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

YULIN SONG - TEST 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 Plan

PROBLEM STATEMENT

Rural regions are home to many industries such as agriculture, renewable energy and manufacturing, and they are major sources of food and energy for our society.

However, rural regions are subject to growing challenges such as population shrinkage and labor shortage, the need to feed a growing population and food demand while subject to climate changes, and the rural-urban education gap.

Rural broadband is the digital superhighway for the broad rural communities and industries, but 39% of the rural US lacks broadband access, and most farms are not connected at all.

Through this project, we are going to develop and prototype advanced wireless solutions for smart agriculture and connected rural communities.

PROJECT OVERVIEW

- 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

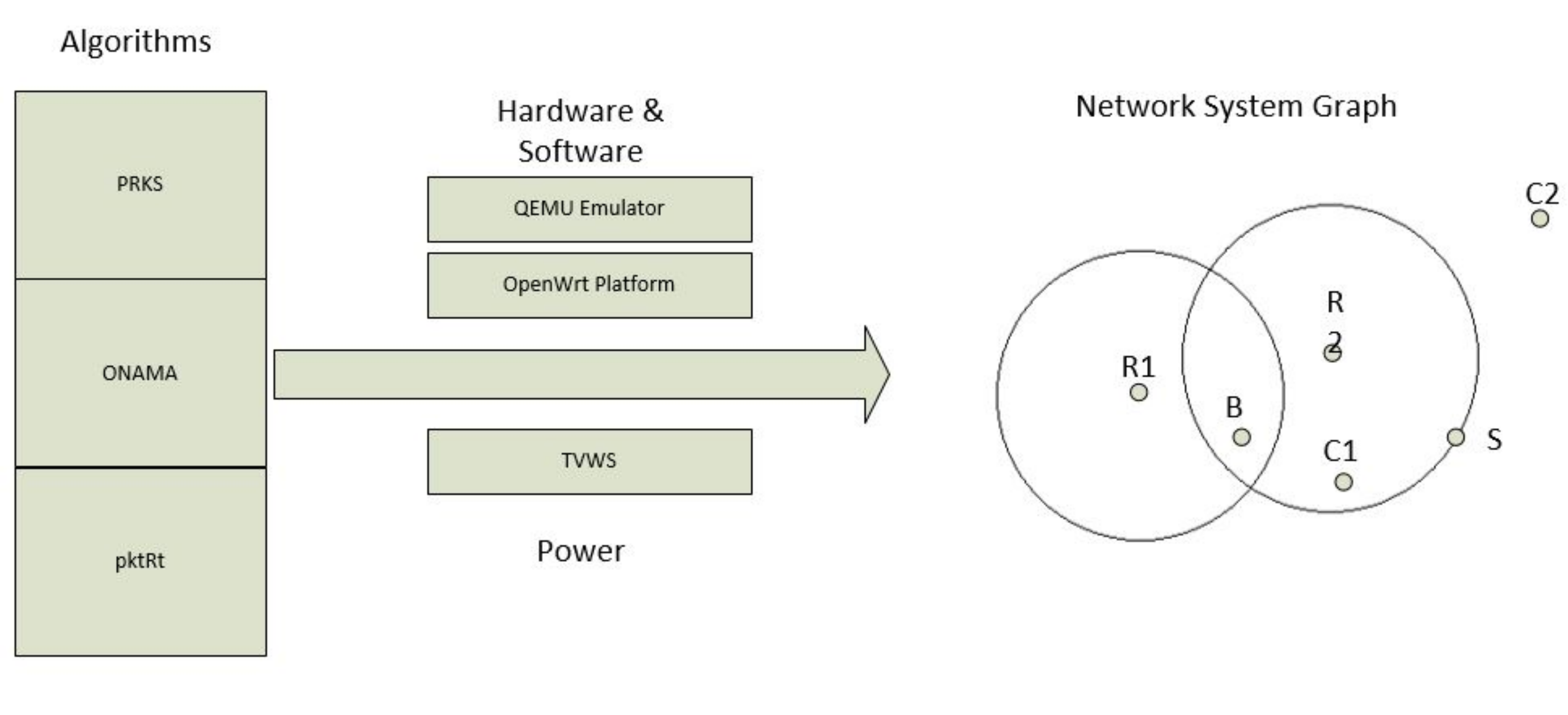
  Implement wireless algorithms (PRKS, pktRT, ONAMA) that aiming on enhancing link reliability, network throughput, decrease package loss rate, and have real-time packet guarantees.

