2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)

Survey of Data Exchange Formats for Heterogeneous LPWAN-Satellite IoT Networks

Ivan I. Lysogor, Leonid S. Voskov, Sergey G. Efremov

Abstract — **This paper discusses data interchange formats in the context of heterogeneous networks for the Internet of Things (IoT). The wide dissemination of IoT technologies into various industries, such as agriculture and mining, reveals data transfer issues in geographically remote locations due to absence of any network infrastructure. Several technologies like LoraWAN and NB-IOT offer extended communication ranges, however they still cannot fully solve the problem. In many cases satellite networks are the only available option for transmitting IoT data to a central collection point. Our research of satellite networks showed that as of today the Iridium Short Burst Data (SBD) network is one of the best technologies suited for IoT applications. However, the SBD imposes a significant limit on the size of transmitted messages, which turns data format selection into a vitally important task. We developed a simulation model as well as a heterogeneous Iridium-LoRAWAN prototype to compare different data exchange formats. Our experiments showed more than 4 times increase in the amount of data transferred with Protocol Buffers, compared to the widely used JSON format.**

Index Terms — **Internet of Things, satellite networks, data exchange formats, LoraWAN, Protobuf, Iridium.**

I. INTRODUCTION

The number of physical IoT objects is growing exponentially and is expected to reach 50 billion by 2020 (Cisco forecast). Several IoT enabling standards and technologies are utilized at the lower OSI stack levels, which include IEEE 802.15.1, IEEE 802.15.4, LPWAN and others. All these wireless networks can be deployed in areas without any backbone infrastructure, however one or several data collection points still need an interface to the Internet.

In our research we focus on a heterogeneous IoT architecture consisting of a LoraWAN network for data collection from wireless sensors and a satellite channel used to transfer all data to centralized aggregation point. This design allows to deploy IoT systems in geographically remote locations for agriculture, mining and transportation [1].

Optimized data exchange in heterogeneous network allows to significantly increase data transmission efficiency and to extend applicability of IoT technologies.

Our paper is organized as follows. We first analyse IoT protocols for long-range communication. We then discuss the possibilities of data transmission through satellite networks. In section IV we give an overview of the main data exchange formats for IoT applications. Finally, we present our simulation model and the main experimental results of comparing different data exchange formats.

II. LONG-RANGE NETWORKS FOR THE INTERNET OF THINGS

When building any distributed system in IoT, one of the main tasks is related to choosing an appropriate underlying technology for sensor data acquisition, since it affects scalability, reliability and energy efficiency of the whole system. In addition, a network protocol imposes limitations on the amount of data that can be sent. There are plenty of IoT network protocols, which differ in transmission range, bandwidth, frequency range, signal modulation and other characteristics.

In our study we used the following criteria to select the best suited protocol for a heterogeneous IoT network:

1. Energy efficiency
2. Applicability in remote areas
3. Network coverage
4. Operation in an unlicensed frequency band

Recent research results [2] show that Low Power Wide Area Networks (LPWANs) satisfy all four requirements. The LPWAN network advantages are:

- network coverage. LPWAN networks were initially designed to support long range data transfer. For example, point-to-point distances can be up to 10 km and higher [4];
- energy efficiency. LPWAN networks are more energy efficient in comparison with other long-range networks.

One of the commonly used protocols in LPWAN networks is LoRa with several extensions, e.g. LoRaWAN. LoRa uses unlicensed frequencies up to 1 GHz (frequency range is country specific). The LoRaWAN protocol provides a solid solution in terms of network coverage and energy efficiency in comparison with other LPWAN protocols.

An alternative protocol used in IoT data networks is the NB-IoT. This protocol was developed by the 3GPP consortium and is integrated into the LTE standard. It is aimed at increasing energy efficiency and providing support for autonomous devices. Compared to the LTE technology, NB-IOT does not have roaming support and aggregation of frequency ranges, which normally improves the overall performance. In addition, it requires an existing cellular network to operate and cannot be used in remote areas.

Since we were looking for a protocol to be used in distant locations with extended area coverage we chose LoRaWAN

978-1-5386-3498-1/18/$31.00 ©2018 IEEE

for our further research.

III. SATELLITE DATA NETWORKS

When no pre-existing ground network infrastructure is available, a satellite channel is the only option for transferring data. As we are targeting a wide scale solution we will only consider global satellite networks.

