OPEN-SOURCE PROTOTYPING OF ADVANCED WIRELESS SYSTEMS FOR SMART AGRICULTURE AND CONNECTED RURAL COMMUNITIES

TEAM MEMBERS - ROLES:

ZEQUN WANG - MEETING SCRIBE

JIAWEI DENG - CHIEF ENGINEER

SHAOHANG HU - TEST ENGINEER

ZHENGWEI SU - REPORT MANAGER

DYLAN SHARP - MEETING FACILITATOR

CLIENT:

IOWA STATE UNIVERSITY

ADVISORS:

PROF. HONGWEI ZHANG

DR. MATTHIAS SANDER-FRIGAU

Team: sdmay20-37 | Website: http://sdmay20-37.sd.ece.iastate.edu/

PROJECT VISION

- Overview

  We are going to use cutting-edge TV white space (TVWS) spectrum and advanced wireless algorithms. At the end of this project, we hope to accomplish an implementation of novel 5G wireless solution for smart agriculture and connected rural communities.

- Target Users

  Wireless engineers / developers

  With the wireless solution, customers in the smart agricultural industry will enjoy better connectivity

- Approach

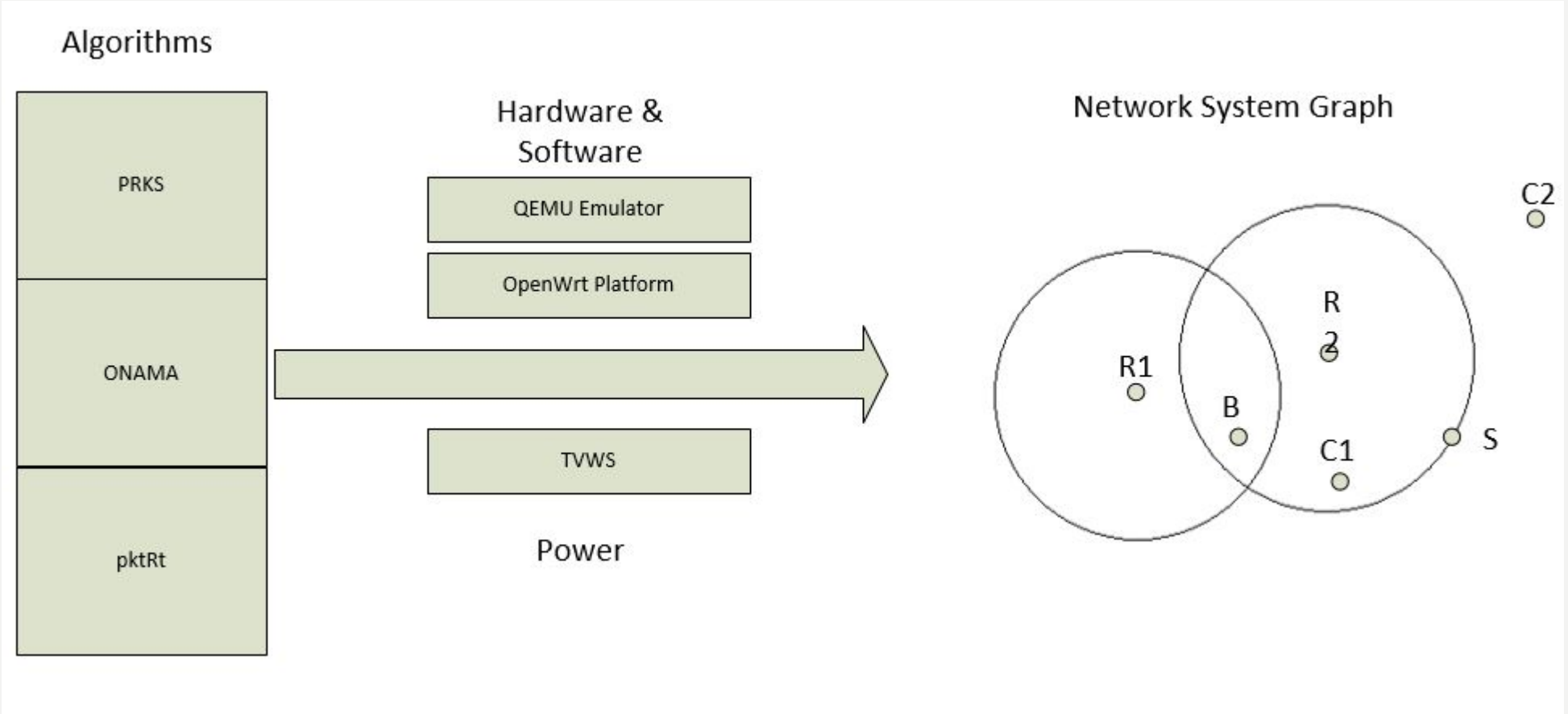
  Implement wireless algorithms (PRKS, pktRT, ONAMA) that aiming on enhancing link reliability, network throughput, decrease package loss rate, and have real-time packet guarantees.

