An Adaptive Solution for Congestion Control in CoAP-based Group Communications

Fathia OUAKASSE, Said RAKRAK
Applied Mathematics and Computer Science
Laboratory (LAMAI)
Cadi Ayyad University
Marrekesh, Morocco

***Abstract*—The use of lightweight devices and constrained resources like Wireless Sensors Network (WSN) makes patterns traffic in the Internet of Things (IoT) different from the ones in conventional networks. One of the most emerging messaging protocols used to address the needs of these lightweight IoT nodes is Constrained Application Protocol (CoAP). CoAP presents a lot of advantages compared to other IoT application layer protocols; it ensures group communication via multicast communications between a server and multiple clients. Nevertheless, it doesn't support a group communication from a client to multiple servers; it relies on multiple unicasts to do so. Regarding the fact that these constrained devices communicate via a large amount of messages and notifications, network congestion occurs. This paper proposes an adaptive congestion control algorithm designed for group communications using unicast between a client and multiple servers. Simulated results show that the proposed mechanism can appropriately achieve higher performances in terms of response time and packet loss.**

***Keywords*—Internet of Things (IoT); Constrained Application Protocol (CoAP); congestion control; group communication; multicast; unicast**

I. INTRODUCTION

Recently, WSNs have been widely deployed in many IoT applications in order to measure, control or detect physical and environmental events like pressure, humidity, temperature and pollution levels, as well as other critical parameters. Applications usually used to send queries to concerned sensors to retrieve values periodically from the measurements or detections. Moreover, it is estimated that by the year of 2020 more than 26 billion devices will be connected to satisfy a wide range of IoT applications [1].

However, in recent critical applications of WSN that require intervention, such as home automation, industry process control, healthcare, environment monitoring, smart grid, and ambient assisted living, the challenge is getting information when an event of interest occurs in order to intervene in real-time. In this context, the publish/subscribe model [2] is the most appropriate model covering these requirements. Furthermore, one of the most important protocols based on this model is CoAP [3]. Indeed, CoAP is the most appropriate protocol for lightweight devices and constrained resources in terms of memory, energy, and computing. Thus, CoAP has been widely used in different application fields for resource constrained networks and M2M applications such as smart grid [4], building and home automation [5], smart cities [6] and in the healthcare industry, in which CoAP presents many applications, as an illustration, a mechanism for health monitoring using a wearable Sensor to provide real-time updates of the patient's status via CoAP protocol is presented in [7].

In many IoT application fields, in addition to unicast communication, nodes should be addressed in groups, so in order to manage the needs of multiple communications between different and several devices, CoAP supports group communication [8].

However, CoAP ensures multicast communication in one sense, from a server to multiple clients, but in the other sense; from a client to multiple servers; it relies on unicast communications. This has led to the problem of network congestion [9]. Network congestion in CoAP represents the great limitation that hinders the proper functioning of this protocol and causes the loss of packets. It can also significantly damage the performance of a network, manifesting in increased packet latencies, while a network may even become useless if the congestion collapse occurs [10].

To resolve this problem, researchers propose to insert a delay between consecutive requests. In this paper, an improved adaptive congestion control for group communication between a single client and multiple servers in CoAP is proposed. The principle of our improvement consists of the estimation of a delay to introduce between two requests; our formula adapts the calculation of the delay to network conditions because it is based on an estimated average link delay. Simulation results show that our proposition can appropriately achieve higher performances in terms of response time and packet loss.

The remainder of this paper is organized as: A brief presentation of the main aspects of CoAP protocol including reliability and security are presented as background in the second section. Then, in the third section, related works to group communication in CoAP including multicast and unicast group communications are described, also the problem of network congestion in group communication is discussed. Afterwards, in the fourth section, the proposed improved congestion control algorithm is detailed and a simulation of our proposition results is drawn using NS2 network simulator is presented. Finally, a conclusion and some future directions are closing up our paper in the fifth section.

