Upper Bound on LoRa Smart Metering Uplink Rate

Christopher Paolini
Deptartment of Electrical and Computer Engineering
San Diego State University
San Diego, CA, USA
paolini@engineering.sdsu.edu

Hrishikesh Adigal
Deptartment of Electrical and Computer Engineering
San Diego State University
San Diego, CA, USA
hadigal@sdsu.edu

Mahasweta Sarkar
Deptartment of Electrical and Computer Engineering
San Diego State University
San Diego, CA, USA
msarkar2@sdsu.edu

***Abstract*— The use of the LoRa PHY and LoRaWAN MAC layer as a wireless communication standard for IoT devices has seen growth over the last few years, with long range, low power, and low cost being the driving force. LoRa is viewed as an alternative to higher power and commercial communication standards used for IoT applications. LoRa is now beginning to see use in low data rate, long range devices for consumer smart metering and demand response applications. Smart meters measure local electrical attributes such as power factor and instantaneous, RMS, and peak voltage, current, and power (real and reactive). These attributes are sampled and transmitted to an electric utility as uplink messages at the highest possible rate to allow the utility to maintain the closest approximation to the real-time state of an electric grid. Smart meters can also measure line frequency and detect voltage sags and current faults, which can be used to predict an impending outage. LoRa transceivers embedded within residential smart meters send uplink messages to local, utility operated neighborhood concentrators that are connected to a WAN and ultimately connect back to a utility's LAN. Concentrators serve as message gateways that copy a received LoRaWAN uplink payload into a MQTT publish message that is sent to a utility's application server. Steady state requires the service rate μ of the combined concentrator and application server to exceed the uplink arrival rate λ. In this paper we derive an upper bound on the measurement uplink rate for LoRaWAN smart metering infrastructure such that the ratio $\rho = \lambda/\mu$, or utilization factor, is maintained under 1, based on LoRaWAN packet time-on-air (ToA) and expected packet delay D_p incurred from the combined concentrator and application server expected service rate, and the concentrator expected packet arrival rate.**

Keywords— LoRa; LoRaWAN; Spreading Factor; Code-Rate; Smart Meter; Smart Grid; Advanced Metering Infrastructure, Data Concentrators; Packet Delay; Packet Service Rate; Packet Arrival Rate

I. INTRODUCTION

The *Internet of Things* or IoT refers to a network of physical devices connected to the Internet that can exchange messages with each other [1]. A *smart grid* is an IoT enabled electric grid which facilitates efficient and reliable power generation [2]. One of the major elements of a smart electric grid is Advanced Metering Infrastructure (AMI) which includes smart metering devices. With the advancement in long-range, low-power communication technology facilitated by the LoRa physical layer, there has been recent interest in deploying LoRaWAN for use in smart electric grid communications, since LoRaWAN can support bidirectional message exchange between energy providers and consumers, which is capability needed to implement demand response [2]. LoRaWAN is now being viewed as one of the most promising wide area network wireless communication technologies for use in home automation applications [3]. In this paper we present our work in progress toward deriving an upper bound on the maximum rate at which LoRa based smart metering infrastructure can transmit uplink messages containing power measurement payloads to an electric utility. We hypothesize the maximum uplink rate is primarily a function of packet *Time on Air* (ToA) and the combined concentrator plus application server service rate μ. ToA is primarily a function of spreading factor (*SF*), payload length (*PL*), and Coding Rate (*CR*). Service rate μ is affected by the Message Queuing Telemetry Transport (MQTT) protocol publish-subscribe time required for a concentrator to publish a message to an application server. We have developed a testbed of LoRa devices configured with different SF values and measured ToA and average packet interarrival time and service rate. To measure ToA under different configurations of SF, we developed testbed measurement code that allows us to dynamically and remotely change the SF of a device on demand using LoRaWAN's Over the Air (OTA) update capability. The ability of a utility to dynamically vary each consumer's residential smart meter SF, CR, *data rate* (DR), and *bandwidth* (BW) parameters will allow the utility to determine the *optimal configuration* for a set of smart meters in a given residential area that report to the closest concentrator in that region. By optimal configuration we mean the set of parameter values that achieve the greatest uplink rate. This paper provides an overview of these four LoRa communication parameters which contribute to the time taken for a LoRaWAN device to transmit an uplink packet to a LoRa concentrator, also known as a *gateway*, and subsequently from the gateway to a service provider's application server over a LAN. We summarize key features of LoRa and LoRaWAN in section II. The testbed and its implementation are discussed in section III. Results and analysis are presented in section IV and section V features our conclusion and future research work.