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.

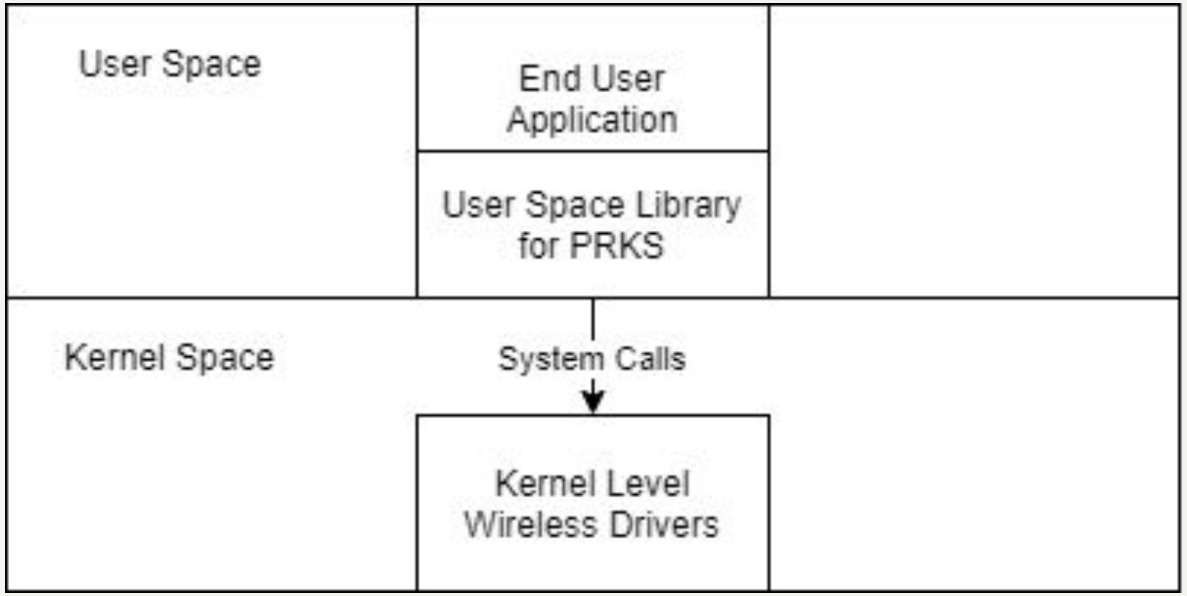
pktRT: Probabilistic Per-Packet real-time guarantee algorithm for wireless networks.

CONCEPTUAL SKETCH



CONCEPTUAL SKETCH

- High level design approach
  - OpenWrt Linux OS



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

CONSTRAINS

- Limitations
  - Limitation on local unused channels.
  - Hardware Requirements (RAM, CPU Speed, etc.).
  - Any noise / interferences within the TVWS Spectrum.

RISKS AND MITIGATION

Risks:

- Technical Understanding
  - wireless networking
  - kernel development
  - advanced algorithms
- COVID-19
- Team Availability

Mitigation:

- Trello, GitLab
- ZOOM
- Assigning jobs
- Weekly advisor meeting
- Weekly team member briefing

System Design

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.