II. CoAP Background

CoAP has been designed by the Internet Engineering Task Force (IETF) to support IoT with lightweight messaging for devices operating in a constrained environment. CoAP is an application layer protocol based on a REST architecture. It defines two kinds of interactions between end-points: 1) The client/server model which provides as well two interaction types: a) a one-to-one interaction which means request/reply and b) a multi-cast interaction; when a Client wants to interrogate servers it makes requests to servers, servers send back responses. Like HTTP, Clients have the ability to manage resources using requests: GET, PUT, POST and DELETE to perform Create, Retrieve, Update, and Delete operations. 2) A publish/subscribe model called the observer model [11], where a server, playing the role of the publisher, sends messages of notifications as publications to an observer, playing the role of subscriber, about a resource (event) that the subscriber is interested in receiving.

Unlike HTTP, CoAP doesn't run over TCP, it runs over UDP. Communication between clients and servers is afforded through connectionless datagrams. Retries and reordering are implemented in the application stack. UDP broadcasts and multicasts are also allowed by CoAP for addressing [12]. Otherwise, CoAP is considered more suitable for the IoT domain, this is going back to the fact that it is possible to build sufficiently basic error checking and verification for UDP to make sure that messages arrive without the significant communication overhead as in the case of TCP [13]. An overview architecture of CoAP protocol is drawn in Fig. 1.

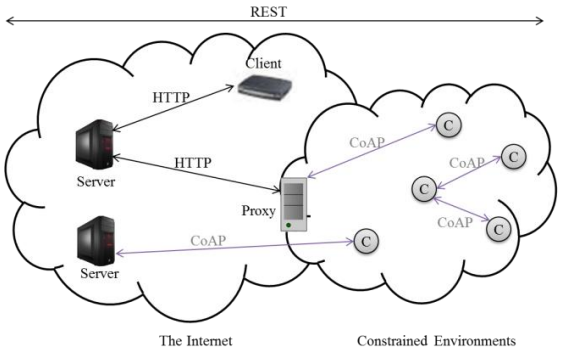

Fig. 1. An overview architecture of CoAP protocol.

CoAP utilizes four message types: 1) confirmable; 2) non-confirmable; 3) reset; and 4) acknowledgment, where two among them concern reliability messages. The reliability of CoAP consists of a confirmable message and a non-confirmable message [14]. In the case of a confirmable message an acknowledgment message (ACK) is sent to the sender from the intended recipient as shown in Fig. 2(a), else the message is retransmitted. This is just a confirmation that the message is received, but it doesn't confirm that its contents were decoded correctly. However, a non-confirmable message is fire and forget, i.e., no reception confirmation as shown in Fig. 2(b) [15].

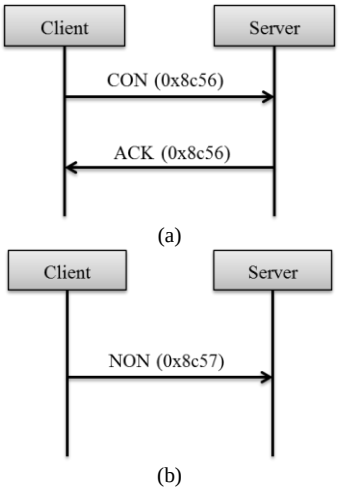

Fig. 2. (a) Reliable message transport (b) Unreliable message transport.

Since SSL/TLS are not available to provide security in UDP, CoAP uses Datagram Transport Layer Security (DTLS) on top of its UDP transport protocol for transfers of data [12].

III. Group Communication CoAP-Based

In the IoT, applications use group communication to make transactions between its different nodes, this goes back to the fact that nodes should be addressed either individually or in groups.

In many IoT applications, nodes addressed in group, i.e., a one to many communication patterns is essential to meet the needs of the application. Furthermore, in some applications, to increase the accuracy and the reliability of gathered data, it is important to collect information from more than one sensor. Moreover, the information gathered at the same time from many sensors may be very crucial to decide the appropriate way to intervene in situations which require real time intervention. So, all these scenarios and others require a communication with a group of sensors as recognized in the Charter of IETF CoRE Working Group [16].

