BMP: A Bounded Message Protocol for the IoT

Frédéric Bergeron
frederic.bergeron2@usherbrooke.ca
Université de Sherbrooke
Sherbrooke, Canada

Kevin Bouchard
Université du Québec à Chicoutimi
Saguenay, Canada
kevin.bouchard@uqac.ca

Sylvain Giroux
Université de Sherbrooke
Sherbrooke, Canada
sylvain.giroux@usherbrooke.ca

Sebastien Gaboury
Université du Québec à Chicoutimi
Saguenay, Canada
sebastien.gaboury@uqac.ca

ABSTRACT

In this paper, we introduce a new communication protocol for machine to machine communication within the Internet of Things (IoT) based on physical distances rather than on addresses. A common challenge with the IoT is to filter the quantity of data that can be collected, especially in centralized solutions. By using distance from the emission point instead of addresses to move messages between objects, this protocol forces the use of decentralized algorithms. It also tackles the problem of the quantity of data by giving each data a time to live, after which messages are deleted by the protocol. The idea is that an object that sends a message should decide who can access it and for how long the data is relevant. We specify the conditions under which such a protocol can be useful and further provide some examples of application that could gain from using this protocol.

CCS CONCEPTS

• **Networks → Network protocol design**.

KEYWORDS

Communication protocol, Internet of Things

ACM Reference Format:
Frédéric Bergeron, Kevin Bouchard, Sylvain Giroux, and Sebastien Gaboury. 2019. BMP: A Bounded Message Protocol for the IoT. In *EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs '19), September 25–27, 2019, Valencia, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3342428.3342696

1 INTRODUCTION

The second half of the 2010s marks the emergence of vast Internet of Things (IoT) systems that promises to change the way many things are done. IoT leads to new way of collecting massive amount of data from a given environment that can it turn be feed to powerful artificial intelligence algorithms. Sensors can be integrated in almost any existing object to turn it into a smart object. Those objects can connect themselves together to form networks of objects, or an Internet of Things [4, 14].

Even thought IoT has not yet reached its full potential, certain challenges are already arising and they need to be addressed in order to fulfill its promise. A first challenge is to be able to process all the data gathered by the ever increasing number of objects. This gets even more challenging when all data are regrouped into a centralized algorithm that must then decide what data are useful for a given task. To palliate this problem, it is sometime possible to decentralize algorithms and make them run directly on the objects. The idea is that some objects require only a small part of the available data to take decisions and execute their tasks [8].

Another challenge with the IoT is communication. Several protocols exist, each having different use cases. This article is about a new communication protocol, a Bounded Message Protocol (BMP), that aims to bring the intelligence to the objects. The novelty of this protocol is that is uses physical distances instead of addresses to send its messages. Messages also have a time to live on hosts that is expressed in seconds.

2 RELATED WORK

This paper is about a new communication protocol for the IoT. Several other protocols already exist for different needs. A popular one is the Bluetooth protocol, either in its version 4 Low Energy or in its version 5. Along with ZigBee and UltraWideBand (UWB), it is part of the IEEE 802.15 specification for wireless personal area networks (WPAN). It works under a master and slaves paradigm in which a master object can have up to seven slaves, grouped into a piconet. Slaves cannot communicate between themselves, only with the master. Since July 2017, it is possible the use Bluetooth in a mesh configuration [1]. It is also possible to use it in broadcast.

Zigbee uses different roles than Bluetooth for the objects. The first object in the network is the coordinator. It is the one that initiates the network. Then, there are the router objects. They can host a Zigbee application and act as intermediary to the last role, the end devices. They are the less capable objects, only able to communicate with their direct parent. Together, they form a mesh [5, 15].

More recently, the Long Range Wide-Area Network (LoRaWAN) protocol has been proposed for small and low energy objects. It is a proprietary protocol. It uses LoRa for its physical layer protocol. It is aimed for long range communication between low energy objects.

