# Wireless Internet of Things – Quantifying Cost to Transmit TCP Data in Lossy Radio Link Conditions

Craig Lee
AT&T Labs - Ecosystems & Innovation
Internet of Things Foundry
Plano, Texas, United States of America
Craig.Lee@att.com

Andrea Fumagalli
Open Networking Adv. Res. (OpNeAR) Lab
The University of Texas at Dallas
Richardson, Texas, United States of America
andreaf@utdallas.edu

**Abstract— IoT wireless connectivity often presents a lossy and error prone radio link layer between the wireless edge device and the wireless base station. Data transacted over the radio link layer often uses TCP, which provides programmatic error mitigation elements, e.g., loss of and out-of-order packet detection, combined with ACK/NACK from far end to trigger packet retransmission. In the use case of subscription based wireless connectivity, such as cellular or satellite, the information payload is often metered and billed per amount of data passed over the radio link. Hence, a wireless edge device can incur significant data overage charges to transmit a given amount of application information due to the increased billable TCP-driven retransmissions.**

**The aim of this paper is to quantify the TCP-retransmission rate caused by a lossy radio link as it is experienced by a typical IoT LTE wireless device. The authors also propose the creation of a functional layer between the application layer and the lower layers on the wireless edge device, referred to as "RF Fidelity Layer". Its purpose is to provide real-time situational knowledge of the radio link layer to the application software, which in turn can "tune" the information flow for maximum efficiency and minimum billable data overage due to TCP-driven retransmissions.**

***Keywords— Internet of Things, TCP over Lossy Link, Billable Data Overage***

## I. Introduction

Many IoT applications rely on edge devices that are connected through radio links [1]. Information types in IoT devices typically fall into three classifications:

1) urgent real-time information that must be transacted at all costs (nominally less than 1kB);
2) normal communications between the device and backend host (nominally less than 10kB); and
3) over-the-air updates & maintenance communications (nominally 1MB or greater).

For transferring data end-to-end, edge devices often make use of TCP. When IP packets cannot be delivered to the other end, TCP retransmissions are triggered to ensure that lost packets are in the end correctly received and delivered to the application layer in the right order. While being effective in achieving this goal, the TCP solution requires IP packets to be retransmitted over the radio channel. In addition, the TCP throughput performance may be compromised by a lossy radio link. This problem has been widely addressed in the literature [2], [3], [4], [5], [6], [7].

For most IoT applications, however, the edge device throughput is typically quite moderate and the TCP throughput performance does not represent an issue. More relevant to IoT applications is the *inefficient utilization of resources* caused by TCP retransmissions. For example, with some wireless ISPs, the subscriber is charged for IP packets being both transmitted and retransmitted. In other wireless ISPs, retransmitted packets represent a loss of revenue as the subscriber is only charged for each packet delivered successfully [8], [9]. From the device point of view, battery consumption is proportional to the amount of retransmitted packets, which may deplete unnecessary energy from the edge device. In summary, while TCP is a well-established packet retransmission mechanism for ensuring end-to-end data delivery, its inefficient resource utilization caused by lossy radio channel is perceived as a serious drawback from an IoT application point of view.

In this paper, the authors argue that the first step towards addressing this drawback is to quantify the TCP "retransmission cost" caused by a lossy radio channel. While qualitatively it is straightforward to expect that a lossy channel is causing more TCP retransmissions over a given data transfer, the aim of this preliminary study is to quantify the effect of some of the radio channel performance indicators [10] on TCP packet retransmission. Based on the outcome of this preliminary study, the authors propose the creation of a *functional layer* between the application layer and the lower layers on the wireless edge device, referred to as "RF Fidelity Layer". Its purpose is to provide real-time situational knowledge of the radio link layer to the application software, which in turn can "tune" the information flow for maximum efficiency and minimum billable data overage due to TCP-driven packet retransmissions.

## II. Rationale for the RF Fidelity Layer

A significant body of research has focused on modifications to the TCP/IP protocol to address situations where a lossy error-

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

prone link, such as wireless, is involved [2], [3], [4], [6], [7]. These methods and algorithms demonstrate excellent improvements in wireless communications but often require modifications to the TCP/IP protocol. As TCP can "mask" a lossy link layer via repeated retransmissions to correct for lost/corrupted information packets, application developers for the edge device often code the functionality assuming that the end-to-end connectivity is error-free. The high layer application "fires and forgets" the information to be transmitted while the lower layer TCP stack is tasked with successfully delivering it to the destination.

