Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 7, 1187-1204
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i7.8874

Markov model-based congestion control for the internet of things

Godfrey A. Akpakwu[1]*, Topside E. Mathonsi[2]
[1,2]Tshwane University of Technology, South Africa; AkpakwuG@tut.ac.za (G.A.A.) mathonsiTE@tut.ac.za (T.E.M.).

Abstract: Internet of Things (IoT) devices use the lightweight RESTful Constrained Application Protocol (CoAP). Resource-constrained IoT networks need efficient congestion control (CC) for reliable communication. Radio channel capacity and device hardware limitations can cause congestion. CoAP, which operates on top of UDP, must independently manage CC. Simple CoAP rules handle congestion but do not react to changing network conditions. IoT applications demonstrate considerable resource limitations that present new issues for the design of CC techniques. Current CC mechanisms, such as Default CoAP and CC with advanced mechanisms (CoCoA), exhibit restricted adaptation to fluctuating traffic conditions. This research presents a three-state Markov model for CC that utilizes probabilistic state transitions to adapt dynamically to network conditions and manage varying congestion levels effectively. The proposed Markov model is validated against Default CoAP and CoCoA using Contiki OS and the Cooja simulator. A comparative performance analysis of the Markov model has shown 199.237 kbps throughput, 22.174% packet loss, and a delay of 126.603 ms compared to the Default CoAP and CoCoA mechanisms at higher transmission intervals. These findings suggest that Markov-based adaptive techniques could improve CC in constrained networks, enabling more reliable IoT communications.

Keywords: *Congestion control, Internet of things, constrained application protocol, Markov model, Resource-constrained networks.*

1. Introduction

THE Internet of Things (IoT), an advanced technology, promises to revolutionize the global world through the interconnection of embedded systems and smart devices. These devices are low-power in nature and communicate over the Internet [1]. IoT applications include smart and remote medical systems, smart cities, smart industries and factories, smart grids and meters, remote asset monitoring and control, smart agriculture, emergency reporting services, vehicle fleet management systems, and many more. These platforms depend on these smart devices and have limited hardware and computing power because they work in an environment with few resources [1-3].

Recent advancements in mobile communication have introduced cellular protocols like Long-Term Evolution (LTE), but they are not well suited for low-power and low-data-rate devices such as IoT sensors. Several new IoT standards have emerged to address this issue. The next generation of mobile networks and beyond is designed to solve the problems of

older cellular standards and be a key part of future IoT developments [4]. The IoT includes many different technologies that work with Wireless Sensor Networks (WSNs) to send and receive data. These technologies work at different levels of communication, such as the physical, network, transport, and application layers. A thorough analysis reveals that a wireless sensor network consists of four fundamental components: A group of sensor and actuator nodes is linked to a group of gateways that connect to a server or broker that many clients can use to obtain send information [5].

The rapid expansion of IoT has generated considerable challenges in network connectivity, especially in resourceconstrained environments. These network devices differ due to their limited

History: Received: 9 May 2025; Revised: 18 June 2025; Accepted: 22 June 2025; Published: 15 July 2025
* Correspondence: AkpakwuG@tut.ac.za

memory and processing capabilities, reduced radio bandwidth, and elevated bit error rates. All of these factors significantly affect congestion in constrained networks [6]. Establishing protocols and standards is essential for devices with limited hardware and computational capabilities to fulfil IoT requirements. The Internet Engineering Task Force (IETF) has developed various communication protocols for the IoT that operate on low-power and lossy networks. The IETF Core Working Group (CWG) standardised the Constrained Application Protocol (CoAP) [7] making it easier for IoT networks to share information. In WSNs, sensors and actuators are often placed in tricky spots where multipath fading causes a lot of bursty packet loss on wireless channels [8]. Because these devices don't have a lot of power, memory, or computing power, protocols like Message Queue Telemetry Transport (MQTT), Extensive Messaging and Presence Protocol (EMPP), Advanced Message Queuing Protocol (AMQP), and Constrained Application Protocol (CoAP) need to handle large amounts of data quickly and efficiently.

The subscription/notification paradigm serves as the foundation for MQTT, a lightweight transport protocol. It is characterised by its compact size, low power consumption, and effective dissemination of information to multiple recipients. Because it relies on the Transport Control Protocol (TCP), its overall application latency cannot work in places where packets are lost a lot due to the need for retransmissions [9]. Through protocol extensions, the IoT has modified EMPP, a communication protocol for message exchange. Because MQTT needs TCP for transport and Extensible Markup Language (EML) for message encoding, it has some limitations that make it ineffective in some situations [9]. For IoT applications, AMQP is a message exchange protocol that offers both reliability and security. Because AMQP depends on TCP for transmission, packet loss has a great effect on it and causes a lot of latency when conditions are not ideal [10].

