# CoCoA+: An Advanced Congestion Control Mechanism for CoAP

August Betzler, Carles Gomez, Ilker Demirkol, Josep Paradells

*Universitat Politecnica de Catalunya, Barcelona*

*Fundacio i2CAT, Barcelona*

## 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.

*Keywords:* CoAP, Congestion Control, Internet of Things

## 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

---

*Email addresses:* `august.betzler@entel.upc.edu` (August Betzler), `carlesgo@entel.upc.edu` (Carles Gomez), `ilker.demirkol@entel.upc.edu` (Ilker Demirkol), `josep.paradells@entel.upc.edu` (Josep Paradells)

(RPL) [2], the IPv6 over Low-Power Wireless Personal Area Network (6LoW-PAN) adaptation layer [3], and the Constrained Application Protocol (CoAP) [4]. These important standards have been adopted by entities such as the Zig-Bee 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 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

| CoAP |
|:---:|
| UDP |
| IPv6 / RPL |
| 6LoWPAN |
| MAC |
| PHY |

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

## 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 acknowledgement (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.

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.

In the following we take a look at how these three aspects are implemented in CoCoA, an Internet-Draft proposal for an advanced CC mechanism [10]. In this paper we refer to CoCoA as it is defined in version 0 of the draft. The changes made in the versions 1 and 2 of the draft are partially based on the results obtained in the research presented in this paper.

## 2.2. CoCoA

In CoAP, the initial RTO is always picked from a fixed interval, a BEB is applied to retransmissions, and the only state information stored is the current RTO value of an exchange for a certain destination. The CoCoA draft proposes different approaches to these three basic mechanisms in order to provide an improved CC for CoAP. The main difference in the behavior of CoCoA compared to default CoAP is the use of Round-Trip Time (RTT) measurements to calculate the RTO of the first transmission of a CoAP message. A source node of reliable CoAP messages uses ACKs to obtain RTT values. Based on RTT measurements, RTO values are calculated, as it is done for the Transmission Control Protocol [14]. Since delays for CoAP messages can be large due to application processing at the destination node, the initial RTO value is increased from 1 to 2 seconds in CoCoA.

According to Karn's algorithm [14], an RTT measurement is considered to be valid if the corresponding message was not retransmitted. However, in the ambit of constrained networks packet losses due to congestion or lossy links are likely. This increases the probability of message retransmissions, resulting in a lower

probability of obtaining valid RTT measurements. CoCoA therefore runs two RTO estimators in parallel for each destination endpoint: A strong and a weak RTO estimator that are updated when measuring *strong RTTs* or *weak RTTs*, respectively. Strong RTTs are measured when an ACK is received after the first transmission of a confirmable CoAP message. Each strong RTT measurement is used to calculate an estimate average of strong RTTs ($RTT_{strong}$), used to update the strong RTO estimate ($\text{RTO}_{strong}$). In the same way, an estimate average of weak RTTs ($RTT_{weak}$) is used to update the weak RTO estimate ($\text{RTO}_{weak}$). Weak RTTs are measured when an ACK is received after at least one retransmission of a confirmable CoAP message. Due to the ambiguity of not knowing to which of the transmissions an ACK belongs to, a weak RTT is always set to be the duration between the first transmission of the CoAP message and the time of the reception of an ACK. The purpose of using a weak RTO estimator in despite of this ambiguity is to be able to benefit from all RTT measurements.

With the following formulas $RTT_{strong}$ and $RTT_{weak}$ are calculated when a new RTT measurement $RTT_{X\_new}$ is obtained:

$$\text{RTTVAR}_X = (1 - \beta) \times \text{RTTVAR}_X + \beta \times |RTT_X - RTT_{X\_new}| \qquad (1)$$

$$RTT_X = (1 - \alpha) \times RTT_X + \alpha \times RTT_{X\_new}, \qquad (2)$$

where X stands for *strong* or *weak* accordingly and with $\alpha = \frac{1}{4}$ and $\beta = \frac{1}{8}$.

The measurement of a $RTT_X$ is used to update $\text{RTO}_X$ as

$$\text{RTO}_X = RTT_X + K_X \times \text{RTTVAR}_X, \qquad (3)$$

with $K_X = 4$.

According to the CoCoA draft, when $\text{RTO}_{strong}$ or $\text{RTO}_{weak}$ is updated after getting a RTT measurement, an overall RTO ($\text{RTO}_{overall}$) is recalculated:

$$\text{RTO}_{overall} = 0.5 \times \text{RTO}_X + 0.5 \times \text{RTO}_{overall} \qquad (4)$$

The newly calculated $\text{RTO}_{overall}$ is used as next initial RTO ($\text{RTO}_{init}$) for a confirmable message to the same destination endpoint. As in default CoAP in the base specification, dithering is applied to the initial RTO of a CoAP message in CoCoA: the $\text{RTO}_{init}$ for the first transmission is randomly chosen from the interval $[\text{RTO}_{overall}, \text{RTO}_{overall} \times 1.5]$.

According to the CoCoA draft, if one of the RTO estimators has a value of less than 1 second and it is left without further update for more than 16 times its current value, its RTO value should be increased. In the draft, several proposals are made on how to increase the value, amongst others doubling it or setting it to $\frac{1}{8}$ of the time passed since the last update. For simplicity, in our implementation of CoCoA, we update the RTO estimator to a value of 1 second.

### 2.3. Shortcomings of default CoAP and CoCoA

In the previous sections, the default CoAP and CoCoA CC mechanisms were explained. In order to improve CC abilities, CoCoA adds state information about individual RTOs for different destination endpoints, based on RTT measurements. Default CoAP does not gather any end-to-end connection information to adjust the RTO values. Thus, it behaves the same in any type of network, independent of the amount of nodes, the current degree of congestion or any other network characteristic. The addition of state information in CoCoA to calculate adapted RTO values can help to achieve a better performance in congested networks. In previous evaluations of CoCoA [9] it could be shown that a network implementing CoCoA can reach a better performance than one implementing the default CoAP in terms of higher packet delivery ratios (PDRs) and lower Media Access Control (MAC) layer drop rates, especially, when the network is heavily congested. However, when observing the behavior of CoCoA, some shortcomings could be found. These are discussed in this section.

In network scenarios with a high number of packet losses due to congestion, subsequent updates of the weak RTO estimator can cause several undesired effects. CoAP allows a total of four retransmissions of a CoAP message, before considering the exchange to have failed. Therefore, a $RTT_{weak\_new}$ might be obtained after the second, third, fourth, or fifth transmission. For the source of a confirmable CoAP exchange it is not possible to know to which transmission intent a CoAP ACK corresponds to. According to the CoCoA draft, the $RTT_{weak\_new}$ is therefore the time between the reception of the CoAP ACK and the first transmission of the CoAP message.

This behavior has multiple consequences for the calculation of timeout values: first, if a $RTT_{weak\_new}$ is measured after applying multiple retransmissions and then it is used to update $RTO_{overall}$, the new timeout value can grow to a multiple of the previous $RTO_{init}$. Second, the probability of a $RTT_{weak\_new}$ measurement to differ from the actual RTT grows with every retransmission, since it is not clear to which transmission the ACK corresponds. Third, consecutively calculated $RTT_{weak}$ values may differ substantially from each other. According to (3), four times the $RTTVAR_{weak}$ value is added to $RTT_{weak}$ to calculate the new $RTO_{weak}$. Thus, large differences between two subsequent $RTT_{weak}$ values may cause a significant increase of $RTTVAR_{weak}$, which reflects in an important augment of $RTO_{overall}$. Fig. 2a shows the Cumulative Distribution Function (CDF) of $RTTVAR_{weak}$ calculated when using CoCoA in a simulation of a 6x6 grid topology (see the details of the simulation environment and scenarios in Section 4). Less than half of the variances are below 5 s, with a measured average $RTTVAR_{weak}$ of 8.36 s. When nodes measure such high $RTTVAR_{weak}$ values, according to (3) they experience a severe augmentation of their $RTO_{weak}$ value. This causes subsequent retransmissions of CoAP messages to spread over time, which helps to reduce the degree of congestion in the network. However, packet losses may not be caused by congestion but by lossy links, due to strong interferences, occluding objects, nodes with mobility, etc. In such a case, contributions of the weak estimator towards a large $RTO_{overall}$

should be avoided, as they would increase the timeouts for following exchanges further. This can result in an underutilization of the available bandwidth.

