Ad Hoc Networks 33 (2015) 126–139

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

CoCoA+: An advanced congestion control mechanism for CoAP

August Betzler *, Carles Gomez, Ilker Demirkol, Josep Paradells

Universitat Politecnica de Catalunya, Barcelona, Spain
Fundació i2CAT, Barcelona, Spain

ARTICLE INFO

Article history:
Received 20 July 2014
Received in revised form 10 February 2015
Accepted 18 April 2015
Available online 29 April 2015

Keywords:
CoAP
Congestion control
Internet of Things

ABSTRACT

The Constrained Application Protocol (CoAP) has been designed by the Internet Engineering Task Force (IETF) for Internet of Things (IoT) devices. Due to the limited radio channel capacities and hardware resources of such devices, congestion can be a serious problem. CoAP addresses this important issue with a basic congestion control mechanism. CoCoA, an Internet-Draft proposal, introduced alternative congestion control mechanisms for CoAP. Yet, there has been limited evaluation of these congestion control mechanisms in the literature. In this paper, we assess the methods applied in CoCoA in detail and propose improvements to address the shortcomings observed in the congestion control mechanisms. We carry out simulations to compare the congestion control performance for default CoAP, CoCoA, and our new proposal, CoCoA+, in a variety of network topologies and use cases. The results show that CoCoA+ outperforms default CoAP and achieves better results than CoCoA in the majority of considered cases.

1. Introduction

IPv6 capable networks of constrained devices play a crucial role in the effort of making the Internet of Things (IoT) [1] part of our everyday lives. The IoT has brought up new challenges for the design of protocols and standards used by devices with limited hardware and communication capacities. The Internet Engineering Task Force (IETF) is developing specifications for different layers of the communication protocol stack that are trimmed to the requirements of networks of constrained, IPv6 capable devices. Amongst others, this resulted in the design of the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [2], the IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) adaptation layer [3], and the Constrained Application Protocol (CoAP) [4]. These important standards have been adopted by entities such as the ZigBee Alliance [5] and Thread [6] to provide IPv6-capable communication protocol stacks for constrained devices in the IoT. Other standard protocol stacks for constrained devices that use IPv6 and 6LoWPAN are ISA 100.11a [7] and IEEE 1451.5 [8].

This paper focuses on the application layer protocol CoAP, designed for networks of constrained devices in the IoT. Such networks suffer from limited memory and processing capacities, as well as low radio bandwidths and relatively high bit error rate. Yet, a major problem for communications in these networks is the phenomenon of congestion. Network congestion can be observed when the generated traffic load in a network gets close to the network capacity or when the queuing and storing capacities of nodes are exceeded. Traffic loads that can cause such congestion are likely to happen in CoAP communications, where messages between large numbers of devices are exchanged.

The CoAP base specification defines a basic congestion control (CC) mechanism to address this important issue. In previous work [9] we analyzed the basic CC mechanisms

* Corresponding author.
E-mail addresses: august.betzler@entel.upc.edu (A. Betzler), carlesgo@entel.upc.edu (C. Gomez), ilker.demirkol@entel.upc.edu (I. Demirkol), josep.paradells@entel.upc.edu (J. Paradells).

http://dx.doi.org/10.1016/j.adhoc.2015.04.007
1570-8705/

implemented by default CoAP and compared them to the alternative CC mechanisms proposed in the CoAP Simple Congestion Control/Advanced (CoCoA) draft version 0 [10]. The results showed that the limited CC capacities of default CoAP can be improved with CoCoA. However, due to several shortcomings that have been detected in its mechanisms and algorithms, CoCoA may perform worse than default CoAP under a variety of network conditions. In this paper, modifications, as well as additions to default CoAP and CoCoA are presented to address these shortcomings, resulting in a new and improved CC mechanism for CoAP, *CoCoA+*.