A. Unicast group communications CoAP-based

In [17], authors propose to use an alternative unicast-based group communication solution for communication between CoAP devices. In order to facilitate the manipulation of a group of resources used by multiple smart objects, they create an intermediate level of aggregation. The group of resources is called an entity, the resources themselves are called the entity members and the component that manages these entities is called the Entity Manager (EM). By using a single CoAP request, an entity can be manipulated and thanks to the EM, entities that are created from groups of resources residing on CoAP servers can be maintained inside the Low power and Lossy Networks LLN. On the other hand, the EM acts as a proxy between the client and the constrained devices, thus clients on the Internet can create new entities and manipulate

them by requests via the EM. This latter analyses and verifies the client requests and then route them to the suitable constrained devices based on CoAP, after receiving responses, EM combines them according to the needs of the client and sends back an aggregated response to the client [18]. Fig. 3 shows an overview of the involved components.

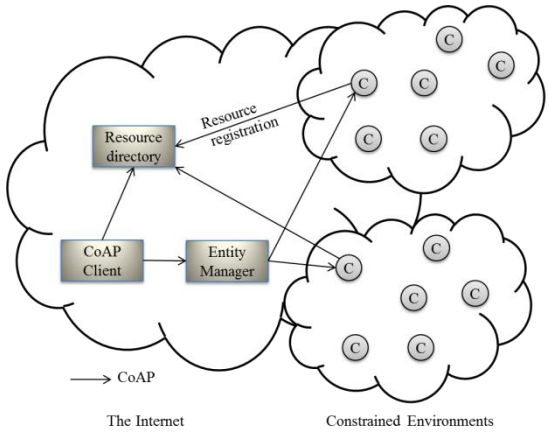

Fig. 3. The process of the creation of entities by clients on the entity manager.

Moreover, in [17] authors have introduced the notion of profiles which allows the client to give more details about the behavior of the created entities. The most advantage of this approach is its reliability; this goes back to the fact that it relies on unicast messages based on CoAP reliability mechanism.

However, in [19], authors gave a solution called SeaHttp for unicast-based group communication, where they proposed two additional methods called BRANCH and COMBINE to substitute the role of the entity manager and enable nodes to join and leave groups by themselves. This will benefit by reducing the number of messages. However, it can present some difficulties from viewpoint of the implementation in existing networks as a lack of flexibility.

B. Multicast group communications CoAP-based

As mentioned above, the IETF CoRE working group has first recognized the need to support a non-reliable multicast message. Thus, they have developed a specification for Group Communication for CoAP in RFC 7390 [20] to explain how we can use the CoAP protocol in a group communication context. Indeed, Group communication based on CoAP consists of sending a single non-confirmable message to multiple nodes grouped into a specific group using UDP/IP multicast for the requests, and unicast UDP/IP for the responses (if there was any). This means that all the nodes grouped in this group receive the same exact message.

It was proved that the use of multicast communication for sending requests is very efficient but it does not impact the number of responses sent by the destination nodes since these are sent as unicasts.

In the same context, authors in [21] presented an alternative lightweight forwarding algorithm for efficient multicast support in LLNs. This allows reducing a number of requests in the LLN since it sends one request to multiple destinations at the same time instead of a unicast for each destination.

C. Congestion control in group communications

The problem of congestion happens when the traffic load offered to a network approaches the network capacity [22]. This phenomenon is one of the main obstacles that still hinder the well-functioning of many protocols and thus impacts directly the efficiency of the communication. On the other hand, requests in group communication using CoAP engender a multitude of responses from different nodes, potentially causing congestion. Therefore, both the group communication multicast-based requests and the group communication CoAP unicast-based responses to these multicast requests must be conservatively controlled.

Indeed, CoAP must handle the congestion control by itself because it is based on UDP. Unlike HTTP which is based on TCP where a proper end-to-end congestion control is provided, CoAP offers a basic congestion control in the case of unicast messages [23].

In addition to the basic congestion control in unicast communications, the core CoAP specification also defines congestion control mechanism to be able to handle congestion control in case of multicast communications (requests from a server to multiple clients). Indeed, it defines a random delay called leisure which consists of a period of time delay inserted between multiple multicast requests. This leisure could be either a default value used by the server or it can be computed according to the following formula:

$$\text{Leisure} = S*G/R \quad (1)$$

Where, G is an estimated group size, R is a target data transfer rate R and S is an estimated response size.

Nevertheless, in the case when a single client is communicating with multiple servers using unicasts, CoAP does not specify a congestion control mechanism. To overcome this situation, authors in [17] proposed a simple solution consisting of a delay inserted between consecutive requests; this led to a limitation in the rate at which requests are sent.