ONAMA (Optimal Node Activation Multiple Access): Scheduling protocol for wireless networks that ensures maximal concurrency and low latency.

PRKS (Physical Ratio K Scheduling): Wireless network scheduling algorithm that ensures predictable link reliability.

pktRT: Probabilistic Per-Packet real-time guarantee algorithm for wireless networks.

CONCEPTUAL/VISUAL SKETCH



REQUIREMENTS

- Functional
  - Protocol must store a buffer of nodes
  - Each node in the exclusive region must update every 1 s
  - Every node must monitor the signal strength
  - Every node must store the local signal map
  - Value of K must update for each exclusive region every 1 s

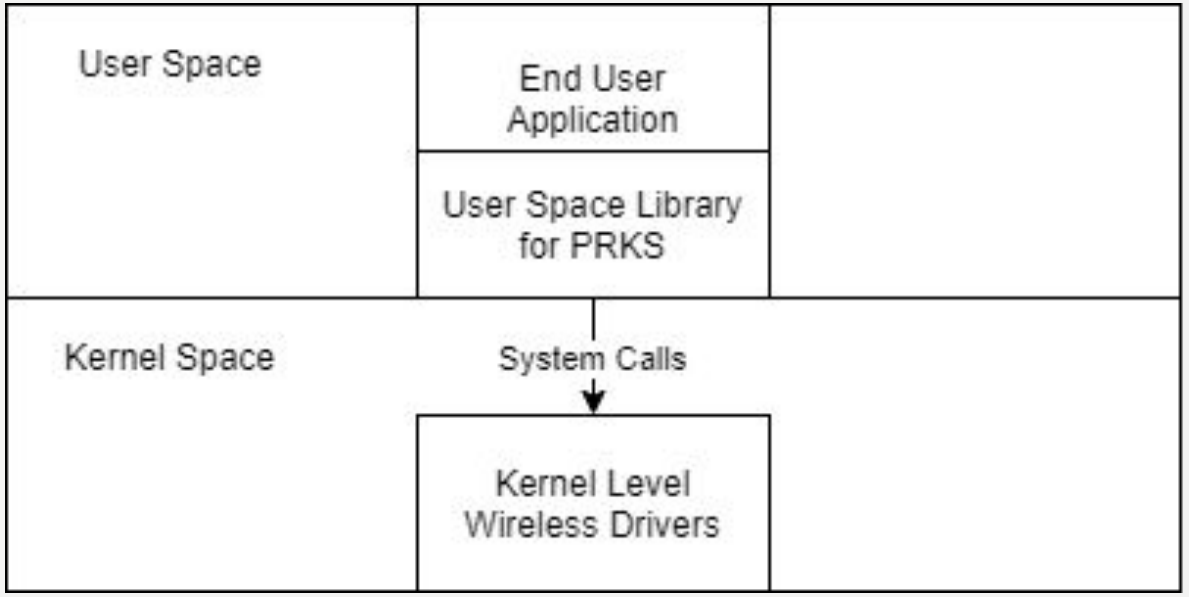
- Non-functional
  - Package loss rate should below 10%
  - Link reliability should be higher than 90%
  - Real time latency should be below 50 ms
  - Network throughput should be higher than 10 Mbs

REQUIREMENTS

- Limitations
  - Local unused channels within the TVWS Spectrum to operate on.
  - Hardware limitations (RAM, CPU Speed, etc.).
  - Any noise / interferences within the TVWS Spectrum.

