ARTICLE TYPE

An optimization-based congestion control for constrained application protocol†

Godfrey A. Akpakwu*[4,1] | Gerhard P. Hancke[2,1] | Adnan M. Abu-Mahfouz[3,1]

[1]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa
[2]Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
[3] Council for Scientific and Industrial Research, Pretoria, South Africa
[4]Department of Electrical and Electronics Engineering, Federal University of Agriculture, Makurdi, Nigeria

Correspondence
*Godfrey A. Akpakwu, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0084, South Africa. Email: godfreyspa@gmail.com

Present Address
Present address

Summary

The Constrained Application Protocol (CoAP) is a lightweight web transfer protocol designed based on the REST architecture standardized by the Internet Engineering Task Force (IETF) to meet and accommodate the requirements of the constrained Internet of Things (IoT) environments. Managing congestion control in a resource-constrained lossy network with a high bit error rate is a significantly challenging task that needs to be addressed. The primary congestion control mechanism defined by CoAP specification leverages on basic binary exponential backoff and often fails to utilize the network dynamics to the best of its traffic conditions. As a result, CoCoA has been introduced for better IoT resource utilization. In addition, CoCoA retransmission timeout (RTO) for network dynamics is based on constant coefficient values. The resource-constrained nature of IoT networks poses new design challenges for congestion control mechanisms. In this paper, we propose a new particle swarm optimization (PSO)-based congestion control approach called psoCoCoA as a variation of CoCoA. The psoCoCoA applies random and optimal parameter-driven simulation to optimize default CoAP parameters and update the fitness and velocity positions to adapt to the traffic conditions. This process is performed for different traffic scenarios by varying the retransmission and max-age values by using the optimization-based algorithm. We carried out extensive simulations to validate the congestion control performance for CoAP with Observe, CoCoA, and psoCoCoA with different network topologies. The results indicate that psoCoCA outperforms or very similar to CoCoA and achieves better performance compared to CoAP with Observe under different network scenarios.

KEYWORDS:
Internet of Things; Congestion Control; Particle Swarm optimization

1 | INTRODUCTION

The ever-growing Internet of Things (IoT) is paving the way for the global network infrastructure, and the number of devices interconnected to the IoT is proliferating, which in turn leads to several diverse smart applications and services [1]. In view of the tremendous potential of the IoT, it is predicted that more than 50 billion Internet-capable smart things will be connected over the Internet shortly and this will tend to revolutionize the global world [2]. The IoT connects numerous heterogeneous devices by applying a wide range of technologies that include communication, networking, and information processing and ensures inter-operability among global Internet services by integrating smart objects into the existing network and information systems [3,4,5].

In the IoT, the application layer protocols are responsible for orchestrating the network that would be utilized by constrained devices with limited hardware and computation capabilities. To this end, the Internet Engineering Task Force (IETF) CoRE Working Group (CWG) standardized the Constrained Application Protocol (CoAP) [6]. CoAP is a lightweight protocol stack which adapts the Representational State Transfer (REST) paradigm [7] for easy compatibility with the current Internet trend. CoAP has specialized features, including multicast support, very low overheads, and suitability for resource-constrained lossy and low powered networks (LLNs) [8,6,9] and improves the ability to work with machine-to-machine (M2M) applications [10]. CoAP end-nodes exchange requests-responses by using messages exchanged asynchronously through User Datagram Protocol (UDP) [11]. The CoAP is designed to meet the requirements of LLN and to provide lightweight inter-operability among the global Internet services [12,13]. By default, the CoAP has no end-to-end congestion control (CC) mechanism, and therefore it includes a separate CC scheme as part of the CoAP base specification. Still, the solution of a CC mechanism is not yet mature, especially when the traffic load on the IoT network approaches the network capacity. The mechanisms supported by the Transmission Control Protocol (TCP) in most conventional Internet applications support end-to-end CC. However, traditional Internet protocols are not suitable for resource-constrained environments due to the dynamic characteristics of LLN [14]. Moreover, the CoAP does not rely on TCP, as it operates over UDP to enable lightweight applications and to implement the CC scheme by itself [15]. As a result, different protocols have been considered [16]. In IoT network, the involvement of heterogeneous devices with a wide range of applications creates network congestion when the traffic load reaches its maximum capacity, and degrades the performance of the IoT system. The main reason for the IoT congestion is bursty traffic, which represents the scenario of high-bandwidth data transmission over a short period, resulting from the huge number of connected devices. Therefore, congestion management is crucial to maintain the network stability backbone while providing the end-user with quality of experience (QoE) to ensure and guarantee optimal quality of service (QoS) delivery. The aim of the scheme is to ensure that necessary management policies are

[†]This is an example for title footnote.

applicable at the appropriate phases and granularities. The traffic pattern in the IoT network is different from that of conventional networks, and congestion is a significant factor that degrades the performance of data communications [17]. Therefore it is necessary to introduce CC mechanisms into IoT to provide real-time QoS.

Even though, the IoT data size is small; the sensor devices communicate periodically to notify their measurements and ACKs. This periodic notification results in network congestion [15]. Typically, memory-constrained and CPU-constrained nodes are used in IoT communications with limited processing and memory capabilities, which makes CoAP suitable for these extreme constraints. Accordingly, RFC 7641 [18], extends CoAP with Observe, a publish-subscribe mechanism for supporting unreliable CoAP communication with the introduction of an upper limit on the permitted rate of outgoing messages. By enabling an observe mechanism, servers can inform the interested parties about any changes in the state of resources. The standard CoAP CC specification doubles the retransmission timeout (RTO) at the expiration of the timeout. CoAP under-performs when it is found to be too conservative in small networks or too aggressive in large networks [19,20], instead of adapting to the network status information. Another extension of CoAP CC specification is CoCoA [21,22] which is currently being standardized by the IETF CWG [23], as an adaptive, advanced congestion control mechanism. CoCoA+ makes the system more sensitive to the network dynamics by incorporating a novel round-trip-time (RTT) estimation scheme which automatically sets the RTO value, along with a variable backoff factor (VBF) and aging mechanism for the transmission of CoAP messages [15]. It has been proven that CoCoA significantly improves the performance over CoAP in congested networks [20,24]. However, CoCoA is still deficient in choosing the exact RTO value during burst traffic because of the inability to determine the RTT of retransmitted request-response accurately [25], thereby resulting in unnecessary spurious retransmissions [26].

In this paper, we present a Particle Swarm optimization (PSO)-based approach to congestion control called psoCoCoA, that significantly improves the performance of the CoAP protocol by tuning the retransmission count to an optimal value, along with a varying max-age value, under different network conditions. The Observe option is enabled in the CoAP protocol for max-age value. The optimal $MAX\_RETRANSMIT$ and max-age value are obtained by applying optimal parameter-driven simulations for different congestion levels, and employing the PSO-based technique, using the Cooja simulation environment, a toolset of the Contiki OS. To summarize, with psoCoCoA, we make the following novel contributions:

1. We propose a PSO-based CC approach for improving the baseline CoAP with Observe parameters which incorporates the random and optimal parameter-driven simulation based on different network congestion scenarios.
2. The PSO-based CC scheme applies the fitness measure obtained from previous scenarios to update the velocity for which the fitness function is computed for optimal number of retransmissions and max-age values.

3. The proposed PSO-based CC mechanism leverages on the $RTTVAR$ which is used to allocate the default values for CoAP timeout parameters in order to estimate the network congestion level, mitigate the network fluctuations, and to enhance the convergence for a stable network condition.

4. We evaluate and compare the performance of psoCoCoA at different traffic scenarios and max-age values with CoAP with Observe and CoCoA in terms of end-to-end delay, packet loss, and packet delivery ratio (PDR) in both random and grid network topology, respectively.

We have structured rest of the paper as follows: Section 2 discuss some related survey works about congestion control in constrained application protocol. In Section 3 we present the various congestion control mechanisms for evaluation and followed by the optimization-based congestion control approach in 3.4. Section 4 introduces the experimental setup and evaluation results. Finally, in Section 5 we set our conclusion and discuss future work on congestion control in a resource-constrained environment.

2 | RELATED SURVEY WORKS

Several works have been presented in Wireless Ad hoc and Sensor Networks for CC in constrained application protocol; recently CC mechanisms for CoAP are [27,28,29,30,31,32,33] and notable contributions on CC for reliable exchanges of CoAP communications include [34,35,36,21,20,24,22]. The Internet Engineering Task Force (IETF) RFC 7641 [18] provides an extension of the CoAP base specification for CC in unreliable CoAP communications as CoAP with Observe, which uses a publish-subscribe mechanism, with the introduction of an upper limit for the permitted rate of outgoing messages. These schemes play an important role in providing CC for constrained networks.

The experimental evaluation of CC for the CoAP protocol is presented in [37] for unreliable CoAP communication without end-to-end reliability between connected devices over emulated GPRS/UMTS links and also in a real IEEE 802.15.4 multi-hop testbed of constrained devices. The experimental results show that CoCoA performs better by maintaining high performance in almost all the scenarios considered, due to its flexibility when compared to default CoAP protocol, whereas, owing to the requirement of an adjustment period, the rate control applied by CoCoA may not perform optimally, a result which is reflected in a slightly lower packet delivery ratio (PDR) due to the varying condition of the network [24,38]. In the bid to improve on the CoCoA algorithm, the authors in [24], have proposed CoCoA 4-state-Strong as a modification to CoCoA that employed 4-state estimators in order to differentiate the wireless losses from the congestion losses. Their results have shown an improvement in the performance of maximizing throughput and the rate of packet loss at the same time. In [39], the authors propose a new rate-based CC for CoAP (CoAP-R) as an improvement to determine the performance of CoAP in bursty traffic environments. To control the network traffic, this mechanism monitors the sending rate of CoAP sources and uses a rate-based scheme instead of the usual window-based mechanisms. This scheme is aimed at achieving maximum bandwidth and network resource allocation based on fairness. It is observed that CoAP-R ensures uniform distribution of network resources amongst all the senders with reduced delay compared to CoAP and CoCoA. For instance, in [25], the authors proposed an enhancement of the CC mechanism for CoAP and CC/advanced which we called (E-CoCoA), that makes use of the RC of the request packet to estimate the actual RTT by matching the request-response packet to determine the RTO value. The results show that the proposed mechanism improves the efficiency of CC when compared to the base specifications of CoAP and CoCoA+.