II. LONG RANGE (LORA) - NETWORK TOPOLOGY

The word LoRa is a portmanteau of "Long Range". LoRa is a spread spectrum modulation scheme developed by Semtech Inc., which is appropriate for wireless communication applications that require low-power, low data-rate, and long range. LoRa is an alternative to modulation schemes such as FSK and PSK, and has become one of the widely used wireless communication technologies for Internet of Things (IoT) devices, which typically require low power and long-range communication between the "things" (i.e. end devices) and an Internet accessible server [4]. LoRa is a proprietary spread spectrum modulation scheme PHY implementation based on Chirp Spread Spectrum modulation (CSS) [4]. The key properties of LoRa modulation are scalable bandwidth, low-

978-1-7281-3893-0/20/$31.00 ©2020 IEEE

power, high robustness, fading resistance, long range capability, enhanced network capacity, and localization [4].

LoRaWAN is a non-proprietary wide area network protocol used with LoRa modulation at the network (MAC) layer. LoRaWAN can operate in one of three different communication classes at a given time – Class A, Class B and Class C. Class A is the default class of communication, is asynchronous, and always initiated by an end device through one uplink transmission followed by two downlink transmissions [5]. Downlinks are short messages received by an end device which are occasional and enable bi-directional communication at minimal power. In advanced metering infrastructure, downlinks provide a mechanism for a utility to initiate demand response. Uplink messages can be sent at any time from an end device to a gateway. Class B and Class C communication is not suitable for battery powered metering applications due to higher power consumption [6]. In addition to SF, BW, and CR, two other primary configurable parameters of a LoRa transceiver are carrier frequency (CF) and transmission power (TP). SF controls the data rate (symbol rate) and can have values between 7 to 12, depending on geographic region. BW is related to SF through the relation

$$BW = 2^{SF} / T_s \text{ , s}^{-1} \text{ or Hz} \quad (1)$$

where T_s is the symbol time in seconds. As SF increases, bit rate decreases and receiver sensitivity increases, which affects device energy consumption. CR determines the proportion of data with respect to error correction bits in the data encoded onto a signal [4]. CR has significant influence on ToA: the higher the CR the more overhead is added to the message length which increases ToA. LoRaWAN permits the four CR configurations of 4/5, 4/6, 4/7 and 4/8. As the carrier frequency CF increases, ToA decreases. The government allowed transmission power (TP) for LoRa devices ranges from 2 to 20 dBm. There are differences between LoRa standards based on geographic region. In the European Union, LoRa is assigned the 863 MHz to 870 MHz band, while in North America devices must operate within the 902 to 928 MHz band. The LoRaWAN packet frame is comprised of a preamble, an optional header, and a payload [4, 7]. There are two types of LoRaWAN packets: *explicit* and *implicit* [4]. Explicit packets contain a short header which incurs additional payload information, whereas the implicit packet does not contain a header [4]. The default mode of operation has the header included (i.e. the explicit packet type is used) [4].

III. TESTBED SYSTEM DESIGN

Our testbed consists of four end devices, a gateway, a network server, and an application Server. Fig. 1 shows an illustration of our system architecture. The end devices in Fig. 1 model a LoRa enabled smart meter. The gateway is used to bridge the communication between LoRa end devices and the network server. Our testbed end devices are MultiTech programmable development boards with a LoRa RF radio module and an Arm Cortex-M4 processor. These four devices are configured to continuously transmit uplink messages that model metering measurements to a MultiTech Conduit LoRa gateway. The Conduit has a LoRa RF module used to receive and send LoRa messages, and an Ethernet interface used to communicate with a network server over a LAN. The gateway uses the MQTT protocol to transmit a LoRaWAN packet payload as a MQTT publish message to an application server. The network server functions as a bridge between the gateway and the application server, and forwards received MQTT messages from the gateway to the service provider's application server and vice versa.

