LiFi BPLR Network

Whitepaper

Version 2.0

January 2022
Abstract

Reliable transmission of large or bulk packet data over Lora is considered to be challenging due to insufficient data rates and payload size.

The current LoRa protocol under service network service providers such as Helium, Loriot and TTN is to transmit bulk data in a sequence of packets, with each packet being individually acknowledged.

This methodology makes LoRa unsuitable for a multitude of applications as the transmission time of bulk packets is significant due to the wait time for each individual acknowledgement. The acknowledgements also increase network traffic.

In order to facilitate the transmission of bulk data including videos, image and text over LoRa, LiFi propose a new protocol entitled “BPLR” (Bulk Packet LoRa).

Used in conjunction with a channel reservation protocol, “BPLR” has been proven to reduce the transmission time of bulk data over LoRa by up to a factor of 70 whilst also remaining within the confines of nation-specific network constraints.

LiFi intend to build a P2P network of gateways to support the global role out of “BPLR” and incorporate blockchain technology to make bulk data available to developers post-transmission.
Introduction

Due to the limited data rate and payload size, a 1MB file would typically take a minimum of 498 seconds to transmit, even at its peak physical layer rate. This incurs high costs, high energy usage, high network load and high collision rate when using the standard LoRa MAC ALOHA protocol.

Furthermore, some jurisdictions enforce duty cycles limitations.

BPLR is designed to be a reliable delivery protocol for LoRa.

BPLR batches data packet transmissions and incorporates bit-vector acknowledgement packets that provide the reception status for each individual packet within the batch. This vastly improves upon the current “stop and wait” protocol of standard LoRa.

BPLR utilises common data compression techniques such as J-LAW and JPEG 2000 to ensure each packet within a batch complies with individual packet limitations whilst also incorporating blockchain concepts to make the data easily obtainable by the data owner and in an immediately usable fashion.

Blockchain is also used to build a reward mechanism for P2P hosts of BPLR gateways thus ensuring a symmetrical network growth, globally.

BPLR has been tested and implemented using a test bed LoRa Network. With no packet loss and point to point transmissions, BPLR reduced bulk packet transmission times by a factor of 42. In instances of packet loss incorporation, the bulk packet transmission times were reduced by a factor of up to 70 times, leading to a 1MB transmission of around 7 seconds which is comparable to most 2.4ghz transmissions.

BPLR also incorporates a channel reservation protocol for when there are situations of isolated gateways handling batched data packets from multiple devices.

BPLR in conjunction with channel reservation facilitates the sending of many more bulk data packets than ALOHA or “Stop and Wait” due to the negation of the network load associated with acknowledgement packets.
LoRa

The approximate signal range of LoRa in a rural environment (line of sight) is approximately 10-15km and 3-5km in an urban environment.

Spreading Factor (SF) and Bandwidth (BW) are the most important considerations in LoRa, and different combinations of these leads to different data rates. The physical layer data of LoRa can be given by the following expression:

\[ SF \times \frac{BW}{2^{SF}} \times CR, \]

Here “SF” is the Spreading Factor, “BW” is the bandwidth and “CR” is the Coding Rate.

Using this equation we can calculate the physical layer data rate of LoRa for given values of SF, BW and CR.

An example is set out below where “CR” is 4/5

<table>
<thead>
<tr>
<th></th>
<th>500 kHz</th>
<th>250 kHz</th>
<th>125 kHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF7</td>
<td>21.88</td>
<td>10.94</td>
<td>5.47</td>
</tr>
<tr>
<td>SF8</td>
<td>12.5</td>
<td>6.25</td>
<td>3.12</td>
</tr>
<tr>
<td>SF9</td>
<td>7.03</td>
<td>3.52</td>
<td>1.76</td>
</tr>
<tr>
<td>SF10</td>
<td>3.91</td>
<td>1.95</td>
<td>0.98</td>
</tr>
<tr>
<td>SF11</td>
<td>2.15</td>
<td>1.07</td>
<td>0.54</td>
</tr>
<tr>
<td>SF12</td>
<td>1.17</td>
<td>0.59</td>
<td>0.29</td>
</tr>
</tbody>
</table>
Protocol Design
Bulk Packet LoRa (BPLR)

BPLR utilises multiple data compression techniques dependent upon the nature of the transmitting device, these include JPEG2000 for image, A-LAW / J-LAW for voice and MJEPG for video.

Despite the best efforts of these data compression techniques, most data sets are too big to fit in a single LoRa packet. Therefore a data file must be transmitted using multiple packets.