The Inmarsat global geostationary satellite network

Inmarsat satellites operate on geostationary orbits and provide coverage up to 70° North and South latitude and hence can be used almost everywhere besides Arctic and Antarctic regions.

The Inmarsat satellite network provides up to 384 Kbit/s data transmission rate and has an off-the-shelf controller, which can be used for M2M communications. In addition, Inmarsat has several predefined solutions (IsatM2M, IsatDataPro and BGAN M2M) that differ in complexity, data transmission rate and throughput.

The Iridium global non-geostationary satellite network

The Iridium satellite network enables data exchange services with speeds up to 128Kbit/s. Iridium provides M2M solutions based on satellite short-burst-data modems 9602, 9603N and 9523 series and allows to send short data messages. The main distinctive feature of SBD modems is their low power consumption.

Mining and transportation industries often require 100% Earth coverage (especially important for Arctic mining regions). In this case the only satellite solution that can be used at the present time is the Iridium satellite network.

A SBD modem cannot be connected directly to IoT sensors; thus it is necessary to have a LoRa-Iridium gateway to handle all data received from IoT sensors and send it to a centralized data collection point. As a result, we end up with a heterogeneous network consisting of IoT sensors, a LoRa network, a LoRa-Iridium gateway and the Iridium satellite network.

IV. DATA EXCHANGE FORMATS

Data exchange formats are important in terms of transmission efficiency when sending information via IoT and satellite networks.

Data messages transferred between a LoRa endpoint and a LoRa gateway can only contain binary sensor data, which does not require any conversion and is standardized by the corresponding protocol. A LoRa endpoint can transmit data from several sensors. In this case sensor identifiers and extra service data is added to the message.

During further transmission to the central data acquisition point we need to encapsulate the LoRa endpoint identifier and certain technical information related to the IoT network.

The most commonly used scheme that fits the nature of IoT data is the key-value representation. Several key-value exchange formats can be used for IoT data transmission including:

- JSON;
- BSON;
- CBOR;
- Msgpack;
- JSONC;
- Protobuf;

JSON [3] conforms to the ASN.1 specification and is commonly used in modern IoT systems. For example, MQTT messages almost always contain JSON encoded data. Textual data representation allows to easily troubleshoot any IoT system without data decoding and a flexible key-value representation without predefined keys gives flexibility, which allows to use this format in practically any application. In addition, JSON is supported by major programming languages in their built-in libraries. However, both textual data representation and predefined keys lead to increase in message size.

BSON [4] is a binary JSON format version, which allows to handle binary data inside JSON messages. It doesn't contain any data encoding optimization and generally matches JSON in message size.

CBOR (RFC7049) [5] and MsgPack [6] use binary data representation and optimize message size by using appropriate data format for transmitted values. These formats don't use space and comma delimiters and convert integer and floating-point values from text to binary form. In general, it allows to optimize the overall payload. Both CBOR and MsgPack formats don’t require to define keys before data transmission and are comparable in message sizes.

The JSONC format [7] uses zlib (RFC1950) compression for JSON data and doesn't optimize any data representation. This format can significantly optimize message size when transmitting large textual data.

The information coming from IoT sensors usually contains fixed attributes/descriptors, and can be optimized during encoding with predefined keys. There are several exchange formats, using this principle.

Protocol Buffers (Protobuf) [8] use a binary format, which allows to decrease message size by optimizing value type encoding and using a predefined key structure. It minimizes the payload by sending key identifiers instead of key names and by converting values to binary format.

Formats with predefined keys are more efficient for IoT data since keys can be send once before regular value transmissions begin. In our study we will use the Protobuf format for data transmission.

V. A HETEROGENEOUS LPWAN-SATELLITE NETWORK MODEL FOR IOT

We developed a simulation model to test efficiency of different data exchange formats for typical IoT applications. The model diagram is presented on figure 1.

