Evaluating IoT Protocols Project Plan and Work Breakdown Structure

Background:

This document serves to provide an outline for what overall work is to be completed for the project. In particular, this document provides steps and specifics that should be completed during each phase of research into the mentioned protocols. This will give better structure and insight into what must be done before moving forward.

Main Requirements and Milestones:

- Investigate and become proficient in the following protocols:
  - MQTT, AMQP, LwM2M, CoAP
- Code up simple development client and server code to interact with each protocol
- Perform packet captures with a sampling of each communication option for each protocol
  - This will be used for understanding how much data is being transferred overtime which can be useful for understanding power consumption
- Understanding security and its overhead cost of consumption in relation to each of the protocols
- Acquire development boards to perform power consumption benchmarking

Breakdown

For each of the following protocols each week, the following main goals will be completed: get working client and server side code to interact with each protocol, conduct packet captures for each communication option for each protocol. At the end of each protocol cycle, there will be an interactive session amongst the project members in which they will go over each of the protocols and packet captures to ensure everyone is on the same page. After each sprint for each protocol, a sample presentation will be prepared and presented to the project mentor for review. This will not only give the mentor a chance to provide feedback to the team regarding the content, but also begin to inform the mentor about what has been researched.

Once the preliminary research into each of the protocols has been completed, obtain development kits and begin to understand how to program/use the different protocols on the development boards. The main objective is to get the protocols running on cellular networks rather than standard wifi/ethernet transmission. There will be some overhead on this part and once the protocols are implemented. Once that understanding has been met, the protocols will then be implemented and a concrete power consumption breakdown can be performed.

Work Breakdown Structure:

Task Name	Duration	Start Date	End Date
1 Initial Protocol Research	7 Weeks	10/12/20	12/5/20
1.1 MQTT	2 Weeks	10/14/20	10/28/20
1.2 AMQP	2 Weeks	10/28/20	11/11/20
1.3 CoAP	1 Week	11/11/20	11/18/20
1.4 LwM2M	1 Week	11/18/20	11/25/20
1.5 Other protocols	1 Week	11/25/20	12/5/20
2 Dev Board Implementation	6 Weeks	12/5/20	2/11/20
2.1 Learn to use/interface with the board	2 Weeks	12/5/20	12/19/20
2.2 Create client simulations for the protocols	1 Week	1/14/20	1/21/20
2.3 Plan/Create evaluation metrics	1 Week	1/21/20	1/28/20
2.4 Perform power consumption testing	3 Weeks	1/28/20	2/18/20
3 Final Report and Presentation	4 Weeks	2/18/20	3/17/20
3.1 Compile results into succinct report	2 Weeks	2/18/20	3/3/20
3.2 Prepare final presentation	1 Week	3/3/20	3/10/20
3.3 Deliver final report and presentation	1 Week	3/10/20	3/17/20

Outlined Research Plan for Investigating Protocols

MQTT (MQ Telemetry Transport):

- Gain a surface level understanding of the publisher/subscriber model and how it pertains to the MQTT protocol
- Understand the protocol from an information perspective by going through the MQTT specification sheet
- Understand the dependencies that MQTT has on other protocols (TCP)
- Understand how to publish/subscribe to topics
    - Understand different wildcards that can be used to subscribe
- Get a working implementation of an MQTT client in python
- Get a working implementation of an MQTT broker/server (mosquitto recommended)
- Write python script(s) to test out the different functionality as a client with regards to the different packets that can be sent to the broker
- Obtain packet captures from each of the different communication methods outlined in the MQTT specification sheet (CONNECT, SUBSCRIBE, PUBLISH...)
    - Get an understanding of how much data is being transferred in each packet and how much gets transmitted overtime
- Understand how to use the MQTT authentication mechanisms for connecting to the broker

- Understand the different mechanisms for recovering sessions in the event of a dropped connection
- Write client code to simulate a sensor pushing data to the broker and application code to record the results from the sensor. Simulate the recovery process by cutting the network connection and see how the recovery process works.

AMQP (Advanced Message Queuing Protocol):

- Understand the protocol on a conceptual level by reading through and understanding the specification sheet for AMQP
- Understand the main differences between AMQP and MQTT
- Understand the main components of AMQP including exchanges, the message queue, and bindings
- Understand the different control packet types for the protocol
- Conceptually understand how consumers consume from the queue and how different use cases play into the standard
- Get a working implementation with the RabbitMQ broker and pika Python client library for AMQP
- Obtain a packet capture using the RabbitMQ broker and test python client
- Understand and get basic implementations for the security aspects built into AMQP

COAP(Constrained Application Protocol):

- Understand how CoAP relates to Http
- Understand what the header of CoAP looks like, and what the different parts of the header serve to do
- Understand what the 4 different message types on CoAP are and what their connections look like
- Understand how asynchronous messaging works in CoAP
- Know what a ‘Gateway’(or proxy) in a CoAP network does, and how it works
- Understand why CoAP goes well on top of RESTful (GET,PUT,POST,DELETE) interfaces
- Make a client and Server using CoAPthon
- Send messages between the client and server, and observe on Wireshark
- Research congestion control In COAP
- Research RFC 7641

LwM2M:

- Uses COAP, so much of the above applies
- Understand the built in objects used for reporting to the server
- Understand what parts of COAP are used and why
- Utilize Leshan server and client to explore the protocol more
- Obtain packet captures to gain an understanding of the various messages sent between client and server

Power Consumption Testing:

- Obtain the development boards to be used for testing purposes
- Write basic clients for each of the protocols evaluated in the preliminary research
- Perform power consumption testing by utilizing the energy analyzer on board and understand how much power will be consumed for each protocol in different cases
- Perform ‘robustness’ tests to see how the protocols perform when unexpected cases occur such as network failures or hardware problems