GoodTechs '19, September 25–27, 2019, Valencia, Spain

ACM ISBN 978-1-4503-6261-0/19/09…$15.00
https://doi.org/10.1145/3342428.3342696

The network is organized as a star-of-star, where many objects connects via LoRa to a gateway that connects itself via IP to the internet. A study found this protocol to be more energy efficient than Zigbee [7]. Given the long range of LoRa, LoRaWan are usually single hop networks and thus have medium access challenges in large networks [3].

Those protocols, or protocol stacks, cover certain needs of the IoT, from network size to encryption. They, however, do not offer any help for data separation and selection and they tend to centralize communication to a central node; namely the master, the coordinator and the gateway in the previously mentioned protocols.

With a more data-centric point-of-view, several protocols have also been implemented for the IoT. Some of the most widely adopted are Message Queuing Telemetry Transport (MQTT) protocol [2], Constraint Application protocol (CoAP) [11] and Advance Message Queuing Protocol (AMQP)[13]. MQTT only implements a publish/subscribe mechanism while the other two also implements a request/response mechanism. MQTT and AMQP both use TCP at the transport layer while CoAP uses UDP. Accordingly, CoAP has a smaller overhead on a reliable network were messages do not need re-transmission when reliability is needed. More complete comparisons can be found in [6, 9, 12] where they are also compared with more protocols (HTTP/REST, XMPP and websockets). Publish/subscribe topics are usually predetermined by the programmers and discovery mechanisms are not provided by the protocols.

3 DESCRIPTION OF THE PROTOCOL

In the introduction, we presented some challenges related to the massive usage of the IoT. A solution to some of those challenges could be to move most of the intelligence, or decision making, closer to the basic objects instead of centralizing it into a main computer, as in fog computing approaches. Bringing intelligence to objects is easier when systems are conceived using small and independent units that need few communication, if any, between them. Those system are close to the ones presented by Resnick in his book on decentralized systems [10]. In this book, global behaviours emerge from independent units following simple rules when interacting with their environment. As presented in section 2, existing communication protocol are not adapted for this kind of decentralized communication as all data always need to pass through a centralizing intermediary.

3.1 Main purpose

The communication protocol we present in this paper is not a general communication protocol for the IoT. Rather, it is oriented toward diffusion of general knowledge information to everyone in a given range. A good analogy might be to mimic a lighthouse with a foghorn. This is useful when you do not know who will get your message and when no answer is required. In fact, as with a lighthouse, an object emitting using the protocol does not even know if anyone will receive the message. Hence, this is not made to communicate critical information that need to reach a given receiver. There are no guaranty of service nor addressing.

3.2 Main characteristics

In the previous subsection, we stated the goal of this communication protocol, to diffuse general information to everyone in a given range. From that goal, the two first characteristics emerge: messages are broadcasted and they have a range. Messages are broadcasted in order to reach every objects while still not knowing if there are any. However, as there can be an arbitrary number of objects in a network, diffusing everything in broadcast can lead to the network jam. To avoid this pitfall, we assign to every message a physical range outside of which every object that receives this message must destroys it and stops forwarding it. It might however still happen in very dense networks.

Adding a physical range on all messages serves as a way to prevent broadcast storms. But, it also serves another purpose: to limit the propagation of information to the objects the sender thinks may need it. The hypothesis is that an object that sends a message should be the one to decide who should receive this information. The idea of a physical limit given to a message is not new. It is already part of the Internet Protocol (IP). It take the form of a number that must be decreased once per second and once per re-transmission by a relay. The physical limit we propose here is completely independent from the number of hop. It comes in two flavors: a measurable distance or a named place. The measurable distance is simply a distance expressed in meters from the emission point. It can also be viewed as the distance at which the message content becomes irrelevant. The named place is intended for smart building where it may be more practical to restrict messages to a given room, like the entrance hall.