A 10kb image for example must be encoded and segmented into at least 56 MTU packets before being transmitted over LoRa.

Under the standard LoRa “stop and wait” method as deployed by network providers such as Helium and Loriot the sender waits for an acknowledgement per packet to verify that the data arrives correctly. The impact of this is that the packet delivery rate is severely limited, power consumption of sender and receiver is increased significantly, the network load increases and in turn the required transmission power of the receiver.

The objective of BPLR is to reduce the amount of acknowledgement messages that need to be sent and the cumulative time spent waiting for these messages.

To achieve this objective, BPLR utilises batched packet transmission and bit vector acknowledgements.

Stop and Wait Protocol (Utilised by all current networks) Fig 1
BPLR Protocol Fig 2
As shown in figure 2, dependant on the size of the transport window, the sender sends a batch of data packets to the receiver consecutively.

The correctness of each packet is verified through the checksum in the protocol header and the forward error correction in the LoRa physical layer.

The status of each packet delivery is returned to the source via a bit vector acknowledgement packet (BVACK) that contains a bit for every packet in the batch. If the bit in a particular index position is “0” then the packet was received and “1” if not. This functionality allows the sender to determine which data packets are lost or corrupted.

**Intended Packet Format (Figure 3)**
The intended packet format consists of the header and the data packet payload.

**The Header**

- 16 Bytes
- 4 Bytes Destination EUI
- 4 Bytes Source EUI
- 1 Byte Service Number
- 2 Bytes Sequence Number
- 1 Byte Flag
- 1 Byte Payload Size
- 1 Byte Batch Size
- 2 Bytes Checksum
Destination EUI and Source EUI are the unique identifiers of destination and source device, respectively. The Service number identifies the service to which the data packet belongs. The Sequence number is used to order the packets. The Flag field indicates the packet type. The possible values are SYN, SYN-ACK, DATA, BVACK, FIN, and ACK. Payload Size and Batch Size describe the length of the data payload and the size of the current batch, respectively. Finally, Checksum is used to check the correctness of the header. The data payload portion still has a space of up to 239 bytes. This header covers the functionalities of both data link and transport layers. The overhead of this 16-byte header is only 6.3%.

Figure 4- BPLR in Single Hop Data

1. Connection established between sender and receiver
2. Sender sends all packets in first batch and waits for BVACK from the receiver
3. The sender sends the next batch including any retransmissions from the original batch
4. When all packets have been sent a 4-way handshake is performed through FIN and ACK packets
5. Receiver returns to listen mode.
Channel Reservation

BPLR can reduce image transmission time significantly however batched packets can increase the duty cycle and the potential for packet collisions if there is a single gateway handling multiple requests from nodes.

Under standard LoRa network protocols such as Helium and TTN, collisions due to congestion from data packets from other devices will affect the goodput of the transmission (rate of correctly received packets).

Under BPLR we can eliminate all packet collisions using a protocol entitled “channel reservation”.

Figure 5 - Data Channel Reservation Timing

The channel reservation protocol can be implemented on gateways that listen on one or multiple channels.

As per figure 5, a gateway operating on BPLR will continuously listen for request packets on the control channel. When the gateway receives a request packet it pseudo-randomly selects a data channel and informs the sending node of this channel. Both the gateway and the sending node switch to the data channel and perform the BPLR BVACK transmission. When the transmission is complete, both the gateway and the node switch back to the control channel and the gateway will sit idle until a new request is sent.

When a gateway is operating on the data channel any requests sent to the control channel will not be received and will instead be routed to the next nearest gateway or declined if there are no other gateways in range.
The data channel is not chosen entirely at random. Gateways learn what data channels have higher packet loss rates and will actively avoid selecting these channels for the data channel. This approach avoids the use of channels that are used by other devices in the receiver range.

**Figure 6 Data Channel Reservation Protocol**

1. Node sends a SYN packet to the gateway on the control channel to request connection.
2. If the request is not received, the node will enter sleep mode for a random time before its next connection attempt.
3. Gateway receives the request and assigns a data channel, this is indicated by the acknowledgment packet.
4. Gateway switches to data channel and waits for node to confirm.
5. Node switches to the data channel after receiving SYN-ACK and also sends ACK.
6. Gateway sends a READY message to ensure that the node does not start broadcasting data before the gateway is ready to receive.
7. Node receives READY message and begins to broadcast.

Implementing this protocol also benefits the reward mechanism as it ensures a wider distribution of data packets that contribute to the compensation tokens.
Transmission Test Results

Image Compression

3280x2464 full-resolution image of 5.8 MB. We use Pillow, a Python imaging library, to resize the image to 480x320 pixels, and to apply JPEG compression.