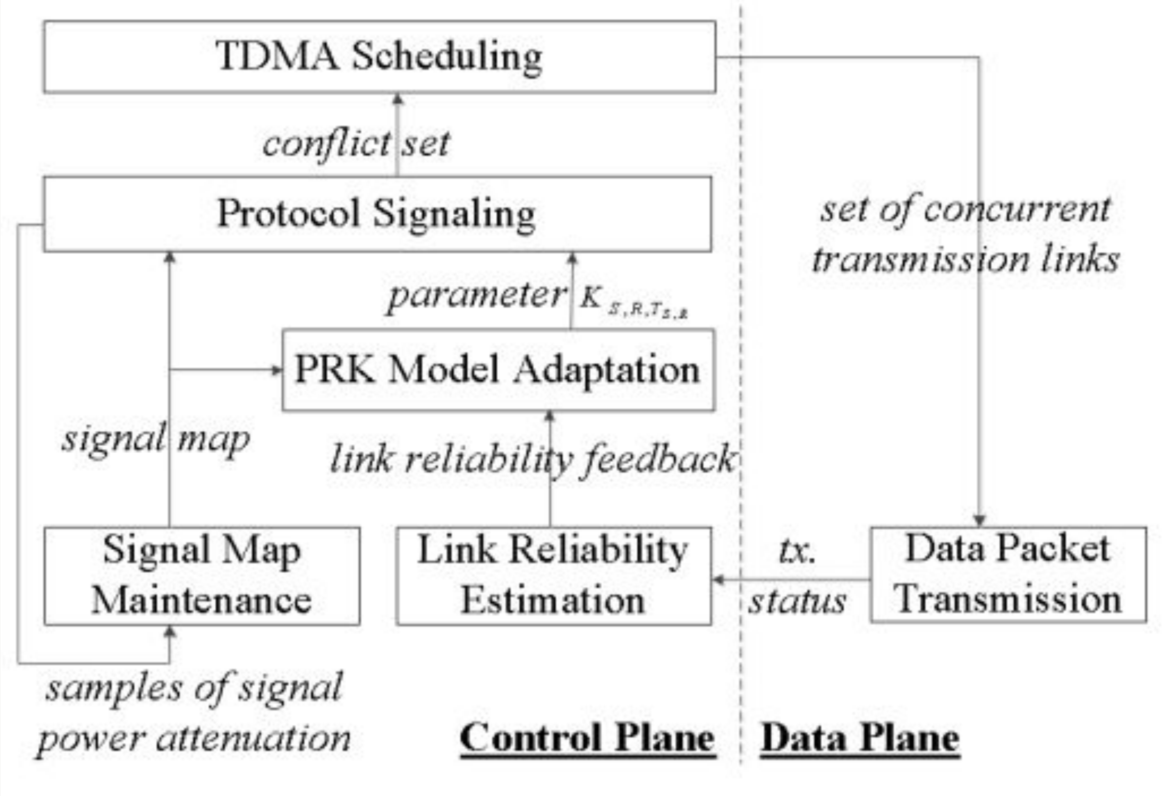


Fig. PRKS Architecture Diagram

SYSTEM DESIGN

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

- Support for the TVWS Spectrum to be supplied as a shared library from industry vender
- Within the mac80211 subsystem we will need to add a PRKS mode to support using the PRKS algorithm.