A new mechanism called Fast-Slow RTO (FASOR) [40] based on retransmission timeout and CC mechanism for CoAP is introduced to address the problem of packet loss, which is either due to the wireless link environment or to congestion, by considering three unique features, including self-adaptive retransmission timer backoff, Slow RTO computation, and Fast RTO computation mechanisms. One promising advantage of the scheme is the fact that, unlike traditional CoAP and CoCoA mechanisms, the FASOR scheme can handle a high congestion level, even in the buffer-bloat environment, but suffers from high-latency cost due to the slow RTOs involved in the scheme.

A Precise CC Algorithm (pCoCoA) [41] for CoAP introduces the modified algorithm of CoCoA+ that overcomes the limitations in the existing algorithm. In pCoCoA, the use of a weak estimator is eliminated, and a transmission count is applied to match the ACK messages with CON messages, even during retransmission. This method also detects unnecessary retransmissions in the network by comparing the transmission counter value of the retransmission with the transmission counter value of the ACK message. The advantage of pCoCoA is the fact that it has reduced retransmissions and the ability to work in bursty traffic occurrence-based scenarios. However, considering the wide range of IoT scenarios, some of the fixed values used in this scheme might not be suitable. In [15], the authors present a context-aware CC (CACC) approach for lightweight CoAP/UDP-Based IoT Traffic which improves the modification of CoCoA+ [21] by utilising mechanisms that include an RTO estimator, RC-based smoothed RTT observation, a lower-bound RTO restriction approach, the aging concept and a 3-RTO estimator to identify the exact network status and provide adaptive CC. The results show that the resulting scheme called CACC, is effective in reducing the overall number of retransmissions, while guaranteeing higher throughput and minimised packet loss compared to those obtained with the base CoAP and CoCoA+ in all the network scenarios. However, to the best of the knowledge on optimal parameter tuning, this is the only work on CC for CoAP communications based on the PSO-algorithm.

A summary of some of the proposed CC mechanisms, with their unique characteristic features, is presented in Table 1

3 | CONGESTION CONTROL FOR COAP

In this section, we present a brief overview of the base CoAP CC specification, the Observe, and CoCoA mechanism which is followed by the proposed psoCoCoA.

TABLE 1 Summary of Congestion Control Algorithms with their unique features [15].

Scheme	RTO aging	Backoff method	RC	RTT estimators	Existing scheme
CoAP	No	BBF	No	No	None
CoCoA	Yes	VBF	No	Yes - 2	LinuxRTO
4-State Strong	Yes	VBF	No	Yes - 4	CoCoA
E-CoCoA	Yes	VBF	Yes	Yes - 1	CoCoA
CACC	Yes	VBF	Yes	Yes - 3	CoCoA

3.1 | Preliminaries of default CoAP CC specification

CoAP uses the Confirmable (CON), Non-confirmable (NON), Acknowledgement (ACK), and Reset (RST) messages for packet transmission. In reliable transmission, the ACKs is required to ensure reliability when selecting the CON packet type by the destination endnode within a retransmission timeout (RTO) interval. The CoAP ensures reliability by retransmitting the message if no ACK is received from the server before the RTO expires.

CoAP implements the primary CC by using an exponential backoff mechanism. Initially, the client sends a CON message to a server node. The initial value of RTO in CoAP is assigned randomly within the interval between ACK_TIMEOUT and ACK_RANDOM_FACTOR, where ACK_TIMEOUT and ACK_RANDOM_FACTOR represent the transmission parameters, and the default values of these factors are 2, and 1.5 s respectively. The sender node keeps on retransmitting the CON message until it reaches the maximum number of retransmissions MAX_RETRANSMIT and moreover the CoAP assigns MAX_RETRANSMIT value as 4. After four unsuccessful attempts of retransmission, CoAP enables the sender node to close the session. CoAP selects the initial RTO value randomly between ACK_TIMEOUT and (ACK_TIMEOUT x ACK_RANDOM_FACTOR).

Baseline CoAP CC specification enables the sender node to transmit the messages after having acknowledged the ACK message from the server. When congestion occurs and the ACK is not received, the RTO interval increases to double. The client node increases the RTO value and retransmits the CON message up to four times and attempts to find the correct RTO value. Another important parameter in CoAP for consideration is freshness in terms of *max-age* value. When a response is considered to be fresh, it is used in further processes without requesting the server again and this improves the communication efficiency of CoAP. The mechanism for determining the freshness in CoAP is the exploitation of the expiry time by using the *max-age* value. The *max-age* value indicates that the response is invalid after its age has expired. By default, CoAP assigns the *max-age* value as 60 s. When a *max-age* value remains the same when its age is greater than 60 s, it is considered to be invalid. The shortcoming of the base specification scheme in $CoAP$ is the fact that both the *max-age* value and $MAX\_RETRANSMIT$ are not set automatically. These parameters have to be assigned based on network congestion to improve the communication efficiency of CoAP. Table 2 presents a summary of default values chosen in the presence of congestion for CoAP delay and timeout parameters, and Figure 1 illustrates the CC mechanism used in base CoAP CC specification.

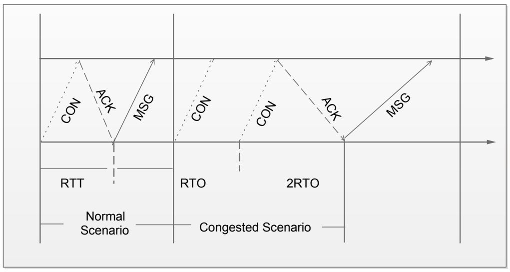


FIGURE 1 CoAP Congestion Control Mechanism

TABLE 2 Base values of CoAP delay and timeout parameters[6]

Parameter	Base value
ACK Timeout	2 s
ACK_RANDOM_FACTOR	1.5
MAX_RETRANSMIT	4
Processing delay	2 s
Data freshness	$\bar{T}_s = 60$ s
MAX_RTT	202
MAX_TRANSMIT_SPAN	45 s
DEFAULT_LEISURE	5 s

3.2 | Observe

The CoAP being a RESTful environment for resource-constrained nodes and networks, the state of a resource on its server is expected to change over time. A unique feature to the base CoAP specification is the addition of a very simple protocol extension that defines publish-subscribe mechanism that enables the CoAP clients to observe resources, i.e., to retrieve a representation of a resource by the server and ensure that it is updated over time[18]. With this feature, an observer (a client) can register itself to be permitted to access or follow some physical variable which is readily available at the IoT subject (server). On acceptance of the observer's registration request, the subject notifies the client whenever there are changes in the observed resource. To enable this process, the subject continues to update and maintain the list of all the registered observers over time in the system. For an observer to be able to follow or access multiple physical variables, such an observer is permitted to indicate its interest in registering with multiple subjects by extending the request-response model to a request/multiple-response model.

Data flows from the subject to client as a stream of notifications (normally transmitted as NONs) which the subject is expected to send to all observers currently on the list. In IoT-based networks with several and large subscribers, the Observe notification process can results to the transmission of voluminous messages: since one notification message is sent per subscriber and observed resource. These CoAP notifications are based on a valid lifetime which are defined by the max-age option and are cacheable. The Observe's conservative approach may not deliver optimal performance in scenarios with varying delays and varying degrees of congestion. To this effect, a dynamic approach to address this conservative rate control is proposed in CoCoA[21], which is summarized in the following subsection.

3.3 | CoCoA

The base CoAP specification is improved upon with an advanced CC for CONs and NONs messages called CoCoA. This mechanism uses round-trip-time (RTT) information based on CONs and its corresponding ACKs to maintain a RTO estimation for each destination endnode. CoCoA introduces mechanisms including adaptive RTO calculation, a variable backoff factor (VBF), and RTO aging mechanism[21].

CoCoA maintains averages of strong and weak RTT measurements, RTT_{strong} and RTT_{weak}, respectively, that are calculated as follows, when a new RTT measurement $RTT_{X_{new}}$ is performed:

$$RTTVAR_X = (1-\beta) \times RTTVAR_X + \beta \times |RTT_X - RTT_{X_{new}} \tag{1}$$

$$RTT_X = (1-\alpha) \times RTT_X + \alpha \times RTT_{X_{new}} \tag{2}$$

Where X represents strong or weak accordingly and using $\alpha = 0.25$ and $\beta = 0.125$. Subsequently, for each update of a RTT_X will lead to an update of RTO_X as

$$RTO_X = RTT_X + K \times RTTVAR_X \tag{3}$$

Where the default values for K_{strong} and K_{weak} are 4 and 1, respectively. To determine the overall RTO value which is maintained for a destination based on the update of either the strong or weak RTO estimators is determined using the $RTO_{overall}$.

$$RTO_{overall} = \lambda \times RTT_X + (1-\lambda) \times RTO_{overall} \tag{4}$$

where λ is 0.5 and 0.25 for strong and weak estimator update, respectively.

The $RTO_{overall}$ is then used in order to determine the initial RTO (RTO_{init}). CoCoA chooses the $RT0_{init}$ randomly from the interval [$RTO_{overall}$, $RTO_{overall} \times 1.5$]. At the point of timeout, contrary to the BEB used in base CoAP CC specification, CoCoA adopts the VBF in order to adjust the backoff value base on the $RT0_{init}$ of a transmission. When the value of $RT0_{init}$ is (<1 s) a backoff factor of 3 is set for retransmission and this reduces the chance of spurious retransmissions that may occur due to network congestion. On the other hand, when the $RT0_{init}$ is above 3 s, a backoff factor of 1.5 is set for retransmissions. Otherwise, when the value of $RT0_{init}$ is between 1 s and 3 s, a backoff factor of 2which corresponds to BEB of base CoAP CC specification is used for retransmission.