In the following paper, we propose an improved formula to calculate the estimated delay to introduce between requests in order to reduce the network congestion.

IV. The Proposed Approach

Experiences show that communications via unicasts between a single client and multiple servers automatically engender a congestion of the network. In order to reduce the problem of congestion, we propose, in this paper, a simple adaptive solution based on the leisure defined in the RFC 7390 [20].

A. Adaptive solution to network condition

Indeed, the fact that the CoAP congestion control, designed for group communication between a single client and multiple servers, doesn't take into consideration the link delay to calculate the delay to insert between consecutive multicast requests, this leads to a congestion control mechanism insensitive to network conditions.

So, in this paper, in order to improve the delay and to adapt the behavior of our solution to network conditions, we propose

a delay between unicast requests depending on the link delay and the estimated group size as shown in the following formula:

$$D = \text{average link delay} * G / G-1 \quad (2)$$

Furthermore, the link delay represents the behavior of the network; if it increases, it means that congestion is more likely to happen, so in order to manage this problem, the estimated delay between unicast requests has to increase. On the other hand, if the link delay decreases, it means that the network is more available and the delays between requests have to be short adapting its behavior to the condition of the network.

B. Simulated results

In order to evaluate the performance of our proposed solution, we perform, in this section, simulations. Moreover, in order to figure out the performance of our proposed estimated delay to insert between unicast requests for group communications between a single client and multiple servers, we carry our evaluations on a NS2 simulator.

Our simulations are performed in terms of the average response; time taken by servers to respond to unicast communication, the jitter; the variation in the delay of the received messages and the packet loss ratio resulted from group communication.

The parameters considered in this simulation are detailed in Table1.

TABLE. I. SIMULATION PARAMETERS CONSIDERED IN OUR APPROACH

Parameter	**Value**
Nodes number	10 to 40 nodes
Packet size	1 Ko
Link speed	3 Mbps
Link delay	5 to 30ms
Simulation duration	10s

Fig. 4 shows the average response time according to several group sizes of servers to respond to unicast communications initiated by a single client. As expected, the average response time increases as the link delay increases proportionally to the group size. Furthermore, initially, in low link delay, all the group sizes have slightly the same average response time. Afterward, graphs for all of the group sizes start to increase following approximately the same curve variation, this is due to the fact that congestion is likely to happen causing more retransmissions delays, the thing that led to the increase in the average response time.

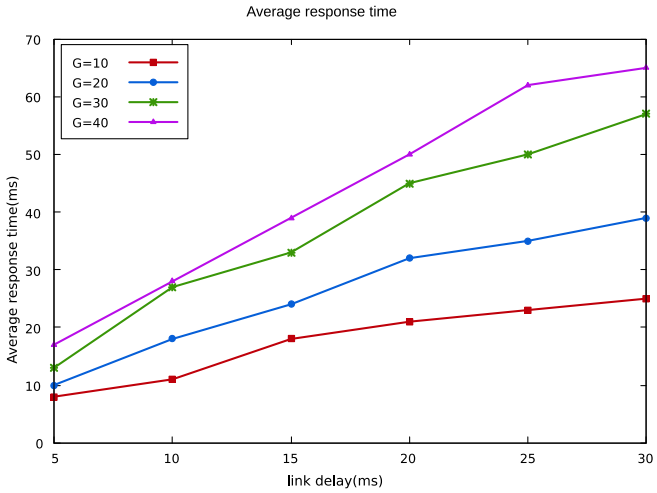

Fig. 4. Average response time according to link delays using several group sizes.

As discussed in the previous figure, the increases in link delay according to the increase of group size have slightly the same impact on the jitter variation as shown in Fig. 5.