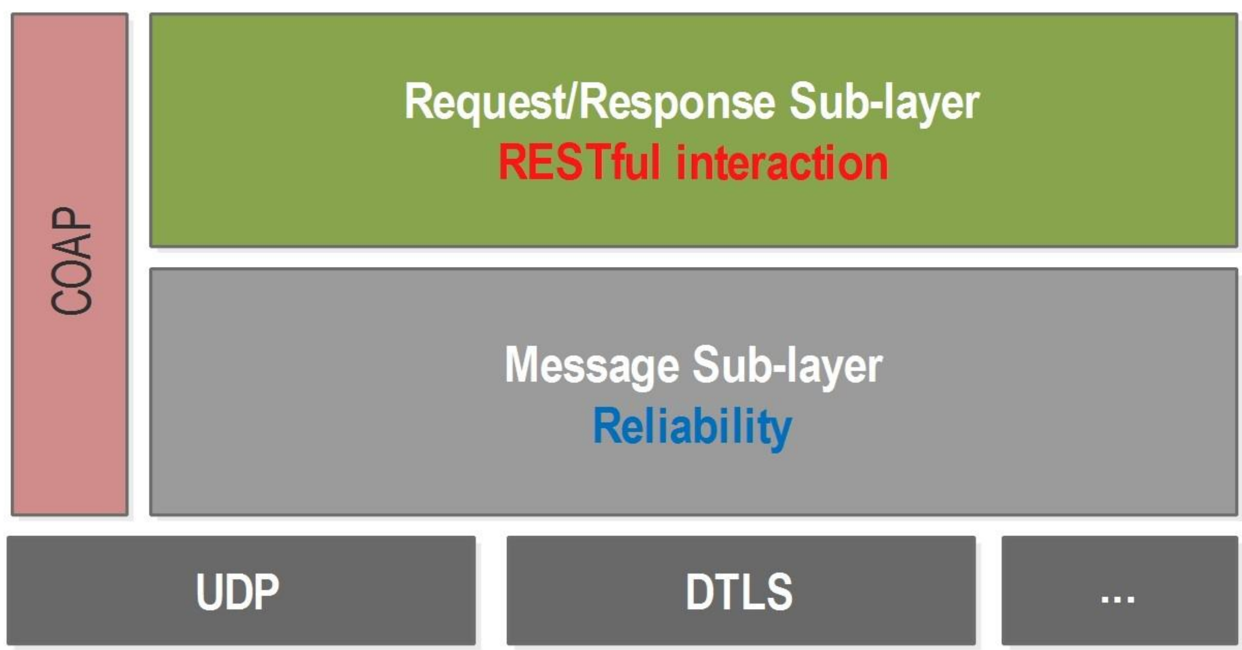

Figure 1.
CoAP Sub-layers.
Source: Akpakwu, et al. [11].

For restricted nodes and networks, COAP is a specialised web transfer protocol. Facilitates a request and response framework between endpoints and enables the inherent discovery of services and resources. CoAP is designed to work smoothly with Hypertext Transfer Protocol (HTTP) for web integration. It also meets certain needs, such as the need for multicasting and low complexity and overhead in some situations [7, 12].

CoAP is expected to meet the needs of low-power and low-loss networks (LLN) and allow global Internet services to use light protocols [13, 14]. CoAP does not have a builtin end-to-end congestion control (CC) mechanism, so it uses a different CC method from the start. The resolution of a CC system is still inadequate, especially as the traffic load on the IoT network approaches its limits. The techniques employed by TCP in the majority of conventional Internet applications allow for end-to-end congestion control. However, due to the dynamic nature of LLNs, conventional Internet protocols are ineffective in resource-constrained environments [15, 16]. Moreover, CoAP operates independently of TCP; it uses the User Datagram Protocol (UDP) for lightweight applications and autonomous congestion control strategies. The conventional CoAP consists of two abstract layers. In the RESTful paradigm, the upper layer facilitates the transmission of requests and responses, while the bottom layer governs the transmission of CoAP messages via the UDP protocol [17] as shown in Fig. 1 [11].

The main cause of IoT congestion is burst traffic, which happens when a lot of devices are connected and networks do not have enough resources to handle large amounts of data being sent quickly. Therefore, it is crucial to manage congestion to maintain network stability and provide optimal quality of experience (QoE), which in turn ensures superior quality of service (QoS). The objective of the plan is to ensure that the essential management rules are relevant in the appropriate stages and levels of detail. In IoT networks, traffic moves differently than in traditional networks, and congestion is a key factor that lows data transfer. Consequently, it is imperative to implement CC methods within IoT to ensure real-time QoS [18].

The CoAP is a lightweight protocol that provides efficiency for IoT devices with restricted computational and energy resources. However, congestion in IoT networks continues to pose a substantial barrier, impacting performance measures such as packet delivery, throughput, and latency. Current methods for controlling traffic jams, such as as Default CoAP and Congestion Control with Advanced Mechanisms (CoCoA), are not very flexible when traffic conditions change.