Finally, CoCoA utilises an aging scheme, known as RTO aging, which helps to prevent the use of outdated $RTO_{overall}$ estimates that may have become obsolete over time without any updates if no new RTT measurements are performed to the RTO estimators.

3.4 | psoCoCoA: the proposed mechanism

CoAP's main limitations rely in the variability of its performance, irrespective of its significant advantages, which is induced by the choice of its default parameters, including the number of retransmissions and max-age value. The inability to assign appropriate values to these parameters and their impact on the network's behaviour is still a difficult process for CC specification. The number of retransmissions is tunable without a clear default value, while the *max-age*, is chosen as 5 and 20 s. Based on this, the proposed system identifies an appropriate $MAX\_RETRANSMIT$ and obtains a considerable configuration better than the default parameters for the various IoT-congested scenarios by employing the PSO-based technique. In the random parameter-driven simulation, the existing CoAP CC specification is performed using default parameters, and the process is then repeated for different congestion scenarios by varying the retransmission and *max-age* values using the optimization approach. Furthermore, the system builds on low, medium, and high- traffic scenarios by varying the number of retransmissions and *max-age* value to analyse the impact of these parameters on base CoAP performance. The variation in these parameters is decided based on a PSO fitness and velocity function. An optimal value of the $MAX\_RETRANSMIT$ is decided on to use the best fitness value. The velocity variation is adaptive to the network congestion level. Thus, the proposed PSO-based CC assigns an optimal value to the number of retransmissions with a fixed set of *max-age* values, according to the network congestion level, and improves performance of the base CoAP CC specification in different IoT scenarios. Figure 2 illustrates the concept and components of the proposed psoCoCoA.

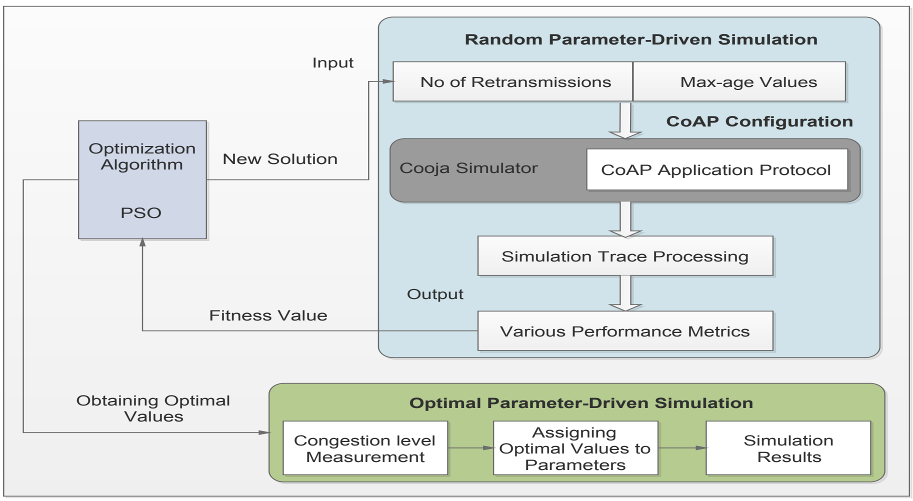


FIGURE 2 Proposed concept and components of psoCoCoA.

3.4.1 | System and network model

The standard CoAP model has two abstract layers. The upper layer implements the request-response communication for RESTful model, whereas the lower layer implements the control mechanism over CoAP message transmission through the underlying UDP protocol, which supports both reliable and unreliable modes in the network. In the Observe option, the NON-messages are mainly utilised during the transmission of notification messages. With the Observe, the server makes use of the *max-age* option to specify the exact expiry time for the CoAP response resource representation. The maximum message size is typically 128 bytes (block-wise) payload with a header size of 48 bytes. The base CC specification of the CoAP lacks in supporting unreliable exchange of messages and is insensitive to network conditions. In order to overcome this limitation, the Observe extension is introduced to support CC in unreliable data communication.

Particle swarm optimization is an optimization base technique designed and developed by[42], and inspired based on the social behaviour of bird flocking and fish schooling. The PSO technique is a very simple concept, and easy to implements with the adjustment of few parameters including the number of particles (solutions), range of solutions, learning factors, inertia weight, V_{max} (which represents the maximum change for a particle during one iteration), global and local best positions. This technique is computationally inexpensive, and by the use of only primitive mathematical operators, the process can be achieved[43]. The basic equations which are generally utilised by the PSO technique are velocity and position update equations to optimize the problem as given by Equation 5, and 6 respectively[44]. These equations are modified in each iteration of PSO algorithm to achieve optimum solution.

$$V = V + C_1 \times RND \times \left(P_{best} - PRS\right) + C_2 \times RND \times \left(G_{best} - PRS\right) \tag{5}$$

$$PRS = PRS + V \tag{6}$$

Where V, RND, PRS, and P_{best} and G_{best} is the particle velocity, random number between (0,1), the current particle (solution), the local best position and global best position respectively, and C_1, C_2 are the leaning factors.

The PSO probem presents possible solutions (particles) that randomly send requests to a server node in the network. In this scenario, there is only one server in the network to respond to requests of the connected CoAP nodes. The PSO technique learned from this, to apply and solve the optimization problem. For this process, each single particle is a CoAP node in the network. This we can also call a solution. All solutions are characterised with fitness values which can be assessed by the fitness function to be optimised, and have velocities which direct the path of the solutions. The solutions request through the network to obtain the current optimum solutions. In the proposed methodology, CoAP with Observe protocol is used to consider both reliable and unreliable CoAP communication. The PSO-based CC scheme considers the process as a random and an optimal parameter-driven simulation. In the random parameter-driven simulation, the first iteration is based on the primary CoAP CC specification with a default transmission parameter. With psoCoCoA, the fitness values in PSO rely on the performance metrics obtained at the end of the simulation results from the perspective of validation. The velocity reflects the distance travelled by this particle at each iteration. Each particle (solution) has to maintain its position P_{best}, known as the local best position and the G_{best}, known as the global best position among all the particles. In the initial iteration of PSO, the current fitness value is considered as P_{best}, and one is assigned as G_{best}. In the progressing iterations, the P_{best} and G_{best} are updated based on the fitness value in consideration of optimal values of performance metrics. The velocity is added with $MAX\_RETRANSMIT$ for optimal tuning along with adjustment of the *max-age* value. During the simulation with adjusted parameters, the resultant value of the metrics is taken as a fitness value for the next iteration. The inertia weight, w, is adjusted, based on the network traffic, which controls the momentum of particle by weighing the contribution of the previous velocity in controlling the new velocity. In the proposed CC mechanism, round-trip-time (RTT) value is employed to determine the congestion level. For further iterations, the optimal number of retransmissions is obtained for different congestion scenarios. In the optimal parameter-driven simulation, the optimal tuned $MAX\_RETRANSMIT$ with the adjusted *max-age* are used, based on the network traffic, to control the congestion in the network.

With the results of random parameter-driven simulations, the optimization approach determines the optimal values for CoAP parameters. Assigning the same value to various congestion scenarios is not valid. Therefore, the proposed PSO-based CC considers the IoT congestion network in three categories, namely, low, medium, and high-level congestion scenarios. For each category, the proposed psoCoCoA assigns the parameter values differently. The random parameter-driven simulation is a way of assigning the appropriate values to the parameters that regulate the performance of base CoAP CC specification. This leads to optimal configurations for this protocol, tailored for various IoT scenarios.

3.4.2 | Fitness measurement based-PSO algorithm and tuning inertia weight

The problem of data insufficiency at the server node during access by multiple clients simultaneously is a major cause of network congestion in resource-constrained networks. The PSO-based scheme creates several test scenarios to determine the network congestion levels. In the initial scenario, the default values are set for the CoAP parameters. The performance results realized from the first scenario are used to tune the default values of these parameters step by step, and continues this process for multiple times. The PSO technique applies the fitness measure of the results retrieved from the first scenario and updates the velocity to continue the process towards achieving an optimal number of retransmissions and max-age value. After the random parameter-driven simulation, it returns the simulation results of the packet delivery ratio PDR, the normalised overhead, and delay of the communication scenario. This information is then used to compute the fitness function F as follows:

$$F = W1 \times (PDR) + W2 \times (-NO) + W3 \times (-D) \times C \tag{7}$$

The objective of the PSO-based function is to maximize the packet delivery ratio and minimise both the packet loss and delay. For this reason, we formulate the system, such that the packet delivery ratio has a positive sign and the others have a negative sign. In Equation 7, factors $W1$, $W2$, and $W3$ represent the influence of each metric on the resulted fitness value. These metrics are assigned with 0.5, 0.3, and 0.2, respectively. By applying such weighting factors, the packet delivery ratio takes priority over other metrics. The negative polarity in the weighing factors is considered for influencing the importance of each performance metric in both individual and combined perspectives. The delay is also multiplied by the constant value ($C = 0.01$) to deal with a similar range of packet delivery ratios and normalised overheads. After the fitness for results of the random parameter-driven simulation has been measured, the velocity value for each parameter is estimated by using the improved PSO based on the level of network congestion.

$$W_\alpha = \frac{\{(W_{max} - W_{min}) - (F_{max} - F_{min})\}}{(F_{avg} - F_{min})} \tag{8}$$

$$V_j^{SOL+1} = WV_j^{SOL} + c_1 r_1 (P_{Best} - X_j^{SOL}) + c_2 r_2 (G_{Best} - X_j^{SOL}) \tag{9}$$

$$IW = \begin{cases} \dfrac{[(W_{min} + (W_\alpha)]}{2} \text{ if a,} \\ W_{min} + (W_\alpha) \text{ if b,} \\ W_{max} + (W_\alpha) \text{ if c.} \end{cases} \tag{10}$$

Where a, b, and c represent when congestion is low or normal, congestion is medium, and congestion is high respectively.