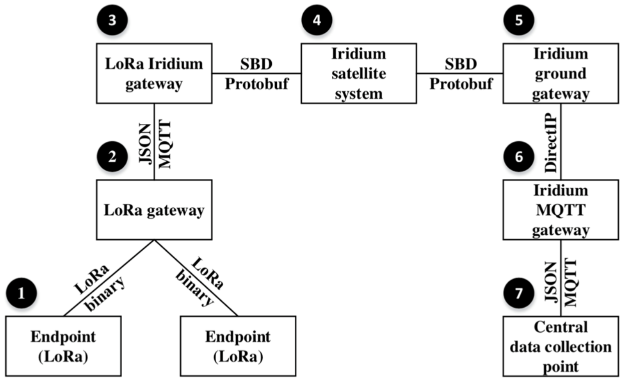

Figure 1. Model of a heterogeneous IoT network

The main elements of the model are:

1. LoRa endpoint;
2. LoRa gateway;
3. LoRa-Iridium gateway;
4. Iridium satellite system;
5. Iridium ground gateway;
6. Iridium MQTT gateway;
7. Central data collection point.

Node connection protocols:

- (1)-(2) – LoRa binary;
- (2)-(3) – MQTT-JSON;
- (3)-(4) – SBD-Protobuf;
- (4)-(5) – SBD-Protobuf;
- (5)-(6) – DirectIP-Protobuf;
- (6)-(7) – MQTT-JSON.

The important limitation to mention is a maximal LoRaWAN network message size for data transfer between nodes (1) and (2). It is equal to 222 bytes for messages forwarded via retransmission device and 242 bytes for messages send directly from an endpoint to a gateway [9]. Furthermore, the LoRaWAN transmission delay can vary from 60 to 1250 ms.

Sensors generate messages independently from each other and have a fixed probability of message generation per sensor. We use the Poisson distribution for the number of sensors per LoRa endpoint [10].

The LoRa gateway (2) adds extra information to each received packet, including the LoRa endpoint MAC address, modulation type, Signal to Noise Ratio and channel number. This information is encoded in JSON and published on the MQTT server.

The Lora-Iridium gateway (3) receives the LoRa gateway (2) messages via MQTT. We don't have any limitations for such kind of transmission since the Lora-Iridium gateway and the Lora gateway are collocated and connected via a Local Area Network. Hence, we don’t need to include any constraints to our model for communication between the Lora-Iridium gateway (3) and the LoRa gateway (2).

The Lora-Iridium gateway (3) transmits all data to a central data collection point (7). It uses the Iridium low speed satellite network with global coverage. The main limitations of the SBD Iridium data service are the maximal message size of 340 bytes [11] and high transmission costs. It is then extremely important to reduce the number of transmitted messages and the overall payload. As previously mentioned, the Protobuf data encoding will be applied.

SBD messages are forwarded via the satellite system (4) and the ground gateway (5) to the Iridium MQTT gateway (6). The ground gateway (5) uses the DirectIP protocol [12] to send data to the Iridium MQTT gateway (6). DirectIP works on top of TCP and does not have IANA reserved port numbers. All the data coming from the Lora-Iridium gateway (3) to the Iridium MQTT gateway (6) are forwarded without any modifications and do not have any additional limitations so we can exclude the Iridium satellite system (4) and the Iridium ground gateway (5) in calculation of payload.

The main purpose of the Iridium MQTT gateway (6) is to receive all data coming from the ground gateway (5), to convert the Protobuf format to JSON and publish it on an external MQTT server for further processing.

VI. VERIFYING EFFICIENCY OF DATA EXCHANGE FORMATS

We will use components (3), (4), (5) and (6) of the model for method verification. To validate method efficiency, we developed a Python script to emulate this part of the model. Python contains pre-build libraries for JSON and Protobuf support as well as SBD message parsing.

The LoRa gateway publishes the following data on MQTT server in JSON format [13]:

```
{
  "phyPayload": "AAEBAQEBAQEBAgICAgICAgJpNbxrAh8=",
  "rxInfo": {
    "channel": 1,
    "codeRate": "4/5",
    "crcStatus": 1,
    "dataRate": {
      "bandwidth": 125,
      "modulation": "LORA",
      "spreadFactor": 7
    },
    "frequency": 868300000,
    "loRaSNR": 7,
    "mac": "1dee08d0b691d149",
    "rfChain": 1,
    "rssi": -57,
    "size": 23,
    "time": "0001-01-01T00:00:00Z",
    "timestamp": 2074240683
  }
}
```

The "*phyPayload*" field is used to encapsulate sensors’ data, the LoRa endpoint MAC address is stored in the "*mac*" field. The remaining fields store parameters of the LoRa network.

