Communication Energy and Protocol Considerations in IoT Deployments

E. Liri, N. Orie, H. Siezar
and K.K. Ramakrishnan
University of California,
Riverside, CA 92521
Email: eliri001@ucr.edu,
norie001@ucr.edu, hsiez001@ucr.edu,
kk@cs.ucr.edu

P.K. Singh, W. Cai,
L. Gerday and K. Kar
Rensselaer Polytechnic Institute,
Troy, NY 12180
Email: singhp6@rpi.edu,
caiw2@rpi.edu, lancegerday@gmail.com,
koushik@ecse.rpi.edu

G. Lyon and P. Sharma
IoT and Mobility Lab,
Hewlett Packard Labs,
Palo Alto, CA 94304
Email: geoff.lyon@hpe.com,
puneet.sharma@hpe.com

Abstract—**In this paper, we experimentally evaluate the impact of protocol features at the transport and application layers on the energy consumed in IoT devices. Our transport layer experiments with TCP and UDP over WiFi indicate that at distances less than 15m between the IoT device and gateway, the difference in energy used by the different transport protocols is not too significant, as long as the number of messages exchanged does not grow significantly. However, adding application layer features through additional message exchanges comes with a significant energy cost. For example, adding DTLS to CoAP is 4x more expensive than using AES with CoAP. In addition, if services at one layer impose a burden on other layers, it may result in significant increase in energy use. We conclude that in an IoT environment, it may be preferable to increase packet sizes rather than number of messages when feasible. Additionally, careful consideration should be given to adding features at a layer if it results in increased energy consumption at another layer.**

I. Introduction

Edge devices deployed in an IoT environment are typically constrained by limited power, memory and compute capability. These devices are often powered by batteries, and deployed in environments where replacing batteries may expensive or infeasible. Since energy used up in communication makes up the bulk of the total energy cost, designing strategies for reducing the communication energy cost is essential for improving device lifetime. In the context of sensor networks, where energy usage is also a key consideration, there is a large body of prior work on optimizing communication energy cost [1]. Most of the prior work in this context focus on optimizing individual layers of the communication stack – using rate and power adaptation at the MAC layer or adjusting sensing rates [2] or duty cycles [3] at the application layer, for example. Cross-layer approaches which are more complex and combine parameters from multiple layers have also been proposed [4]. However, similar issues have not been investigated comprehensively for IoT devices, and in the context of the emerging IoT communication protocols.

In this work, we experimentally investigate the impact of IoT protocol features at the transport and application layers on communication energy cost in IoT devices. IoT protocol features refer to the number and types of services provided; examples of such services include reliability and congestion control, encryption and authentication etc. At the transport layer, we re-look at TCP and UDP - which most application layer IoT protocols tend to build upon - in terms of their energy usage. Our consideration of TCP and UDP is also guided by the fact that these two protocols represent the two main alternatives in terms of transport protocol features provided to the application layer - while TCP provides end-to-end reliability, flow and congestion control and a host of other features, UDP does not provide any of these services. Our results show the strong dependence of the transport layer energy consumption on the distance between the IoT edge device and the IoT gateway[1]. Further, we make the two following interesting observations: (i) below a certain distance between the IoT device and gateway (<15m), there is not much difference between using simple and complex transports (e.g., UDP vs. TCP respectively); (ii) above this critical distance, an exponential increase in energy is observed though at a lower rate for the simpler transport protocol . In the application layer, we consider the Constrained Application Protocol (CoAP) [5] in the confirmable mode, and show that adding encryption and authentication features at the application layer can have a significant effect on device communication energy especially if additional messages are involved. In particular, as compared to vanilla CoAP with no encryption or authentication, using CoAP+DTLS increases power consumption by 23.7% - or approximately 4x the additional energy incurred due to simple AES encryption.

The core contribution of our work is in the rigorous *experimental* evaluation of energy consumption in IoT devices (measured at Raspberry Pi devices) over WiFi, for different transport layer protocols, and security options at the application layer protocol. The results may help practitioners to determine which transport protocol to use in IoT deployments, depending on the distances between the IoT devices and the gateways. Further, it would help in the understanding of the additional energy costs associated with adding security features at the application layer. The results also indicate that practitioners

[1] The IoT gateway is the bridge between the IoT device and the Internet or external networks.

and protocol designers interested in energy efficiency should consider protocol features (including cross layer interactions and impact on number of transmitted messages) when making IoT protocol implementation and deployment decisions.