Here, the particles represent the possible values of several retransmissions and the max-age value. It is important to note that the PSO generates k solutions (SOL) and among them, the best solution is considered to be G_{Best}. The previous position of a particle acts as P_{Best} for the next solution. Moreover, X_j^{SOL} represents the position of a particle in a solution, and r_1 and r_2 are the random values, which are selected in the range of $[0, 1]$. The factors c_1 and c_2 represent the acceleration constant. The fitness represents the quality of a solution for every particle movement. The inertia weight IW factor W, is estimated by using Equation 8, 9 and 10, where W_{max} and W_{min} denote the maximum and minimum values of W, and F_{min} and F_{avg} denote the minimum and average fitness of particles in previous solutions. Each particle moves from the current position to the next one at the estimated velocity used in Equation 9. The inertia weight controls the velocity of a particle in the next solution. Notably, the inertia weights for the first and second particles are selected randomly, since the terms $(W_{max} - W_{min})$, $(F_{max} - F_{min})$, and $(F_{avg} - F_{min})$ are zero for the second solution. From the third solution, the inertia weight starts to update according to the fitness of its solutions and improves the efficiency of the PSO-based algorithm.

A PSO algorithm employs the inertia weight to mitigate the effect of the previous velocity on the current one. Significant inertia weight leads the particle to a global search, while the smaller factor guides the particle to a local search by using the previously estimated best solution. In Equation 10, the inertia weight is updated based on the level of network congestion. For low network congestion, the PSO-based CC scheme perform a local search only by increasing the inertia weight and velocity at a minimum level, compared to the medium and high congestion levels. The network with a high congestion level increases the inertia weight and the velocity of a particle significantly to perform a global search. Low inertia weight controls the velocity of a particle towards the solution. In the PSO-based algorithm, it is crucial to determine a proper termination criterion as this affects performance, solution quality and algorithmic efficiency. The termination of the algorithm is based on the closeness of the potential performance to the tuned retransmission count, provided that solution constraints are resolved. The number of retransmissions is controlled while tuning the parameter to avoid the possibility of getting a larger retransmission count, which in turn affects the performance efficiency of the system. Network congestion aware PSO-based factor tuning improves the efficiency of deciding on the base CoAP CC specification parameters.

3.5 | Network congestion level measurement

The random parameter-driven simulation controls the network traffic to create various congestion scenarios differently. During the simulation of the real-time environment, it is hard to predict how many sensors will be connected with the server at a time. This creates significant challenges for the detection of network congestion levels. The key idea of the optimization-based approach to estimate the level of network congestion is to utilise RTTVAR. The growth of RTT variance indicates the network state as a contention, and it leads to high packet loss and higher variability in communication delay. Therefore, the PSO-based approach leverages on the RTTVAR variable are proposed to define the characteristics of the network state, which are used in allocating the default values for the CoAP parameters, such as $MAX\_RETRANSMIT$ and $max\text{-}age$ value.

$$RTO_{x(y)} = \begin{cases} \dfrac{\left[(SRTT_x - \gamma \times RTTVAR_x)\right]}{SRTT_x} \\ \dfrac{\left[(SRTT_x + \gamma \times RTTVAR_x)\right]}{SRTT_x} \end{cases} \tag{11}$$

Where: γ value is 1 if ACK is received in the first transmission, otherwise γ value varies from 1 to 3 if ACK is not received in the first transmission.

In Equation 11, the $RTO_{x(y)}$ represents the confidence level of the network congestion, as it is a measure based on RTT variability. By using Equation 10, we identify three characteristic regions as follows:

Case 1 (Low congestion or normal scenario): $1 > RTO_{x(1)}$

In this state, the $RTO_{x(1)}$ denotes that there is no or less congestion in the network, as the negative RTTVAR leads to high $RTO_{x(1)}$. Consequently, the sending rate can be slightly increased.

Case 2 (Medium congestion scenario): $RTO_{x(1)} > 1 << RTO_{x(2)}$

This state represents the short-term RTT variability increment and this indicates the network congestion. Therefore, there is a need for transmitting messages in a controlled manner to avoid congestion.

Case 3 (High congestion scenario): $1 << RTO_{x(3)}$

In this state, the term $RTO_{x(3)}$ indicates the high congestion scenario, due to the huge variation in RTTVAR. Therefore, the CoAP has aggressively to reduce the sending rate and mitigate the impact of the congested network on message transmission.

3.5.1 | Tuning the optimal number of retransmissions with fixed max-age values

The proposed PSO-based CC scheme initially assigns a single sending rate to each message exchange and updates the sending rate on every ACK reception, based on the estimated level of network congestion. Specifically, the CoAP sender always maintains the maximum number of retransmissions and max-age value as 4 and 60 s, respectively. Since the proposed psoCoCoA is aimed at keeping the values of those parameters optimal, it employs the PSO-based optimal value search to achieve optimal maximum retransmission. The scheme uses max-age value 5 and 20 as these provide better performance with a tuned optimal maximum retransmission value. This approach is known to mitigate the effect of network condition fluctuations and to facilitate the convergence to stable network behaviour.

Algorithm 1 explains how the inertia weight factor in PSO is updated based on network congestion. By applying the algorithm, PSO-based parameters are measured. The solution with the highest fitness is considered to be the optimal value in each network category. The proposed psoCoCoA estimates the congestion level of the network and sets the parameters of CoAP optimally. As $MAX\_RETRANSMIT$ is freely adjusted, including the possibility of obtaining more significant value than the default, it requires further adjustments to the time values such as ACK_RANDOM_FACTOR and $ACK\_TIMEOUT$ to avoid unnecessary waiting time. These parameters are adjusted based on the optimal $MAX\_RETRANSMIT$ value achieved during the optimal parameter-driven simulation. Therefore, the PSO-based CC scheme significantly improves the efficiency and performance of the IoT network, compared to the existing CC schemes in various congestion network scenarios.

Algorithm 1 Network Congestion-Based Inertia Weight Update

Input: Measurement of Network Congestion Level

Output: Allocation of Optimal Values to Number of Retransmissions and $Max - age$ value

Function for optimal allocation ($SRTT_x$, $RTTVAR_x$)
Measurement of $RTO_{x(y)}$;
if $RTO_{x(1)} > 1$ **then**
  Low Network Congestion;
else
  if $RTO_{x(1)} > 1 << RTO_{x(2)}$ **then**
    Medium Network Congestion;
  else
    if $1 << RTO_{x(3)}$ **then**
      High Network Congestion;
    end if
    End function
    if (network congestion = Low) **then**
      Update the inertia weight using $(W_{min} + (W_\alpha))/2$;
      Measurement of Velocity and Fitness;
    else
      if (network congestion = Medium) **then**
        Update the inertia weight using $W_{min} + (W_\alpha)$;
        Measurement of Velocity and Fitness;
      else
        if (network congestion = High) **then**
          Update the inertia weight using $W_{max} + (W_\alpha)$;
          Measurement of Velocity and Fitness;
        end if

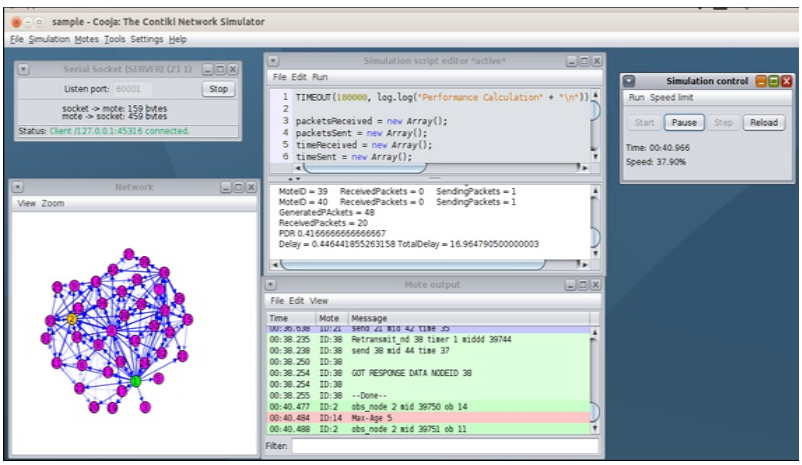
4 | EXPERIMENTAL SETUP AND EVALUATION

4.1 | Experimental setup

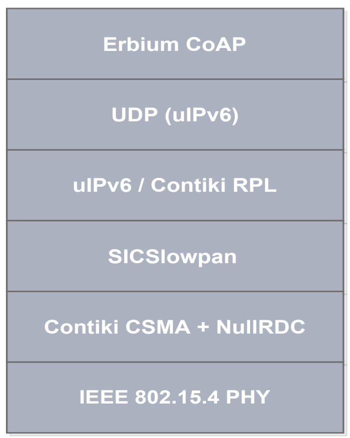
Cooja is a Java-based Contiki operating system (OS) simulation tool as an open source for both traditional wireless sensor networks and for the IoT, to connect tiny low-cost, and low-power micro-controllers to the Internet. This platform enables the execution of Contiki OS-based code and a simulation output that can be collected and processed, and obtain the performance metrics. With this platform, a Cooja inbuilt java-script simulator editor called gedit can be used to set the simulation time and also to calculate the performance metrics. This network simulator supports the simulation of sensor networks at three different levels which are: the machine code instruction set, the application level and the operating system[45]. This platform enables Cooja motes to emulate an off-the-shelf wireless sensor node that takes into consideration the capability functions of real nodes in the simulation environment. The Cooja simulation tool, including the script editor, the control panel and the serial socket (server) port connection plugin, is shown in Figure 3a.

The performance evaluation of the proposed psoCoCoA utilises the Cooja simulator of the Contiki-OS for constrained devices and the IoT to check the network performance of the proposed work. Figure 3b indicates the Contiki communication protocol stack used for evaluation. This platform permits the execution of Contiki OS-based code and enables Cooja motes to emulate an off-the-shelf wireless sensor node that takes into consideration the capability functions of real nodes in the simulation environment. The proposed work creates the sensor nodes and border router by using two of the IEEE 802.15.4 capable motes, Wismote and SkyMote, in a Cooja simulator, respectively. The simulation model of the proposed scheme consists of varying nodes in terms of 20, 40, and 50 with the transmission range of each node as $50m$. The network is constructed with $100m \times 100m$ area. For radio transmissions, the simulation exploits a Unit Disk Graph Medium (UDGM) radio model with circular transmission and applies interference areas. When applying the UDGM radio model, a static Link Delivery Ratio (LDR) setting is applied. The 802.15.4 MAC layer protocol and UDGM propagation model are used in the MAC and physical layer, respectively.

