection for ath9k

Approach taken

- worked out GPL-compliant NDA for Atheros' driver DFS source code
- port pulse detection chip set-up to ath9k
- interface pulse reporting to pattern matcher

Current Status

- chip set-up ported to and usable in ath9k
- some post-filtering ported for significant decrease of false pulses

Pattern Detection

Detection Complexity

The latest update from ETSI v1.4 to v1.5 introduced a leap for pattern detection complexity:

- v1.4: discrete values for pulse widths and PRI
- v1.5: ranges for widths and PRI

$\Rightarrow$ computational complexity grew from $O(n^2)$ to $O(n^3)$

Detection Factors

A DFS pattern detector must handle pulse patterns facing

- missing pulses
- false pulses
- limited timing accuracy

Detection Parameters

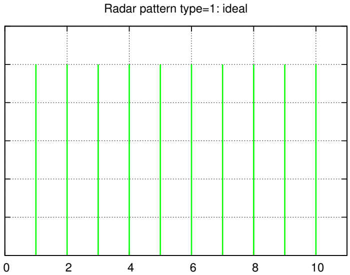
Challenge

Based on the detection factors

- identify pattern types to expect
- derive detection parameters

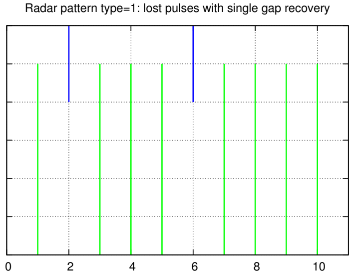
Reference

The following considerations will use ETSI radar tests signal 1 with 10 PPB as reference.



Detection Parameters: Missing Pulses

Closing single Gaps

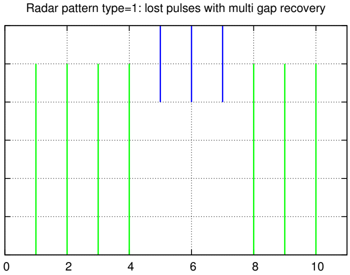


- many PRIs ⇒ high confidence

Detection Parameter

No. of single gaps to close / pattern

Closing multi Gaps



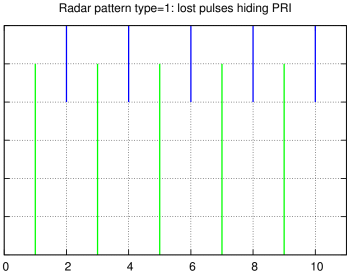
- multi PRI gap backed by PRIs

Detection Parameter

Max gap width to tolerate

Detection Parameters: Missing Pulses

Not closing PRI hiding Gaps

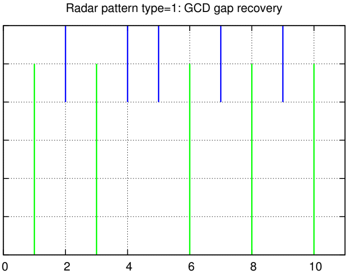


- no indication for PRI ⇒ recovery purely speculative

Detection Parameter

Enable speculation

Closing GCD Gaps

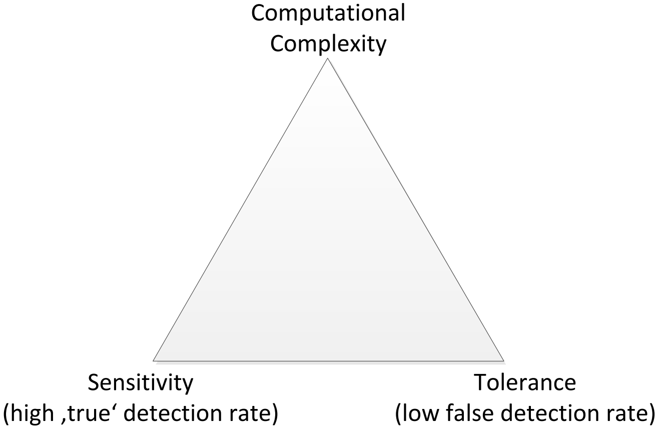


- different multi PRIs indicate PRI ⇒ highly speculative

Detection Parameter

Max GCD threshold

Pattern Detection: Bottom Line

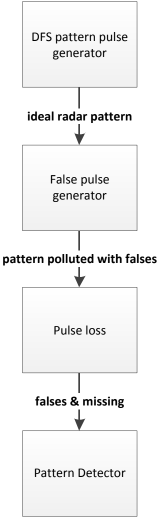


Testing: Statistical Detection Analysis

Evaluating DFS detection performance is done with statistical analyses.

Test Framework

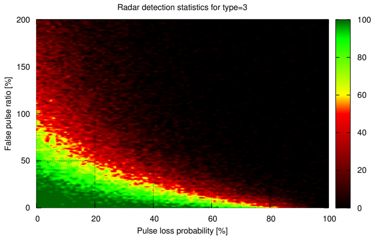
- aggregates detection statistics for test patterns
  - given radar test patterns
  - within a given parameter space
  - for a given detection algorithm
- supports visualization for quick inspection
- provides semi-automated detector parameter optimization



Testing: Example Results

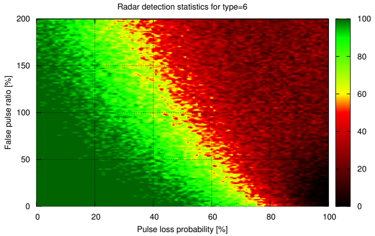
Well balanced Detector

- reliably detects more than 60 % with reasonable pulse loss rate
- robust against false pulses
- small false detection ratio



Over-sensitive Detector

- very high detection rate
- false pulses lead to unacceptable high false detections



Testing: Detectors' Regulatory Compliance

For DFS compliance we need the detector performance under defined conditions.