This study introduces and evaluates a Markov model-based congestion control strategy designed to predict and proactively respond to network congestion. By modelling network conditions through discrete states and using probabilistic transitions informed by observed metrics, the Markov model enables earlier and more precise adjustments to transmission behaviour.

The remainder of the paper is organised, as follows: Section II includes a review of similar studies; Section III gives information about the benchmark protocols and the analytical model of the proposed three-state Markov model. Section IV discusses the simulation setup and the performance metrics used for the evaluation. Section V presents the simulation and discussion results. Section VI concludes and presents possible suggestions for future work on CoAP CC for constrained networks.

2. Related Work

Several works have been conducted in ad-hoc wireless and sensor networks for CC. However, in our work, we base this on the analytical model for CC mechanisms in CoAP.

The authors in Chen and Kunz [19] looked at latency and loss in a WSN based on CoAP while taking into account multi-hop routing. The study assesses the efficacy of a stack comprising CoAP and 6LoWPAN over an IEEE 802.15.4 radio connection utilising the Contiki OS and Cooja simulator in conjunction with the CoAP framework Californium (Cf) while evaluating the quality of service parameters, including throughput, endto-end delay, and packet loss. In Collina, et al. [20] the authors assess latency, loss and throughput in a severely damaged wireless network and compare the performance of CoAP and MQTT. This study shows that the protocols work well, with great speed and reliability, making them a strong choice for medical IoT applications and beyond. Betzler, et al. [21] looked at how well CoAP and MQTT handled congestion by measuring delay, loss, and throughput under a range of network conditions. This work looks at how well the chosen IoT protocols work under different amounts of traffic, the likelihood of losing packets, and delays to find the best protocol for the application's needs.

In Akpakwu, et al. [22] the authors introduced CoAP Simple Congestion

Control/Advanced (CoCoA), a sophisticated CC technique for CoAP that is being standardised by the IETF CoRE working group. In addition to a variable backoff factor (VBF) and aging processes, CoCoA offers new ways to estimate RTT. This lets you make dynamic and regulated retransmission timeout (RTO) changes, which is perfect for IoT communications. They did an analysis of the performance of CoCoA and a number of other algorithms, including the most advanced mechanisms made for TCP. Experiments conducted in real testbeds form the basis of the research. A testbed made up of nodes running the IETF's IPv6-based IoT protocol stack over IEEE 802.15.4 is used for the first set of experiments. The second series of studies based on GPRS, a prevalent M2M solution for IoT. Their findings show that, compared to the other methods they looked at, CoCoA always outperforms the default CoAP CC in all of the situations they tested it in Bhalerao, et al. [23]. Likewise, Herrero [24] examines the constraints of CoCoA and suggests enhancements that are assessed through performance evaluation. This study looks at CoCoA and, based on what they found, the authors suggest and create CoCoA 4-state-Strong, which uses a 4-state estimator for variable backoffs to greatly improve throughput, even in very unreliable networks, while maintaining a higher goodput. In Hartke [25] the authors present a two-state analytical model that connects how well applications perform with loss of network packets and helps to estimate how lossy wireless channels affect CoAP situations. This study examines a mathematical model to analyse the effects of the transport mechanisms employed by CoAP for the transfer of sensor data. This model calculates the expected loss and delays of application packets, which are then compared to the actual values obtained through a set validation process.

The aforementioned work emphasises performance and a two-state Markov model applicable to real-time estimation of latency and loss in applications. We propose a three-state Markov model for congestion control that includes an intermediate stage to more efficiently regulate varying congestion levels.

3. Coap Congestion Control Mechanisms

This section delineates the CC algorithms used in our evaluations. Alongside Base CoAP [7] we carry out evaluations using CoCoA [6].

3.1. CoAP Traffic and Base CC Specification

CoAP is a Representational State Transfer (RESTful) protocol that facilitates resource manipulation on servers using the operations GET, PUT, POST, and DELETE. The objective of CoAP is to serve as a lightweight substitute for HTTP. Wireless networks with limited memory, processing capabilities, and radio infrastructure were the target audience for the creation of CoAP.

CoAP functions in two main modes, employing UDP for reliable data transmission: (a) a confirmable (CON) transmission scheme (as illustrated in Fig. 2(a)), in which a packet is considered delivered upon receipt of an acknowledgment (ACK) from the remote endpoint within the retransmission timeout (RTO). If the server does not provide an ACK before the RTO expires, CoAP ensures reliability by retransmitting the message. When using the non-confirmable (NON) transmission scheme (as illustrated in Fig. 2(b)), packets with sensor data, such as temperature and humidity readings, are sent one way without waiting for an ACK, there is no guarantee of successful delivery. According to RFC 7641 [27], this scenario follows the CoAP extension for resource observation. In this case, a single subscription forces the server to continue to send NON-sensor data.