Moreover, RPL and UDP are used in the network and transport layer respectively. Different traffic loads are created by varying the packet intervals 4 s, 8 s, and 15 s for low, medium, and high traffic scenarios respectively. The total simulation time taken is 180 s. The proposed scheme shows the performance analysis for the variation of nodes in terms of 20, 40, and 50 for the three traffic scenarios. The traffic scenarios are varied based on the number of packets generated per second, and for each scenario, their respective tuning factors such as the number of retransmissions and fixed *max-age* values of 5 and 20 are considered to achievev the performance of the proposed scheme. In the core CoAP with Observe extension, the default values of transmission parameters such as $MAX\_RETRANSMIT$, $ACK\_TIMEOUT$, and ACK_RANDOM_FACTOR are 4, 2, and 1.5 s respectively. The default *max-age* value is 60. The three traffic scenarios with the tuned parameters deliver improved maximum outputs in terms of packet delivery ratio, delay and packet loss. The simulation deploys randomly distributed clients in two different test scenarios, such as random topology and grid topology. The hardware specifications with their unique capability features, and details of the Cooja simulation parameter setup are presented in Table 3, and Table 4, respectively.



(a) Contiki Cooja simulation environment showing the random network topology. 1 and 2 represent RPL border router and sink node respectively, while all other motes in the network represent nodes for running the full Contiki OS stack and the Erbium implementation of CoAP.



(b) Showing different layers of the Contiki protocol stack.

TABLE 3 Hardware Simulation Parameters for the WisMote and Moteiv Tmote Sky Wireless Sensor Nodes

Features	Wismote	Tmote Sky
RAM	16 KB	10 KB
ROM	256 KB	48 KB
MCU	MSP430	MSP430F2617
RADIO	CC2520	CC2420

TABLE 4 Summary of Cooja Simulation Setup Specifications

Parameter	Value Specification
Congestion mechanisms	Base CoAP with Observe, psoCoCoA
Wireless channel model	Unit Disk Graph model, TR = $50m$
Transport and network	UDP + uIPv6 + 6LoWPAN
Routing protocol	Routing Protocol for LLN (RPL)
Radio duty cycling (RDC)	Null-RDC
Media access control	(CSMA/CA)
$Max - age$ value	5, 20
Network area	$100m \times 100m$
Radio band	$2.4GHz$
Physical	$IEEE$ 802.15.4 PHY
Simulation time	180 s
Traffic loads	4, 8, and 15 s
Cooja simulation speed	200 %

4.2 | Performance metrics

A set of performance metrics are considered in order to measure the performance of PSO-based CC approach with existing CC schemes. The packet delivery ratio (PDR), the packet loss, and the end-to-end delay are used to evaluate performance for the different congestion control approaches.

4.2.1 | Packet delivery ratio (PDR)

The overall system PDR indicates the reliability of the network and remains a critical QoS metric which is generally applicable in a resource-constrained networks. The PDR is calculated as the ratio of the number of successfully received data packets at the destination node to the total number of data packets generated by the clients nodes during an experiment. The PDR in percentage can be calculated as shown in Equation (12).

$$PDR\,(\%) = \frac{\sum packets\ received}{\sum packets\ sent} \times 100 \tag{12}$$

4.2.2 | End-to-end delay

End-to-end delay is the average time that taken by data packets from being sent by the CoAP client nodes to being successfully received by the destination node during an experiment. The end-to-end delay is an important indicator to signify the behaviour of the different congestion control schemes. This is a very critical QoS metric that is considered in a resource-constrained IoT networks. The end-to-end packet delay can be expressed by using Equation (13).

$$Delay\ = \frac{\sum (Time\ received - Time\ sent)}{\sum packets\ sent} \tag{13}$$

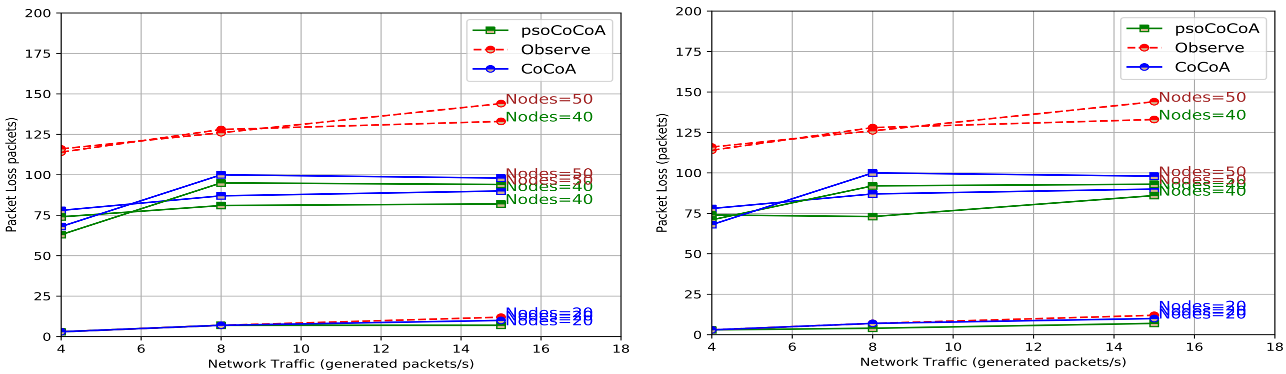
4.2.3 | Packet loss

The packet loss indicates the total number of undelivered data packets to the destination end node.

4.3 | Evaluation results

In this section we organize the validation of psoCoCoA against CoAP with Observe, and CoCoA for different network traffic scenarios as low (L), medium (M), and high (H) in 2 subsections to evaluate the performance results for the different schemes to congestion control in a random, and grid network topology respectively. The set-up includes two 180 s simulations running at 200 % speed by varying the *max-age* values of 5 and 20, with tuned retransmissions, each with 20, 40, and 50 connected nodes, to analyse the performance to the different schemes to congestion control. The simulation set generates results necessary to evaluate the performance in terms of packet loss (packets), end-to-end delay (s), in the 20, 40, 50-node, and PDR % measurement for 20 and 50-node based on the network traffic scenarios for random and grid topology, respectively.

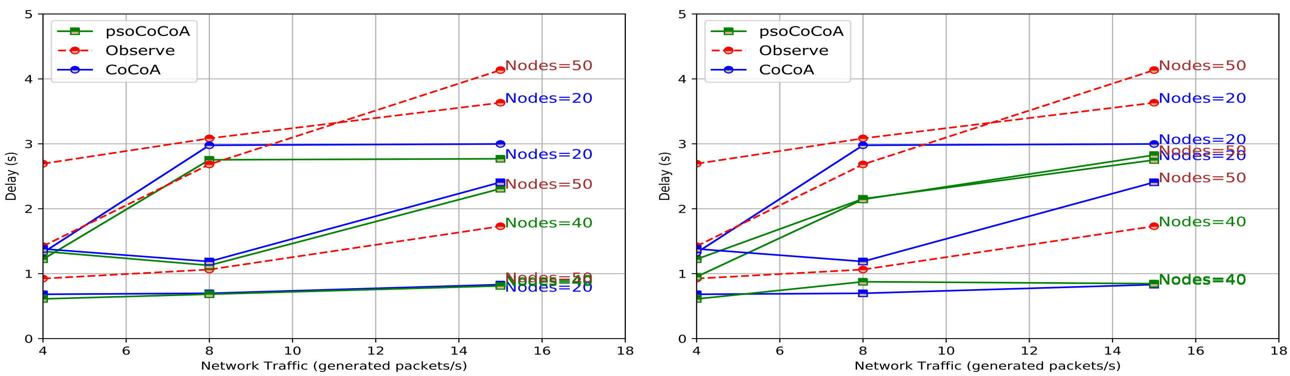
The results obtained for the different performance metrics are presented in Figures 4, 5, 6, and 7 for random-topology as discussed in subsection 4.3.1, and Figures 8, 9, 10, and 11 for grid-topology as discussed in subsection 4.3.2.



(a) Packet loss measured with max-age 5 for different traffic scenario

(b) Packet loss measured with max-age 20 for different traffic scenario

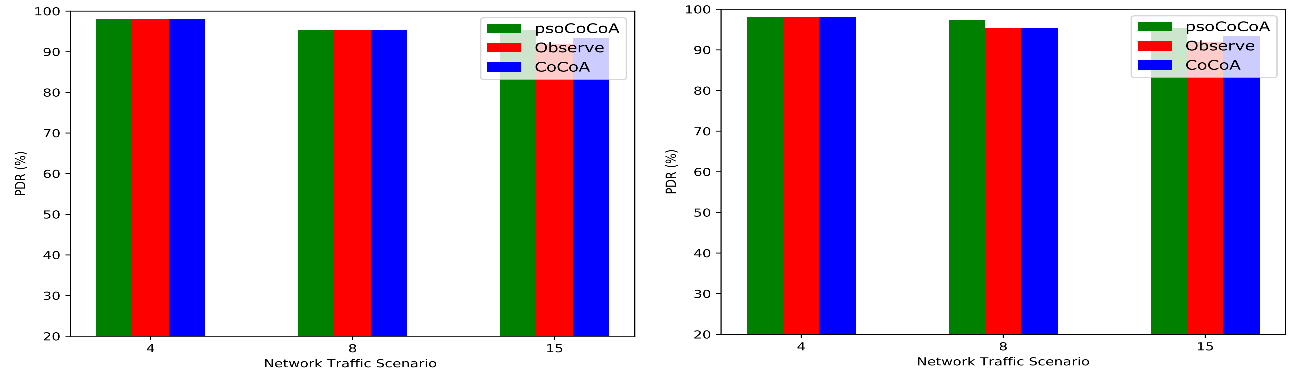
FIGURE 4 Packet loss measurement based on different network traffic scenarios for random topology .