CONCEPTUAL DESIGN DIAGRAM

- High level design approach
  - OpenWrt Linux OS



PROJECT PLAN : RISKS

- Technical Understanding
  - wireless networking
  - kernel development
  - advanced algorithms
- Team Availability
  - regular meeting

PROJECT PLAN : RISK MITIGATION

- Trello
- Assigning jobs
- Weekly advisor meeting
- Weekly team member briefing

SYSTEM DESIGN

PRKS algorithm architecture

This diagram displays the different modules within the PRKS algorithm and what relational arguments are passed from one module to another. All of these modules will be implemented and tested individually in a series like fashion. Note that instead of using TDMA, for scheduling we will implement the ONAMA scheduling algorithm.

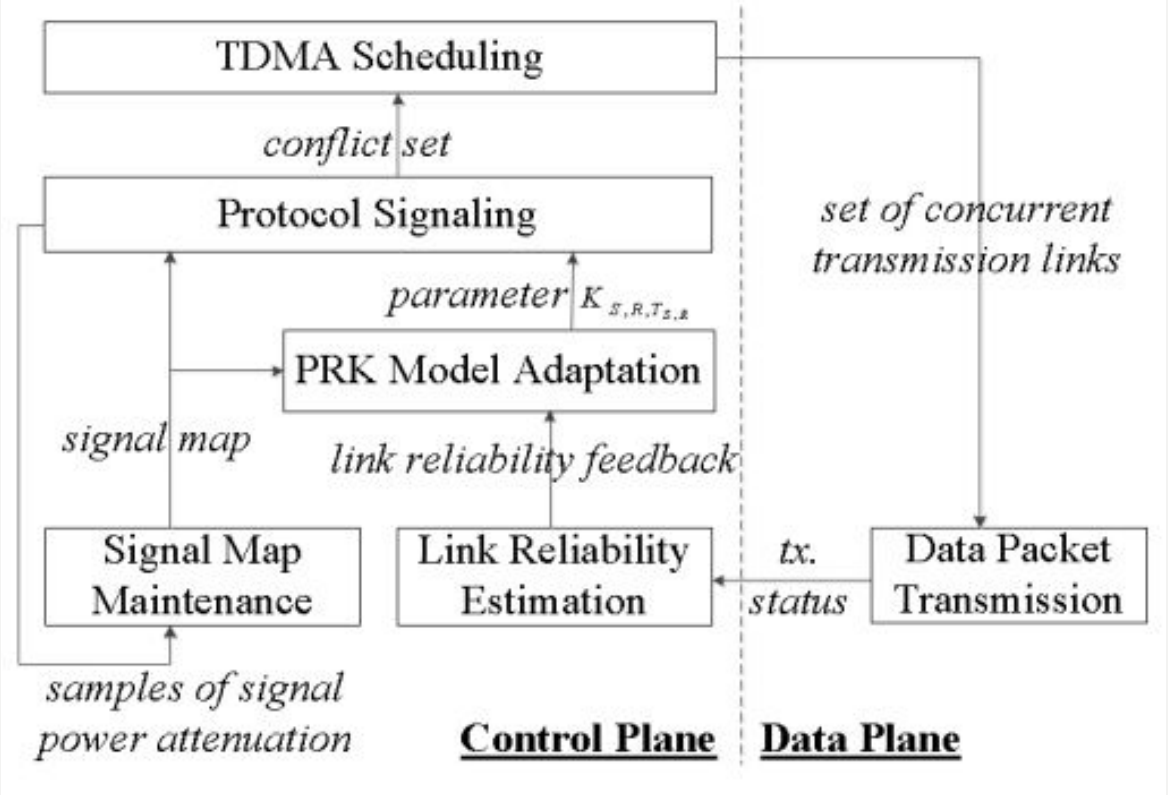


Fig. PRKS Architecture Diagram

SYSTEM DESIGN

The implementation will be within this Linux kernel IEEE 802.11 Wi-Fi system.

- Modification to the ath9k device driver to allow operation of IEEE 802.11af (an amendment to IEEE 802.11 for the use of the TVWS spectrum. This includes enabling the frequency bands needed for the TVWS spectrum.

- Within the mac80211 subsystem we will need to add a PRKS mode to support using the PRKS algorithm.

Fig. IEEE 802.11 Linux kernel implementation architecture

SYSTEM DESIGN

- Within the cfg80211 we will need to add a PRKS configuration mode to support our mac80211 PRKS mode.
- Add commands to nl80211 to add the support of joining and leaving the PRKS network on the TVWS spectrum.
- Edit iw (user space application) to support the new PRKS modes.