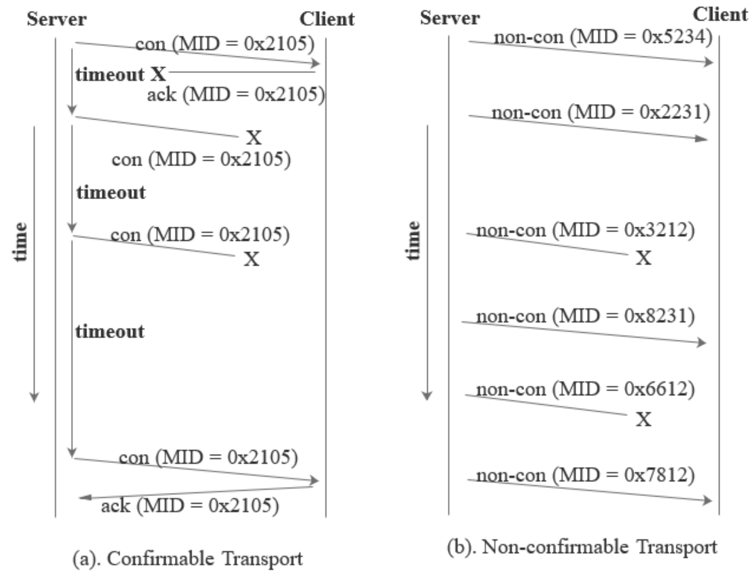

Figure 2.
CoAP Traffic Modes.
Source: Hartke [25].

Congestion frequently occurs in networks with limited devices due to the inadequate processing power of wireless nodes and the slow speeds of radio links. Congestion leads to two primary issues: packet collisions and saturated packet buffers, both of which result in packet loss and increased delays. Loss of confirmed CoAP messages necessitates retransmissions, leading to increased traffic and potential exacerbation of congestion. To address network congestion resulting from frequent packet retransmissions, CoAP employs a basic congestion control method, outlined as follows:

When transmitting a CON message, CoAP randomly chooses an initial RTO value ranging from 2 to 3 *s* for the initial message transmission. If the message transmission initiator who sent the message doesn't get an ACK from the destination endpoint by the end of the timer set for this RTO value, a loss is assumed and the CoAP message has to be sent again. CoAP reduces network congestion by doubling the retransmission timeout (RTO) value of the resent packet. This is done with a binary exponential backoff (BEB) strategy. This CC technique applies to all CON CoAP communications, irrespective of the target endpoint. The NSTART parameter in the CoAP base specification controls the maximum number of exchanges that can happen at the same time with a certain destination endpoint. The specification advises configuring NSTART to 1, which is adequate for the majority of CoAP applications.

The foundational specification of CoAP delineates three essential components that constitute the CoAP CC mechanism:

- The initial transmission of a CON CoAP message uses the RTO computation.
- Before retransmitting a CON CoAP message, the RTO applies the backoff mechanism.
- The state information concerning the destinations of CON COAP messages is retained.

3.2. CoCoA

CoCoA is an adaptive technique for round-trip time (RTO) estimation, similar to the RTO estimation mechanism used in TCP [26]. CoCoA employs two RTO estimators that utilise TCP RTO estimations: a strong estimator that relies on ACKs from the original transmission and a weak estimator that relies on ACKs from retransmissions.

CoCoA computes the averages of the strong and weak RTT measurements, RTT_{strong} and RTT_{weak}, respectively, which are determined as follows when a new RTT measurement (RTT_{Xnew}) is executed by using Equations13 and 2:

$$RTTVAR_X = (1 - \beta) \times RTTVAR_X + \beta \times |RTT_X - RTT_{Xnew} \quad (1)$$

$$RTT_X = (1 - \alpha) \times RTT_X + \alpha \times RTT_{Xnew} \quad (2)$$

X denotes strong or weak correspondingly, utilising $\alpha = 0.25$ and $\beta = 0.125$.
As a result, each update of RTT_X will lead to a corresponding update of RTO_X as given by Equation 3

$$RTO_X = RTT_X + K \times RTTVAR_X \quad (3)$$

The default values for K_{strong} and K_{weak} are 4 and 1, respectively. The overall RTO value for a destination is determined by updates from either the strong or weak RTO estimators, calculated as $RTO_{overall}$ by Equation 4.

$$RTOoverall = \lambda \times RTTX + (1 - \lambda) \times RTOoverall \quad (4)$$

where λ equals 0.5 for a strong estimator update and 0.25
for a weak estimator update.