(a) Delay measured with max-age 5 for different traffic scenario

(b) Delay measured with max-age 20 for different traffic scenario

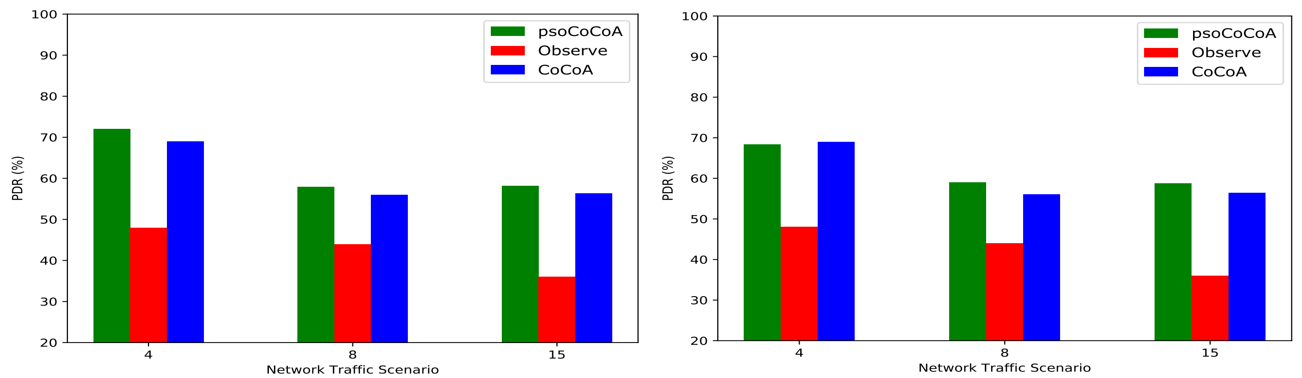
FIGURE 5 Delay measurement based on different network traffic scenarios for random topology.



(a) PDR measured with max-age 5 for different traffic scenario.

(b) PDR measured with max-age 20 for different traffic scenario.

FIGURE 6 PDR measurement for different network traffic scenarios in the 20-node random topology.



(a) PDR measured with max-age 5 for different traffic scenario.

(b) PDR measured with max-age 20 for different traffic scenario.

FIGURE 7 PDR measurement for different network traffic scenarios in the 50-node random topology

	Observe			psoCoCoA			CoCoA		
Nodes	L (%)	M (%)	H (%)	L (%)	M (%)	H (%)	L (%)	M (%)	H (%)
20	98	95.3	92	98	95.3	95.3	98	95.3	93.3
40	43	36	33.5	63	59.5	59	61	56.5	55
50	48	44	36	72	57.7	58	69.7	55.5	56

TABLE 5 PDR measured for different traffic scenarios with max-age 5 in random topology.

	Observe			psoCoCoA			CoCoA		
Nodes	L (%)	M (%)	H (%)	L (%)	M (%)	H (%)	L (%)	M (%)	H (%)
20	98	95.3	92	98	97.3	95.3	98	95.3	93.3
40	43	36	33.5	63	63.5	57	61	56.5	55
50	48	44	36	68.4	59	58.7	69.7	55.5	56

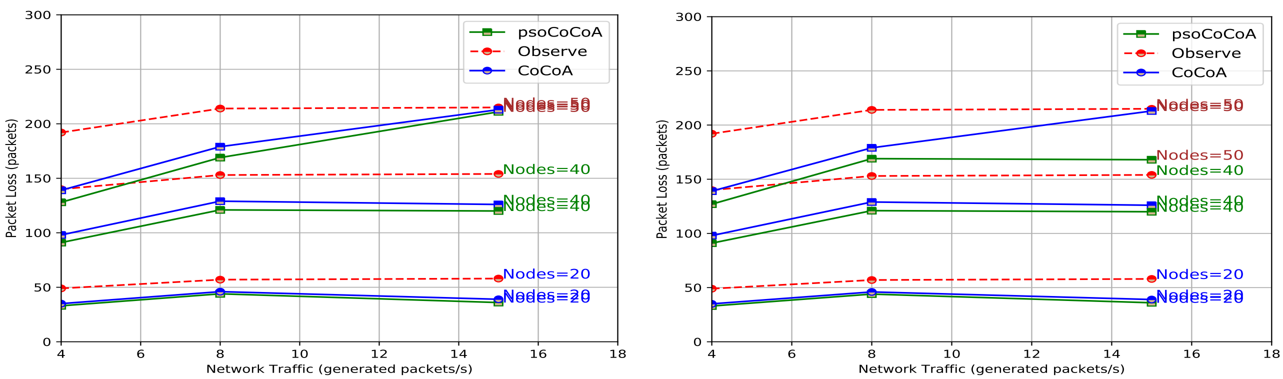
TABLE 6 PDR measured for different traffic scenarios with max-age 20 in random topology.

4.3.1 | Results for random network topology

The results obtained show the improvement of successful transmission of packets with different traffic scenarios and max-age for different number of connected nodes, taking CoAP with Observe as reference. The optimization-based congestion control approach considers the RTT and RTO values during packet retransmission in order to determine the congestion levels, which helps in the selection of an optimal retransmission count with efficient resource utilisation. Whereas CoAP with Observe lacks in adjusting transmission parameters based on the network congestion levels, thereby resulting in high packet loss in all traffic scenarios. The psoCoCoA and CoCoA show a notable improvement in terms of packet loss, compared to CoAP with Observe, as the connected nodes are increased from 20, 40, and 50 by setting the optimal retransmission count with a *max-age* value of 5 and 20 for different traffic scenarios, as depicted in Figure 4a and 4b respectively. In a 20-node with different traffic scenarios, the performance of all three CC schemes looks similar. However, when the number of connected nodes increases to 40 and 50 for different traffic scenarios with max-age values, psoCoCoA and CoCoA significantly perform better than CoAP with Observe. For a maximum number of nodes considered, the packet loss in the psoCoCoA is less for 45 %, 24 %, and 34 % for the L, M and H congestion scenarios compared to CoAP with Observe.

Similar results are obtained for the end-to-end delay with different traffic scenarios and max-age values for 20, 40, and 50-nodes, as can be seen in Figure 5. For the end-to-end delay with max-age values of 5 and 20 as depicted in Figure 5a and 5b respectively. The psoCoCoA and CoCoA clearly perform better with a minimised delay compared to CoAP with Observe in all traffic scenarios. This shows a significant reduction in the end-to-end delay to 20 % and 32 % for medium and high-traffic scenarios respectively, even with the maximum number of connected nodes. In baseline CoAP with Observe, the $MAX\_RETRANSMIT$ is fixed at 4, irrespective of the network traffic scenarios, which leads to high packet loss and end-to-end delay. However, the optimization-based congestion control considers the results obtained for each varied number of retransmissions to choose the optimal $MAX\_RETRANSMIT$, based on each network condition, and this significantly improves on the performance of the congestion control scheme.

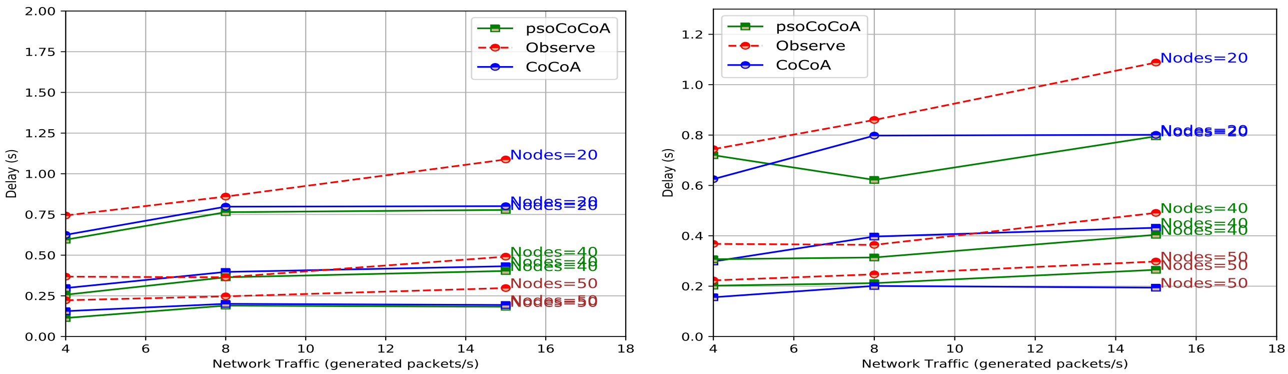
Figure 6, and 7 show the improvement of the PDR measurements with different traffic scenarios and max-age values for 20 and 50-connected nodes, respectively. For the different traffic scenarios, the PDR performance of all three CC schemes is very similar as depicted in Figure 6. However, as the network traffic scenarios increases with max-age values, psoCoCoA shows an improvement in PDR or very similar to CoCoA as shown in the 20-node random topology. Figure 7 shows the improvement of the PDR measurements during the experiments with different traffic scenarios in the 50-node random topology. Tables 5, and 6 show a summary of the PDR values generated with max-age values of 5 and 20, respectively, for varying traffic loads.