While the CoCoA draft states that RTO estimators should increment their RTOs if their RTO values are below 1 s, the document does not state if large RTO values should decay over time. The RTT information that has been used to calculate an RTO may have become obsolete over time and the network conditions may have changed.

Another issue arises if $RTO_{init}$ drops to very low values or exceeds the default RTO of 2 s. We observed that when a confirmable message transmission is initiated with a very low RTO, a node may spend all CoAP retransmissions within a very short amount of time. This is not recommended in the CoCoA draft, however, there is no indication on how to address this problem. On the other hand when an exchange initiates with a very large RTO and applies several retransmissions with BEBs, the duration of the exchange extends significantly in comparison with that of default CoAP. CoCoA also does not include a decay mechanism for long RTOs that may become obsolete after longer periods of not obtaining fresh RTT measurements.

All the shortcomings of the CC mechanisms for CoAP introduced in this section can cause a performance deterioration. In the next section we propose changes and additions to default CoAP and CoCoA to provide an improved CC protocol that addresses these issues.


## 3. CoCoA+: Improved CC for Constrained Device Networks

In the previous section we pointed out several inefficient aspects of default CoAP and CoCoA. In this section we propose changes to the three fundamental aspects of CC mechanisms for CoAP that were derived in Section 2.1, with the goal to improve network performance. The combination of all proposals results in a new advanced CC mechanism for CoAP that we call *CoCoA+*.

CoCoA+ comprises the following proposals to address the issues that have been observed when CoCoA is used as CC mechanism:

1. A modification of the weak estimator calculations to reduce the impact of $RTT_{weak}$ variations and its impact on $RTO_{overall}$.
2. A replacement of the BEB used for retransmissions by a Variable Backoff Factor (VBF).
3. A new approach for aging large $RTO_{overall}$ values.

### 3.1. Modification of the weak estimator

To dampen the impact of large $RTTVAR_{weak}$ values caused by strong fluctuations of $RTT_{weak}$, as explained in Section 2.3, the value of K in the calculation of $RTT_{weak}$ (3) is reduced from 4 to 1. A reduction of the K value to a value greater than 1 is not sufficient, considering the distribution of $RTTVAR_{weak}$ that shows a high probability for large values (Fig. 2a).
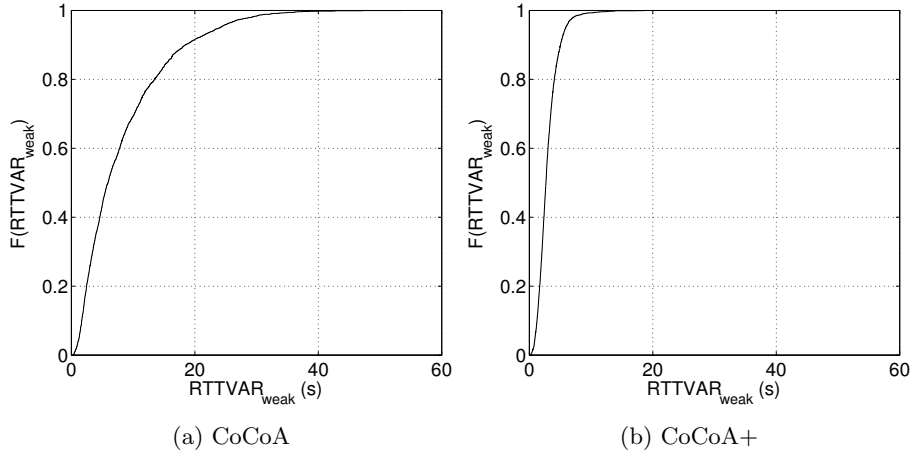
Figure 2: CDF for RTTVAR$_{weak}$ values calculated during simulations of a 6x6 grid at an overall traffic rate of 6 kbps.

Also, in CoCoA+ we halve the weight of the weak estimator to limit the effect of ambiguities inherited in the weak RTT measurements and the potential strong fluctuations by updating the RTO$_{overall}$ formula for RTO$_{weak}$ as

$$\text{RTO}_{overall} = 0.25 \times \text{RTO}_{weak} + 0.75 \times \text{RTO}_{overall} \tag{5}$$

Reducing the weight of the weak estimator further reduces its impact, rendering it inconsequential.

Apart from reducing the impact of RTTVAR$_{weak}$ in the formulas and reducing the weighting of RTO$_{weak}$ in (4), we propose to limit $RTT_{weak}$ measurements to be obtained only from the first transmission and the first retransmission. RTT measurements obtained after the second retransmission are ignored when this limitation is applied. Therefore, large increments of RTO$_{overall}$ due to long $RTT_{weak}$ measurements or large RTTVAR$_{weak}$ values that do not reflect current network conditions are avoided more efficiently. Yet, this mechanism allows to utilize the weak estimator in a more reasonable manner.

As a result of the improvements proposed in this subsection, the RTTVAR$_{weak}$ values are considerably smaller in CoCoA+, when compared to those of CoCoA. The CDF of RTTVAR$_{weak}$ values measured by CoCoA+ is shown in Fig. 2b.

### 3.2. Variable Backoff Factor

The introduction of the VBF is an important change to the backoff behavior used by default CoAP and CoCoA, replacing the BEB applied to retransmissions. Instead of doubling the previous RTO value (RTO$_{previous}$) to obtain the RTO applied to the next retransmission (RTO$_{new}$), it is multiplied by a variable factor

$$\text{RTO}_{new} = \text{RTO}_{previous} \times \text{VBF}, \tag{6}$$

where VBF depends on the $\text{RTO}_{init}$ of a CoAP exchange as follows:

$$\text{VBF}(\text{RTO}_{init}) = \begin{cases} 3, & \text{RTO}_{init} < 1s \\ 2, & 1 <= \text{RTO}_{init} <= 3s \\ 1.3, & \text{RTO}_{init} > 3s \end{cases} \qquad (7)$$

The VBF mechanism introduces alternative backoff values for low and high values of $\text{RTO}_{init}$:

1. If $\text{RTO}_{init}$ of a confirmable message transmission is larger than 3 s, we reduce the backoff factor from 2 to 1.3. By applying this lower backoff factor, the time between two transmissions does not increase as much as in CoCoA, where a high $\text{RTO}_{init}$ could lead to unreasonably large backoff values. At the same time the maximum duration of an exchange is reduced, avoiding a long term blocking that could affect other exchanges with the same destination endpoint.

2. If $\text{RTO}_{init}$ of a confirmable message transmission is smaller than 1 s, we prevent the retransmission mechanism to spend all available retransmissions in a very short time interval. This helps to avoid further congestion, since retransmissions would be carried out very quickly.

The VBF is a novel addition to CoCoA. The choice of the backoff values is based on two criteria: For large $\text{RTO}_{init}$ values, the backoff factor should be smaller than 2 to avoid large idle times. It should, however, not lie below 1 to ensure a safe mechanism in terms of CC. For small $\text{RTO}_{init}$ values, the backoff factor should be greater than 2 to avoid spurious retransmissions but not too large to avoid long idle times. Based on these criteria, the values 1.3 and 3 were chosen, respectively. Several alternative backoff values for small and large $\text{RTO}_{init}$ values were evaluated, where 1.3 and 3 delivered the best results.