Determining the distance or the name to emit to is a task left to the application using the protocol. Validating the distance of an inbound message is, however, a task for the protocol. When using named distance, the protocol simply looks if the name is in a list given by the running applications. It is a publish/subscribe system where topics should be physical places. Using physical distances, on the other hand, requires to know the origin of the message in a given system of coordinates. This origin can be found in many ways. To allow every user to use the system of their choosing, acquiring the position of an object works using a plugin system that simply outputs a position. The distance between objects is computed using the euclidean distance.

As stated before, we designed this protocol with the though that the agent that emits a message is the one to decide who should receive it. We go further on this logic by forcing the emitter to also decide for how long his message should exists. Indeed, general knowledge information usually have a lifetime until the information ceases to be valid. In a IoT context, this could represent the time until a new value is collected. As for the distance, the time limit comes in two flavors. In the first one, the emitter gives a date-time at which to delete the message. The second one gives a lifespan in second. As there is no routing in the protocol, there is no way of knowing how much time a message took to reach an object. Therefore, the lifespan only begins to diminish once it has been received. Given the re-transmission mechanism explained in the next paragraph, this means that a message can live forever under certain conditions, no matter the original lifespan it was given. Such conditions could arise in a city network were cars come and go

fast enough to perpetually re-transmit the message to new cars without ever decreasing the timer or effectively decreasing it at a slower pace. An object could even receive the same message twice without ever noticing it as the first one would have expired and been deleted. A workaround to this pitfall is to add the original emission date-time to the header of all messages so the lifespan can be adjusted at each re-transmission. We recommend usage of the Coordinated Universal Time (UTC). It is assumed that most computer chips can keep track of the passage of time with a decent precision for lifespan of a few seconds. However, we cannot assume that all computer chips have access to the current date-time and therefore we cannot assume that this new header information can be added by all objects. In that case the first object that can fill the missing header should do it before re-transmitting the message. There are still no guaranties that the receiving object will have access to the date-time to confront the time to live header with the original emission time header. In that case it should simply be ignored.

In the previous paragraph, we hinted some of the mechanisms that occur when re-transmitting a message. Point-to-point protocols usually come with a routing protocol that establishes a best path or avoid packet collision on the medium of transmission to ensure each messages reach its destination. As there is no routing in our protocol, transmissions and re-transmissions needs to follow different rules. The whole re-transmission process, along with the process of storing a message is presented in Figure 1. Each message is assigned a unique ID. This ID is used to make sure a object only keeps one copy of a message if it is received many times. It is also used to make sure a message is only re-transmitted once, unless we are adding the emission date-time header. The decision flow is quite simple. We first check if this is the first time we see the message, then if the time and the distance are valid we re-transmit it, adding the emission time if needed.

There are three main characteristics to BMP:

- All messages are broadcasted.
- Message are limited by a physical range.
- Messages have a time to live on each objects.

Those characteristics can be implemented on most layer of the OSI protocol stack if sub-layer protocols can be chosen. All form of unconstrained wireless signal can be used for the actual transmission of data. BMP aims to be deployed on totally unstructured network as it does not suppose the existence of any network. As BMP can work on many physical layer, it can also work on many at the same time, provided that some objects can access many radios. A computer equipped with a UWB radio and a WI-FI radio can emit the same BMP message on both medium, thus linking together an UWB network and a WI-FI network. BMP can be placed at any OSI layer, as long as the three previously mentioned requirements can be met.

3.3 Limitations and constraints

The protocol, as described in this section, has some inner limitations and constraints. First of all, it is assumed that each relay in the network uses the protocol so the protocol alone decide whether or not to further transmit the messages. This might be easy to ensure in a line-of-sight wireless IoT network. This is however harder in a wired network where the hidden network topology might cause some nodes to be outside the physical range while being to only one connected to an object in range.

There can also be some problems with the notion of position. Getting its own position can be hard and long for some objects. Fixed object could have it encoded in a static file or in a configuration. Moving objects, on the other hand, needs to compute it periodically. Moving objects pose the question on how messages are accessible. The initial propagation of a message is explained in Figure 1. Still, it does not provide an answer to the question: How can a moving object be informed of all valid messages as it moves to a new zone? This is still an open question, with many possible solutions.