The general $RTO_{overall}$ is used to determine the initial RTO (RTO_{init}). CoCoA randomly selects $RT0_{init}$ from the range
[$RTO_{overall}$, $RTO_{overall}$ ×1.5]. At timeout, CoCoA uses the variable backoff factor (VBF) to modify the backoff value based on $RT0_{init}$ of a transmission, in contrast to the BEB employed in the base CoAP CC specification. When $RT0_{init}$ is less than 1 *s*, a 3 backoff factor is set for retransmission. This reduces the chance of unwanted retransmissions that could happen because the network is busy. In contrast, when $RT0_{init}$ exceeds 3 *s*, a backoff factor of 1.5 is established for retransmissions. If the value of $RT0_{init}$ is between 1 *s* and 3 *s*, a backoff factor of 2, corresponding to the BEB of the base CoAP CC specification, is used for retransmission.

The integration of aging RTO aligns the RTO value with the default RTO value of 2 *s* over extended intervals without changes in the RTO. If the RTO exceeds 3 *s* and remains constant for a duration of 4×*RTO*, the new RTO is determined using the formula *RTO* = 1+0.5×*RTO*. The approach sets the new RTO to double the existing RTO if it is less than 1 *s* and is constant for a duration of 16 × *RTO*.

3.3. Three-State Markov Model for Congestion Control

This section provides a detailed description of formulating the proposed CC as a Markov process with three congestion states. The three-state Markov model consists of a number of separate parameters that define its network behaviour in CC.

1) Congestion States of the Model: Figure 3 illustrates the proposed CC approach, which is based on a three-state Markov model. The network model represents the system behaviour through the Normal, Medium, and High congestion states, respectively. While the channel is in the Normal state, there is a successful transmission of packets with fewer delays than when the channel is in the High congestion state with a high probability of loss.

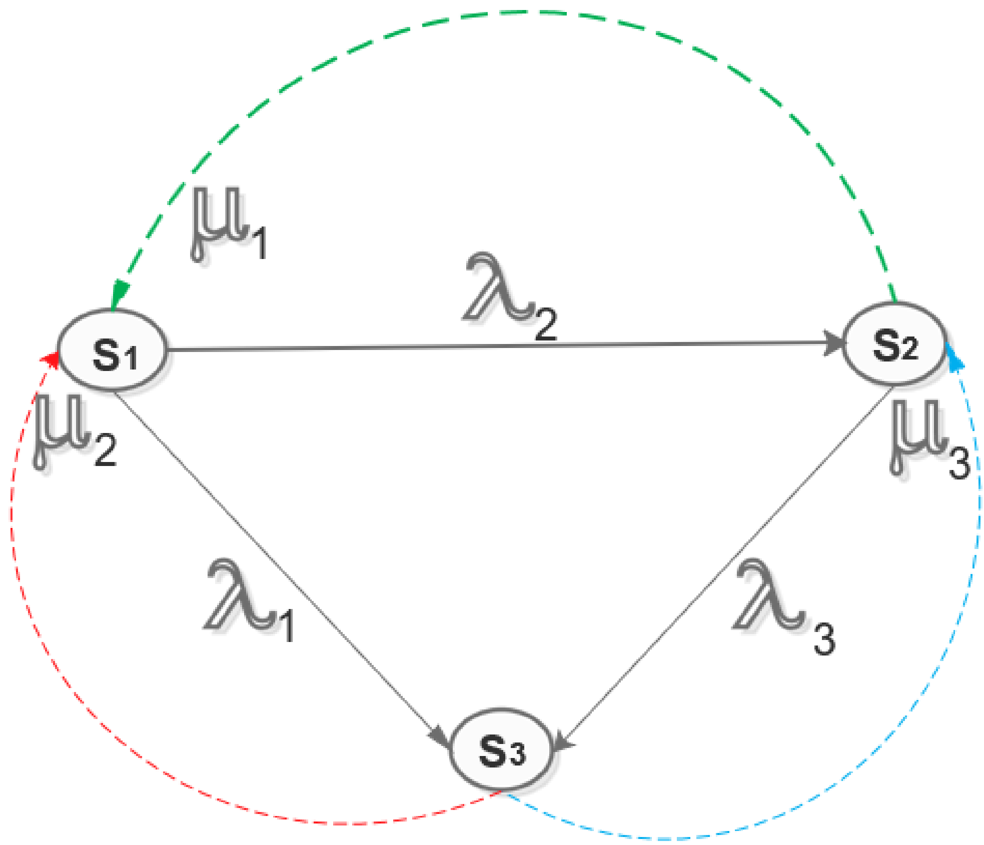

Figure 3.
A three-state Markov model for congestion control in CoAP for IoT-based networks.

Where S_1, S_2, and S_3 represent the congestion states, respectively [23].