Using the Cooja simulator [11], different network setups and application scenarios are assessed for a comparative performance evaluation of the three CC methods. An open-source implementation of an IoT stack by ContikiOS [12] has been used to achieve a holistic evaluation. Results show that the improved mechanisms of CoCoA+ promise significant performance improvements for the majority of the evaluated cases: PDR improvements of up to 19.8% and a reduction of average delays during bursts of notification messages of up to 31.2% are measured in comparison to default CoAP.

The rest of the paper is organized as follows. In Section 2, we summarize how CoAP applies CC and we derive three core mechanisms from it. We then analyze how these three core mechanisms are implemented in CoCoA. In Section 3, we present a new approach to CC for CoAP: CoCoA+. In Section 4, we introduce the simulation setup and communication protocol stack configuration that is used to carry out performance evaluations of the new advanced CC approach for CoAP. The results of these evaluations are presented in Section 5. The conclusions of this paper are given in Section 6.

2. CC mechanisms for CoAP

In this section, the way default CoAP implements CC is explained and the mechanisms added by CoCoA are introduced. Based on an analysis of default CoAP CC and CoCoA and through previously obtained results, the shortcomings of these CC mechanisms are identified.

CoAP is a Representational State Transfer style (RESTful) [13] protocol that offers the operations GET, PUT, POST, and DELETE to manipulate resources on servers. CoAP is intended to be a lightweight alternative to HTTP, especially designed for wireless networks of constrained devices that have very limited hardware capacities in terms of memory, processing power, and radio technology. Fig. 1 shows the default communication protocol stack developed by IETF used by a wireless device in a constrained network implementing CoAP. Because CoAP operates on top of UDP, CoAP assumes (optional) end-to-end reliability and CC. Any CoAP exchange, that is the transmission of a CoAP message and the (optional) reception of a reply from a destination node, can be chosen to use confirmable or non-confirmable messages. A confirmable CoAP message, indicated by setting the confirmable flag in outgoing CoAP messages, requires the destination endpoint to reply with an acknowledgment (ACK). On the other hand, if no end-to-end reliability is required by the application, non-confirmable messages, which do not require an end-to-end ACK, may be used.

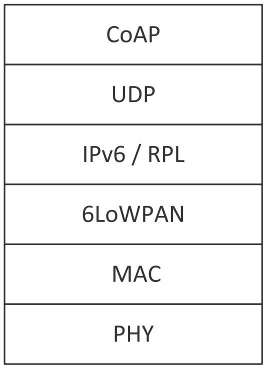

Fig. 1. The default IETF communication protocol stack used for the evaluations in this paper.

Due to the reduced hardware capacities of wireless nodes and the limited capacities of the radio links, congestion can be a common problem in networks of constrained devices. Two problems caused by congestion are packet collisions and full packet buffers, both leading to packet losses and increased delays. If confirmable CoAP messages are lost, retransmissions are used, resulting in additional traffic that may lead to congestion. To avoid contributing to network congestion as a result of frequently retransmitting packets, CoAP applies a basic CC mechanism that is explained in the following.

2.1. CoAP CC

When a confirmable message is sent, CoAP randomly chooses an initial retransmission timeout (RTO) value from the interval between 2 s and 3 s for the initial message transmission. If the timer set to this RTO value expires and the initiator of the message transmission has not received an ACK from the destination endpoint, a loss is assumed and the CoAP message is retransmitted. To avoid network congestion, a binary exponential backoff (BEB) is applied, doubling the RTO value of the retransmitted packet. This CC mechanism applies to all confirmable CoAP messages, independent of the destination endpoint. The parameter NSTART from the base specification of CoAP determines how many exchanges are allowed in parallel towards one particular destination endpoint. The specification states that NSTART should be 1, which is sufficient for most applications running CoAP.

Based on the base specification of CoAP, three basic aspects that make up the CC mechanism for CoAP are identified:

1. The RTO calculation for the initial transmission of a confirmable CoAP message.
2. The backoff behavior applied to the RTO before retransmission of a confirmable CoAP message.
3. The state information stored about destinations of confirmable CoAP messages.