Fig. 3 illustrates how RTO values evolve for three different $\text{RTO}_{init}$ values (0.5 s, 1.5 s, and 6 s) when applying the BEB or VBF mechanisms to retransmissions. While the RTO evolves in the same way for default CoAP and CoCoA+ when $\text{RTO}_{init}$ is 1.5 s, clear differences are visible for $\text{RTO}_{init}$ values of 0.5 s and 6 s. For the small $\text{RTO}_{init}$ value of 0.5 s, the RTO grows faster with the VBF than with the BEB to avoid spurious retransmissions. Contrarily, when the $\text{RTO}_{init}$ value is 6 s, the RTO grows slower with the VBF than with the BEB to avoid unnecessarily large idle times in case of losses. Note that in Fig. 3, for example, the 4th transmission intent is the 3rd retransmission.

Fig. 4 compares the total duration of a message transmission with a short $\text{RTO}_{init}$ used with BEB and VBF. With the BEB mechanism, all retransmissions are spent quickly in under 5 s, whereas with the VBF the last retransmissions starts after 10 s. The total duration of the transmission before it times out with a BEB is 7.75 s, while with a VBF it is 30.25 s, respectively. The extension of the total duration of a message and the additional delay between retransmissions gives the destination node more time to reply and helps to avoid further congestion due to spurious retransmissions, respectively.
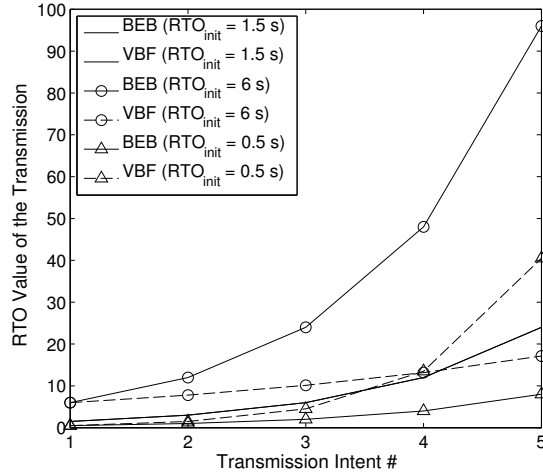
Figure 3: Evolution of the RTO timeout value for each transmission when applying a BEB or VBF to initial RTO values of 0.5 s, 1.5 s, and 6 s.
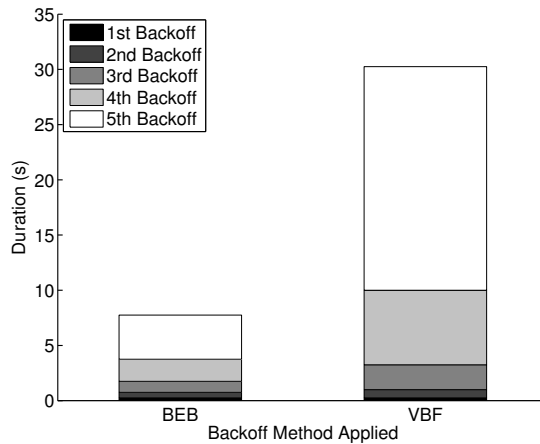


Figure 4: Backoff durations for up to 5 message transmissions, starting with $RTO_{init} = 0.25$ s. The maximum duration of all transmission intents extends from 7.75 s to 30.25 s.

### 3.3. RTO aging

Another improvement of CoCoA+ is an RTO aging mechanism that is applied when RTO estimators with a large $RTO_{overall}$ value are left without updates for a certain time. It is plausible to assume that RTO information may become obsolete after a certain time if no new RTT measurements are carried out to update the RTO estimators. Thus we propose that if $RTO_{overall}$ is larger than the base RTO from the default CoAP specification of 2 s, and it is not updated during more than 30 s, on the next transmission the $RTO_{overall}$ is
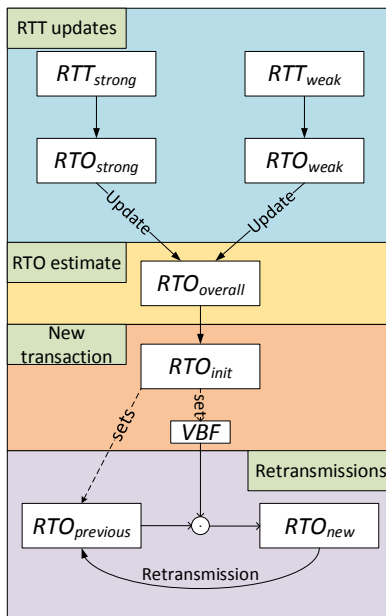
10

Figure 5: An overview of the different RTO variables used to maintain and update the RTO state information for a destination endpoint in CoCoA+.

updated as

$$\text{RTO}_{overall} = (2 + \text{RTO}_{overall})/2 \; s. \tag{8}$$

Applying this formula updates the $\text{RTO}_{overall}$ value of an estimator to a value that is closer to the default value of 2 s. The limit of 30 s for the aging of RTOs was chosen after evaluating several options, including 45 s and 60 s. Applying the 30 s threshold setting in a constant traffic scenario, an average PDR improvement of 0.92% was observed in comparison with the alternative settings of 45 s and 60 s in different network topologies. Thus, the better performing 30 s threshold was chosen for the RTO aging mechanism. However, in other network scenarios and for other use cases, a different aging threshold may work better. An evaluation to determine alternative values or an algorithm to calculate a dynamic threshold is to be carried out as future work.

In this section, the modifications of the weak estimator, the VBF as a new backoff method, and an RTO aging mechanism were introduced to address the shortcomings of default CoAP and CoCoA the CC mechanisms. An overview of the different RTO variables used to maintain and update the CoCoA+ RTO estimates and timers is shown in Fig. 5.

In the next section we define network scenarios and performance metrics for a simulation-driven evaluation of this new proposal, CoCoA+.

## 4. Evaluation Setup

In this section we give the details on the simulation setup used to carry out performance evaluations of the three CC mechanisms presented. This includes the configuration of the simulator, the traffic scenarios, the network topologies, and the performance metrics used to carry out the performance evaluations.

### 4.1. Simulation Setup

In this paper, Cooja [11], as part of the ContikiOS [12] toolset, is used to simulate networks of constrained devices and carry out performance evaluations. One of the key features of Cooja is the emulation of the off-the-shelf wireless sensor node hardware that takes into account the hardware specifications and processing capabilities of real nodes. The compiled binary image file of a node can be uploaded into the simulated nodes, where the compiled program code is then executed with the emulation model of the selected node type during the simulations, as if it were a real node.

For the simulations, the default ContikiOS implementation of the stack as depicted in Fig. 1 is used. At the physical (PHY) and MAC layers, the nodes implement IEEE 802.15.4, using a data transmission rate of 250 kbps in the 2.4 GHz radio band. Radio duty cycling (RDC) of the MAC layer is turned off, since the implementation of RPL routing protocol has been proven to not work properly with RDC in our simulations. MAC layer retransmissions are enabled by default to increase one hop delivery probability. Two types of nodes supporting this configuration of the ContikiOS stack are used in the simulations: Z1 motes from Zolertia [15] and TMote Sky motes from Moteiv [16]. Some key hardware specifications are given in Table 1.

Table 1: Memory, Microcontroller (MCU), and Radio Specifications of the Zolertia Z1 and Moteiv TMote Sky Motes.

|       | TMote Sky    | Z1          |
|-------|--------------|-------------|
| RAM   | 10 kB        | 8 kB        |
| ROM   | 48 kB        | 92 kB       |
| MCU   | MSP430F1611  | MSP430F2617 |
| Radio | CC2420       |             |

The evaluations of the CC mechanisms will be analyzed for three different types of traffic scenarios that are common in the IoT:

1. Constant traffic scenario: a scenario where nodes periodically generate CoAP messages directed towards a sink node. In the IoT, such traffic patterns are observed when data is generated from sensor readings, keep-alive notifications, or other periodical events. Generally, such data needs to be forwarded to a database or service that is capable of storing and processing the data. This network scenario allows to analyze how the CC mechanisms perform for different amounts of offered traffic load.