BPLR VS ALOHA

No Packet Loss

28kb

SF7  BW 500  BPLR- 0.36 seconds | ALOHA- 25.37 seconds
SF8  BW 500 BPLR- 0.91 seconds | ALOHA- 53.57 seconds
SF9  BW 500 BPLR 1.57 seconds | ALOHA- 88.51 seconds
SF10 BW500 BPLR 1.88 seconds | ALOHA- 106.17 seconds
SF11 BW500 BPLR 23.02 seconds | ALOHA- 207.87 seconds

The net result of introducing BPLR as a LoRa protocol is the ability to transmit larger data packets within a much smaller transmission timeframe, this also allows for more bandwidth options and spreading factors whilst retaining a much more efficient transmission over the standard LoRa.
Applications

At the peak transmission times, BPLR can transmit over LoRa at a rate of 82kb per seconds, when utilising decompression, this equates to 17.4mb per second, a speed which is close to the average 2.4ghz and 4G speeds (real world).

Being able to transmit over a “free-to-air” frequency at such a rate has significant potential for next gen applications and hardware as well as conversion of current applications operating on “paid” frequency bands.

These applications and hardware are the focus of LiFi in respect of potential clients with examples including:

Hardware for image/video streaming including CCTV, doorbells, dashcam, drones, imaging nodes, video nodes;

Applications for secure text/data distribution including database routing and web 3.0;

Applications that facilitate image/video messaging/distribution that require instantaneous (relative) delivery with high level security (established only over sub 1ghz frequencies) including P2P messaging, social media and business to business functions.

The “marketplace” for potential applications and devices operating on BPLR are approximately 20 times greater in volume than standard networks such as Helium and Loriot due to the inclusion of devices and applications currently unsuitable for LoRa.

In the mid-term it is entirely feasible for communication devices to be available that operate solely on BPLR.

Of significant interest is the ability to deliver a fully decentralised communications network that has all the functionality of cellular.
Blockchain

LiFi BPLR network utilises blockchain technology in a number of ways.

NFT

One of the core fundamentals of LoRa and indeed any transmission network is to verify the authenticity of both sender and receiver, especially where sensitive/personal data is being transmitted.

Under LoRa (and referenced in figure 3) each data packet contains the sender device (source) EUI and also the intended recipient (destination) EUI. “EUI” being a unique identifier.

BPLR takes authentication one step further by converting each source node/device and recipient device into an NFT (Non-Fungible Token). This not only acts as an incorruptible EUI but also provides ownership credentials which are vital inside applications/devices that are built for the individual consumer.

The use of NFT’s also negates the requirement for addressing Byzantine faults within the verification protocol.

Data Post and Extraction

To be able to deliver a consumer ready, data transmission network it is vital to be able to deliver the transmitted data in an easy to obtain format, especially where application developers have a focus on end-user functionality.

Typically LoRa data is simply posted to databases or routed to servers in the raw message format. This approach is fine if the data payload arrives in its original format, however the reality of LoRa is that most data payloads must be compressed, this means that the data arrives in a format that must be decompressed before being available for general use. Likewise including server/database costs adds cost and vulnerability to applications that ultimately makes 2.4-5ghz a more cost effective option for transmission.

BPLR utilises blockchain to deliver a highly cost efficient storage and retrieval medium, whilst also combining the assignation of NFT’s and a decompression API to offer an “off the shelf” service that delivers bulk LoRa data in a readable format directly to the end-user.

When a recipient device receives the data packet under BPLR it records the receipt (BVACK) on our blockchain. The transaction data contains the source EUI, recipient EUI and the data payload. The data payload is the compressed bulk data. The data payload is entered as a “note” on each blockchain transaction.
The “note” section containing the compressed data is then decompressed by the BPLR API and the final data output is extracted for use by the end-user.

The holder of the “source” NFT is able to set (or have set for them by the device/app developers) routing rules for the final data output, these are dependent on the actual use case of the app/device.

**Reward**

To offer BPLR as an “off the shelf” network service there must be minimal entry costs to a developer.

The major entry costs to any LoRa network are the establishment of sufficient “recipients” (otherwise known as “gateways”).

To negate the entry costs of BPLR for designers, a P2P network of gateways will be established that provide the required recipients.

To incentivise peers to acquire and host a “gateway” BPLR again utilises blockchain by providing a reward incentive in the form of a token issuance that is directly tied to the BVACK data stored on the blockchain.