(a) Packet loss measured with max-age 5 for different traffic scenario

(b) Packet loss measured with max-age 20 for different traffic scenario

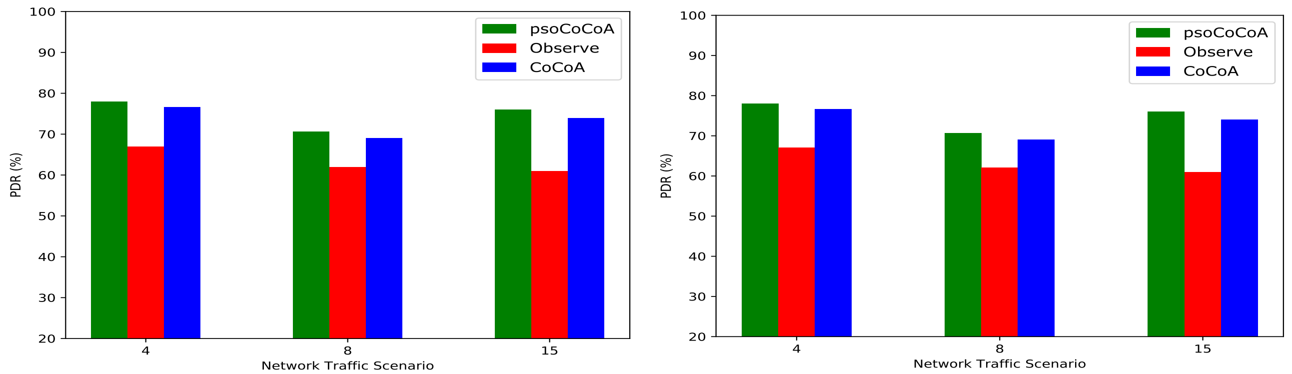
FIGURE 8 Packet loss measurement based on different network traffic scenarios for grid topology.



(a) Delay measured with max-age 5 for different traffic scenario

(b) Delay measured with max-age 20 for different traffic scenario

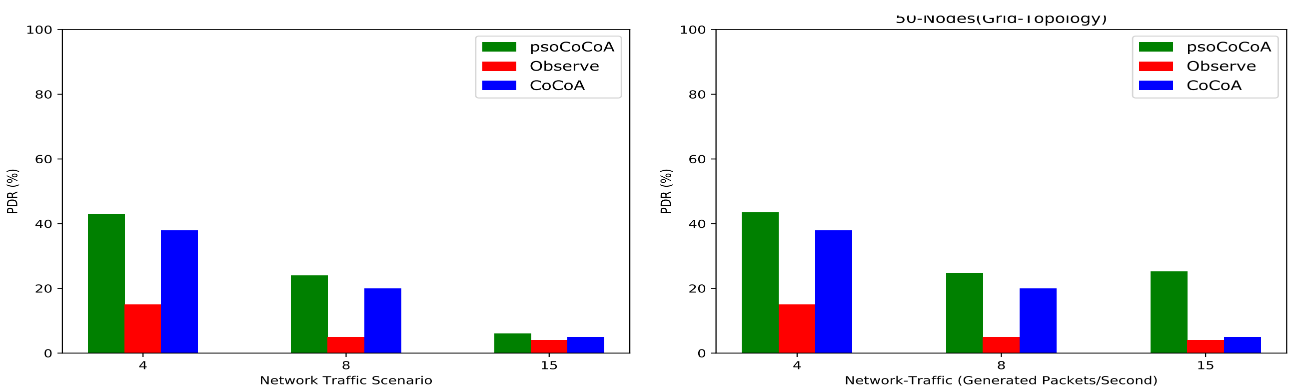
FIGURE 9 Delay measurement based on different network traffic scenarios for grid topology.



(a) PDR measured with max-age 5 for different traffic scenario.

(b) PDR measured with max-age 20 for different traffic scenario.

FIGURE 10 PDR measurement for different network traffic scenarios in the 20-node grid topology.



(a) PDR measured with max-age 5 for different traffic scenario.

(b) PDR measured with max-age 20 for different traffic scenario.

FIGURE 11 PDR measurement for different network traffic scenarios in the 50-node grid topology.

Nodes	Observe L (%)	Observe M (%)	Observe H (%)	psoCoCoA L (%)	psoCoCoA M (%)	psoCoCoA H (%)	CoCoA L (%)	CoCoA M (%)	CoCoA H (%)
20	67	62	61	78	70.7	76	76.7	69.3	74
40	30	23.5	23	54.5	39.5	40	51	35.5	37
50	14.6	4.9	4.4	43	24	6	38	20.4	5.3

TABLE 7 PDR measured for different traffic scenarios with max-age 5 in grid topology.

Nodes	Observe L (%)	Observe M (%)	Observe H (%)	psoCoCoA L (%)	psoCoCoA M (%)	psoCoCoA H (%)	CoCoA L (%)	CoCoA M (%)	CoCoA H (%)
20	67	62	61	78	70.7	76	76.7	69.3	74
40	30	23.5	23	54.5	39.5	40	51	35.5	37
50	14.6	4.9	4.4	43.5	24.8	25.3	38	20.4	5.3

TABLE 8 PDR measured for different traffic scenarios with max-age 20 in grid topology.

4.3.2 | Results for grid network topology

In a resource-constrained network, packet loss remains one of the critical performance metrics, most especially for applications that require immediate reactions, short notification periods, and data integrity. The results obtained show that the optimization-based congestion control improves the performance of CoAP with Observe. The performance analysis shows a significant improvement for packet loss in the proposed psoCoCoA and CoCoA, compared to CoAP with Observe, as the nodes are increased from 20, 40, to 50 by setting the optimal retransmission count with max-age values for different traffic scenarios, as depicted in Figure 8a, and 8b respectively. For instance, in a 20-connected node, the psoCoCoA and CoCoA experience packet losses in low, medium and high traffic scenarios as 33, 44, 36 and 35, 46, 39 packets respectively, whereas, in the CoAP with Observe there are 49, 57, 58 packet loss respectively, in a similar network scenario. It is obvious that the significant decrease in the packet loss realized in the PSO-based and CoCoA congestion control mechanisms is due to the avoidance of both steep increment and excessive shrunk of the RTO, by utilizing the field of retransmission count.

The psoCoCoA and CoCoA congestion control show a very similar significant performance in the end-to-end delay even as the nodes are increased from 20, 40 to 50 in different network traffic scenarios as shown in Figure 9. In Figure 9a, psoCoCoA clearly outperform better than CoAP with Observe, most especially at small packet intervals, where the congested network environment starts to build up and affect the protocol performance. For instance, in a 20-node topology, the end-to-end delay experienced between the psoCoCoA and CoAP with Observe congestion control approaches are 0.14 s, 0.09 s, and 0.31 s for the different traffic scenarios, respectively. The fixed parameter values in CoAP with Observe and insensitivity to network conditions lead to longer delays and packet loss. However, the consideration of the congestion level and application of optimal transmission parameters provide efficient and better packet delivery with minimized end-to-end delay as observed in psoCoCoA, which is better and very similar in performance with CoCoA congestion control mechanism.

Figure 10, and 11 show significant improvement in the PDR measurements with different traffic scenarios and max-age values for 20 and 50-connected nodes, respectively. For the different traffic scenarios in the 20-node topology, the PDR performance of psoCoCoA congestion control approach indicates an improvement or very similar to that of CoCoA as shown in Figure 10 for the max-age values of 5 and 20, respectively. It is obvious that the proposed psoCoCoA and CoCoA congestion control approaches perform significantly better than CoAP with Observe as can be seen in Figure 10 for the different traffic scenarios. Figure 11 shows the PDR performance measurement in the 50-node topology, and with the psoCoCaA significantly performing better than CoCoA and CoAP with Observe in all traffic scenarios. Tables 7, and 8 present a summary of the values with *max-age* values of 5 and 20, respectively. The use of optimal transmission parameters by the PSO-based congestion control approach helps to improve the default CoAP parameters, improves the delivery of packet loss with reduced delay, and minimises the computational overhead, thereby improving the efficiency of the overall system performance.

5 | CONCLUSION AND FUTURE WORK

Congestion control remains a critical issue that demands continuous research for reliable communication in constrained networks for efficient resource utilisation and for optimal network performance. In this paper an optimization-based solution has been proposed for congestion control that significantly improves the reliability of base CoAP protocol at various congestion levels with a varying number of connected nodes. The proposed psoCoCoA utilises PSO-based solution for transmission parameters such as $MAX\_RETRANSMIT$ with varying $max-age$ values by applying random and optimal parameter-driven simulation. The optimization-based congestion control determines the optimal number of retransmissions for different traffic scenarios, and maintains relatively high quality of service, even when the network condition fluctuates.

The performance validation of psoCoCoA shows significant improvement or almost similar in packet loss, end-to-end delay, and PDR with CoCoA in all traffic scenarios. In addition, the adaptive nature of psoCoCoA and CoCoA clearly perform better or show improvements over the results obtained with CoAP with Observe in almost all traffic scenarios. Possible future work would be further enhancement of the performance of congestion control mechanisms to develop a system framework for service provisioning as a three-state system base on queuing analysis and an intelligent machine-learning-based congestion control approach to Internet of Things traffic that will improve the performance of certain Internet communication components for an efficient user-friendly service experience.

6 | ACKNOWLEDGMENTS

We sincerely acknowledge the author, August Betzler, for making available the Erbium implementation of CoCoA.

References

1. Paul A, Jeyaraj R. Internet of Things: A Primer. *Human Behavior and Emerging Technologies* 2019; 1(1): 37–47.

2. Akpakwu GA, Silva BJ, Hancke GP, Abu-Mahfouz AM. A survey on 5G networks for the Internet of Things: Communication technologies and challenges. *IEEE Access* 2017; 6: 3619–3647.

3. Atzori L, Iera A, Morabito G. The internet of things: A survey. *Computer networks* 2010; 54(15): 2787–2805.

4. Siow E, Tiropanis T, Hall W. Analytics for the Internet of Things: A survey. *ACM Computing Surveys (CSUR)* 2018; 51(4): 74.

5. Colitti W, Steenhaut K, De Caro N, Buta B, Dobrota V. Evaluation of constrained application protocol for wireless sensor networks. In: Proceedings of the IEEE 18th Workshop on Local & Metropolitan Area Networks (LANMAN). ; 2011: 1–6.

6. Shelby Z, Hartke K, Bormann C. The constrained application protocol (CoAP). 2014.

7. Bormann C, Castellani AP, Shelby Z. CoAP: Application protocol for billions of tiny internet nodes. *IEEE Internet Computing* 2012(2): 62–67.