2. Global event scenario: a scenario where, during the simulation, a global event requires all nodes of the network to transmit one or several packets at the same moment to a sink node. Before and after the global event, the network constantly generates a low amount of background traffic directed to the same sink node. In this scenario it is analyzed how the network reacts to the state of sudden congestion and how long it takes for the network to operate normally again. An example scenario in the IoT is a network where all nodes of the network register a global event simultaneously. For example, this can be heavy vibrations caused by a seismic event that are registered by nodes with accelerometers. Data that is generated in such events needs to be delivered to a sink, where it can be processed. In remote networks with sparse or non-existent infrastructure, the sink can be a gateway device that provides Internet access to the network. This allows the data to be forwarded to external destinations, for example a cloud service.

3. Mixed traffic scenario: a scenario where nodes generate a low amount of constant traffic for a primary sink node but they also generate periodic bursts of notification messages for a secondary sink node. In this scenario the CC mechanisms need to be able to resolve repeatedly states of sudden congestion and at the same time maintain the performance for constant traffic. An example application scenario is a network that collects environmental data, such as temperature, humidity, etc., that is periodically sent to a control center responsible for checking and reacting to the gathered data. Additionally, every minute the nodes send notifications to a proxy node with updated values for a cloud service, that allows web users to check the current state of the network sensor readings. In the IoT, mixed traffic is common when network devices have multiple sensors that can generate different types of traffic for different destinations.

All CoAP messages are POST messages with a size of 96 bytes that are used to update the state of a variable at their destination.

We define four different simulation network topologies with a variable amount of static nodes for these traffic scenarios. The network layout inter alia affects how many direct neighbors each node has, how many nodes compete concurrently for the radio channel, and it determines the diversity of available links between any two nodes of the network. The topologies used for the performance analysis are i) a chain of 17 nodes, ii) a dumbbell topology with 21 nodes, iii) a grid of 36 nodes (6x6), and iv) a grid of 49 nodes (7x7).

Each node is assigned a role that determines the type of CoAP messages it generates and whether the node is a possible destination of CoAP messages from other nodes. Fig. 6 shows the four topologies with one RPL border router (dark node), message generator nodes (bright nodes), primary sink nodes (nodes with a continuous circle), and secondary sink nodes for notifications in the mixed traffic scenario (nodes with a dashed circle). Due to space limitations, the 6x6 grid topology is illustrated as a part of the 7x7 grid topology, however, in the simulations these topologies are separated network scenarios.
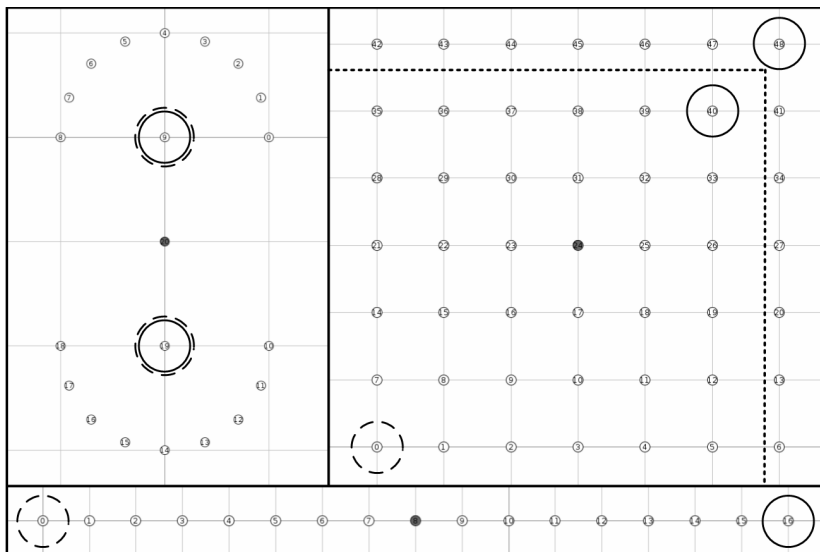
13

Figure 6: The four network topologies used for performance analysis. Starting at the upper left topology and going clockwise: dumbbell, 7x7 grid, 6x6 grid, and chain. The 6x6 grid is depicted as a subset of the 7x7 grid. Nodes with circles are sink nodes for CoAP messages. Dark nodes are RPL border routers. The edges of the unit squares are 10 m long.

The RPL border router is responsible for initiating the Destination Oriented Directed Acyclic Graph (DODAG) (i.e. the network structure created and maintained by RPL) and storing the collected routing information for all nodes that are part of the DODAG [2]. For the simulations, the RPL border router is defined to serve only as relay for CoAP messages: it does not create CoAP messages and is not the destination endpoint of CoAP messages. ContikiOS implements only the storing mode for RPL [2]. RPL border router RAM requirements are higher than those of the rest of the network nodes, which is why RPL border routers are assigned to be TMote Sky motes. In comparison to Z1 motes, which are used for all other nodes of the network, a TMote Sky offers 2 kbyte more of RAM that is used to store routing information. The Z1s, on the other hand, offer a larger amount of ROM, leaving more room for the coding of the applications and CC control mechanisms for CoAP. Since it requires a certain amount of time for RPL to set up the DODAG across the whole network, each simulation starts with a RPL initialization phase of 60 s. During this phase, the network does not generate any traffic apart from RPL control messages.

As soon as the RPL network is set up, nodes periodically generate CoAP messages that are directed towards a primary sink node. In the global event traffic scenario, additionally all nodes send a notification message to the same sink when a global event is detected. In the case of the dumbbell topology, nodes on one side of the dumbbell generate packets for the primary sink on the other side of the dumbbell. Notifications in the mixed traffic scenario are directed

14

towards a secondary sink node that represents a border router/proxy that then forwards the notifications towards an external destination, like a cloud service. In the dumbbell scenario, the notifications are always sent to the secondary sink node that is closest to the origin of the notification (that is, on the same side of the dumbbell). Since external entities like a cloud service cannot be simulated in Cooja, the border router only creates ACKs for the notifications it receives, indicating that the notification to external destination will be handled by the proxy.

For the analysis of the CC mechanism carried out in this paper, a destination endpoint is assumed to be a single IPv6 address, which in the simulations is a single network node. We adopt the recommendation of the CoAP base specification to set NSTART to 1, meaning that only one exchange per destination endpoint is allowed at any time.

For radio transmissions, Cooja's default Unit Disk Graph Medium (UDGM) radio model with circular transmission and interference areas is applied. When a node transmits a packet, other nodes within the transmission range are able to receive the packet. Nodes in the interference range are unable to receive the packet, but it can cause collisions with other, simultaneously transmitted packets. The transmission ranges of the nodes are set to be 10 m, which corresponds to a unit square edge in the grids depicted in Fig. 6. The interference range of the nodes is set to be 20 m.

When applying the UDGM radio model, Cooja calculates a link delivery ratio (LDR) for a radio transmission, determining with which probability a packet can be received correctly by a destination node within the transmission range. Outside of the transmission range, the LDR is always 0. In this paper, we apply two LDR models to the simulations, a static LDR setting and a dynamic LDR setting. When the static setting is applied, the LDR is configured to be always 100%. With this setting, packets are only lost due to packet collisions or due to packet drops as consequence of full buffers.

The dynamic setting is based on a modification of the default UDGM model, where for each transmission a new LDR is calculated. We let the simulator calculate a random LDR for each packet transmission that follows the CDF (Fig. 7) obtained from measurements in a real indoor sensor grid built with 60 TelosB nodes [17]. We limit the LDR for each packet transmission to lie in the interval between 75% and 100%, corresponding to the range of LDR values observed to be normally chosen for data transmissions by link quality aware routing protocols in the grid. Using the dynamic setting, it is possible to observe how the CC mechanisms perform if packet losses occur due to the use of lossy links.