II. Background and Related Work

A. Background

Three of the more popular IoT protocols used today include Constrained Application Protocol (CoAP) [5], Message Queue Telemetry Transport (MQTT) [6], and MQTT for Sensor Networks (MQTT-SN) [7]. CoAP, is a light-weight request/response protocol designed for resource-constrained devices. Initially designed for a UDP transport, message overhead is kept small by specifying that the payload fit into one packet. CoAP uses a RESTFul API and is standardized in RFC 7252 [5] by the IETF. It uses four types of messages for interactions: *Confirmable*, *Non-confirmable*, *Reset* and *Acknowledgement* messages. CoAP implements reliability using Confirmable messages which use a simple stop-and-wait ARQ mechanism, and an exponential back-off timer for retransmissions. CoAP has also been integrated with TCP [8] to enable CoAP traffic in networks that do not forward UDP traffic. In this case, only TCP manages reliability and message duplication detection as well. For security, CoAP over TCP uses the Transport Layer Security protocol (TLS) (RFC 5246) [9] to provide privacy and data integrity between two communicating applications. CoAP over UDP uses the Datagram Transport Layer Security (DTLS) protocol [10] which is based on TLS and supports authentication, data integrity, confidentiality and automatic key management[11]. MQTT relies on a TCP transport and follows a publish-subscribe model to implement one-to-one, one-to-many and many-to-many connections between IoT devices. It uses a broker to interface between publishers who publish sensor data and subscribers who are interested in receiving this data. MQTT has three reliability levels (QoS levels) at the application layer, fire-and-forget, deliver at least once and deliver exactly once. MQTT-SN is a variant of MQTT that reduces the TCP overhead by using UDP. Comparison of how these protocols operate under network impairments can be found in [12].

B. Related Work

Mechanisms used to manage energy efficiency include using different optimization strategies at different layers of the communication stack. For example MAC layer strategies include using collision avoidance or output power control, network layer strategies include using efficient routing and data dissemination protocols [13] while application layer strategies include alternate operating schedules of the redundant nodes [14] and adjusting sensing rates [2]. However cross-layer strategies which are more complex and combine parameters from multiple layers have also been proposed [4]. A survey on some of these approaches is in [1].

Node placement is another strategy to optimize energy efficiency, and [1] discusses different approaches to determine the best node placement to maximize energy efficiency.

Prediction approaches seek to proactively manage device energy, typically by identifying power used by different components of the IoT device e.g., by experimentation, modelling it and using the model to predict future energy use. For example, in [15] the authors model the power used by the IoT communications, acquisition and processing systems. In [16] the authors present a power model (Power Pi) to derive possible power saving strategies for the Raspberry Pi (RP) when used as home gateways. [17] shows that a large fraction of energy is used by frames when they cross the protocol stack (OS, driver, NIC) and [18] extends this to show that this cross factor energy is device independent.

III. Evaluation

A. Transport Layer

At the transport layer, we use TCP and UDP to illustrate the impact of features with respect to reliability. Since TCP and UDP are the most widely used protocols, our insights will be applicable to the many IoT protocols that use them. These two protocols represent two extremes of reliability and have been studied extensively. For IoT protocols that implement their own reliability separately, e.g., CoAP over UDP, their implementation typically lies between the two extremes of full (TCP) and no (UDP) reliability. Therefore evaluating transport complexity using UDP and TCP may give a fair indication of the range of performance possible. Finally, reliable delivery is usually enforced through retransmissions. Therefore, retransmission rates and the mechanisms used for it have a large impact on communication energy.

B. Application Layer

We focus on the application layer because some IoT protocols implement additional services at the application e.g., reliability or security that affect the number/size of messages to be transmitted, and therefore, the communication energy. In order to demonstrate the impact of protocol features on the device energy we use CoAP as an example IoT protocol. An alternative to CoAP is MQTT. However, since MQTT relies on TCP as the transport, our transport layer results with TCP provide guidance on the impact on MQTT as well. Our consideration of CoAP is also guided by the fact that it has been standardized by the IETF [5], and is finding increasing adoption.