**Hardware**

BPLR is not feasible without a gateway network and as such the hardware requirements for gateways are minimal with a view to driving rapid growth.

As BPLR is based on the open sourced “Chirpstack” network server there are two fundamental hardware requirements of which at least one must be present on any device wishing to join as a fee-earning gateway.

**Requirement 1- Semtech**

Any device wishing to join the BPLR network as a fee earning gateway must contain a Semtech LoRa chip/concentrator. BPLR can run on the “Semtech Packet Forwarder” firmware and this is available to any device operating on a Semtech LoRa chip/concentrator.

**Requirement 2- Chirpstack**
The device must carry the functionality of being able to install firmware entitled “Chirpstack Gateway Bridge” directly on to it, or alternatively have built in Chirpstack connectivity. This will allow the device to direct data packets to BPLR via a broker service called MQTT.

Providing one of the two prerequisites are met, any device, including self-builds can be added as a “fee-earning gateway to the BPLR network”.

**Subscription Model**

Launching a gateway on BPLR requires an annual subscription currently set at $240.

**Helium Gateway Conversion**

Almost all Helium gateways can be converted to BPLR but this will require image flashing to a default Semtech package and so recording an image of the Helium settings is advised.

**Tokenomics**

The incentivisation model for gateway hosting is a dual token methodology with built in protocols to drive the token price and issue protection to holders.

**Immediate Reward Token**

The immediate reward token is “LFX”.

LFX is hosted on BSC (LFX).

Initial minting: 50,000,000

Burn only protocol

5% awarded to founders upon minting
LFX token is the immediate reward token for gateway holders.

Token Release Schedule

Years 1-3: 8,500,000
Years 3-6: 4,250,000
Years 6-9: 2,125,000
Years 9-10: 1,062,500

Daily Rewards

20% Awarded to “Heartbeat”. This is the reward for the gateway simply being online and awaiting contact from nodes.

Heartbeats are verified by mobile BPLR nodes on a random basis.

80% awarded to “Data”.

“Data” is determined by what proportion of the entire network traffic a specific NFT (gateway) has registered in a set 24 hour period.

Burning Mechanism

Users of BPLR pay a “per packet” rate. This is a fixed value per MTU packet.

The price per MTU packet is $0.0000001

(A 5.8mb image compressed using JPEG2000 and without losing quality is around 10Kb.
10Kb = 56 MTU Packets)

28% of the daily accrued payments for MTU packets will be used to acquire LFX token. This volume of tokens will then be burned from supply.

Liquidity

A fixed value of liquidity (LFX:BUSD) will be provided by LiFi, this will be equivalent to 16% of the daily accrued payments, with a minimum contribution of $100 per day.

Liquidity can be provided via token holders adding to the LFX:BUSD liquidity pool.
Mezzanine Reward Token

The mezzanine reward token is “LFXR”

LFXR is hosted on BSC

Mint only protocol

LFXR is a reward token that is generated by staking or farming (via LP token) LFX

Staking or Farming (via LP) has a time dependant variable APY

The APY generated daily determines the volume of LFXR minted daily.

Incentivised Staking/Farming

Incentivising staking results in a more stable token;

Incentivising farming results in an increased liquidity ratio which directly impacts the token price

Incentives values are determined by the length of time that LFX token is committed to a staking or farming initiative.

LFXR Payment

LFXR is minted daily in direct correlation with the APY rewards generated.

LFXR is paid at the end of a fixed time period for staking/farming.

A maximum sale of 2% of LFXR balance can be sold per day.

Liquidity
To place a gateway on the BPLR network and qualify for rewards, an annual subscription of $240 must be paid. 47% of the subscription cost will be committed to the liquidity of LFXR (LFXR:BUSD) with a minimum daily commitment of $1500. The LP token for LFXR:BUSD will be assigned to a vested, locked contract.

LASCO

Project LASCO is the raison d’etre behind the creation of BPLR.

LASCO is the development of a P2P light fidelity network using Free Space Optical (FSO) devices (open source build).

Crucial to the development of LASCO is the ability for the FSO devices to adapt their positioning to micro climate changes and location specific events.

The only way of monitoring these events is via live video streaming in conjunction with AI.

LoRa in its current form could not support live video, even when using neural network filtering, however LoRa as a protocol, operating on sub 1ghz bands without reliance is ideal for FSO.

BPLR was derived from a desire to operate sensors over LoRa but with a requirement for filtered, compressed video.

Contributors

Chirpstack
Mosquitto
Zeromq
Semtech
Algorand
AWS
Vladimir Jiostoc
Slowmist
JAM

www.lifi-network.com