The simulations of the constant traffic and mixed traffic scenarios have a duration of 10 minutes, while the global event scenario has a duration of 5 minutes. For each evaluated configuration, the simulations are repeated 8 times with different random seeds. Using the simulation setup and the network scenarios described in this section, the different CC approaches for CoAP are evaluated using several performance metrics.
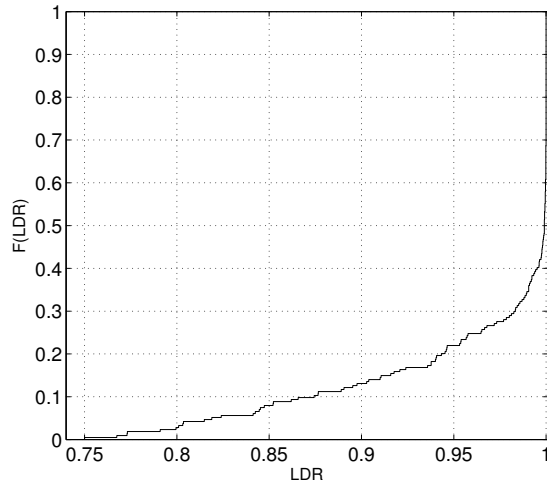
Figure 7: Empirical CDF of LDR values observed in a real testbed in the range of 75% to 100%

*4.2. CC Performance Metrics*

To evaluate and compare the performance of CC mechanisms for CoAP, we chose a set of performance metrics. These metrics, which include qualitative and quantitative metrics shall indicate how well the network performs in the network topologies and traffic scenarios introduced in the previous subsection when using different CC mechanisms.

The first performance metric used to evaluate the simulation results is the overall PDR. We define the overall PDR to be the total amount of received CoAP messages over the amount of sent CoAP messages in a given time interval. The PDR is an indicator of how reliable the network is and whether losses are to be expected. Losses of CoAP messages may lead to network malfunctioning at the application level (e.g. if an alarm is triggered and the message carrying the alarm notification does not reach its destination). As pointed out before, during a congestion state, the probability of losses is high. An effective CC mechanism should be able to detect the state of congestion and take measures to dilute it. As a consequence of applying these measures, we expect CC mechanisms to increase the robustness of CoAP against message losses, thus delivering higher PDR values. From the overall PDR, the network throughput in terms of carried load versus offered load can be derived.

The second performance metric is the end-to-end delay of CoAP messages, which is evaluated by measuring the average delay and the CDF of delays. The end-to-end delay is the time it takes for a CoAP message to reach its destination from the moment it is sent at the application layer of the source node to the moment it is received by the application layer at the destination node.

The third performance metric we choose for our analysis of CC mechanisms for CoAP is the Settling Time (ST). ST is the time in seconds it takes for the

network to revert to a stable state of operation after an event causes a burst of notifications throughout the network. There is no specific formula to calculate ST, thus we define it as follows: we use the average end-to-end delay $D_B$ of CoAP messages measured prior to the event that causes the burst of messages to determine whether the network reverted to a stable state. We define ST to be the duration it takes for the average delay $D_A$ after the burst to get back to a value $D_A <= D_B \times 1.1$. $D_A$ is always calculated for a window of 20 s that is shifted forward in time, starting at the time of the global event, until the aforementioned condition is met and ST can be determined. The ST metric is used as performance metric in the global event scenarios, indicating how resilient the network is against peaks of congestion and how quickly it responds to changes of the network state.

With this set of performance metrics, it can be analyzed how crucial problems caused by congestion, namely packet losses, large delays, and long ST values are addressed by the different CC mechanisms. In the next section the simulation results are presented.

## 5. Performance Evaluation

In the previous section, the simulation setup, as well as the analyzed scenarios, and the performance metrics were introduced. In this section the results for the CC performance analysis of default CoAP, CoCoA and CoCoA+ are presented. First, the results for the constant traffic scenario are given and discussed, followed by the results for the global event scenario and the mixed traffic scenario.

### 5.1. Constant Traffic Scenario

In the constant traffic scenarios the CC algorithms are evaluated in 4 different topologies and for varying traffic loads. The traffic generation rate at the nodes is adjusted to vary the average network-wide traffic load from 1 kbps to 10 kbps in steps of 1 kbps.

Considering the simulation results with the lossy link model as detailed in Section 4 (Fig. 7), in the majority of cases CoCoA and CoCoA+ increase network performance. In the 6x6 grid topology, the three CC mechanisms perform similarly up to 4 kbps. At higher offered loads CoCoA achieves the highest throughput, followed by CoCoA+ and default CoAP. In the 7x7 grid an analogical behavior of the CC mechanisms as in the 6x6 grid can be observed. In the chain topology, CoCoA+ is able to perform better than default CoAP and CoCoA up to 9 kbps, then it is surpassed by CoCoA. The use of advanced CC mechanisms in the dumbbell topology does not improve performance up to higher offered load (6 kbps). Until reaching this mark, default CoAP performs slightly better than CoCoA+, and CoCoA performs worst.

This means, CoCoA is generally able to provide a good performance compared to default CoAP in the considered conditions (Constant Traffic Scenario, and with MAC layer reliability enabled). The only exception has been observed
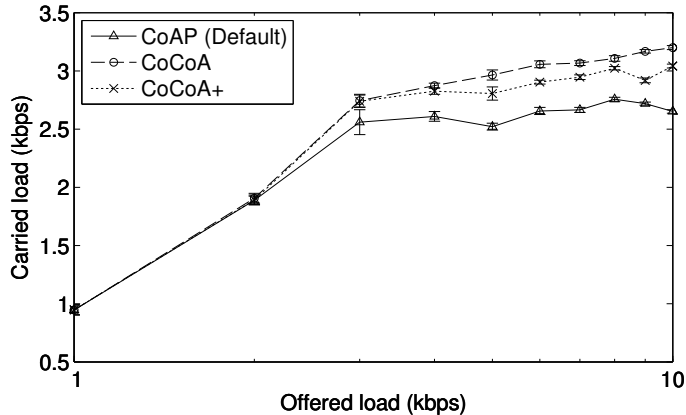
17

Figure 8: Average throughput with standard deviation achieved for different offered loads in the 6x6 grid topology with lossy links.

in the dumbbell topology at low offered traffic rates, where CoCoA underperforms the other CC mechanisms. On the other hand, CoCoA+ performs always better or at least very similar to default CoAP and also performs better than CoCoA in the chain and dumbbell topology. Due to space limitation, only the results for the 6x6 grid topology are presented in detail, as an example for a network setup where CoCoA is able to outperform default CoAP and CoCoA+. We explain under which conditions this is possible and show why in other setups the behavior of CoCoA can lead to degradation of performance. The detailed results for the other topologies are available online [18].

In the 6x6 grid topology (Fig. 8), up to offered loads of 2 kbps the three CC mechanisms perform similarly. Both advanced CC mechanisms outperform default CoAP at traffic rates above 2 kbps, where congestion starts having a relevant impact on the overall performance of the network. As with the traffic rates congestion increases, a higher performance is achieved with CoCoA than with CoCoA+. While CoCoA initializes RTOs with large values due to the contribution of the weak estimator to the $RTO_{overall}$, CoCoA+ normally does not initialize exchanges with such large RTO values. In Figs. 9a and 9b the values of $RTO_{init}$ in the 6x6 grid scenario with an offered traffic load of 6 kbps are displayed for CoCoA and CoCoA+. While the initial timeouts for a message transmission accumulate below 10 s with CoCoA+, the values spread widely above this value when using CoCoA. The mean $RTO_{init}$ for CoCoA in this case is 9.1 s, while the average value of CoCoA+ is 2.8 s, which is very close to default CoAP's average of 2.5 s.