We created a proto-definition to comply with the provided format [14]:

```
syntax = "proto3";

import "google/protobuf/timestamp.proto";

message phypayload {
  repeated float data = 1;
}

message rate {
  int32 bandwidth = 1;
  enum mod {
    LORA = 0;
  }
  mod modulation = 2;
  int32 spreadFactor = 7;
}

message info {
  int32 channel = 1;
```

```
  string codeRate = 2;
  int32 crcStatus = 3;
  rate dataRate = 4;
  int64 frequency = 5;
  int32 loRaSNR = 6;
  string mac = 7;
  int32 rfChain = 8;
  sint32 rssi = 9;
  int32 size = 10;
  google.protobuf.Timestamp time = 11;
  google.protobuf.Timestamp timestamp = 12;
}

message iotj {
  phypayload phyPayload = 1;
  info rxinfo = 2;
}
```

This Protobuf definition allows to transmit the same data we receive in JSON format encoded in Protobuf via satellite network.

We generated 1000 random messages from LoRa endpoints with different amount of sensor data in each message (from 0 to 13 sensors – number of sensors defined by Poisson distribution) and transmitted them with JSON and Protobuf encoding. Results are shown on figure 2. The average message size decreased from 453 bytes with JSON encoding to 93 bytes with Protobuf.

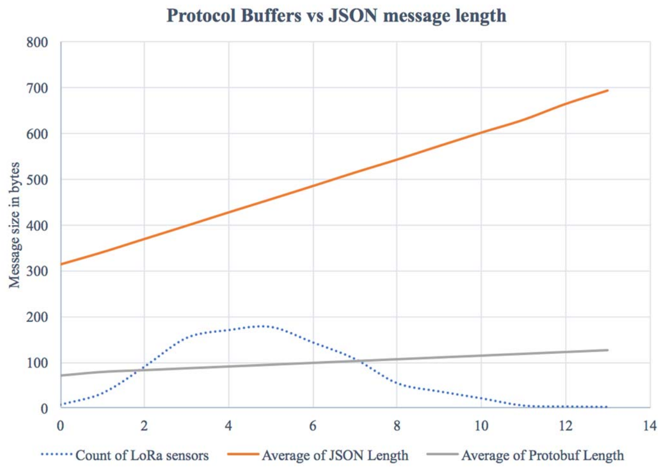

Figure 2. Protocol buffers vs JSON message length

VII. RESULTS VERIFICATION

To validate simulation results we built a hardware-software prototype comprising the following components:

- LoRa endpoint based on the Laird DVK-RM186 LoRa development kit;
- LoRa gateway built using a Raspberry Pi3 computer, a iC880A LoRa radio interface and a LinkLab Lora Gateway shield;
- Iridium 9602 SBD modem, connected to the LoRa gateway;
- DirectIP (SBD) and MQTT servers installed on a virtual cloud server.

Sensor data was generated on the DVK-RM186 LoRa endpoint and sent to the Raspberry Pi3 LoRa gateway. Data was then published on a local MQTT server (installed on Raspberry Pi3) and was parsed by Python script, which encoded data and sent it via the Iridium 9602 SBD modem to the Iridium satellite system. Information was received by the DirectIP server hosted on cloud virtual machine and was parsed by Python script to decode data to JSON format and publish it on the remote MQTT Server for final analysis.

In the end the experimental results we got on the prototype were consistent with initial estimations.

VIII. CONCLUSION

In this paper we highlighted the importance of choosing a proper data exchange format for IoT applications running on hybrid network infrastructures. Aiming at IoT solutions for geographically remote locations we developed a simulation model and a prototype of a LoRaWAN – Irridium network.

Simulation and experimental studies were conducted to compare efficiency of the widely used data exchange formats. Taking JSON as the benchmark, we achieved an almost 5 times decrease in average message size for data transmission over a low speed satellite network.

The results can be used for IoT systems in geographically remote locations with the Iridium SBD network as the gateway to the Internet.

We plan to extend our research by further optimizing data transferring techniques in heterogeneous networks, in particular, by using compression methods for SBD messages and finding a trade-off between the number of messages sent and repetition of information for different IoT applications.

