A TCPlp implementation based on: *Performant TCP for Low-Power Wireless Networks*

KUMAR ET AL. (2020)

Presented by Alex Gavin & Brandon Lavinsky

Outline

- TCPlp Background Info

- Implementation Details

- Demo

Background

What is TCPlp?

TCPlp: Full-scale TCP stack designed for resource constrained low-power and lossy networks (LLNs)

LOW-POWER AND LOSSY NETWORKS (LLNs)

As the name describes LLNs are a class of networks which have limited:

1. Processing power

2. Memory

3. Energy/Battery power

LLNs Sometimes Need Reliable Communication

Instances such as **voice commands,** and **vibration monitoring,** and **updating firmware over the air** need reliable communication

1. UDP-based protocols

2. Application specific protocols

3. TCP

UDP and other protocols aren't reliable and TCP is too heavy-weight

Standard TCP is not Suitable for LLNs

TCP deemed **too heavy-weight** for LLNs

1. Memory usage non-deterministic

2. TCP headers take up half of IEEE 802.15.4 frames

3. Expected power usage poor

Queue TCPlp

TCPlp is fully compatible TCP, interoperable w/ other devices, includes standard TCP features, and **lightweight**

1. Deterministic memory usage

2. TCP headers not problem
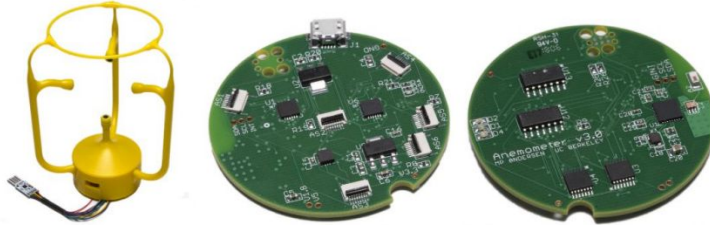
3. More efficient power usage

Implementation Motivation/Goals

- Implement TCPlp on **commodity** hardware
  - Paper leverages custom-built hardware

- Recreate Upper Bound Single-Hop Goodput

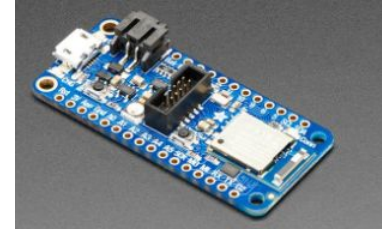- Recreate Single-Hop Maximum Segment Size experiment (didn't get to this)

Hardware Comparison

(a) Anemometer   (b) Hamilton-based PCB (bottom and top)

Adafruit Feather nRF52840 Express

- 512 KiB Flash

- 64 KiB SRAM

- 48MHz Cortex M4

- AT86RF233 Radio

- 1 MiB Flash

- 256 KiB SRAM

- 64 MHz Cortex M4F

- nRF52840 Radio

Single-Hop Goodput Upper Bound

Experiment:

1. Send a one way bulk transfer of 78 KiB of data from one nRF52840 board to another using TCPlp

2. Measure the upper bound of goodput during the transfer
   a. Goodput is only a measure of useful data
   b. TCP/IP headers on packets for example are not counted as goodput since headers do not contain application data

Expected Results:

- Paper achieved **75 kb/s** goodput for single-hop

- We aim to see similar however not identical results given the differentiation in hardware

Implementation Details

Hardware

- Adafruit nRF52840 Feather Express (~$25/unit)

  - ARM Cortex M4F w/ Nordic nRF52840 radio

  - Lots of other features we won't use as well

Software

- OpenThread – TCPlp implementation

- Nordic Connect SDK – board SDK (if needed)