Before discussing addition of security features we consider how many messages are typically needed for the basic CoAP (with UDP transport) and no security. If we use Confirmable CoAP messages, and include the ACK in response messages, then the average number of messages for a single request and response is 2 (the best case). AES encryption with CoAP, does not increase the number of messages sent, but may increase the message size slightly due to padding to align the data to the size of a block. On the other hand, adding DTLS to a CoAP message generally requires an additional 6 messages (3 additional round trips) for providing encryption, authentication and message integrity. Adding TCP/TLS to a CoAP message generally requires an additional 3 round-trips

TABLE I
CoAP Messages

CoAP Setup	Num Round-trips	Total Messages
Basic CoAP	1	2
CoAP with AES Encryption	1	2
CoAP+DTLS	4	8
CoAP+TCP/TLS	5	10

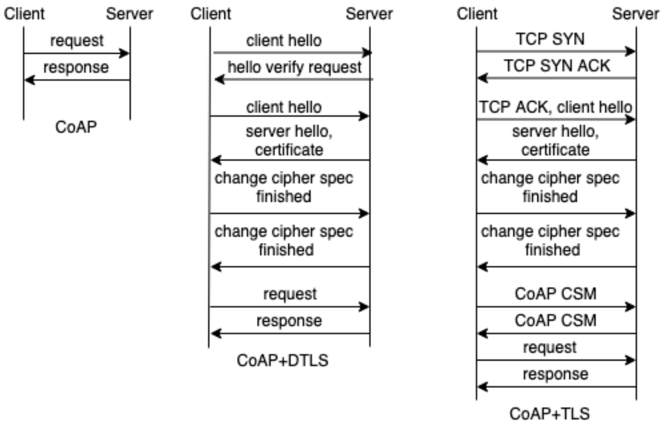

Fig. 1. Call flows comparing CoAP, CoAP with DTLS and CoAP with TLS

for the TCP and TLS Handshakes and an additional CoAP round trip for the Capabilities and Settings message (CSM) exchange. All of this is prior to sending the CoAP request, thus incurring 4 additional round trips. Table I compares the number of messages in each case and Fig. 1 illustrates the differences. Adding features with increasing complexity which increases both the amount of data to be sent and the messages exchanged, has an impact on energy consumption. A balance needs to be found between the additional desirable features that require complexity and the energy cost of those features. This needs to be done from the perspective of both implementation and operation. For example in terms of implementation, AES provides only encryption and increases the packet size while DTLS requires additional messages but provides authentication as well. In terms of operation, with CoAP+TCP/TLS, every new TCP connection will incur the extra messages required for setting up the encryption. Therefore if using TCP/TLS with CoAP, practitioners need to decide between utilizing keep alive messages (CoAP Ping and Pong messages) to maintain the TCP connection (thus keeping the IoT device active throughout, and not going to sleep) for a longer time versus having rather short TCP connections and incurring the setup cost of a new connection and TLS session. We perform experiments to understand the quantitative impact of additional features at both the transport and application layers.

C. Experimental Setup

Transport layer experiments compared energy required for UDP and TCP while application layer experiments compared CoAP with and without security mechanisms. The two types of experiments were performed with different devices, in slightly different environments and restricted to 1 client and 1 sink to compare energy use patterns and trends across devices. 2.4GHz WiFi was used and a power meter (AVHZY CT-2 USB Power Meter Load Tester Voltage Detector) connected to the client measured power consumption.

We recognize that there are a range of communication technologies available for IoT solutions including short range (e.g., RFID and Bluetooth), mid-range (e.g., WiFi and Zigbee), and long range (e.g., NB-IoT, LoRaWAN, SigFox etc). We used WiFi primarily for convenience. WiFi itself is also evolving, with alternatives, such as those being proposed by the WiFi Halow project to design a more energy efficient WiFi protocol for IoT devices based on the original WiFi protocol being a candidate for a range of IoT solutions. We believe that the experimental methodology used here for determining energy consumption for the networking subsystem could be applied to other communication technologies as well.

IoT applications are evolving, with some requiring IoT devices having increased functionality, more compute and communication capability. Agriculture applications that were traditionally measuring humidity and temperature seek to now use image and video data to monitor plant growth. Smart home monitoring and surveillance also incorporate video or images in addition to motion sensors. The use of low-cost, relatively lower power consuming Raspberry PI devices with adequate compute capability are therefore likely to the type of IoT devices used in these applications. Nonetheless, the one common characteristic is that they are generally energy constrained, with many running on battery power. We consider such energy constrained IoT devices, that can see increased lifetime and functionality through energy management techniques.

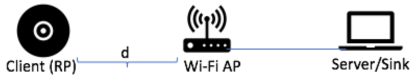

Fig. 2. Transport layer experimental setup

1) Transport Layer Experiments: Fig. 2 illustrates the experimental setup. Iperf3 was used to generate upload traffic from a client (RP3B+) to a server (MacBook Air) in an open field. The sink was connected to a Netgear Nighthawk AC1750 (Model: R6700v2) router (WiFi AP) via a short Ethernet cable. The distance between the client and AP was increased in steps from 0.9m to 61m and at each step separate experiments were run with TCP and UDP and the client uploading 1, 3, 5, 10, and 10,000 packets. Each experiment was repeated 200 times, apart from the 10,000 packet case at distances greater than 15m, which was only repeated 30 times due to increased execution time. The primary metric of interest used was the energy to transmit 1 MB from the client to the server (energy per MB). A number of related metrics were also collected to help understand the root cause for the communication-related power consumption.