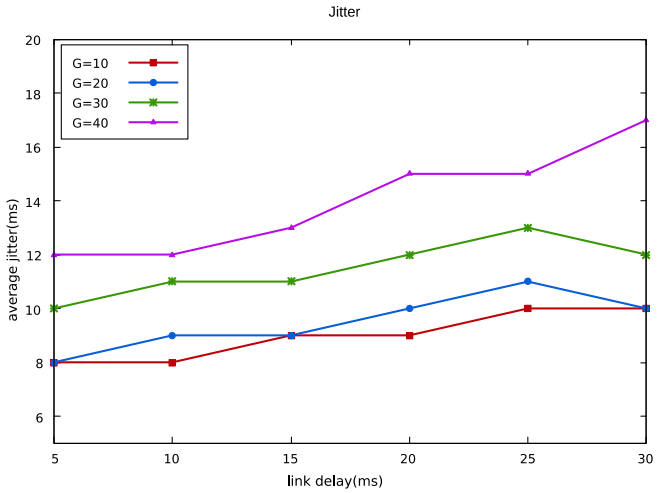

Fig. 5. Average jitter time according to link delays using several group sizes.

Indeed, using large groups can cause immediately network congestion. The reason for this is that with the increase in group size, the density of the nodes typically also increases, and as a result, congestion occurs in the network. Nevertheless, Fig. 6 shows that the average of packet loss stays less than 20% under the worst network conditions (link delay = 30 ms

and group size = 40), this is thanks to the use of the adaptive delay inserted between consecutive requests proposed in this paper.

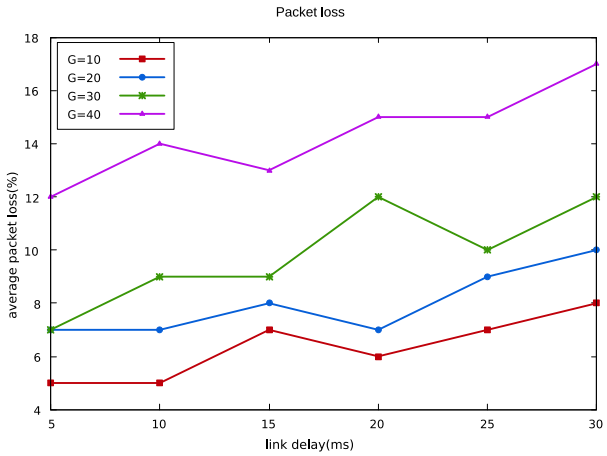

Fig. 6. Average packet loss time according to link delays using several group sizes.

Thanks to its flexibility and its ability to adapt its behavior to different network conditions, our proposition consistently presents high performances and short response times; it has the ability to increase the number of successful transactions and to decrease the packet loss ratio. Consequently, the proposed congestion control algorithm can maintain high performance and reduce the network congestion in almost all the considered scenarios.

V. Conclusion

The Internet of Things (IoT) is now offering the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. It is connecting different devices in our entourage through the use of WSN based on different protocols. One of the most appropriate protocols for lightweight devices and constrained resources in terms of memory, energy, and computing is CoAP. However, in such a network, the problem of congestion is very frequent especially in the case of group via unicast communications. Nevertheless, authors propose solutions for congestion control insensitive to network conditions, the thing which lowers its performances. The challenge is to design a congestion control mechanism for CoAP group communications between a single client and several servers suitable to ensure safe network operation while using network resources efficiently. Thus, in this paper, we present an improved congestion control algorithm, adaptive to network condition for the calculation of the delay to introduce between consecutive requests. In order to evaluate the performance of our proposed solution, we draw simulations under NS2 simulator. Simulated results show that our proposition can appropriately achieve higher performances in terms of response time and packet loss. Future works will consist of applying the idea of the paper to devices in a mobile environment in order to evaluate its performance in such environment.

Acknowledgment

I acknowledge the support provided by my supervisor Pr. Said RAKRAK and the members of the laboratory LAMAI (Laboratory of Mathematics Applied and Informatics) of the Faculty of Science and Technology-Cadi Ayyad University-Marrakesh.

References

[1] P. Middleton, P. Kjeldsen, J. Tully, “Forecast The Internet of Things,” Worldwide, Gartner, Inc., Tech. Rep. 2013.

[2] P.T. Eugster, P.A. Felber, R. Guerraoui, A.M. Kermarrec, “The many faces of publish/subscribe,” ACM Computing Survey. 35: 114–131, 2003.

[3] Z. Shelby, H. Hartke, C. Bormann, “Constrained Application Protocol (CoAP) drafietf core coap 18,” RFC 7252, Ver. 17, 18, 2013.