ACKNOWLEDGEMENTS

The article was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2017 (grant №17-05-0017) and by the Russian Academic Excellence Project «5-100».

IX. REFERENCES

1. SATELLITE TECHNOLOGIES FOR IOT APPLICATIONS / IoTUK // [online]: Home - IoTUK – Available from: https://iotuk.org.uk/wp-content/uploads/2017/04/Satellite-Applications.pdf (accessed 15.10.2017).
2. LPWAN White Paper / Leverge // [online]: Leverge – Available from: https://www.leverege.com/research-papers/lpwan-white-paper (accessed 15.10.2017).
3. Introducing JSON / JSON // [online]: JSON – Available from: http://www.json.org (accessed 15.10.2017).
4. BSON - Binary JSON / BSON // [online]: BSON – Available from: http://bsonspec.org (accessed 15.10.2017).
5. Concise Binary Object Representation (CBOR) / C. Bormann, P. Hoffman // [online]: Internet Engineering Task Force (IETF) – Available from: https://tools.ietf.org/html/rfc7049 (accessed 15.10.2017).
6. MessagePack / Sadayuki Furuhashi // [online]: MessagePack – Available from: http://msgpack.org (accessed 15.10.2017).
7. JSON compressor and decompressor / tcorral //

[online]: Github – Available from: https://github.com/tcorral/JSONC (accessed 15.10.2017).

8. Protocol Buffers / Google Developers // [online]: Google Developers – Available from: https://developers.google.com/protocol-buffers/ (accessed 15.10.2017).
9. LoRaWAN 1.0.2 Regional Parameters / LoRa Alliance Technical committee // LoRa Alliance – 2017.
10. Modeling Interference for Wireless Sensor Network Simulators / Umber Noreen, Ahcène Bounceur, Laurent Clavier // [online]: Universitè de Bretagne Occidentale – Available from: http://pagesperso.univ-brest.fr/~bounceur/anr/persepteur/articles/bdaw_2016_umber.pdf (accessed 15.10.2017).
11. Iridium 9602 SBD Transceiver Developer's Guide / Iridium Communications Inc. // [online]: Iridium Communications Inc.– Available from: https://github.com/whitef0x0/week1example/blob/master/iridiumsbd/Iridium-9602-SBD-Transceiver-Product-Developers-Guide.pdf (accessed 15.10.2017).
12. Iridium Short Burst Data Service Developers Guide // Iridium Satellite // Iridium Satellite LLC – 2012.
13. LoRa Server documentation / The LoRa Server project // [online]: The LoRa Server project. – Available from: https://docs.loraserver.io/lora-gateway-bridge/use/data/ (accessed 15.10.2017).
14. Language Guide (proto3) / Google Developers // [online]: Google Developers – Available from: https://developers.google.com/protocol-buffers/docs/proto3 (accessed 15.10.2017).

Ivan I. Lysogor received the B.S. degree in Automated Data Processing and Management from Adygeya State University, Maykop, Russia, in 2002. He is currently pursuing the M.S. degree in Computer Systems and Networks at MIEM HSE, Moscow, Russia.

He works for Juniper networks and has more than 15 years of experience in computer networks design and implementation. His areas of interest are Internet of Things and Data Center technologies.

Ivan I. Lysogor's awards and honors include Cisco Routing and Switching CCIE certification, Cisco CCIE Voice certification and Juniper JNCIP certification.

Leonid S. Voskov received his degree in CAD systems from the Moscow State Institute of Electronics and Mathematics in 1968 and his PhD in the same field in 1974. In 1978-1979 he was a visiting research fellow at the Laboratory of Artificial Intelligence at the University of Edinburgh. He is currently a Professor of the School of Computer Engineering at the National Research University Higher School of Economics, Moscow, Russia.

His research interests include wireless sensors networks, Internet of Things, energy efficient communication.

Sergey G. Efremov received his BSc and MSc degrees in Computer Science from the Moscow State Institute of Electronics and Mathematics in 2010 and his PhD in Mathematical Modelling, Numerical Methods and Software Complexes from the Higher School of Economics in 2013. He is currently an Associate Professor of the School of Business Informatics at the National Research University Higher School of Economics, Moscow, Russia.

His research interests include network protocols, Internet of Things and data analysis.