The main reason for the distribution of $RTO_{init}$ values lies in the limitation of the weak estimator to allow RTT measurements of only up to the first retransmission and the reduced range of $RTTVAR_{weak}$ in CoCoA+ (Fig. 2b). However, with packet drops as a result of congestion, at higher traffic rates the use of multiple retransmissions due to packet losses is likely. Therefore,
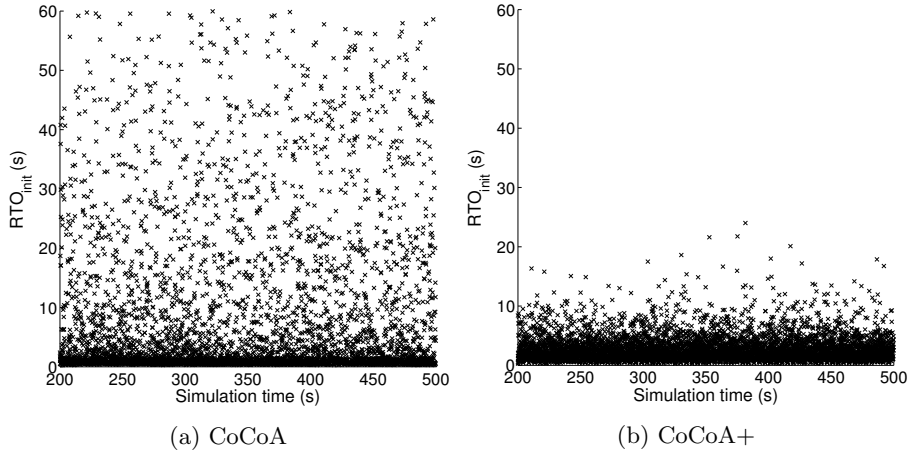
Figure 9: Comparison of $RTO_{init}$ applied to message transmissions during 5 minutes in the simulation of the 6x6 grid topology with lossy links at a traffic rate of 6 kbps.

CoCoA+ does not update its RTO timers with weak RTTs frequently and when it does, the measured RTTs are considerably smaller than the ones measured with CoCoA. As a consequence, $RTO_{overall}$ values calculated by CoCoA+ are not as large as those calculated by CoCoA. This results in larger $RTO_{init}$ values and in larger backoffs for retransmissions in CoCoA, leading to a reduction of the traffic across the network, which has a positive effect on the congestion and is the reason for CoCoA to perform better than CoCoA+ in this scenario. However, in other network scenarios, the RTO behavior of CoCoA can lead to severe underperformance, which will be shown later.

The simulation results for the 100% LDR links setting are very similar to the ones obtained for lossy links [18]. The main reason for the similar results is the MAC layer reliability mechanism that allows up to three retransmissions of a frame at the link layer. This leads to high one-hop delivery ratios, even with lossy links. However, normally, the application layer does not have any information about characteristics of lower layer (e.g. radio interface) protocols used. Since the MAC layer reliability (which is optional in IEEE 802.15.4) might have a relevant impact on the network performance, we want to observe how the results may change when MAC layer retransmissions are disabled. Network performance with disabled MAC layer retransmissions and a 100% LDR changes significantly when compared to the case with enabled MAC layer retransmissions. Fig. 10 shows a completely different evolution of the throughput in the 6x6 grid topology, where CoCoA+ outperforms default CoAP and CoCoA. The latter performs noticeably worse than default CoAP and CoCoA+ in this network scenario.

The amount of message transmissions that start with large RTO values in CoCoA is high and the timeouts get larger when retransmissions are necessary. While delaying retransmissions by using large timeouts may help to reduce
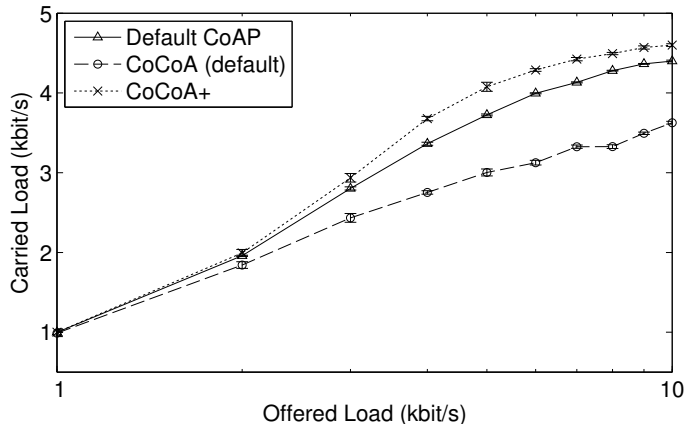
Figure 10: Average throughput with standard deviation achieved for different offered loads in the 6x6 grid topology without MAC layer retransmissions and 100% LDR links.

congestion, waiting for ACKs that will not arrive in case of packet losses blocks the exchange of other messages to the destination node due to the NSTART = 1 limitation. This behavior can lead to a substantial reduction of the PDR due to packet drops at application layer. The average timeouts applied by CoCoA+ are larger than those of default CoAP, but their range is reduced in comparison to CoCoA, since $RTT_{weak}$ measurements are only allowed up to the first retransmission. Above that, CoCoA+ uses the VBF on large $RTO_{init}$ values to limit the growth of the retransmission backoff. As a result of the improvements included in CoCoA+, when it comes to providing end-to-end reliability without additional per-hop reliability, CoCoA+ turns out to be the best solution, while CoCoA performs worst. The same tendency for the scenario with disabled MAC layer retransmissions also applies to the other topologies [18].

The results for the constant traffic scenario show that in networks with lossy and error-free links CoCoA normally delivers a good performance. Within these scenarios, CoCoA+ is generally able to outperform CoCoA at low traffic rates and performs very similar to or noticeably better than default CoAP. In scenarios with disabled MAC layer retransmissions, CoCoA+ outperforms the other CC mechanisms, and CoCoA underperforms default CoAP to a significant extent.

## 5.2. Global Event Scenario

In the next step of the performance evaluation, the settling time (ST) is analyzed for the global event scenarios. In these scenarios the nodes produce an overall average network load of 4 kbps in the grid topologies and 2 kbps in the other topologies towards the primary sink. After 2 minutes a global event causes all CoAP nodes to send a notification message to the same sink. This burst of notifications causes a sudden state of congestion across the network

20

that needs to be resolved by the CC mechanisms. After the burst, the network continues to operate normally again. The performance of the CC mechanisms in this scenario is measured by calculating the ST, which indicates the number of seconds that have to pass until a state close to the initial network state is recovered.

When the global event occurs, all nodes send notifications to the sink, which results in heavy congestion with packet losses and thus retransmissions are necessary. During such a burst, CoCoA and CoCoA+ measure many weak RTTs, leading to an increase of the overall RTO. After transmitting all the notifications, the grid starts operating normally again. Default CoAP has no state memory and applies the default $RTO_{init}$ to all message transmissions. Upon a state of sudden congestion it therefore does not adapt its RTO values. The RTO timers of CoCoA and CoCoA+ after the burst are still affected by the RTT measurements obtained during the state of heavy congestion and need to adapt to the normal traffic again.

Table 2 lists the STs for the three CC mechanisms in all topologies considered for the static 100% LDR setting and the lossy link setting (75% - 100% LDR) with enabled MAC layer retransmissions. In addition to the average ST, the 95% confidence intervals are given, indicating how much the ST values oscillate for the repetitions of the burst traffic simulations. In the grid and chain topologies, CoCoA+ recovers from congestion faster than the other CC mechanisms, except for the 7x7 grid scenario with dynamic LDRs, where default CoAP recovers as fast. When using CoCoA, it takes longer for the network to get back to its original state, since the RTOs are set to very high values and packet losses lead to large delays. Also, high RTO values calculated during the notification burst require the exchange of several CoAP messages in the not congested network to get back to low values. Default CoAP does not increase its RTO timers during the burst event and therefore continues transmitting messages during and after the burst with default RTOs. As a result, it does not adapt its initial RTOs like CoCoA and CoCoA+ do. One one hand this means that it avoids the issues CoCoA has when adapting to the congestion by heavily incrementing the RTOs. On the other hand, it does not increase the RTO values at all. As a consequence of behaving independently from network conditions, it can be observed that the performance of default CoAP in comparison to the other CC mechanisms varies with each topology. In the grid topologies it performs better than CoCoA and in the chain and dumbbell topologies it performs worse than CoCoA. However, CoCoA+ always outperforms default CoAP or performs at least as well.