An edge device connecting through a lossy radio link can incur significant data overage charges to transmit a given amount of application information due to the increased billable retransmissions required to correct for radio link induced errors. The authors propose to introduce an RF Fidelity Layer as a method to reduce the anticipated expense of bidirectional error retransmissions when the edge device determines that it is in a high link loss/error situation. The wireless edge device often has real time situational knowledge of the current fidelity of the radio link layer that it is utilizing, e.g., RSSI, RSRQ, SINR, RF interference, antenna capability, link loading, reduced power and mobility speed.

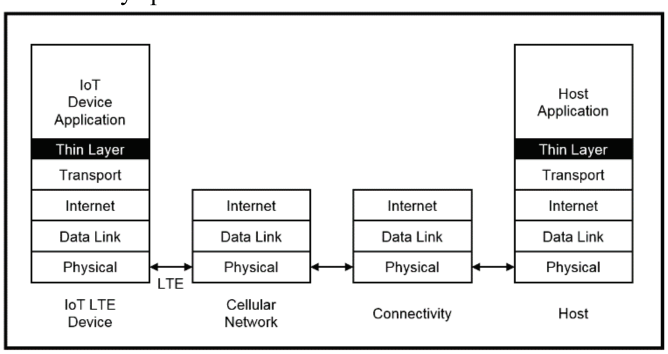


Fig. 1. Proposed end-to-end TCP stack diagram with thin layers at each end.

The RF Fidelity Layer is a thin application layer element running in the stack of both the wireless edge device and host application server (Fig. 1). The RF Fidelity Layer is directly or indirectly aware of the current fidelity of the radio link between the wireless device and the wireless base station. This thin layer may be implemented as a device library called by the device application. Alternatively, the application logic may take into account the algorithmic fidelity prediction directly.

At the wireless device, the RF Fidelity layer collects real-time information from various radio stack probes to develop an algorithmic view of the current wireless link's susceptibility to introducing errors during transmission. As the algorithm determines the real-time radio link fidelity, the RF Fidelity Layer may decide which classifications of datatypes be sent and which classifications may be preempted until wireless link fidelity reaches a state that would ensure less error-prone transmissions.

As already mentioned, information types in Internet of Things devices typically fall into three classifications:

1) urgent real-time information that must be transacted at all costs (nominally less than 1kB);

2) normal communications between the device and backend host (nominally less than 10kB); and

3) over-the-air updates, data logging and maintenance communications (nominally 1MB or greater).

For example, if the algorithm determines that the radio link fidelity is poor, then the RF Fidelity Layer of the edge device can choose to only send the urgent classification of information with the awareness that radio link errors may induce TCP retransmission incurring additional costs.

At the host application server, unbidden data can also be sent to the wireless device. As the wireless device can be made aware of the real-time radio link fidelity, the far end host will not be aware unless the wireless device regularly communicates the current radio link fidelity situation. This additional communication not only increases overall TCP metered data costs, but as the fidelity of the radio link degrades, the ability of the wireless device to send fidelity updates to the far end host will also be challenged or compromised.

A complementary RF Fidelity Layer at the host can indirectly infer the fidelity of the radio link based upon several factors.

1) If recent communications from the wireless device have solely consisted of the first classification, i.e., urgent real-time, when a mix of classification communications is expected, then the host side element will infer that the wireless device's thin layer has algorithmically determined that the radio link fidelity is poor and for that reason is only sending urgent classification of information. The host side element can then mirror its unbidden communications to the wireless device by constraining transmission of data in the same level of classification.

2) The thin layer in the wireless device can append into the application data packet the current real-time algorithmic fidelity prediction values. This information can be removed by the thin layer in the host prior to passing the now original application data packet on to the host application. The host side element uses a complementary algorithm to determine which classification to send to the wireless device.

It is worth noticing that no modifications to the TCP/IP protocol or wireless communications is required for this method to function. The thin layer elements reside entirely at the application layer at both endpoints such that intermediary hops in the communications network need not be aware.