Some limitations regarding the notion of time have already be addressed previously. The main ones are the absence of a real-time clock on many objects and time-drifting on most CPUs.

Imprecision in the measure of time and distance restrict the protocol the certain values of time and distance. Those limit values should always be chosen so that the imprecision is several order of magnitude smaller than the boundary. It should also be noted that the transmission time is considered to be null and is therefore part of the time imprecision.

Despite all those limitations, we believe the proposed protocol to be adapted for many real-life situations. Its interoperability makes it interesting for many kind of applications

4 EXAMPLE APPLICATION

The protocol described in the previous section imposes some constraints on the application using it. It mostly targets ruled-based systems that can operate in a decentralized fashion with only local information. Such systems are common within the IoT. This section describes an example application that could communicate through BMP.

The example application is one of smart housing. It monitors appliance usage in a smart kitchen. Many sensors are distributed in the environment. They can be motion sensors, pressure plates, contact sensors, thermometers, humidity sensors, power supply sensors, and so on. They all broadcast their current state to the named distance "kitchen" with a life time corresponding to their sensing frequency. There can then be multiple agents organizing different aspects of the kitchen. An oven security agent could be interested in using motion sensors data with a thermometer inside the oven the determine if someone is monitoring the cooking and interrupting it if needed. A recipe assistant agent could monitor the steps in a recipe to perform basic activity recognition using the information broadcasted by the environment. In this example, many technologies can coexist. Some sensors are wired, others can be wireless over Bluetooth and Zigbee. All readings can be simultaneously sent to a MQTT broker for logging and external analysis. Security alert should cause immediate reaction from the environment using our protocol, while still being directly sent to a neighborhood monitoring center using LoRaWAN. Inversely, this neighborhood monitoring center could send power usage statistics to nearby houses with BMP so they can independently decide to distribute high power consuming appliance usage to another moment of the day.