In the dumbbell topology, CoCoA performs better in terms of ST than CoCoA+. This can be explained by comparing the PDR values of the two settings. While in the majority of scenarios the PDR obtained by the three CC mechanisms differs only slightly ($< 2\%$), in the dumbbell topology a difference in the PDR of more than 10% in favor of CoCoA+ has been observed. In this specific case, the higher reliability comes at the cost of a larger settling time, since a higher delivery ratio is achieved after the burst, at the expense of increased delivery delay. Leaving this specific case apart, CoCoA+ is the mechanism that adapts the fastest to short term changes of the network conges-

tion and performs similarly or better than default CoAP in all the considered scenarios.

Table 2: Average Settling Times and 95% Confidence Intervals of the Settling Times for Different Topologies and LDR Settings (the best performing mechanism is highlighted with bold letters).

| Topology / LDR setting | Default CoAP | CoCoA | CoCoA+ |
|---|---|---|---|
| 6x6 Grid / 100% | $47 \pm 6$ s | $57 \pm 6$ s | $\mathbf{38 \pm 7}$ **s** |
| 6x6 Grid / Lossy | $33 \pm 10$ s | $42 \pm 9$ s | $\mathbf{21 \pm 6}$ **s** |
| 7x7 Grid / 100% | $44 \pm 11$ s | $48 \pm 7$ s | $\mathbf{36 \pm 7}$ **s** |
| 7x7 Grid / Lossy | $23 \pm 7$ s | $33 \pm 6$ s | $\mathbf{19 \pm 8}$ **s** |
| Chain / 100% | $\mathbf{15 \pm 8}$ **s** | $17 \pm 9$ s | $15 \pm 9$ s |
| Chain / Lossy | $14 \pm 5$ s | $12 \pm 8$ s | $\mathbf{10 \pm 6}$ **s** |
| Dumbbell / 100% | $19 \pm 8$ s | $\mathbf{8 \pm 5}$ **s** | $12 \pm 6$ s |
| Dumbbell / Lossy | $17 \pm 5$ s | $\mathbf{8 \pm 4}$ **s** | $13 \pm 2$ s |

*5.3. Mixed Traffic Scenario*

The last part of the performance analysis focuses on the mixed traffic scenario with two separate sinks. While the traffic load directed to the primary sink is fixed at an overall network average of 3 kbps, the notifications for the secondary sink are generated every 60 s. This results in parallel cross traffic for this more complex scenario, in which the CC mechanisms need to be able to adapt to repeated global bursts of notifications that compete with the constantly generated traffic.

This scenario is tested with the static and dynamic link settings with enabled MAC layer retransmissions. Moreover, it is evaluated how the network performance changes, if different amounts of notifications (1, 2, or 3) have to be transmitted every 60 seconds. The results shown in the following only consider the case for 1 notification, the rest of the results can be found online [18]. The performance of the CC mechanisms in this scenario is determined by comparing the overall PDR, the average delay from the start of the notification bursts up to 10 s after the burst, and the distribution of end-to-end delays.

Table 3 shows the overall PDR values for all three CC mechanisms, including the 95% confidence intervals for both LDR settings when a single notification is sent per burst. As seen in the table, except a minor deterioration of 0.4% in one scenario, in all other scenarios, CoCoA+ achieves a better performance than default CoAP and CoCoA.

Delay is also an important metric, especially for applications that require fast reactions and short notification times, such as alarms. In the mixed traffic scenario the average delays of message transmissions initiated at the beginning of each burst up to 10 s after each burst are calculated. Table 4 shows the average delays with their corresponding 95% confidence intervals for all three CC mechanisms. In the majority of cases, CoCoA suffers from larger delays than default CoAP, unlike CoCoA+ which is able to deliver CoAP messages faster.

Table 3: Overall PDR Values with 95% Confidence Intervals for Different Topologies and LDR Settings (the better performing mechanism is highlighted with bold style).

| Topology / LDR setting | Default CoAP | CoCoA | CoCoA+ |
|---|---|---|---|
| 6x6 Grid / 100% | $85.2 \pm 0.8\%$ | $82.7 \pm 1.8\ \%$ | $\mathbf{85.8 \pm 0.9\%}$ |
| 6x6 Grid / Lossy | $76.7 \pm 2.1\%$ | $73.9 \pm 3.2\%$ | $\mathbf{79.2 \pm 3.9\%}$ |
| 7x7 Grid / 100% | $80.6 \pm 2\%$ | $78.3 \pm 1.5\%$ | $\mathbf{82.3 \pm 2\%}$ |
| 7x7 Grid / Lossy | $67.1 \pm 1.9\%$ | $70.3 \pm 1\%$ | $\mathbf{73.6 \pm 1.9\%}$ |
| Chain / 100% | $\mathbf{92.7 \pm 0.2\%}$ | $91.9 \pm 0.8\%$ | $92.3 \pm 0.6\%$ |
| Chain / Lossy | $88 \pm 0.8\%$ | $86.8 \pm 1\%$ | $\mathbf{89.1 \pm 0.8\%}$ |
| Dumbbell / 100% | $33.5 \pm 0.5\%$ | $30.3 \pm 0.9\%$ | $\mathbf{40.2 \pm 0.5\%}$ |
| Dumbbell / Lossy | $36.5 \pm 3.5\%$ | $31.7 \pm 4.6\%$ | $\mathbf{38.2 \pm 3.1\%}$ |

An exception to this pattern has been observed in the dumbbell topology. In this topology, the mechanisms used by CoCoA+ are providing a substantially higher PDR than default CoAP, which comes at the cost of larger delays. Also, it needs to be taken into account that the average delays without congestion in the dumbbell topology are rather short (less than 100 ms), thus, differences of the average delay values between the three CC mechanisms are rather small. Despite of the slightly worse performance in terms of average delays in the dumbbell topology, using CoCoA+ is the best option if CoAP messages need to be delivered fast. Relative delay decrease of more than 30% is possible with CoCoA+ in comparison to default CoAP, while providing a higher reliability in all of the considered cases.

Table 4: Average Delays and Correspondent 95% Confidence Intervals During Bursts and 10 s After the Burst for Different Topologies and LDR Settings (the best performing mechanism is highlighted with bold style).

| Topology / LDR setting | Default CoAP | CoCoA | CoCoA+ |
|---|---|---|---|
| 6x6 Grid / Static | $1444 \pm 50$ ms | $2051 \pm 123$ ms | $\mathbf{1350 \pm 44}$ ms |
| 6x6 Grid / Dynamic | $2838 \pm 122$ ms | $3283 \pm 271$ ms | $\mathbf{2392 \pm 188}$ ms |
| 7x7 Grid / Static | $2676 \pm 215$ ms | $3083 \pm 287$ ms | $\mathbf{2259 \pm 172}$ ms |
| 7x7 Grid / Dynamic | $5014 \pm 287$ ms | $5401 \pm 187$ ms | $\mathbf{3820 \pm 170}$ ms |
| Chain / Static | $\mathbf{476 \pm 29}$ ms | $545 \pm 29$ ms | $486 \pm 34$ ms |
| Chain / Dynamic | $976 \pm 31$ ms | $997 \pm 46$ ms | $\mathbf{939 \pm 26}$ ms |
| Dumbbell / Static | $63 \pm 3$ ms | $\mathbf{55 \pm 3}$ ms | $76 \pm 6$ ms |
| Dumbbell / Dynamic | $92 \pm 10$ ms | $\mathbf{88 \pm 13}$ ms | $120 \pm 23$ ms |

The delay information given so far does not depict information about the distribution of the delays. Therefore, we evaluate the CDF of the delays for the different CC mechanisms. End-to-end delays of 60 s and more appear as 60 s in the statistic. Unsuccessful CoAP exchanges with an 'infinite' delay are integrated in the CDF. Since analyzing the CDFs for each combination of network topology and traffic scenario is not possible due to space restrictions, only the CDFs of an example scenario are shown. This scenario is representative
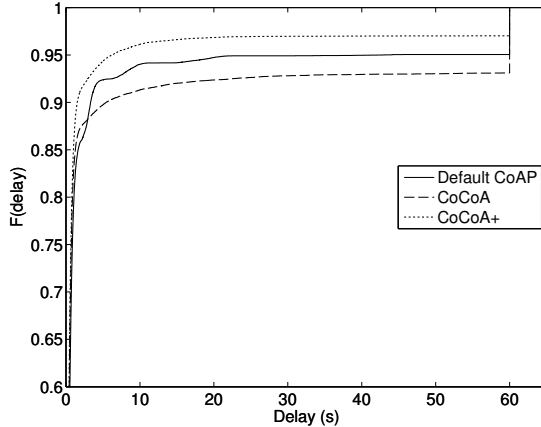
Figure 11: CDF of delays for the 7x7 grid topology with lossy links and bursts of 1 notification in the mixed traffic scenario.

and shows a general tendency for cases where CoCoA+ achieves higher PDR values than the other two CC mechanisms. According to the previously shown results, this is applies to a large part of the considered cases.