## III. Experimental System Description

A typical IoT wireless device consists of an LTE radio module, SIM, the controlling processor, a set of I/O sensors and actuators, antenna structure and power supply as in Fig 2. The LTE radio module contains the RF chip set and driving software stack along with RF front-end, SIM interface and command-and-control interfaces. Control of the LTE radio module is driven through the command interface which utilizes an extended form of the AT command set. A sequence of AT commands from the processor (microcontroller) to the LTE radio module instructs the module to attach to the LTE network and establish a TCP data session. The processor can also query the status of the radio module via the AT command set. The 3GPP TS 36.214 specification defines many of the physical measurements available [1]. The radio module utilized for collecting the test results is the Sierra Wireless AirPrime model EM7455. The AT command utilized in this research is “at!gstatus?”. Upon issuing this command, the radio module replies with a formatted sequence of radio state information as shown in Table 1.

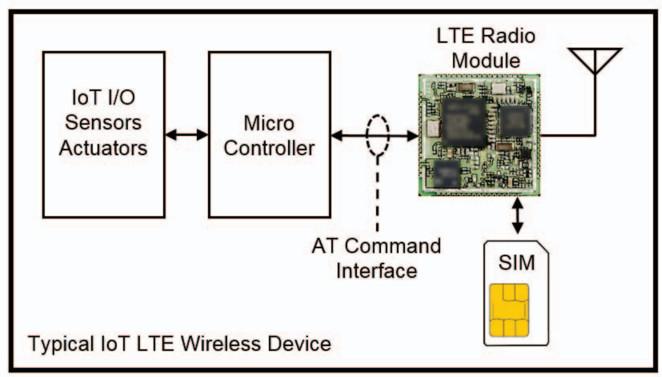


Fig. 2. Typical IoT LTE wireless device components highlighting the AT command interface where current radio chip set probe information is exchanged.

TABLE I. AT COMMAND !GSTATUS OUTPUT

| at!gstatus? | |
|---|---|
| !GSTATUS: | |
| Temperature: | 31 |
| Mode: | ONLINE |
| System mode: | LTE |
| PS state: | Attached |
| LTE band: | B5 |
| LTE bw: | 10 MHz |
| LTE Rx chan: | 2525 |
| LTE Tx chan: | 20525 |
| EMM state: | Registered |
| Normal Service | |
| RRC state: | RRC Connected |
| PCC RxM RSSI: | -83 |
| RSRP RxM (dBm): | -120 |
| PCC RxD RSSI: | -98 |
| RSRP RxD (dBm): | -140 |
| Tx Power: | -20 |
| TAC: | 0001 (1) |
| RSRQ (dB): | -20.0 |
| Cell ID: | 00000100 (256) |
| SINR (dB): | -3.6 |

The thin application layer element “RF Fidelity Layer” implemented within the processor code periodically queries the radio module to collect real-time link status information and processes collected information through an algorithm to determine which of the three classifications of IoT data is to be communicated.

## IV. RESULTS

Testing is conducted in a controlled screen room to isolate external interference during this phase of testing. A block diagram of the test-bed is shown in Fig. 3. The wireless device comprises an LTE radio module, Sierra Wireless AirPrime model EM7455, which is mounted on a companion development board. The AT commands are generated using HyperTerm on a standalone PC mimicking the device microcontroller. One of the two RF diversity input ports on the radio module is terminated with 50 ohms and the other connected to a three-way RF coupler. A Rohde & Schwarz CMW 500 Wideband Radio Communication Tester is connected to the coupler and provides cellular carrier emulation. The third port on the coupler is connected to a R&S SMBV100A Vector Signal Generator to inject controlled interference signals. Wireless LTE communications is established between the CMW 500 and the radio module. An FTP session is then established between the FTP host within the CMW 500 and the standalone PC. Testing is conducted over the lower range of power levels to emulate impaired RF conditions and a swept signal is introduced by Vector Signal Generator to further impair the RF link condition.

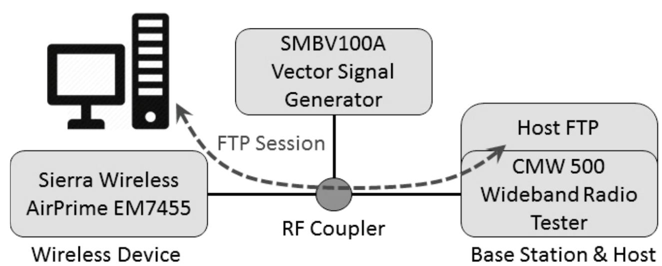


Fig. 3. Test equipment set up in shielded screen room.