IoT traffic is generally expected to be either a periodic or event based traffic pattern. A majority of our experiments assumed continuous packet transmission and ignored the sleep cycle in between. We run the experiments consecutively since we were only concerned with the upload energy. For this, Iperf3 was used, as it enabled bandwidth management and conveniently generates the statistics we needed for our experiment. However we verified our models validity and results by performing additional periodic traffic with an idle period between bursts of transmission, to be more representative of IoT traffic. These results are discussed in Section III-D.

2) Application Layer Experiments: For the application layer experiments three types of CoAP experiments are considered, simple CoAP (with no security features), CoAP with Encryption only (AES 128), and CoAP+DTLS. The setup is similar to Fig. 2 except an RP3B is used as as sink instead of a laptop. Two different IoT devices with different power capabilities were used as the client i.e. ESP32-DevKitC and RP3B. Experiments at distances 3m, 15m and 50m were performed outdoors in an open field while the 0.3m experiments were done indoors. Lightweight Tiny DTLS was used, and in each experiment the CoAP client sent 100 Confirmable requests.

D. Transport Layer Results

The maximum loss free rate, device base power requirements and the iperf energy overhead were first determined by experimentation. The maximum loss free rate for uploading data from the IoT devices to the sink was set to 44Mbps. Using this rate avoids wasteful work on the transmitting node if packets are dropped at the receiver because of UDP not having flow control. The WiFi channel was fixed to channel 6 for all the experiments. In addition, the RP3B+ idle base power with no peripherals connected and with WiFi active was 2.024W (with WiFi disabled it was 1.815W). Correspondingly, for the RP3B it was 1.165W (with WiFi disabled 1.099W). The power difference between the two systems is because the RP3B+ has a higher CPU clock rate of 1.4GHz compared to 1.2GHz for RP3B and supports both 2.4GHz and 5GHz WiFi. The fixed energy overhead contributed by iperf3 was 0.0001Wh. One issue to note is that although iperf3 periodically sends back delivery status updates to the sender the energy cost of transmitting the status updates could not be accurately obtained, therefore a fixed value for the iperf overhead was used. The results presented are limited to the experiments that exchange 10,000 packets, to amortize a number of the fixed overheads introduced by iperf3.

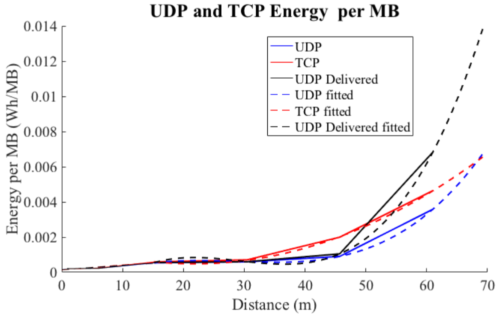

Fig. 3. Energy per MB with TCP and UDP (for both transmitted and delivered data

Fig. 3 shows the change in energy per MB as distance to the AP is increased. The UDP plot first shows the energy per MB to transmit 1MB of data i.e., energy per MB of data transmitted. But some of that data can be lost (channel or at the receiver). The $UDPDelivered$ plot shows the energy per MB for data delivered data i.e., energy per MB according to data actually delivered (lost packets not included). TCP contains reliability mechanisms to ensure data sent is delivered, and therefore the TCP plot for data sent and delivered is the same and includes all data sent, including retransmissions. Although the signal strength decreases with distance (results not shown due to space), the energy cost per MB is fairly constant, initially, in all three plots at distances less than 15m and can be modeled as a linear relationship. However, as the distance grows beyond 15m, there is a significant increase in the energy required in all cases modeled by Equations 1, 2 and 3. In the equations, d is the distance (m) between the IoT device and AP and E_{UDP}, E_{TCP} and E_{UDP_D} are the energy per MB in each case. We consider the energy per MB cost from the perspective of data that was actually delivered by comparing TCP and $UDPDelivered$. At larger distances, the energy per MB is higher for UDP than TCP, because of UDP's packet loss which reduces the amount of data that actually reaches the sink. TCP's retransmissions effectively increase the amount of data that reaches the sink, thus effectively using the energy for delivering a given amount of data. We also consider the energy per MB cost from the perspective of all data sent from the IoT device, for TCP and UDP. At larger distances, TCP requires significantly more energy than UDP i.e., it is more costly to send the same amount of data with TCP than UDP. A similar overall trend is seen with the experiments with less than 10,000 packets per experiment. The factors that are the underlying cause for the increase in transmission energy per MB as distance increases include retransmissions and link-layer ARQs.