Fig. 11 shows the CDF for end-to-end delays in the 7x7 grid scenario with lossy links, where 1 notification per burst is transmitted. The CDF comparison reveals that CoCoA+ clearly has the highest probability of achieving short delays ($< 10\,s$), while CoCoA clearly underperforms default CoAP except for very short delays ($< 3\,s$). The CDF curves of default CoAP show a staircase pattern that repeats throughout all obtained CDFs. Since default CoAP applies a BEB to initial RTO values from a fixed interval, retransmissions are carried out also within a fixed time interval, leading to the visible accumulation of delays. CoCoA and CoCoA+ do not show these patterns, as their initial RTO is not chosen from a fix interval, but dynamically changes depending on the measured RTT values. The RTO calculations of CoCoA+ result in the highest probability of delays below 10 s. CoCoA also has higher probability of providing delays lower than about 3 s, however, due to the lower PDR achieved by CoCoA when compared to the other CC mechanisms, the probability of delays greater than 3 is the smallest. Eventually, all curves reach an asymptotic value after some time, to jump to 100% at 60 s. For the observed scenario this indicates that there are very few measurements with large delays shorter than 60 s. Most of the delays larger than 60 s are caused by lost packets that have an 'infinite' delay. Judging from the CDF, CoCoA+ offers the best performance with the highest probability for short delays.

In this section the performance of three CC mechanisms for CoAP in three traffic scenarios and four topologies has been evaluated. The results show that there is no CC mechanism that always performs best in all scenarios and for all performance metrics. While CoCoA performs well in a variety of scenarios,

it often performs noticeably worse than default CoAP. Our new proposal for an advanced CC mechanism for CoAP, CoCoA+, performs better than default CoAP in the majority of scenarios or at least very similarly. The results in this paper are an important and necessary step towards the definition of a solid alternative to the CC mechanism provided by default CoAP.

## 6. Conclusions

CoAP plays an important role in the ambit of constrained networks and the IoT, yet, so far only limited effort has been put into the evaluation of its CC mechanisms and into finding a suitable advanced CC mechanism. Congestion is an important issue in networks of constrained devices and can lead to malfunctioning or poor network performance. In this paper, by analyzing default CoAP and CoCoA, several CC key mechanisms are identified and the shortcomings of their implementation are exposed. Through the proposal of modifications of these mechanisms and the addition of new mechanisms, we create a new advanced CC mechanism for CoAP: CoCoA+. The performance analysis of default CoAP, CoCoA, and CoCoA+ reveals that CoCoA+ is able to outperform default CoAP and CoCoA in most of the evaluated topologies and scenario settings. CoCoA+ then provides a higher degree of reliability and lower delays. In the analyzed scenarios and topologies, a PDR improvement of up to 19.8% and a reduction of average delays during bursts of notifications of up to 31.2% were observed in comparison to default CoAP. Additionally, it is resilient against sudden changes of network traffic and adapts quickly to different states of network congestion. In some scenarios, CoCoA+ does not achieve the same performance as CoCoA, however, it then performs still better or very similar to default CoAP. On the contrary, CoCoA is unable to provide a consistently better performance than default CoAP, often underperforming it significantly.

Considering that a main criterion for the design of an advanced CC mechanism for CoAP is to assure that it does not underperform default CoAP, CoCoA is not a suitable candidate. Based on this criterion and the results obtained in this paper, we consider CoCoA+ to be a very solid and promising proposal for an advanced CC mechanism for CoAP. A large part of the findings of this study has been used to shape version 1 and 2 of the CoCoA draft [10]. In further work, evaluations in real networks of constrained devices and the implication of interactions with devices over Internet will be necessary to confirm the improvements achievable by CoCoA+. Furthermore, investigations will be required to determine optimized parameter settings for some of the newly introduced CoCoA+ mechanisms, such as the backoff factors for the VBF and the threshold setting for the RTO aging mechanism. The evaluation of these parameters should involve the exploration of dynamic values that adapt to specific network conditions or use cases.

## 7. Acknowledgments

## References

[1] Z. Shelby, C. Bormann, 6LoWPAN: The Wireless Embedded Internet, Wiley Publishing, 2010.

[2] T. Winter, P. Thubert, J. Hui, P. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, Tech. Rep. 6550, RFC Editor, Fremont, CA, USA (Mar. 2012).
URL http://www.rfc-editor.org/rfc/rfc6550.txt

[3] N. Kushalnagar, G. Montenegro, C. Schumacher, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, RFC 4919 (Informational) (August 2007).
URL http://www.ietf.org/rfc/rfc4919.txt

[4] Z. Shelby, K. Hartke, C. Bormann, The constrained application protocol (CoAP) (RFC 5741) (June 2014).
URL https://tools.ietf.org/html/rfc7252

[5] ZigBee Alliance, ZigBee (2015).
URL http://www.zigbee.org

[6] Thread Group, Thread (2015).
URL http://www.threadgroup.org/Home.aspx

[7] ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: Process control and related applications, Tech. rep. (2011). doi:10.1109/MSP.2010.49.
URL http://dx.doi.org/10.1109/MSP.2010.49

[8] IEEE, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats (October 2007).

[9] A. Betzler, C. Gomez, I. Demirkol, J. Paradells, Congestion control in reliable coap communication, in: Proceedings of the 16th ACM International Conference on Modeling, Analysis & Simulation of Wireless and Mobile Systems, MSWiM '13, ACM, New York, NY, USA, 2013, pp. 365–372.
URL http://doi.acm.org/10.1145/2507924.2507954

[10] C. Bormann, A. Betzler, C. Gomez, D. Ilker, CoAP Simple Congestion Control/Advanced (work in progress) (July 2014).
URL http://tools.ietf.org/id/draft-bormann-core-cocoa-02.txt

[11] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with cooja, in: Local Computer Networks, Proceedings 2006 31st IEEE Conference on, 2006, pp. 641–648.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4116633

[12] A. Dunkels, B. Grönvall, T. Voigt, Contiki - a lightweight and flexible operating system for tiny networked sensors (2004).

[13] R. T. Fielding, Architectural styles and the design of network-based software architectures, Ph.D. thesis (2000).

[14] C. Paxson, M. Allman, J. Chu, M. Sargent, Computing TCP's Retransmission Timer (RFC 6298) (June 2011).
URL http://tools.ietf.org/html/rfc6298

[15] Zolertia, Z1 low-power wireless sensor network module (March 2010).
URL http://www.zolertia.com/products/z1

[16] Moteiv Corporation, TMote Sky: Ultra low power IEEE 802.15.4 compliant wireless sensor module (June 2006).
URL http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

[17] C. Gomez, A. Boix, J. Paradells, Impact of LQI-based routing metrics on the performance of a one-to-one routing protocol for IEEE 802.15.4 multihop networks, EURASIP J. Wirel. Commun. Netw. 2010 (2010) 6:1–6:20. doi:10.1155/2010/205407.
URL http://dx.doi.org/10.1155/2010/205407

[18] A. Betzler, Extended simulation results for CoCoA+ (all topologies and traffic scenarios) (March 2014).
URL https://sites.google.com/site/augustbetzler/cocoa-simulation-results