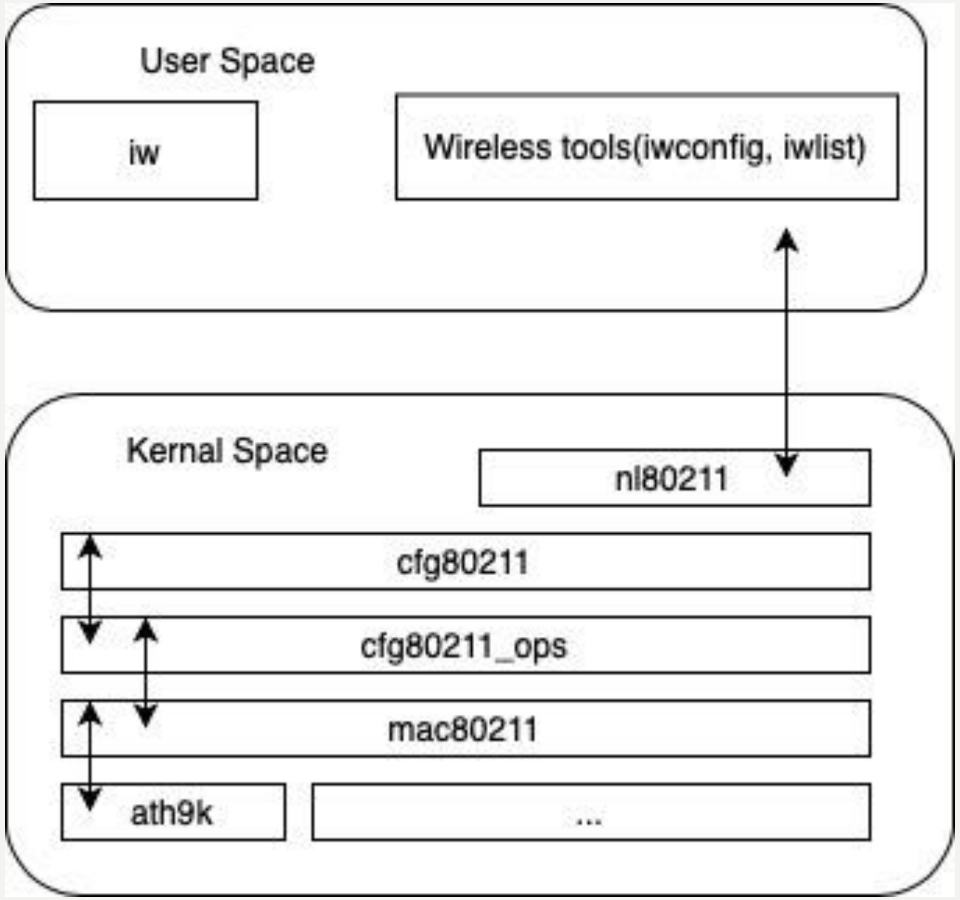


Fig. IEEE 802.11 Linux kernel implementation architecture

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.

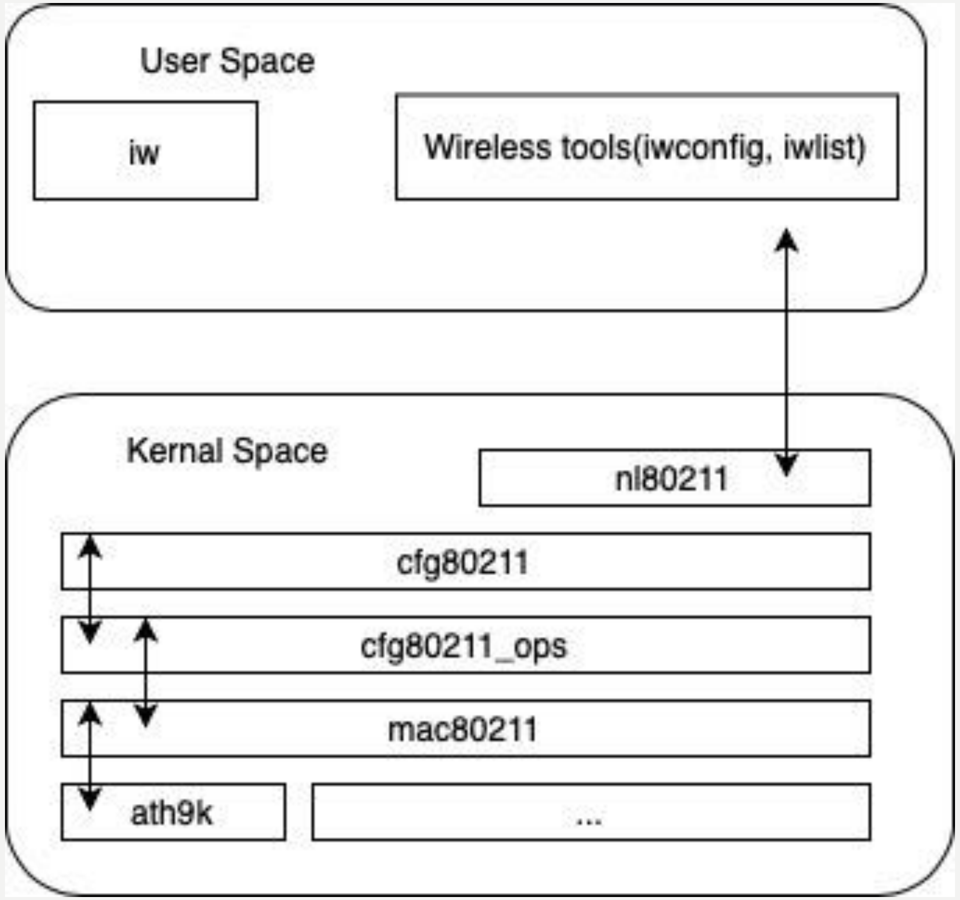


Fig. IEEE 802.11 Linux kernel implementation architecture

SYSTEM DESIGN

- FUNCTIONAL MODULES DESIGN
  - Pseudo Code
  - Structure of PRK Scheduling
    - Developing pseudo code by each part