The free space path loss model (Friis Transmission Formula) [19] indicates that path loss is proportional to the square of the distance d between the transmitter and receiver, or is proportional to $log(d)$ if working in dBm. The Okumura-Hata model [20] shows a similar log relationship between path loss and distance. A transmitter needs to adjust transmit power to ensure the received signal power is always greater than the receiver sensitivity, since path losses reduce the power of the signal received at the receiver. From the log plots of path loss vs. distance in the Friis and Okumura-Hata models, we observe two regions, which is also seen in our results. In the first region, path loss increases quickly but since this loss does not exceed the receiver sensitivity, the additional transmission energy needed per MB is small. In the second region, the rate of increase in path loss (in dBM) has reduced but because of the log relationship and the fact that the receiver sensitivity has been exceeded, even a small change in path loss incurs much higher transmission energy. This is seen as the exponential increase in energy per MB beyond 15m in our results. Note that although the Okumura-Hata model is typically used in a cellular environment, and for $d \geq$ 1km, our model at larger distances can also be modeled by a log relationship and would

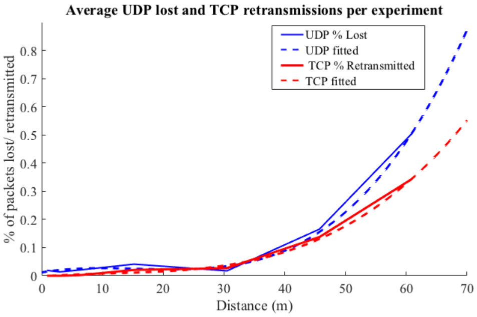

Fig. 4. Average lost/retransmitted packets with distance (% of packets sent per experiment

therefore exhibit similar behavior to the Okumura-Hata model.

$$\mathrm{E}_{UDP} = \begin{cases} 2.356 \times 10^{-05} d + 0.0001538, \text{ if } d \leq 15m \\ 9.883 \times 10^{-08} d^3 - 8.491 \times 10^{-06} d^2 \\ +0.0002313 d - 0.001362, \text{ otherwise.} \end{cases} \quad (1)$$

$$\mathrm{E}_{TCP} = \begin{cases} 2.809 \times 10^{-05} d + 0.0001386, \text{ if } d \leq 15m \\ 2.704 \times 10^{-06} d^2 - 0.0001181 d + 0.00177, \text{ otherwise.} \end{cases} \quad (2)$$

$$\mathrm{E}_{UDP_D} = \begin{cases} 2.446 \times 10^{-05} d + 0.0001515, \text{ if } d \leq 15m \\ 2.302 \times 10^{-07} d^3 - 2.018 \times 10^{-05} d^2 \\ +0.0005516 d - 0.003979, \text{ otherwise.} \end{cases} \quad (3)$$

The first factor causing increase in transmission energy that we consider is the observed throughput. From the experiments, the receive rate remains high for small distances, but then drops drastically from approximately 40Kbps to 20Kbps above a distance threshold i.e. approximately 15m (results not shown due to space limitation). The drop in throughput results in longer transmission time to send the same data (i.e., 10,000 packets) and thus more energy is spent by the IoT device.

The second factor affecting the energy per MB, (in the case of TCP), are the transport layer retransmissions due to the residual packet loss after link layer ARQs. Fig. 4 shows the transport layer losses for UDP and retransmissions for TCP with increasing distance as the average percentage of the total packets sent that are lost or retransmitted per experiment (i.e., the percentage of packets lost with UDP or retransmitted with TCP when sending 10,000 packets). For both transports, the loss/retransmission rate is fairly constant and low up to around 30m before there is an exponential increase at longer distances. At 60m there is a 0.5% loss for UDP and 0.34% retransmission rate for TCP. This residual packet loss recovered by retransmissions at the TCP layer, although small, still requires additional energy compared to UDP.