Multiple experiments are carried out using the test-bed in Fig. 3. The test plan consists of adjusting the CMW 500 to increasingly lower power output levels while maintaining a constant interference signal level. This provides a simulation of an IoT wireless LTE device in a fringe or challenged RF coverage area in the presence of a localized interference signal – a common challenge with IoT devices that might be installed in the basement of a building for example. For each experiment, the CMW 500 output power level is set to predetermined values (-115, -117, -118, -119, and -120dB). Then, a script in HyperTerm is used to evoke the at!gstatus command of the radio module, and RSRQ and SINR values are collected by the standalone PC. For each experiment set up, four data files are transferred from the host to the wireless device over the LTE link. The four files have increasing size, i.e., 1kB, 10kB, 100kB, and 1MB. The TCP/IP packet exchange is captured at the CMW 500 and analyzed using Wire Shark. TCP traffic traces are processed to determine the ratio of successful TCP packets to the total TCP data transacted. For example, a message transfer with no TCP retransmissions would have a ratio of 1:1. A message transfer with 20% retransmission rate would have ratio of 1:1.2. This means that a 100kB message would incur a total of 120kB of TCP traffic or 20kB of overage.

Fig. 4 shows the correlation of increased TCP traffic as the RF fidelity is impaired along with the corresponding SINR and RSRQ radio state information obtained through the at!gstatus AT command. Table II lists the actual test measurement results obtained using the above methodologies.

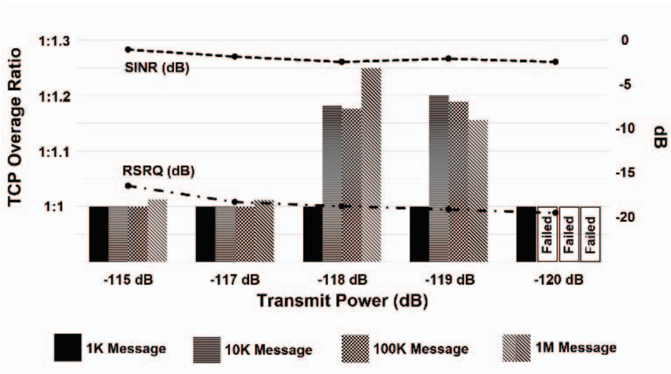


Fig. 4. Graph correlating transmit power to SINR and RSRQ and corresponding TCP overage ratios by message size. Results are averaged over 20 independent experiment runs.

TABLE II. TEST MEASUREMENT RESULTS

| | TRANSMIT POWER | | | | | | |
|---|---|---|---|---|---|---|---|
| | **-100dB** | **-110dB** | **-115dB** | **-117dB** | **-118dB** | **-119dB** | **-120dB** |
| **SINR(dB)** | 8.13 | 2.43 | -1.09 | -1.87 | -2.48 | -2.11 | -2.46 |
| **RSRQ(dB)** | -11.23 | -13.70 | -16.43 | -18.26 | -18.75 | -19.11 | -19.49 |
| **1kB Message** | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| **10kB Message** | 1.00 | 1.00 | 1.00 | 1.00 | 1.18 | 1.16 | Fail |
| **100kB Message** | 1.03 | 1.00 | 1.00 | 1.00 | 1.18 | 1.19 | Fail |
| **1MB Message** | 1.02 | 1.01 | 1.01 | 1.01 | 1.25 | 1.15 | Fail |

The results of the testing show that TCP overages occur predominantly when the RF link power from the CMW 500 falls below -117 dB. From -118 dB to -119 dB the larger TCP messages incur from 16% to 25% of additional overage.

It is interesting to note that the 1kB message consistently achieves a 1:1 ratio throughout even at the -120 dB level where the larger messages fail to transfer. Beyond -120 dB the radio module and CMW 500 can no longer reliably maintain a connection. The 1kB message size is representative of the RF Fidelity Layer urgent classification of real-time information that must be transacted at all costs (and without much TCP overage).