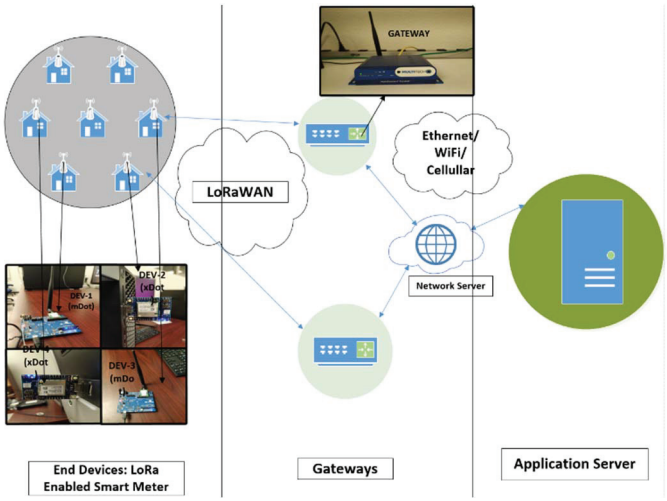

Fig. 1. Our testbed system architecture.

The *application server* is a system connected to a LAN reachable from the Ethernet interface of a gateway and is used to process measurement data sent from metering end devices via a gateway and network server. An application server can respond to an end device uplink message with a downlink message by publishing the downlink payload to the gateway's MQTT service.

We implemented an Over the Air (OTA) update application for end devices which configures a desired SF, CR and TP. Based on the SF, a fixed PL is assigned before the start of transmission. The payload data consists of the end device transmit timestamp and variable-length data based on the SF value. A 10B(yte) timestamp is assigned in the packet payload field before transmission. The devices in our testbed are configured to operate using class A communication mode. We have used Node-Red to deploy an application on the gateway to receive LoRa packets and forward them to the application server. The Node-Red flow on the gateway is comprised of a LoRa Rx node, a special function node to parse the LoRa packet and create a payload for an MQTT publish message, and a MQTT publish node to forward the MQTT message to the application server. The flow also consists of a MQTT subscribe node to receive messages published from the application server. The MQTT message consists of the LoRa payload, the timestamp the LoRa message was received at the gateway, and the end device ID to differentiate LoRa packets among multiple end devices. There are two applications running on the application server: the MQTT subscribe process and the MQTT publish process. The MQTT publish process is used to transmit the server timestamp to configure each end device real-time clock (RTC) and gateway with respect to the application server time. This step is done during initial setup before the MQTT subscribe application is run on the server in order to synchronize the RTC on each end device with respect to the application

server. The MQTT subscribe process on the application server stores (1) the timestamp of the LoRaWAN packet transmitted from the end device, (2) the timestamp of when the LoRaWAN packet was received by the gateway, and (3) the timestamp of when the MQTT publish message was received by the application server, all to a log file for analysis.

IV. RESULTS AND ANALYSIS

We have used scatter plots to show the difference in time between the three timestamps and compare our results with theoretical ToA, which is the sum of preamble time $T_{preamble}$ and payload time $T_{payload}$ [4]:

$$ToA = T_{packet} = T_{preamble} + T_{payload} = \left[\underbrace{(N_{PP} + 4.25)}_{\text{preamble symbols}} + \underbrace{\left(8 + \max\left(ceil\left(\frac{8PL - 4SF + 28 + 16CRC - 20H}{4(SF - 2DE)} \right)(CR+4), 0 \right) \right)}_{\text{payload symbols}} \right] T_s \quad (2)$$