The three-state Markov model represents the network's behaviour during congestion based on the following states, which are characterised below with the following parameters:

- State 1 (S_1): Low Congestion - This state represents a situation where the network is not congested and there is sufficient communication capacity. The system operates normally, with little to no need for congestion control measures. The network is operating normally, with minimal latency and successful packet delivery.
- State 2 (S_2): Moderate congestion - In this state, the network is experiencing moderate congestion, and some form of congestion control (e.g. packet drop, rate limiting) might be needed to maintain network performance. The network encounters moderate congestion, resulting in increased queueing delays and sporadic packet loss; however, it remains manageable.
- State 3 (S_3): High Congestion - In this state, the network is highly congested, leading to significant delays, packet losses, and potential timeouts. Aggressive congestion control measures are required to avoid further degradation. The network is experiencing extreme congestion, leading to considerable packet loss and prolonged queueing delays.

2) State Transition Probabilities: The transitions between these states are regulated by a Markov process, in which the likelihood of transitioning from one state to another is based solely on the present state. We delineate the transition probabilities as follows:

where P_{ij} denotes the likelihood of transitioning from state S_i to S_j, with the constraint (see Equation 5):

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 7: 1187-1204, 2025
DOI: 10.55214/25768484.v9i7.8874

$$\sum_{j=1}^{3} P_{ij} = 1, \quad \forall i \in [1,2,3] \tag{5}$$

Where:
P_{11} : Probability of remaining in S_1 (Low State)
P_{12} : Probability of transitioning from S_1 to S_2
P_{13} : Probability of transitioning from S_1 to S_3
P_{21} : Probability of transitioning from S_2 to S_1
P_{22} : Probability of remaining in S_2 (Moderate Congestion)
P_{23} : Probability of transitioning from S_2 to S_3
P_{31} : Probability of transitioning from S_3 to S_1
P_{32} : Probability of transitioning from S_3 to S_2
P_{33} : Probability of remaining in S_3 (High Congestion).
The transition matrix P is given by Equation 6:

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix} \tag{6}$$

where each row sums to 1.

3) Steady-State Probabilities: The steady-state probabilities π_1, π_2, and π_3 represent the long-term probability that the network will be in states S_1, S_2 and S_3, respectively. We calculate these probabilities by solving the system of linear equations.

$$\begin{aligned} \pi 1 &= \pi 1 P11 + \pi 2 P21 + \pi 3 P31, \\ \pi 2 &= \pi 1 P12 + \pi 2 P22 + \pi 3 P32, \\ \pi 3 &= \pi 1 P13 + \pi 2 P23 + \pi 3 P33, \\ \pi_1 &+ \pi_2 + \pi_3 = 1. \end{aligned} \tag{7}$$

Equation 7 can be represented in matrix form as follows in Equation 8:

$$\begin{bmatrix} P_{11} - 1 & P_{21} & P_{31} \\ P_{12} & P_{22} - 1 & P_{32} \\ P_{13} & P_{23} & P_{33} - 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{8}$$

This system of linear equations can be solved using Gaussian elimination or numerical methods.

4. Evaluation Setup

This section provides details on the simulation setup that we used to analyze performance evaluations of the three CC mechanisms presented. This procedure includes the simulator settings, the network topologies considered for the traffic patterns, and the various performance indicators that were used to validate the performance evaluations.

4.1. Simulation Setup

Designed for traditional WSN and IoT, Cooja [27] is an open source simulation tool for Java-based Contiki OS [28] used to emulate devices with limited network resources and perform performance measurements. A fundamental characteristic of Cooja is its simulation of the hardware of commercially available wireless sensor nodes, which considers the hardware specifications and processing capabilities of the actual nodes. You can upload the compiled binary image file of a node into the simulated nodes, then execute the compiled program code using the selected node type's emulation model during the simulations, just like a real node would.

This platform utilizes an integrated JavaScript simulator editor, gedit, to configure simulation time and compute performance metrics. This network simulator facilitates the simulation of sensor networks at three distinct levels: the machine code instruction set, the application level, and the operating system [29]. From left to right, Fig. 4 compares the IETF communication protocol stack and the Contiki implementation platform used for evaluation [11].