8. Kushalnagar N, Montenegro G, Schumacher C, others . IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals. *(RFC4919 Information), Internet Engineering Task Force* 2007.

9. Ouakasse F, Rakrak S. A Comparative Study of MQTT and CoAP Application Layer Protocols via. Performances Evaluation. *Journal of Engineering and Applied Sciences* 2018; 13(15): 6053–6061.

10. Karagiannis V, Chatzimisios P, Vazquez-Gallego F, Alonso-Zarate J. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud computing* 2015; 3(1): 11–17.

11. Misic J, Ali MZ, Misic VB. Protocol architectures for IoT domains. *IEEE Network* 2018; 32(4): 81–87.

12. Kovatsch M, Duquennoy S, Dunkels A. A low-power CoAP for Contiki. In: Proceedings of the IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems. ; 2011: 855–860.

13. Borgia E. The Internet of Things vision: Key features, applications and open issues. *Computer Communications* 2014; 54(12): 1–31.

14. Shang W, Yu Y, Droms R, Zhang L. Challenges in IoT networking via TCP/IP architecture. *Technical Report NDN-0038. NDN Project* 2016.

15. Akpakwu GA, Hancke GP, Abu-Mahfouz AM. CACC: Context-Aware Congestion Control Approach for Lightweight CoAP/UDP-Based Internet of Things Traffic. *Transactions on Emerging Telecommunications Technologies* 2019(12): 1–19.

16. Verma S, Kawamoto Y, Fadlullah ZM, Nishiyama H, Kato N. A survey on network methodologies for real-time analytics of massive IoT data and open research issues. *IEEE Communications Surveys & Tutorials* 2017; 19(3): 1457–1477.

17. Kiani F. A Survey on Management Frameworks and Open Challenges in IoT. *Wireless Communications and Mobile Computing* 2018; 2018.

18. Hartke K. RFC 7641: Observing Resources in the Constrained Application Protocol (CoAP). *Internet Engineering Task Force (IETF)* 2015.

19. Pötsch T, Kuladinithi K, Becker M, Trenkamp P, Goerg C. Performance evaluation of CoAP using RPL and LPL in TinyOS. In: Proceedings of the IEEE 5th International Conference on New Technologies, Mobility and Security (NTMS). ; 2012: 1–5.

20. Järvinen I, Daniel L, Kojo M. Experimental evaluation of alternative congestion control algorithms for Constrained Application Protocol (CoAP). In: Proceedings of the IEEE 2nd World Forum on Internet of Things (WF-IoT). ; 2015: 453–458.

21. Betzler A, Gomez C, Demirkol I, Paradells J. CoCoA+: An advanced congestion control mechanism for CoAP. *Ad Hoc Networks* 2015; 33: 126–139.

22. Betzler A, Gomez C, Demirkol I, Paradells J. CoAP congestion control for the internet of things. *IEEE Communications Magazine* 2016; 54(7): 154–160.

23. Bormann C, Betzler A, Gomez C, Demirkol I. CoAP simple congestion control/advanced: draft-ietf-core-cocoa-01. 2017.

24. Bhalerao R, Subramanian SS, Pasquale J. An analysis and improvement of congestion control in the CoAP Internet-of-Things protocol. In: Proceedings of the IEEE 13th Annual Consumer Communications & Networking Conference (CCNC). ; 2016: 889–894.

25. Lee JJ, Kim KT, Youn HY. Enhancement of congestion control of Constrained Application Protocol/Congestion Control/Advanced for Internet of Things environment. *International Journal of Distributed Sensor Networks* 2016; 12(11): 1–13.

26. Bolettieri S, Vallati C, Tanganelli G, Mingozzi E. Highlighting some shortcomings of the CoCoA+ congestion control algorithm. In: Proceedings of the International Conference on Ad-Hoc Networks and Wireless. ; 2017: 213–220.

27. Seo J, Lee S, Khan MTR, Kim D. A new CoAP congestion control scheme considering strong and weak RTT for IoUT. In: ; 2020: 2158–2162.

28. Yadav RK, Singh N, Piyush P. Genetic CoCoA++: genetic algorithm based congestion control in CoAP. In: IEEE. ; 2020: 808–813.

29. Bansal S, Kumar D. Distance-based congestion control mechanism for CoAP in IoT. *IET Communications* 2020; 14(19): 3512–3520.

30. Vallati C, Righetti F, Tanganelli G, Mingozzi E, Anastasi G. Analysis of the interplay between RPL and the congestion control strategies for CoAP. *Ad Hoc Networks* 2020; 109: 102290.

31. Quwaider M, Shatnawi Y. Neural network model as Internet of Things congestion control using PID controller and immune-hill-climbing algorithm. *Simulation modelling practice and theory* 2020; 101: 102022.

32. Quwaider M, Shatnawi Y. Congestion control model for securing internet of things data flow. *Ad Hoc Networks* 2020; 106: 102160.

33. Deshmukh S, Raisinghani VT. AdCoCoA-Adaptive Congestion Control Algorithm for CoAP. In: IEEE. ; 2020: 1–7.

34. Eggert L. Congestion Control for the Constrained Application Protocol (CoAP). *draft-eggert-core-congestion-control-01 (work in progress)* 2011.

35. Betzler A, Gomez C, Demirkol I, Paradells J. Congestion control in reliable CoAP communication. In: Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems. ; 2013: 365–372.

36. Betzler A, Gomez C, Demirkol I, Kovatsch M. Congestion control for CoAP cloud services. In: Proceedings of the IEEE Emerging Technology and Factory Automation (ETFA). ; 2014: 1–6.

37. Betzler A, Isern J, Gomez C, Demirkol I, Paradells J. Experimental evaluation of congestion control for CoAP communications without end-to-end reliability. *Ad Hoc Networks* 2016; 52: 183–194.

38. Ancillotti E, Bruno R. Comparison of CoAP and CoCoA+ congestion control mechanisms for different IoT application scenarios. In: 2017 IEEE Symposium on Computers and Communications (ISCC). ; 2017: 1186–1192.

39. Ancillotti E, Bolettieri S, Bruno R. RTT-Based Congestion Control for the Internet of Things. In: International Conference on Wired/Wireless Internet Communication. ; 2018: 3–15.

40. Jarvinen I, Raitahila I, Cao Z, Kojo M. FASOR Retransmission timeout and congestion control mechanism for CoAP. In: Proceedings of the IEEE Global Communications Conference (GLOBECOM). ; 2018: 1–7.

41. Bolettieri S, Tanganelli G, Vallati C, Mingozzi E. pCoCoA: A precise congestion control algorithm for CoAP. *Ad Hoc Networks* 2018; 80: 116–129.

42. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the IEEE 95th International Conference on Neural Networks (ICNN). ; 1995: 1942–1948.

43. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science (MHS's). ; 1995: 39–43.

44. Hu X. PSO Tutorial. *Internet: http://www. swarmintelligence. org/tutorials. php, [Mar. 24, 2013].*

45. Osterlind F, Dunkels A, Eriksson J, Finne N, Voigt T. Cross-level sensor network simulation with cooja. In: Proceedings of the 31st IEEE Conference on Local Computer Networks. ; 2006: 641–648.

46. Bahashwan AAO, Manickam S. A Brief Review of Messaging Protocol Standards for Internet of Things (IoT). *Journal of Cyber Security and Mobility* 2019; 8(1): 1–14.

47. Sehgal A, Perelman V, Kuryla S, Schonwalder J. Management of resource constrained devices in the internet of things. *IEEE Communications Magazine* 2012; 50(12): 144–149.

48. Demir AK, Abut F. mlCoCoA: a machine learning-based congestion control for CoAP.. *Turkish Journal of electrical engineering & computer sciences* 2020; 28(5).

AUTHOR BIOGRAPHY

Godfrey A. Akpakwu received his B.Eng. degree (Hons.) in electronic engineering from the University of Pretoria, Pretoria, South Africa, in 2016, B.Eng. degree (Hons.) in electrical and electronic engineering from the Federal University of Agriculture at Makurdi, Nigeria, in 2008, and MSc. degree in electronic communications and computer engineering from the University of Nottingham, Malaysia, in 2011. He graduated with a PhD degree in electronic engineering from the University of Pretoria, Pretoria, South Africa, in 2020.

He is a Lecturer with the Federal University of Agriculture at Makurdi, Benue State, Nigeria. Dr. Akpakwu's research interests include a wide range of Internet of Things (IoT) and Nano-IoT smart technologies, fifth-generation mobile networks and beyond, wireless sensor networks, network management, and network security.

Gerhard P. Hancke received the B.Eng. and M. Eng. degrees from the University of Pretoria, Pretoria, South Africa, in 2002 and 2003, respectively and the Ph.D. in computer science with the Security Group at the niversity of Cambridge's Computer Laboratory, Cambridge, U.K. in 2008.

He is an Assistant Professor with the City University of Hong Kong, Hong Kong SAR. Dr. Hancke's research interests include system security, embedded platforms, and distributed sensing applications.

Adnan M. Abu-Mahfouz (M'12-SM'17) received his MEng and PhD degrees in computer engineering from the University of Pretoria, Pretoria, South Africa.

He is currently the Centre Manager of the Emerging Digital Technologies for 4IR (EDT4IR) research centre at the Council for Scientific and Industrial Research (CSIR), Extraordinary Professor at University of Pretoria, Professor Extraordinaire at Tshwane University of Technology and Visiting Professor at University of Johannesburg. His research interests are wireless sensor and actuator network, low power wide area networks, software-defined wireless sensor network, cognitive radio, network security, network management, sensor/actuator node development.

He is an associate editor at IEEE Access, IEEE Internet of Things, and the IEEE Transactions on Industrial Informatics, Senior Member of the IEEE and Member of many IEEE Technical Communities. He participated in the formulation of many large and multidisciplinary R&D successful proposals (as Principal Investigator or main author/contributor). Abu-Mahfouz is the founder of the Smart Networks collaboration initiative that aims to develop efficient and secure networks for the future smart systems, such as smart cities, smart grid and smart water grid