where N_{PP} is the number of programmed preamble symbols in a LoRa message packet and T_s is the symbol duration in seconds. $T_{payload}$ is the number of symbols required to create the payload and message header [4]. Data Rate Optimization *DE* is 0 when low optimization data rate is enabled and 1 when disabled. *H* is 1 to configure implicit mode and 0 when the header is disabled (explicit mode). The header is always enabled for LoRaWAN. *CRC* is 1 if cyclic redundancy check is enabled and 0 if disabled. *CRC* is enabled by default for LoRaWAN. Using equation (2) and the values of SF and CR used in end device configuration, we computed the theoretical ToA for packets transmitted in our testbed and compared this value with the actual time on air according to timestamp differences between end devices and the gateway.

Fig. 2 is a scatter plot that shows the measured latency between packets sent by end devices (indicated as *LoRa Tx*) and received by the gateway (*LoRa Rx*), and the latency between end devices and the application server (*Application Server Rx*). These latency measurements provide information about how frequently an end device can transmit a metering uplink message to the gateway at a particular SF. In Fig. 2 starting from top, the blue colored dots represent the end device's transmit timestamp of each packet sent. The x-axis represents time and a dot is rendered according to the timestamp value. The black dots represent the gateway (GW) receive (Rx) timestamp based on theoretical ToA. The green dots represent the actual GW Rx timestamp and red dots represent the MQTT subscription timestamp. The scatter plot compares the performance of LoRa enabled end devices operating at SF8BW125 (spreading factor 8 with a bandwidth of 125 MHz), SF9, SF10 and SF7. We have plotted data for 20 LoRa packets received at the gateway. Fig. 3 shows a comparison between theoretical ToA computed using eq. (2) and actual ToA from differences in timestamps, at different SF values and payload lengths.

From the results shown Fig. 2 and Fig. 3, end devices configured to transmit at SF8 and SF10 with a maximum payload length of 125B and 11B, respectively, have more accurate results compared to other SFs. The number of packets transmitted successfully at SF8 and SF10 at 125kHz show a lower delay between two successful transmissions compared to others. The actual packet delay was seen to be lower for SF8 and SF10, and a greater number of packets were successfully received by the gateway at these two SF configurations. We kept the CR value at 1 (a 4/5 configuration) for all our measurements, as a higher CR will increase ToA according to eq. (2).

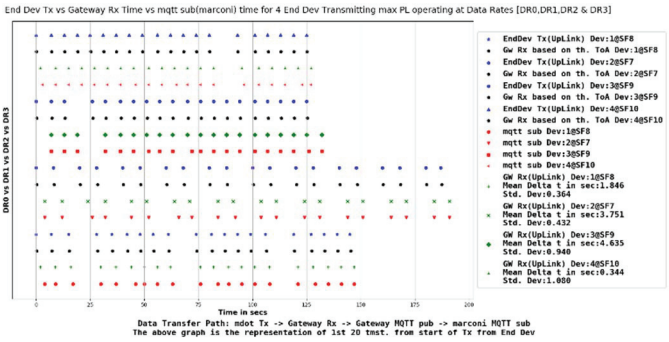

Fig. 2. Scatter plot showing end device LoRa packet transmit time (Tx), gateway receive time (Rx), and application server MQTT subscribe time, for end devices operating at four different values of SF.

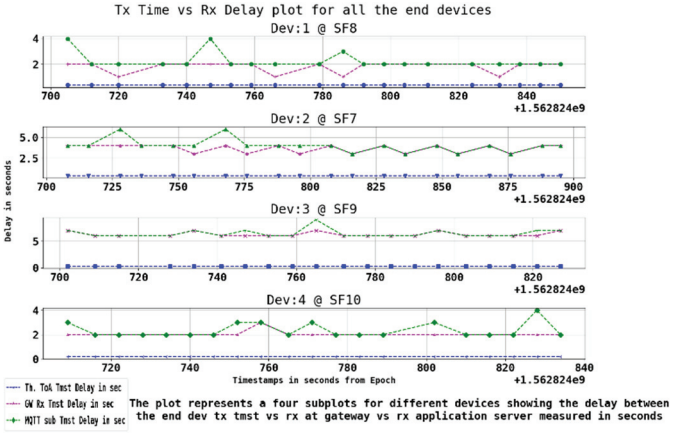

Fig. 3. Comparison of theoretical and measured LoRa packet delay between an end device and a gateway, and an end device and an application server, for four different values of SF.