Fig. IEEE 802.11 Linux kernel implementation architecture

PROTOTYPE IMPLEMENTATIONS

- Pseudo Code
- Proof of Concept Implementation
  - Define implementation in January 2020.
  - Smart Agriculture Industry
  - Results will help prove field implementation of PRKS, pktRT, and ONAMA and non-functional requirements.

```
vector ex_region[] = [];  // Vector List of nodes currently in the exclusive region

int t; //time

Vector P[][] = [][]; //vector 2D array of average signal strength from node to Receiver at time t, example.
P[0][4] means average signal strength from node A to Receiver at time t = 5;

Vector K[] = []; //vector list of K over time



main() {
        t = 0;
        E[t] = get_exclusive_region[t];
        K[t] = 0;
        While(true)
        {
                T = t + 1;
                E[t] = get_exclusive_region(t);
                //when exclusive region expands
                If(E[t].size > E[t-1].size)
                {
```

Fig. Part of the Pseudo Code

PROJECT PLAN – MILESTONES

- OpenWrt Kernel Development Learning
- Algorithm Learning
- Design Phase
- Linux Implementation
- Proof of Concept Implementation

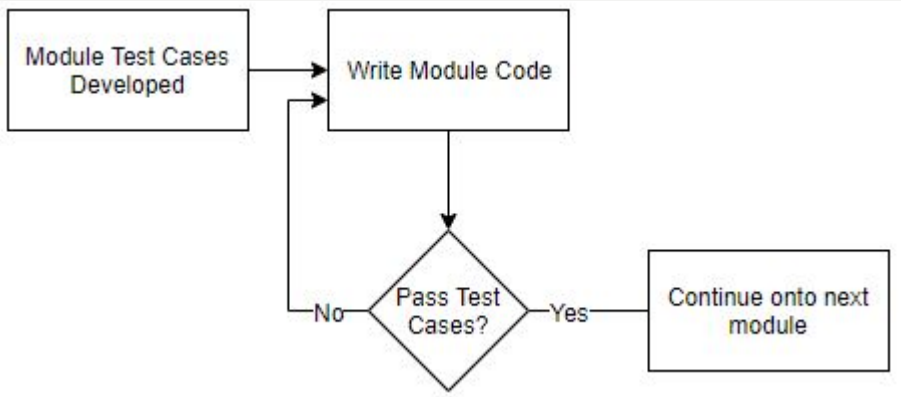
PROJECT PLAN – SCHEDULE

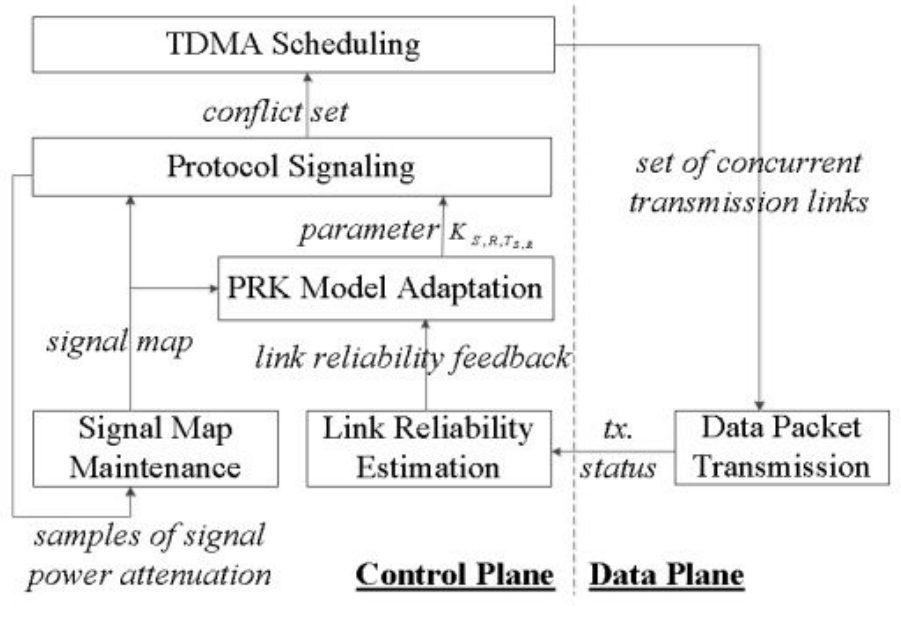
Task Name	Start Date	Finish Date	Estimated Hours
Semester 1			
OpenWrt Kernel Development Learning			
Virtualize OpenWrt	9/16/19	9/23/19	8
Cross Compile Hello World	9/16/19	9/30/19	10
Hardware Virtualization w/ QEMU	9/23/19	9/30/19	16
Linux Kernel Wifi System Understanding	10/21/19	10/28/19	25
Hello World Kernel Module	10/1/19	10/28/19	20
Algorithm Understanding (Learning)			
Define Algorithm Parameters	9/1/19	10/27/19	110
Map Algorithm Parameters to OpenWrt	10/28/19	11/18/19	16
Design			
Design Document	11/17/19	12/8/19	30
Proof of Concept Design Defined	1/13/20	1/20/20	8