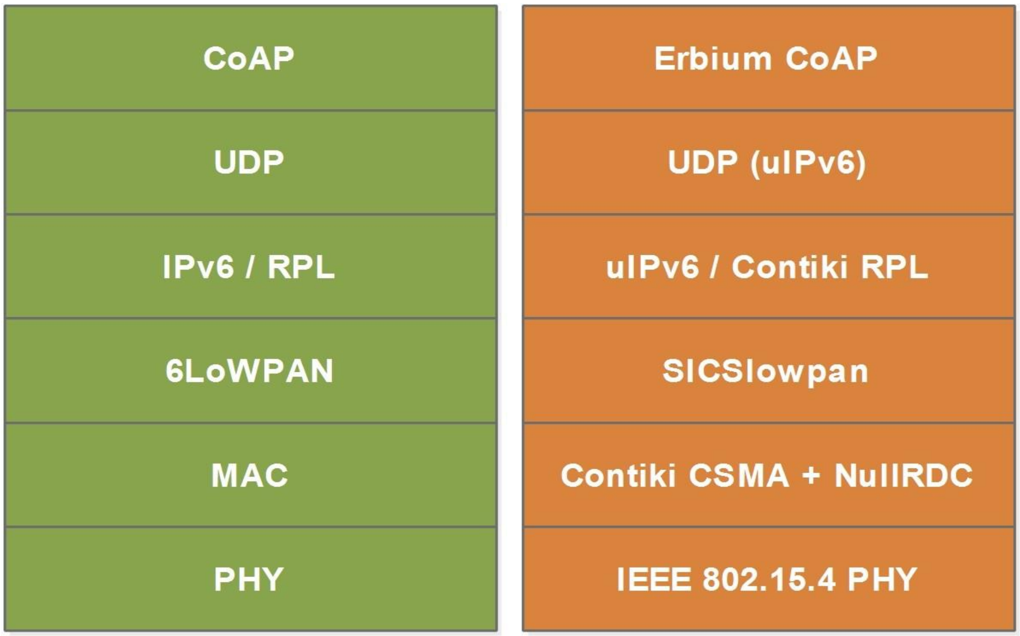

Figure 4.
The default IETF communication protocol stack against its implementation in Contiki platform.
Source: Akpakwu, et al. [11].

We carefully selected two IEEE 802.15.4 capable motes with the same radio transceiver (CC2420) [29] for this simulation: the Moteiv Tmote Sky [30], a hardware equivalent of the TelosB motes, and the Zolertia Z1 mote. The relevant features for the different motes used are defined in Table I, showing their unique capabilities [11].

Table 1.
Hardware simulation parameters for the Wismote and Moteiv Tmote sky wireless sensor nodes [17].

Features	Z1	Tmote Sky
RAM	8 KB	10 KB
ROM	92 KB	48 KB
MCU	MSP430F2617	MSP430F1611
RADIO	CC2420	CC2420

The evaluations of the CC methods will be examined while considering the constant traffic scenario:

- Constant bit rate (CBR) pattern: This describes a scenario in which nodes send messages at regular, predetermined intervals directly to a sink node. In the IoT, the packet generation rate remains constant over time, making it predictable and ideal for applications requiring consistent data flow, such as streaming or telemetry. The packet inter-arrival times are homogeneous and straightforward to analyze due to their steady load on the network. Generally, we need to forward these data to a database or service that can store and process them.

In the performance evaluation of the CC mechanisms, we consider the grid topology for simulations. In this deployment, the positioning and number of nodes vary, revealing specific differences, such as the number of direct neighbours, the distance between the source and destination of the CoAP request, and the number of nodes concurrently competing for the radio channel.

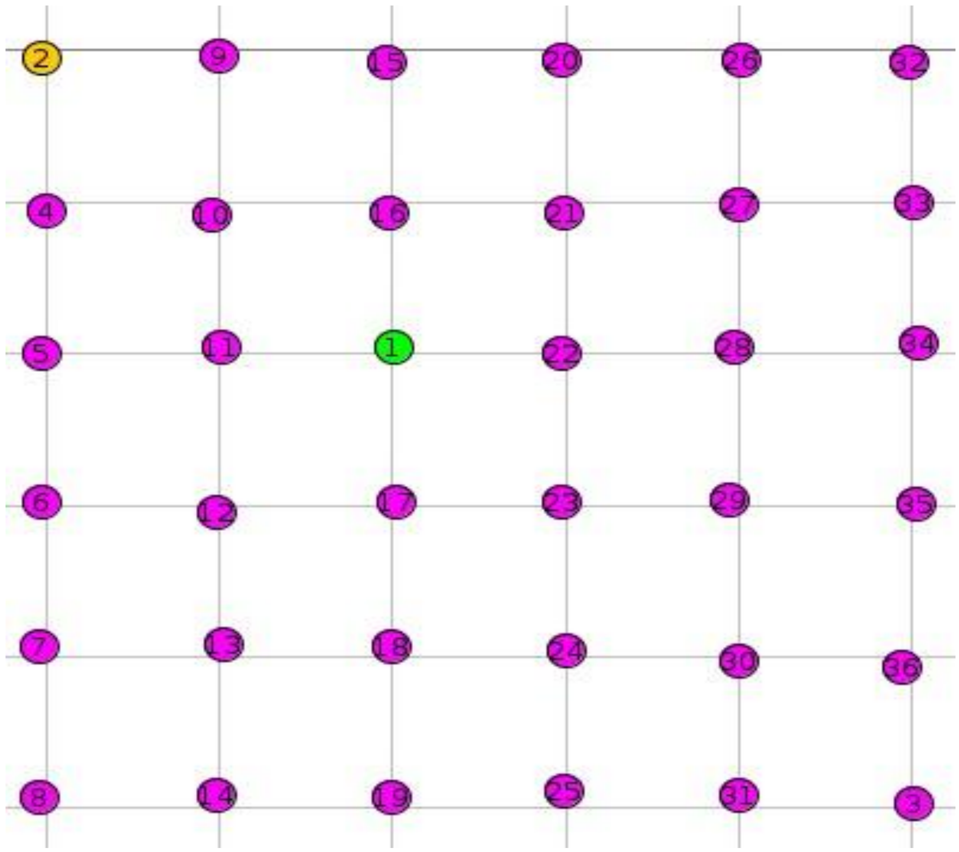

Figure 5.
A 36-grid network topology. 1 and 2 represent the RPL border router and sink node, respectively.