Table 1 compares theoretical ToA with respect to actual delay in LoRa message reception.

TABLE I. THEORETICAL TOA VS ACTUAL TOA

Spreading factor and bandwidth	Tx data rate, kbps	Theoretical ToA, millisecond	Actual average ToA in seconds with 5 sec delay between 2 transmissions
SF8 500kHz	12.50	163	5.64
SF7 125kHz	5.47	374	3.75
SF8 125kHz	3.16	369	1.89
SF9 125kHz	1.76	308	4.63
SF10 125kHz	0.98	206	0.34

We have used latency measurements from Table 1 to model the expected service time for a metering uplink message. We model the combined gateway plus application server as a M/M/c/∞/∞ queue to compute the average system wait time,

where the first M is packet arrival rate λ [packets/hour] modeled as a Poisson process with exponential inter-arrival times, the second M is packet service rate μ [packets/ hour], c is the number of servers (1 combined gateway/application server, therefore c = 1), ∞ represents an infinite queue length, and the second ∞ represents an infinite population. We measured the number of packets arriving at the gateway per hour to compute interarrival rate λ, and measured the elapsed time from gateway reception to application server acknowledgement to estimate service rate μ. We used the relation between the arrival rate and service rate to compute the traffic intensity ρ, which is the ratio of arrival rate to service rate, λ/μ. For an infinite queue length system, this ratio must remain less than 1 to operate in steady state, as a value greater than 1 indicates there would be at least one packet which will remain unserved at the end of the queue, making the system unstable. We computed the interarrival time $1/\lambda$ in seconds between two packets. The expected number of packets L_s in the system is computed as a sum of products of the number of packets j and the probability of exactly j packets present in the system, P_j,

$$L_s = \sum_{i=0}^{\infty} jP_j \tag{3}$$

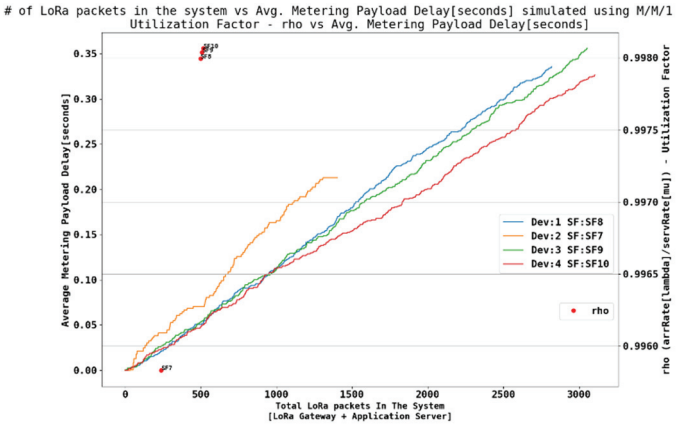

Fig. 4. Expected packet delay D_P, or wait time, as a function of the total number of packets in the system, for end devices operating at four different values of SF. Red dots show utilization $\rho<1$ for SF7-10, with λ bounded by μ.

The $M/M/1$ queuing model is a memoryless system and does not depend on earlier states of the system to define the current state. P_j is based on the number of packets in the system at time $(t+h)$, where h is a small interval of time during which only one event can take place (i.e. either the arrival or service of a packet, but not both). The probability of there being n number of packets at time $(t+h)$ is given by [8]

$$P_n = \rho P_{n-1} = \rho^n P_0 \tag{4}$$

where P_0 is the probability of zero packets present at time $(t+h)$. From (3) and (4), the expected number of packets in the system L_s is

$$L_s = \rho/(1-\rho) \tag{5}$$

From (5), according to Little's law, the expected wait time W_s in seconds for a packet in the system is [8]

$$W_s = L_s/\lambda \tag{6}$$

Using (6) we computed the average wait time W_s, or equivalently the expected packet delay D_p, for a packet in the system at different spreading factors assigned to end devices.