There is a positive correlation to increasing TCP overages when the received signal power is less than -117 dB and the SINR measurement falls below -2dB. These conditions could indicate to the RF Fidelity Layer algorithm that only urgent classification messages should be transacted. Above this range would indicate that the RF fidelity is within the normal classification of messages. Moving up further – to allow a guard band in the case of changing RF conditions during larger transfers – would indicate condition suitable for larger messaging as needed for OTA and maintenance files.

## V. Conclusion and Future Work

This paper presented preliminary lab testing to quantify the TCP-retransmission rate caused by a lossy radio link as it is experienced by a typical IoT LTE wireless device. These results indicate that the proposed RF Fidelity Layer is a promising approach to lowering subscriber overcharges and unnecessary battery consumptions at the wireless device. A complementary RF Fidelity Layer at the host is also proposed, which can discern the fidelity of the radio link based upon analysis of data classifications from the wireless device, hence providing similar benefits when carrying out unbidden host communications.

While these results indicate that the proposed research method has merit, the range and sensitivity to radio link fidelity may be improved with additional radio stack probes. The next phase of research is going to expand the scope of available probes into the algorithmic fidelity prediction procedure focusing for example upon developing additional radio stack probe information to incorporate into the predictive algorithm in order to improve the detectable band of degraded wireless link fidelity. Finally, collection of real-world field data – by building out test units that can log real-time probe information and send out measured TCP messages of various sizes to a back-end host – will be used to validate the proposed concept in the field. Comparisons between various LTE module manufactures will also be studied. This validation will enable synchronization of the TCP traffic traces collected at the host router using Wire Shark with the carrier billing record. Common RF interference sources will be field tested to quantify the impact on TCP data overages with and without the proposed RF Fidelity Layer. One such source of interference can be a nearby brushed motor as it was experienced during the lab testing when a drill press was turned on in an adjoining room. Even in the isolated screen room the testing was adversely impacted with higher than expected TCP errors and overages.

## References


[1] 3GPP TS 36.214 version 11.1.0 Release 11 (2013), “LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements.”

[2] Y. Go, J. Won, D.F. Kune, E.Y. Jeong, Yongdae Kim, and KyoungSoo Park, “Gaining Control of Cellular Traffic Accounting by Spurious TCP Retransmission,” in *Proc. 21st Annual Network and Distributed System Security Symposium (NDSS’14)*, San Diego, CA, USA, February 2014.

[3] Y. Go, D.F. Kune, S. Woo, K.S. Park, Y. Kim, “Impact of Malicious TCP Retransmission on Cellular Traffic Accounting,” in *Proc. 5th Annual Wireless of the Students, by the Students, for the Students Workshop (ACM S3 2013)*, Miami, FL, USA, September 2013.

[4] S. Biaz, N.H. Vaidya, “Distinguishing congestion losses from wireless transmission losses: a negative result,” in *Proc. 7th International Conference on Computer Communications and Networks*, Lafayette, LA, USA, October 1998.

[5] X. Chen, H. Zhai, J. Wang, Y. Fang, “A survey on improving TCP performance over wireless networks,” in *Resource management in wireless networking*, Boston, MA, Springer US, 2005, vol. 16 of the series Network Theory and Applications, pp. 657-695.

[6] C.-F. Chiasserini, M. Meo, “A Reconfigurable Protocol Setting to Improve TCP Over Wireless,” *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, pp.1608-1620, Nov 2002.

[7] H.Jiang, Z. Liu, Y. Wang, K. Lee, I. Rhee, “Understanding bufferbloat in cellular networks,” in *Proc. ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, Helsinki, Finland, August 2012.

[8] C. Peng, G.-H. Tu, C.-Y. Li, S. Lu, “Can we pay for what we get in 3G data access?,” in *Proc. Mobicom 2012* , Istambul, Turkey, August 2012.

[9] Y. Go, D.F. Kune, K.S. Park, Y. Kim, “Towards Accurate Accounting of Cellular Data for TCP Retransmission,” in *Proc. 14th Workshop on Mobile Computing Systems and Applications*, Jekyll Island, GA, USA, February 2013.

[10] F. Afroz, R. Subramanian, R. Heidary, K. Sandrasegaran, S. Ahmed, “SINR, RSRP, RSSI AND RSRQ Measurements in Long Term Evolution Networks,” *International Journal of Wireless & Mobile Networks (IJWMN)*, vol. 7, no. 4, August 2015.