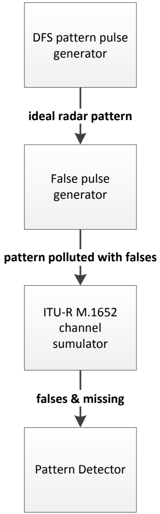
Channel Simulator (ITU-R M.1652)

The probability based pulse loss generator is replaced with an channel simulator, that

- calculates channel occupancy times based on statistical profiles for
  - listening times
  - sending times, based on model profiles for
    - frame sizes
    - data rates
- simulates DFS testing lab conditions

Output

- detector is / is not compliant



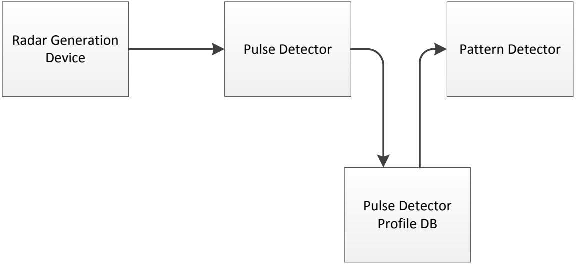
Testing: Closing the Loop

... and testing the real thing.

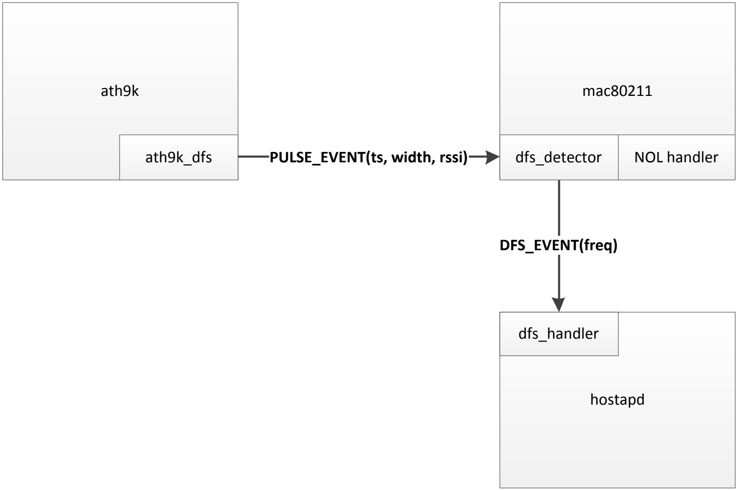
Problem

DFS generating devices not that affordable to the common Open Source developer.

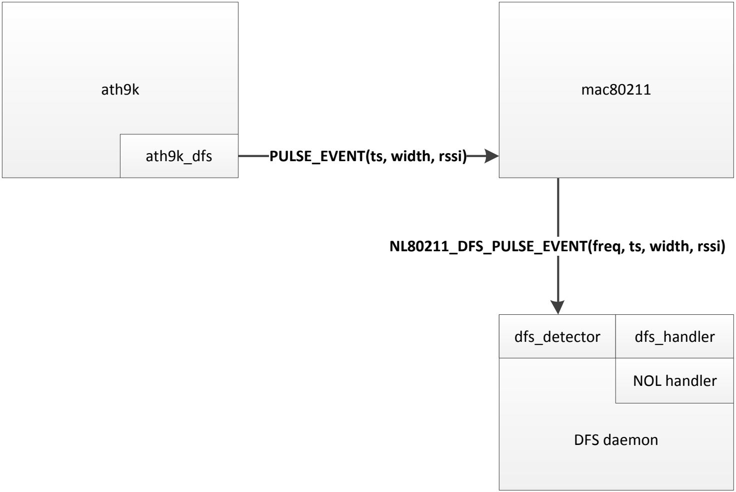
Decoupling Generator



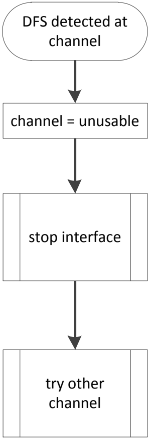
DFS Integration into ath9k/mac80211

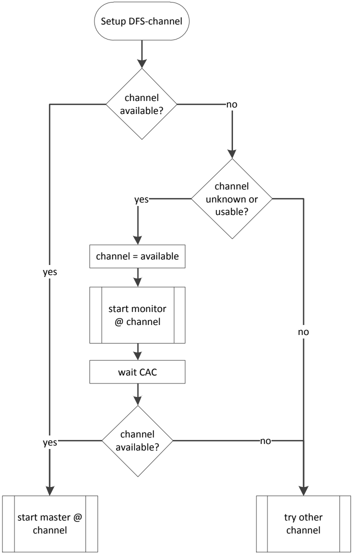


DFS fully controlled form User Space



Workflow of the DFS Daemon





Summary

Status Quo

- designed two-tier DFS detection
- implemented test framework supporting fully simulated and HW recorded profile processing for
  - statistical analyses of pattern detector performances
  - fine-tuning of detection parameter towards certification
- implemented and evaluated different classes of pattern detectors
- proposed proof-of-concept implementations for DFS with
  - pulse detection provided by ath9k and
  - pattern matching implemented as part of mac80211 or fully in user space

Future Work

- further studies and improvements on pattern detector
- focus on workable and certifiable DFS support
  - short term: finalize user space controlled DFS handling
  - long term: support community getting full DFS support into mac80211
- support testing by
  - recording and providing radar pulses detected from HW
  - share DFS compliance related information

Thank You!