PROJECT PLAN – SCHEDULE

Task Name	Start Date	Finish Date	Estimated Hours
Semester 2			
Implementation & Testing			
Linux Implementation	1/13/20	2/23/20	100
Functional Testing	1/13/20	2/23/20	30
Proof of Concept Implementation			
Proof of Concept Implementation	2/24/20	4/5/20	85
Functional Testing for Proof of Concept	2/24/20	4/5/20	30

TEST PLAN - FUNCTIONAL TESTING





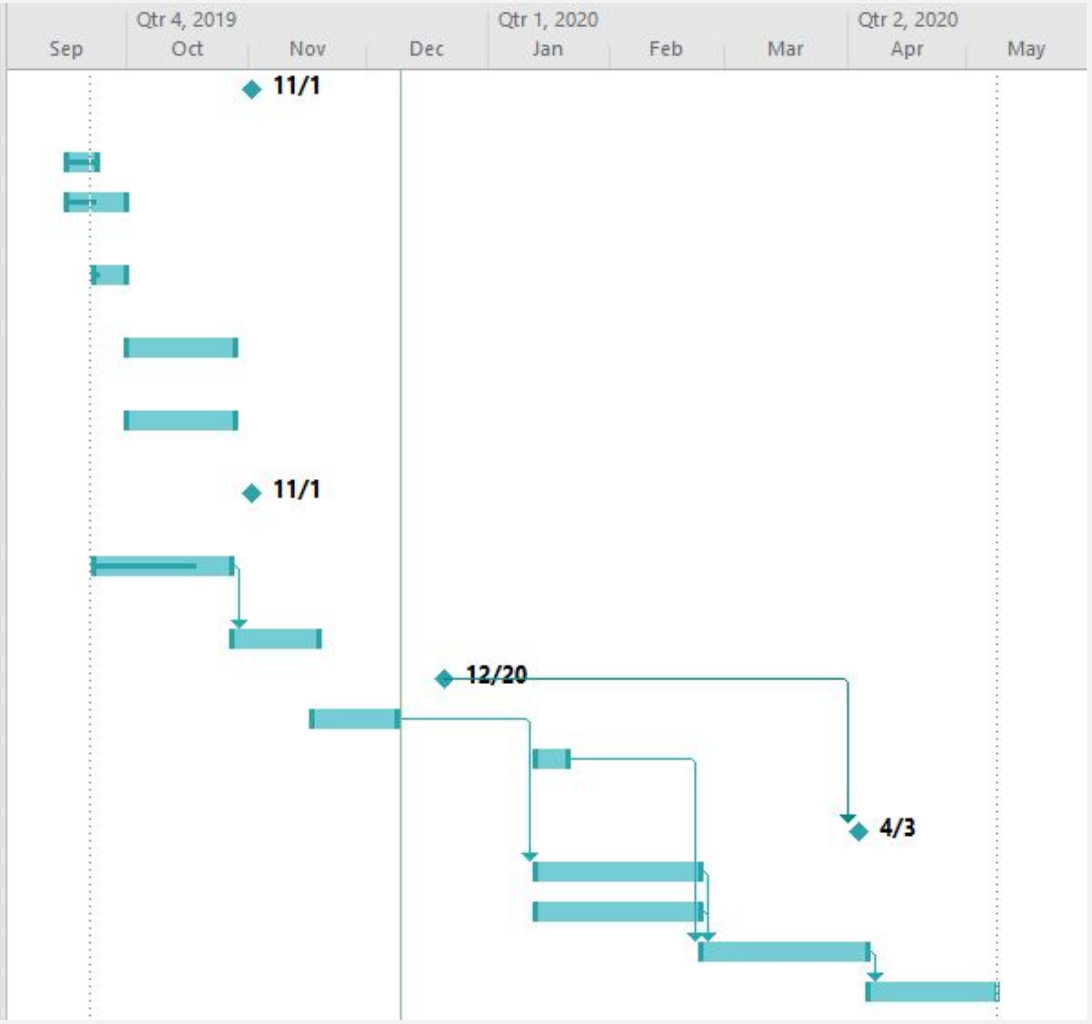
- Develop test cases for each module
- Implementing the code for each module
- Acceptance tests to determine whether module behaves well.
    - If it failed, go back to develop and try again.
    - If it past, continue to the next module.