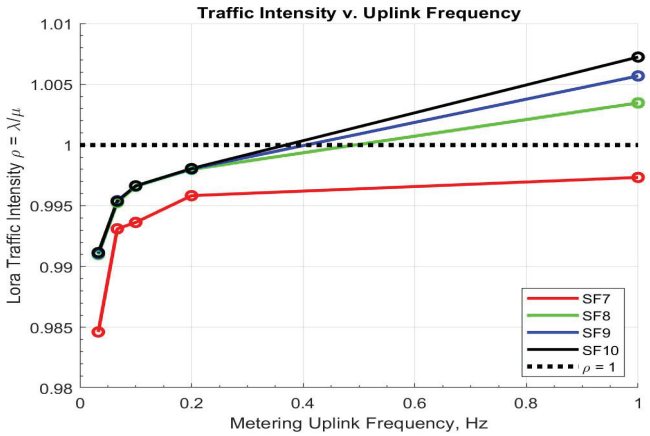

Fig. 5. Traffic intensity ρ as a function of uplink frequency (i.e. rate of transmitting uplink packets with 11B metering data), at four spreading factors.

Fig. 4 shows a plot of expected packet delay as a function of the total number of packets in the system.

We measured traffic intensity at five uplink frequencies by varying the rate our testbed end devices transmitted 11B metering updates to our application server. At $\rho<1$, the gateway/network server and application server will experience some idle time with all uplink packets processed. For frequencies where $\rho=1$, indicated by the black dashed line in Fig. 5, all metering uplink packets are processed while the servers are always busy. For frequencies where $\rho>1$, the servers are always busy and not able to process all arriving uplinks, which ultimately leads to system failure and loss of grid state information.

V. CONCLUSION

To maintain awareness of electric grid state, utilities require near real-time updates from smart metering devices residing at consumer homes and businesses to make predictions about future peak demand and perform demand response. We measured traffic intensity at different LoRaWAN uplink frequencies and showed the maximum metering uplink rate is bounded by $\rho\leq1$ and will vary by end device spreading factor.

REFERENCES

[1] Oracle. "What Is IoT?" https://www.oracle.com/internet-of-things/what-is-iot.html (accessed 09/21/2019.

[2] S. Aman, Y. Simmhan, and V. K. Prasanna, "Energy management systems: state of the art and emerging trends," *IEEE Communications Magazine,* vol. 51, no. 1, pp. 114-119, 2013, doi: 10.1109/MCOM.2013.6400447.

[3] J. Laveyne, G. Van Eetvelde, and L. Vandevelde, *Application of LoRaWAN for smart metering: an experimental verification*. 2017.

[4] Semtech Corporation, "LoRa Modem Design Guide SX1272/3/6/7/8," July 2013.

[5] LoRa Alliance, "A technical overview of LoRa and LoRaWAN," in "Technical Marketing Workgroup 1.0," November 2015.

[6] N. Varsier and J. Schwoerer, "Capacity limits of LoRaWAN technology for smart metering applications," in *2017 IEEE International Conference on Communications (ICC)*, 21-25 May 2017 2017, pp. 1-6, doi: 10.1109/ICC.2017.7996383.

[7] A. Augustin, J. Yi, T. Clausen, and M. W. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors,* vol. 16, no. 9, 2016, doi: 10.3390/s16091466.

[8] R. B. Sørensen, D. M. Kim, J. J. Nielsen, and P. Popovski, "Analysis of Latency and MAC-Layer Performance for Class A LoRaWAN," *IEEE Wireless Communications Letters,* vol. 6, no. 5, pp. 566-569, 2017, doi: 10.1109/LWC.2017.2716932.