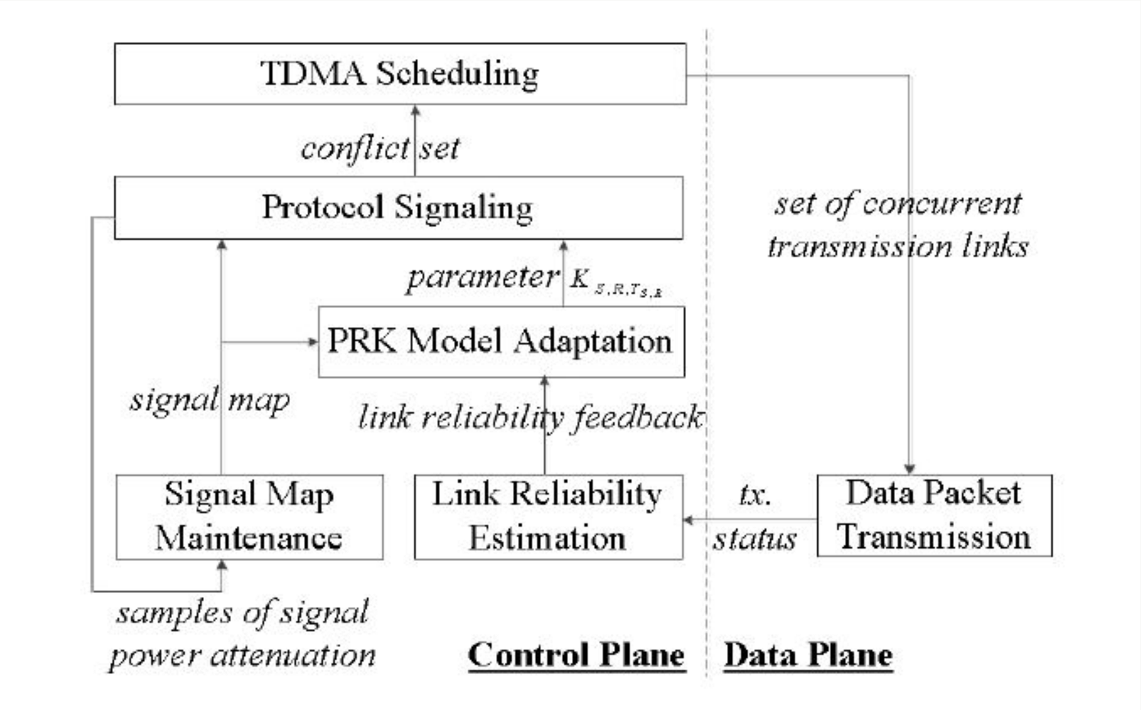


Fig. Structure of the Pseudo Code

SYSTEM DESIGN

- FUNCTIONAL MODULES DESIGN
  - Pseudo Code Sample

```
/*
This function describes how nodes rebroadcast itself and deliver the value of K within its sensing range.
And how other nodes react after receiving the signal.
*/

Main(exclusive_region E)
{

    While(true)
    {
        For Receiver R in exclusive_region E
        {
            Broadcast(R, K)
        }
    }
}

//node broadcast in their carrying sense range
broadcast(node R, K)
{
    For every node C in sensing range of node R
    {
        If P(C, R) >= P(S, R) / K
            C.broadcast(C, K, signal_map)
    }
}
```

Fig. Part of the Pseudo Code for Protocol Signaling

HARDWARE AND SOFTWARE

- OpenWrt
  - a project for embedded system on Linux
  - route network traffic
  - a frame to create application and driver
- QEMU
  - open source machine emulator and virtualizer
  - run OpenWrt with commands