- Zephyr – RTOS

- CMake/Ninja, West – Build system/programming board
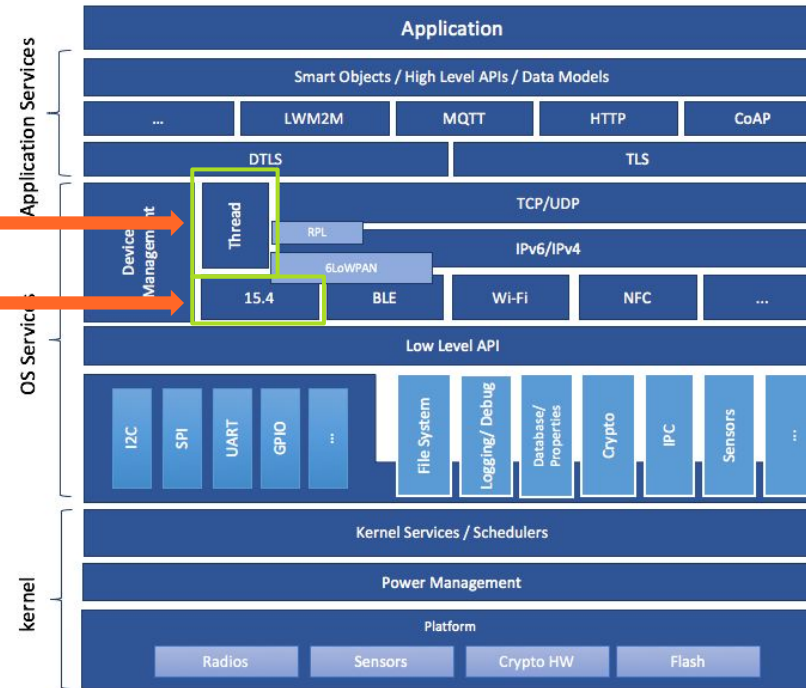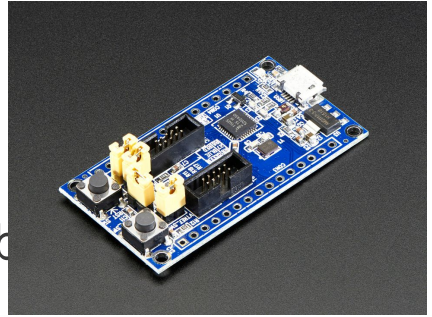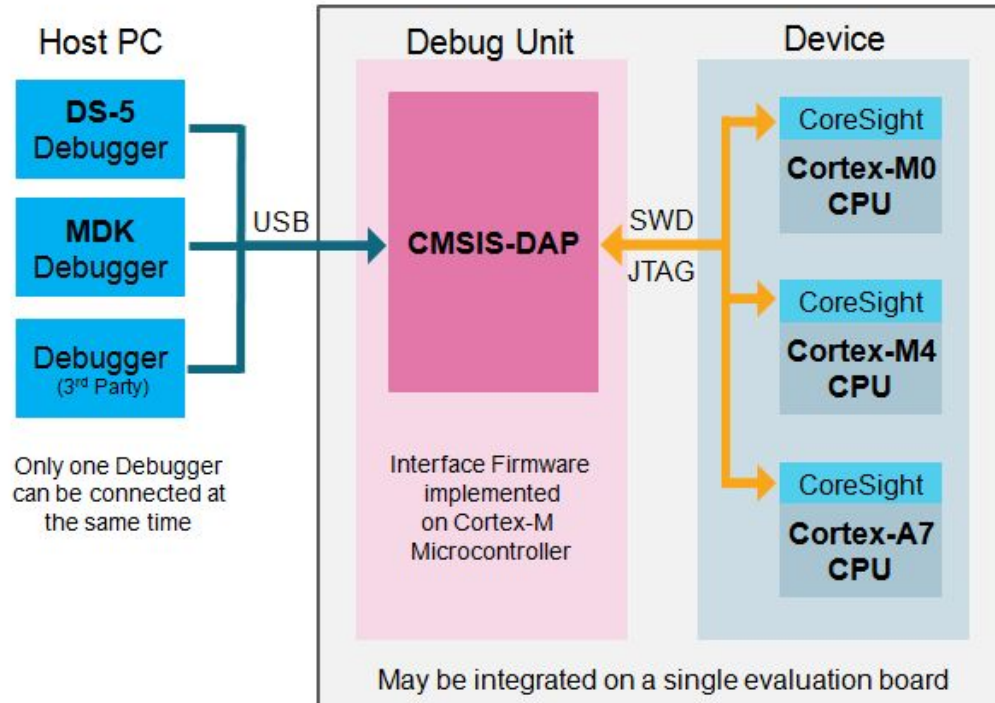
Software

We're here

More Hardware....

- nRF52840 Feather UF2 bootloader
  - Drag n drop programming of boards
- West requires addtl. device to program b
  - Can use GDB now!
- Luckily Embedded Sys. prof had compatible board!
  - Others sold out or super expensive

Flashing/Debugging

Demo