TEST PLAN - NON-FUNCTIONAL TESTING

- Test under our proof of concept
  - System level acceptance tests
  - Print out debug statistics and aggregate results

CONCLUSION - CURRENT STATUS

Task Name	Start	Finish	Predecessors
OpenWrt Kernel Development (Learning)	Sun 9/1/19	Fri 11/1/19	
Virtualize OpenWrt	Mon 9/16/19	Mon 9/23/19	
Cross Compile Hello World	Mon 9/16/19	Mon 9/30/19	
Hardware Virtualization w/ QEMU	Mon 9/23/19	Mon 9/30/19	
Linux Kernel WiFi System Understanding	Tue 10/1/19	Mon 10/28/19	
Hello World Kernel Module	Tue 10/1/19	Mon 10/28/19	
Algorithm Understanding (Learning)	Sun 9/1/19	Fri 11/1/19	
Define Algorithm Parameters	Mon 9/23/19	Sun 10/27/19	
Map Algorithm Params to OpenWrt	Mon 10/28/19	Mon 11/18/1	8
Design	Mon 11/4/19	Fri 12/20/19	
Design Document	Sun 11/17/19	Sun 12/8/19	
Proof of Concept Design Finalized	Mon 1/13/20	Mon 1/20/20	
Implementation, Testing	Mon 1/6/20	Fri 4/3/20	10
Linux Implementation	Mon 1/13/20	Sun 2/23/20	11
Functional Testing	Mon 1/13/20	Sun 2/23/20	
Proof of Concept Implementation	Mon 2/24/20	Sun 4/5/20	14,15,12
Final Report and Demostration	Mon 4/6/20	Fri 5/8/20	16



CONCLUSION - FUTURE PLANS

- Finalize our proof of concept design
- Functional Implementation / Testing
- Proof of concept implementation / Testing
- Non-Functional Testing
- Final Report

QUESTIONS?

Linux WiFi Kernel Implementation

- **ath9k**: Wireless driver for Atheros WiFi chips
- **mac80211**: framework for SoftMAC wireless drivers. (SoftMAC is a type of WNIC where MLME is managed in software).
- **cfg80211**: Configuration API for mac80211
- **nl80211**: bridge between userspace and kernel space configuration API
- **iw**: wireless tool for configuring wireless devices

WNIC = Wireless Network Interface Controller

MLME = Media Access Control (MAC) Sublayer Management Entity

Fig. IEEE 802.11 Linux kernel implementation architecture

Algorithm Papers

PRKS: http://www.ece.iastate.edu/~hongwei/group/publications/PRKS-TWC.pdf

pktRT: http://www.ece.iastate.edu/~hongwei/group/publications/pktRT-TII.pdf

ONAMA: http://www.ece.iastate.edu/~hongwei/group/publications/ONAMA.pdf

PROJECT PLAN : TASKS (Semester 1)

- Algorithm understanding
  - pktRT
  - PRKS
  - ONAMA
- Algorithm Parameter mapping for each of the algorithm
  - Define each algorithm's parameters
  - Map each parameter to OpenWrt
- Algorithm pseudo code
  - Create pseudo code for each algorithm
- Virtualizing Development
  - Emulate Hardware
    - Set up QEMU to Emulate MIPS 24K devices with Atheros QCA9533 chip
  - Virtualize OS
    - Set up OpenWrt with QEMU above to setup for development

PROJECT PLAN : TASKS (Semester 2)

- Implementation
  - Edit kernel drivers to support the algorithms
  - Edit user level library to interact with kernel drivers for the algorithms
- Proof of concept
  - Define proof of concept application
  - Write application
  - Test application (potentially on physical devices)
- Testing
  - Non-functional tests
  - Functional tests