TEST PLAN - FUNCTIONAL TESTING

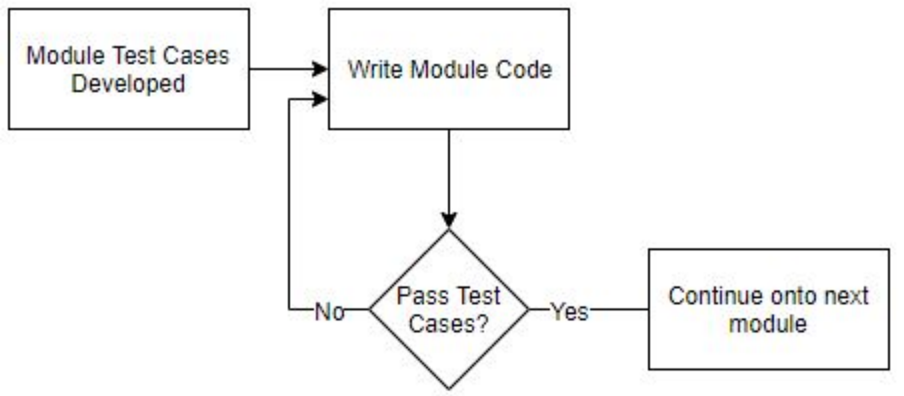


Fig. Test Flow Diagram

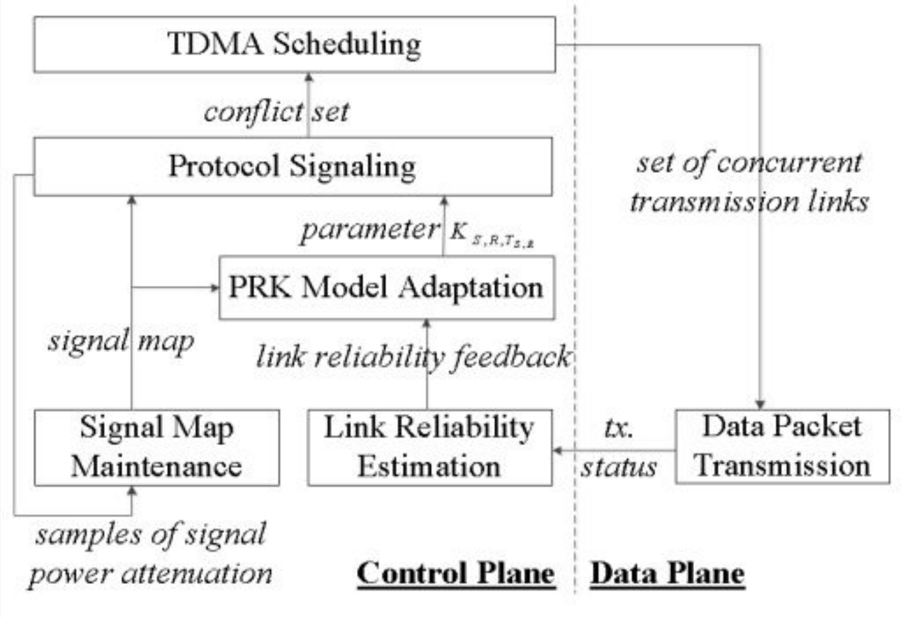


Fig. PRKS Architecture Diagram

- 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

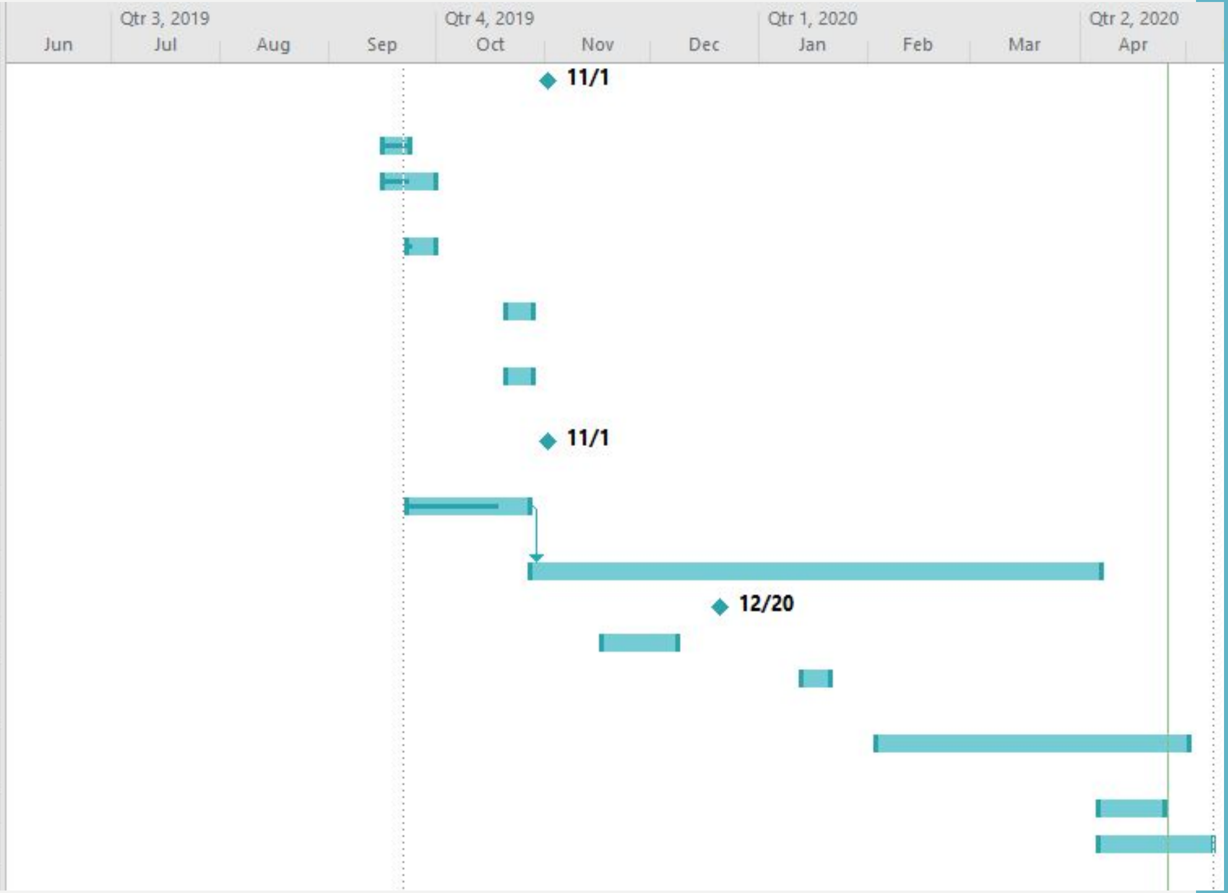
Engineering Standards and Design Practice

- pktRT
  - Per packet real-time guarantees
- PRKS
  - Predictable link reliability
- IEEE 802.11
  - The standard for port-based Network Access Control
- Agile Development

Conclusion

TIME DISTRIBUTION

Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Adc
	OpenWrt Kernel Development (Learning)	46 days	Sun 9/1/19	Fri 11/1/19			
	Virtualize OpenWrt	6 days	Mon 9/16/19	Mon 9/23/19			
	Cross Compile Hello World	11 days	Mon 9/16/19	Mon 9/30/19			
	Hardware Virtualization w/ QEMU	6 days	Mon 9/23/19	Mon 9/30/19			
	Linux Kernel WiFi System Understanding	6 days	Mon 10/21/19	Mon 10/28/19			
	Hello World Kernel Module	6 days	Mon 10/21/19	Mon 10/28/19			
	Algorithm Understanding (Learning)	46 days	Sun 9/1/19	Fri 11/1/19			