The third factor affecting the energy per MB is the MAC layer ARQ that implements reliability at the MAC layer. Fig. 5 shows the link layer retransmission count (ARQ) as a percentage of the total data traffic sent from the RP3B+ to the sink and this increases for both TCP and UDP as distance increases. The ARQ percentage per experiment exceeds 10% at 6m and increases to approximately 24% for UDP and 31% for UDP at 60m. The values for TCP are larger than UDP, due to additional ARQs for the TCP retransmissions. Although the ARQ % increases from 3m onward, we only see a significant jump in energy per MB from 15m, when the ARQ is about 12% for UDP and 20% for TCP. Thus, only having a large % of ARQs (above a threshold) begins to have a significant impact on energy. At shorter distances (<15m) the energy per MB, bit rates and percentage loss/retransmission rates are similar for UDP/TCP and the differences in terms of the reliability mechanisms do not have too much of an impact on energy. The 15m distance is the maximum distance at which the energy cost is agnostic to the transport protocol selected and is dependent on the communication technology, device characteristics and application tolerance for loss. Below this maximum distance, the energy per MB with distance can potentially be modeled by a linear relationship. In this work we have selected 15m from Fig. 3 but depending on the application tolerance, 30m is also a possible distance. At distances larger than 15m, there is an exponential increase in energy per MB, but UDP consistently requires less energy than TCP because of the TCP retransmissions. For IoT applications that have some loss tolerance, UDP can be an attractive and energy efficient option at distances greater than 15m.

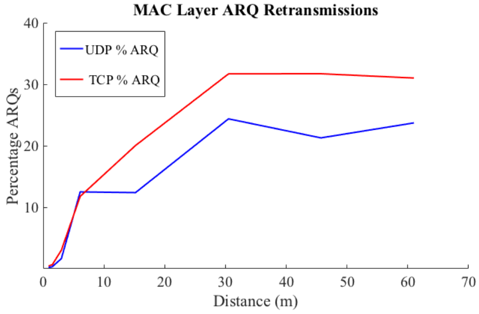

Fig. 5. MAC Layer ARQ retransmissions as a % of total data traffic

The high ARQ percentage for UDP (24%) and TCP (31%) at 61m indicates that the MAC layer is working hard to achieve reliability. For UDP this may be a waste of energy but if ARQ was not implemented, UDP would probably see a much higher, potentially unacceptable loss rate. In the case of TCP, the MAC ARQs help keep the residual loss rate low, but there is still duplication of functionality. For example, where TCP sends a retransmission and the MAC ARQ also has a retransmission for that retransmission, this increases energy consumption more. However, without ARQs, the high loss rate would result in very poor TCP performance. The effect of TCP retransmissions on ARQ retransmissions illustrates the cascading effect on energy across layers due to additional features. This is exacerbated in some protocols like MQTT which have retransmissions being generated at three layers: application, transport and MAC with retransmissions at the lower layer compounding the energy cost of retransmission from the higher layer. Some protocols like CoAP over TCP have tried to address this by having the transport layer handle reliability but the ARQ retransmission issue still remains. A possible solution to this is an adaptive reliability mechanism that reduces the duplication by TCP and ARQ retransmits and

TABLE II
AVERAGE ENERGY PER MB FOR DATA UPLOAD

Experiment	TCP (Wh/MB)	UDP (Wh/MB)
Estimated energy using model	0.000166	0.000176
512B	0.003248	0.001183
1MB	0.000115	0.000100

removes excessive ARQs for UDP. This is the focus of our future work.

IoT traffic is typically characterized by periodic transmissions and therefore we examined how valid our results with Iperf3, and the model derived from those measurements are, by performing additional IoT-like traffic which generates traffic periodically. Using Python scripts and TCP and UDP sockets, we transmitted data periodically from the RP3B+ to the sink with an idle period of 4s between successive transmissions. This was repeated multiple times and for data sizes 512KB and 1MB.

When using TCP traffic, our results show that with small amounts of data the overhead due to the data transfer protocol's setup and tear down as well as TCP connection setup and close is significant. These cause the energy per MB cost to increase very significantly. However, sending larger files, e.g., 1MB and above, this overhead is amortized and the energy per MB cost is similar to the values seen within our model derived from Iperf3 measurements. Table II compares the energy per MB values obtained from these experiments with the estimated energy cost from our model given in Equation (2) in the first row of the table. Since our focus is on IoT devices that are energy constrained but have more compute and communication capability than the traditional resource constrained sensors, data transmissions of 1MB and above are very possible especially when multimedia data is required. Therefore our model will prove useful in such scenarios. Nonetheless, with the same methodology, we will be able to derive a model for shorter transfers as well.