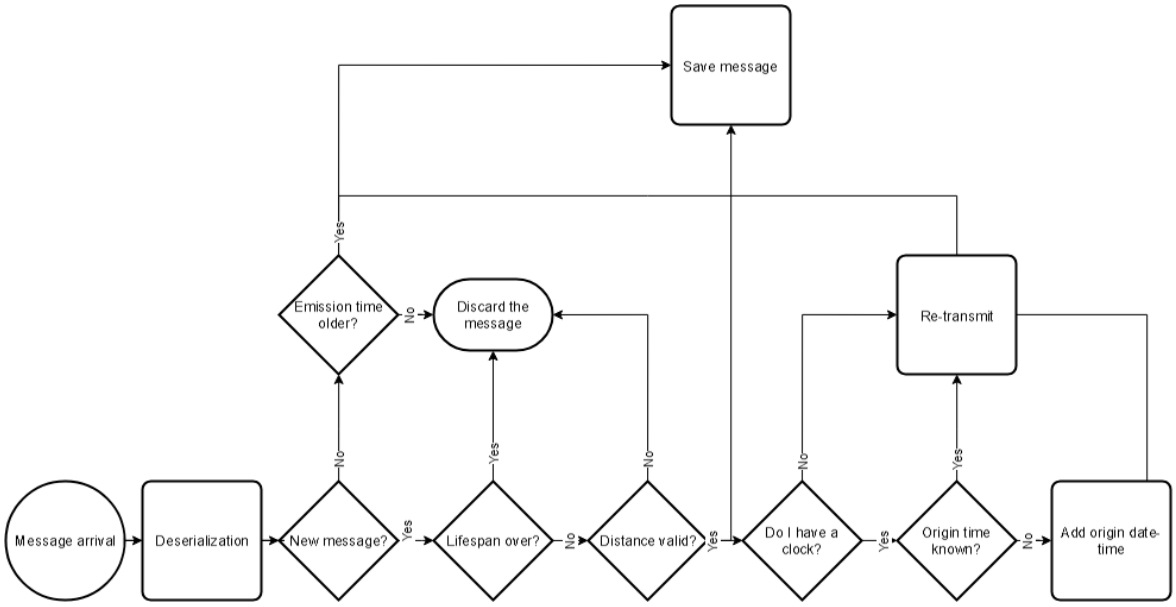

Figure 1: Reception and re-transmission flow of a message

This example emphases that the protocol is about unorganized and emergent behaviours. Objects using it are waiting for information collected by other objects to perform rule-based actions. Objects collecting data are not concerned of what is done with the information, they only want to provide it so others can do meaningful usage of it.

5 CONCLUSION

In this paper, we introduced a new communication protocol. We name it Bounded Message Protocol, BMP. Its key features are that it limits the propagation of broadcasted messages to a physical zone defined by a radius or by a name. The messages also have a time to live set by the sender. Those features aims to control the quantity of data available in the network. The idea is that the object who sends a message knows the pertinence radius of the message. Then, all objects in this radius receive the message through broadcast, as there are no routing table, no addressing. Rule-based expert systems are expected to work well under those conditions. This protocol has yet to be deployed in real life, but computer simulations shows that the basic principles are working. The next steps are to deploy the protocol in a real-life environment to test its capabilities and limitations. As it can also works with any physical layer capable of broadcasting messages, this capability will be experimented in a future work on Arduino boards with UWB and RF22 radios.

ACKNOWLEDGMENTS

The authors would like to thanks the National Sciences and Engineering Research Council of Canada for their financial support to this project.

REFERENCES

[1] Alexandre Adomnicai, Jacques JA Fournier, and Laurent Masson. 2018. Hardware Security Threats Against Bluetooth Mesh Networks. In *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–9.
[2] Andrew Banks and Rahul Gupta. 2014. MQTT Version 3.1. 1. *OASIS standard* 29 (2014), 89.
[3] Nicolas Gonzalez, Adrien Van Den Bossche, and Thierry Val. 2018. Specificities of the LoRaâĎć Physical Layer for the Development of New Ad Hoc MAC Layers. In *International Conference on Ad-Hoc Networks and Wireless*. Springer, 163–174.
[4] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.
[5] Jose A Gutierrez, Marco Naeve, Ed Callaway, Monique Bourgeois, Vinay Mitter, and Bob Heile. 2001. IEEE 802.15. 4: a developing standard for low-power low-cost wireless personal area networks. *IEEE network* 15, 5 (2001), 12–19.
[6] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, and Jesus Alonso-Zarate. 2015. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud computing* 3, 1 (2015), 11–17.
[7] Abdullah Kurtoglu, Joan Carletta, and Kye-Shin Lee. 2017. Energy Consumption in Long-Range Linear Wireless Sensor Networks using LoRaWan and ZigBee. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 1163,1167.
[8] Shancang Li, Li Da Xu, and Shanshan Zhao. 2015. The internet of things: a survey. *Information Systems Frontiers* 17, 2 (2015), 243–259.
[9] Nitin Naik. 2017. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In *2017 IEEE international systems engineering symposium (ISSE)*. IEEE, 1–7.
[10] Mitchel Resnick. 1997. *Turtles, termites, and traffic jams: Explorations in massively parallel microworlds*. Mit Press.
[11] Zach Shelby, Klaus Hartke, and Carsten Bormann. 2014. *The constrained application protocol (CoAP)*. Technical Report.
[12] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. 2014. Performance evaluation of MQTT and CoAP via a common middleware. In *2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*. IEEE, 1–6.
[13] Steve Vinoski. 2006. Advanced message queuing protocol. *IEEE Internet Computing* 6 (2006), 87–89.
[14] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of things for smart cities. *IEEE Internet of Things journal* 1, 1 (2014), 22–32.
[15] Jianliang Zheng and Myung J Lee. 2006. A comprehensive performance study of IEEE 802.15. 4. *Sensor network operations* 4 (2006), 218–237.