	Define Algorithm Parameters	26 days	Mon 9/23/19	Sun 10/27/19			
	Map Algorithm Params to OpenWrt	116 days	Mon 10/28/1	Mon 4/6/20	8		
	Design	35 days	Mon 11/4/19	Fri 12/20/19			
	Design Document	17 days	Sun 11/17/1	Sun 12/8/19			
	Proof of Concept Design Finalized	6 days	Mon 1/13/20	Mon 1/20/20			
	Documentation for Implementation Strategies	65 days	Mon 2/3/20	Fri 5/1/20			
	Router Hands On	15 days	Mon 4/6/20	Fri 4/24/20			
	Final Report and Demostration	25 days	Mon 4/6/20	Fri 5/8/20			



INDIVIDUAL CONTRIBUTION

Documents and Reports - Zhenwei Su

Device Setup/Algorithm Understanding - Dylan Sharp

Source Code Analysis - Zequn Wang

Algorithm Understanding - Yulin Song

Hardware Virtualization/ Cross Compilation - Shaohang Hu

Pseudocode/Algorithm Mapping/Implementation Strategies - Jiawei Deng

FUTURE PROSPECT

- What we haven't done.
- What could be done in the future.
- What we did for future development.

Thank you!

Team, Faculty Advisors, IRP members and Instructors