Results using UDP as the transport show a similar pattern, with the energy per MB cost for small data transfers, i.e., 512B being higher than our estimates based on Equation (1) in Row 1. This is due to the overhead for marshalling and packetizing the data. However, as the data size transferred increases, the measured energy per MB value increases, approximating our estimate since the added overheads are amortized. Based on the model we developed in our field measurements, at very small distances, the energy consumption for UDP is slightly higher than TCP. This is reflected in our estimates. However, the actual measurements for IoT-type traffic show that using TCP for data transfer consumes more energy. This is due to overheads for setting up the data transfer as well as the TCP connection setup and tear down for each transfer.

E. Application Layer Results

For the CoAP experiments, we conduct experiments similar to the transport experiments described in Section III-D. Note that delay and loss were implemented using Linux Traffic Control (TC) and these experiments were run in CoAP Confirmable mode. These experiments also showed a similar increase in energy per MB beyond 15m and with the increase in the distance from 0.3m to 50m, the energy per MB increases by 280.1 % (Fig. 6). However, because of constraints on experiments with distances beyond 15m and the larger step size, the data from the measurements are insufficient to show the exponential increase adequately. Fig. 7 shows the power consumption (in Watts) for CoAP under three different security settings: (i) no security (CoAP), (ii) encryption only (CoAP with Encryption), and (iii) with DTLS (CoAP-DTLS). Addition of simple encryption increases the power consumption slightly (5.53 % over no encryption case). However, the use of DTLS increases the power consumption quite significantly (23.7 % over no encryption case) which is approximately 4x the cost of simple AES. This is due to the extra messaging overhead associated with DTLS, as illustrated in Fig. 1. Experiments performed comparing the energy used with CoAP (no security) and CoAP-DTLS show

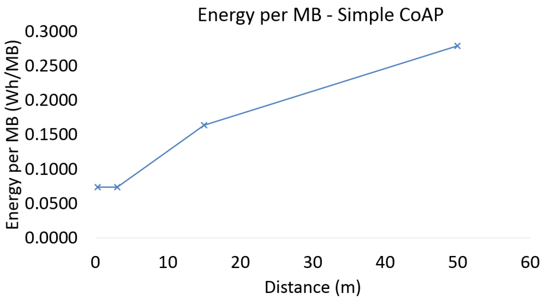

Fig. 6. CoAP: Energy per MB vs Distance

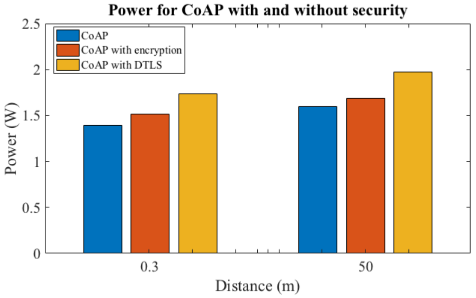

Fig. 7. Power consumption at 0.3 and 50m

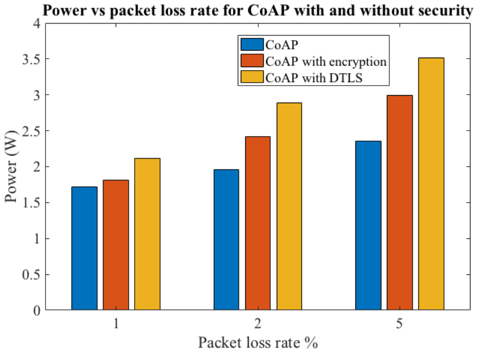

Fig. 8. Power variation with loss

that the CoAP-DTLS overhead is approximately 2x CoAP (no security). We repeat the experiment with ESP32 as the client and observe similar results. From Fig. 7, we observe that the power consumption increases slightly with distance. For CoAP, this corresponds to an increase of 15.7 % in power consumption (Watts) as the distance increases from 0.3m to 50m. However, when measured in terms of energy per MB, the impact of distance is more pronounced, as seen earlier in Fig. 6.

Network impairments were emulated using Linux Traffic Control and Fig. 8 shows the impact of loss on power consumption when we use CoAP with and without security. Generally as the loss increases, all three CoAP configurations show an increase in required power due to additional transmissions required by the protocols. DTLS still requires more power than simple encryption and experiments with delay also show a similar trend. Loss and delay exacerbate the cross layer energy effect. Therefore, one recommendation is in cases of poor network conditions, complex protocols that require fewer messages to be transmitted or protocols that have less cross layer energy effect may be preferred to save energy.