[4] S. In-Jae, E. Doo-Seop, S. Byung-Kwen, “The CoAP-based M2M gateway fordistribution automation system using DNP3.0 in smart grid environment,” IEEE International Conference on Smart Grid Communications (SmartGridComm), Miami, Florida, 2015.

[5] O. Bergmann, K.T. Hillmann, S.A. Gerdes, “CoAP-gateway for smart homes,” IEEE International Conference on Computing, Networking and Communications (ICNC), Maui, Hawaii, pp 446-450, 2012.

[6] J. Krimmling, S. Peter, “Integration and evaluation of intrusion detection for CoAP in smart city applications,” IEEE Conference on Communications and Network Security (CNS), San Francisco, CA, USA, pp 73-78, 2014.

[7] J. Joshi, D. Kurian, S. Bhasin, S. Mukherjee, P. Awasthi, S. Sharma, S. Mittal, “ Health Monitoring Using Wearable Sensor and Cloud Computing,” International Conference on Cybernetics, Robotics and Control (CRC), Hong Kong, China, pp 104 – 108, 2016.

[8] I. Ishaq, J. Hoebeke, I. Moerman, P. Demeester, “Observing CoAP groups efficiently,” Ad Hoc Networks, vol. 37, P2, pp. 368-388, 2016.

[9] H. Yuan, N. Yugang, G. Fenghao, “Congestion Control for Wireless Sensor Networks: A survey,” Control and Decision Conference, Changsha, China, pp 4853-4858, 2014.

[10] V. Paxson, M. Allman, “Computing TCP's Retransmission Timer,” RFC 2988, 2000.

[11] K. Hartke, “Observing Resources in CoAP Draft-Ietf-Core-Observe-06,” RFC 7641, Ver. 06, 2012.

[12] T. Jaffey, “MQTT and CoAP, IoT Protocols,” Eclipse, 2014.

[13] P. Masek, J. Hosek, K. Zeman, F. Kröpfl, “Implementation of True IoT Vision: Survey on Enabling Protocols and Hands-On Experience,” International Journal of Distributed Sensor Networks, Article ID 8160282, pp. 1-18, 2016.

[14] E.G. Davis, A. Calveras, I. Demirkol, “Improving Packet Delivery Performance of Publish/Subscribe Protocols in Wireless Sensor Networks,” Journal of sensors, vol. 13, pp. 648-680, 2013.

[15] X. Chen X, “Constrained Application Protocol for Internet of Things,” 2014.

[16] Constrained RESTful Environments charter charter-ietf-core-02.

[17] I. Ishaq, J. Hoebeke, F. Van den Abeele, I. Moerman, P. Demeester, “Flexible Unicast-Based Group Communication for CoAP-Enabled Devices,” Sensors, vol. 14, no. 6, pp. 9833-9877, 2014.

[18] I. Ishaq, J. Hoebeke, I. Moerman, P. Demeester, “Experimental Evaluation of Unicast and Multicast CoAP Group Communication,” Sensors, vol. 16, no. 7, pp. 9833-9877, 2016.

[19] C.D. Hou, D. Li, J.F. Qiu, H.L. Shi, L. Cui, “SeaHttp: A Resource-Oriented Protocol to Extend REST Style forWeb of Things,” Computer Sciences Technology Journal, vol. 29, pp. 205–215, 2014.

[20] A. Rahman, E. Dijk, “Group Communication for the Constrained Application Protocol (CoAP),” RFC 7390, 2014.

[21] M. Antonini, S. Cirani, G. Ferrari, P. Medagliani, M. Picone, L. Veltri, “Lightweight multicast forwarding for service discovery in low-power IoT networks,” In Proceedings of 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19, pp. 133–138, 2014.

[22] R. Bhalerao, S.S. Subramanian, J. Pasquale, "An Analysis and Improvement of Congestion Control in the CoAP Internet-of-Things Protocol," Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, USA, pp. 889 – 894, 2016.

[23] C. Bormann, A. Betzler, C. Gomez, I. Demirkol, "CoAP Simple Congestion Control/Advanced, draft bormann-core-cocoa-00," Ver. draft-bormann-core-cocoa, 2016.