IV. Conclusion

In this paper, we experimentally investigate how increasing protocol features related to increased energy usage in IoT devices. Our transport layer experiments with TCP, UDP over WiFi indicates that distance between the IoT device and the gateway plays a critical role in this energy consumption: when this distance is less than 15m, the difference in energy used by TCP and UDP is not too significant. However, at larger distances, energy to send the same amount of data increases exponentially for both protocols, with UDP requiring less energy to send the same amount of data than TCP for the same distance. This result is important because it is indicative of the expected behaviour when these transports are used with other IoT application protocols, e.g., MQTT over TCP. This knowledge can then be used when making decisions regarding the desired IoT protocol and its features. Our experiments at the application layer includes studying the energy usage of CoAP with additional security (encryption and authentication) features. The results show that the energy cost of adding DTLS to CoAP can be quite significant.

Overall, our experiments show that adding features incurs significant energy, especially when these features require additional message exchanges and when features at one layer impose a cross layer energy consumption in another. In an IoT environment, it may be preferable to increase packet sizes rather than number of messages. Careful consideration should be made when adding features at one layer to avoid the cross layer energy effect. This requires balancing the energy use against the benefit of a more complex protocol stack to optimize performance.

Acknowledgment

The authors would like to thank NSF for support provided through grants CNS 1619441 and CNS 1618344.

References

[1] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104 – 122, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614001418

[2] H. Liu, A. Chandra, and J. Srivastava, "eSENSE: Energy Efficient Stochastic Sensing Framework Scheme for Wireless Sensor Platforms," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, ser. IPSN '06. New York, NY, USA: ACM, 2006, pp. 235–242. [Online]. Available: http://doi.acm.org/10.1145/1127777.1127815

[3] S. Ganeriwal, I. Tsigkogiannis, H. Shim, V. Tsiatsis, M. B. Srivastava, and D. Ganesan, "Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 843–856, June 2009.

[4] L. Mendes and J. Rodrigues, "A survey on cross-layer solutions for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, pp. 523–534, 03 2011.

[5] C. Bormann, K. Hartke, and Z. Shelby, "The Constrained Application Protocol (CoAP)," *RFC 7252*, 2015.

[6] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-s—a publish/subscribe protocol for wireless sensor networks," in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*. IEEE, 2008, pp. 791–798.

[7] A. Stanford-Clark and H. L. Truong, "Mqtt for sensor networks (mqtt-sn) protocol specification," *International business machines (IBM) Corporation version*, vol. 1, 2013.

[8] C. Bormann, s. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, and B. Raymor, "Coap (constrained application protocol) over tcp, tls, and websockets," *RFC 8323*, 02 2018. [Online]. Available: https://tools.ietf.org/html/rfc8323

[9] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. [Online]. Available: https://tools.ietf.org/html/rfc5246

[10] E. Rescorla and N. Modadugu. Datagram Transport Layer Security Version 1.2. [Online]. Available: https://tools.ietf.org/html/rfc6347

[11] T. A. Alghamdi, A. Lasebae, and M. Aiash, "Security Analysis of the Constrained Application Protocol in the Internet of Things," in *Second International Conference on Future Generation Communication Technologies (FGCT 2013)*. IEEE, 2013, pp. 163–168.

[12] E. Liri, P. K. Singh, A. B. Rabiah, K. Kar, K. Makhijani, and K. K. Ramakrishnan, "Robustness of iot application protocols to network impairments," in *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, June 2018, pp. 97–103.

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," 2000, pp. 3005–3014.

[14] A. Liu, X. Jin, G. Cui, and Z. Chen, "Deployment Guidelines for Achieving Maximum Lifetime and Avoiding Energy Holes in Sensor Network," *Information Sciences*, vol. 230, pp. 197 – 226, 2013, Mobile and Internet Services in Ubiquitous and Pervasive Computing Environments. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025513000297

[15] B. Martinez, M. MontÃn, I. Vilajosana, and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, Oct 2015.

[16] F. Kaup, P. Gottschling, and D. Hausheer, "PowerPi: Measuring and Modeling the Power Consumption of the Raspberry Pi," in *39th Annual IEEE Conference on Local Computer Networks*, Sep. 2014, pp. 236–243.

[17] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi, "Energy consumption anatomy of 802.11 devices and its implication on modeling and design," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. New York, NY, USA: ACM, 2012, pp. 169–180. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413197

[18] P. Serrano, A. Garcia-Saavedra, G. Bianchi, A. Banchs, and A. Azcorra, "Per-frame energy consumption in 802.11 devices and its implication on modeling and design," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1243–1256, Aug. 2015. [Online]. Available: https://doi.org/10.1109/TNET.2014.2322262

[19] H. T. Friis, "A note on a simple transmission formula," *Proc. IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[20] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Trans. Veh. Tech.*, vol. 29, no. 3, pp. 317–325, 1980.