The RPL border router's job is to start the Destination Orientated Directed Acyclic Graph (DODAG), which is the network setup created and maintained by RPL, and to keep the routing information collected for all nodes in the DODAG [30]. In the simulations, the RPL border router is designated solely as a relay for CoAP communications; it neither generates CoAP messages nor serves as the ultimate endpoint for them. Contiki OS exclusively provides the storage mode for RPL [30]. RPL border routers require more RAM than other network nodes, leading to their designation as TMote Sky motes. Compared to the Z1 motes used in all other network nodes, a TMote Sky provides an additional 2 kbytes of RAM for storing routing information. The Z1s, on the contrary, provide an increased ROM capacity, allowing greater allocation for application coding and CC control methods for CoAP.
The simulation employs a Unit Disc Graph Medium

(UDGM) radio model for radio transmissions, which features circular transmission and incorporates interference zones. The UDGM radio model employs a static link delivery ratio (LDR) configuration. When a node transmits a packet, other nodes within the transmission range can receive it. Nodes within the interference range cannot receive the packet; nevertheless, it may induce collisions with other packets delivered concurrently. The grids illustrated in Fig. 5 establish the transmission ranges of the nodes at 10 m, which corresponds to a square unit edge. We employ the 802.15.4 MAC layer protocol and the UDGM propagation model in the MAC and physical layers, respectively.

The simulations of the constant traffic scenario have a duration of 10 minutes. We repeated the simulations five times, each with a different transmission interval, for this evaluation. The simulation setup and network scenario of this section serve as the basis for evaluating the various CC approaches for CoAP using several performance metrics.

4.2. Congestion Control Performance Metrics

To evaluate and compare the performance analysis of CC mechanisms for CoAP, we decided to consider a set of performance metrics. We selected these metrics to determine the performance of the network topology under the CBR traffic pattern. Using steady-state probabilities, the expected values for key performance metrics can be derived. We use the following metrics to evaluate the performance of the Markov model:

1) *Throughput* (T_p) : The successful delivery rate data packets over the network is known as throughput. It is defined by Equation 9:

$$(T_p) = \frac{Total\ Packets\ Received}{Total\ Time} \quad (9)$$

Markov model computes the expected throughput as follows using Equation 10:

$$Tp = \pi 1 Tp1 + \pi 2 Tp2 + \pi 3 Tp3. \quad (10)$$

where T_{p1}, T_{p2}, T_{p3} represent the throughputs in states S_1, S_2, S_3, respectively.

2) *Packet Loss Rate* (P_l) : The packet loss rate indicates the ratio of lost packets to the total packets transmitted on the network. It takes the following form, as indicated by Equation 11:

$$(P_l) = \frac{Packets\ Lost}{Packets\ Sent} \quad (11)$$

In the Markov model, the total packet loss rate is calculated as a weighted summation of the loss rates throughout each state, as indicated by Equation 12:

$$Pl = \pi 1 Pl1 + \pi 2 Pl2 + \pi 3 Pl3, \quad (12)$$

where P_{l1}, P_{l2}, P_{l3} are the loss rates in states S_1, S_2, S_3, respectively.

3) *Round-Trip Time (RTT):* The round trip time (RTT) is the duration required for a packet to cross from the sender to the receiver and return. It encompasses the delays attributable to transmission, propagation, processing, and queueing. The RTT is calculated as follows in Equation 13:

$$\text{RTT} = \text{Transmission Delay} + \text{Propagation Delay} + \text{Processing Delay} + \text{queueing Delay} \quad (13)$$

The Markov model calculates the expected RTT as follows using Equation 14:

$$RTT = \pi_1 RTT_1 + \pi_2 RTT_2 + \pi_3 RTT_3. \quad (14)$$

where RTT_1, RTT_2, RTT_3 represent the RTT in states S_1, S_2, S_3, respectively.

V. PERFORMANCE EVALUATION

This section compares how well the Default CoAP, CoCoA, and the new Markov Model perform based on three important measures: throughput, packet loss, and latency. Transmission intervals ranging from 1 to 25 were used to simulate different network conditions, reflecting variations from low to high congestion. The results clearly show that the Markov model performs better in all areas, proving it can effectively adjust to changing network conditions using its probabilistic state-based system.

4.2.1. Throughput vs Transmission Interval

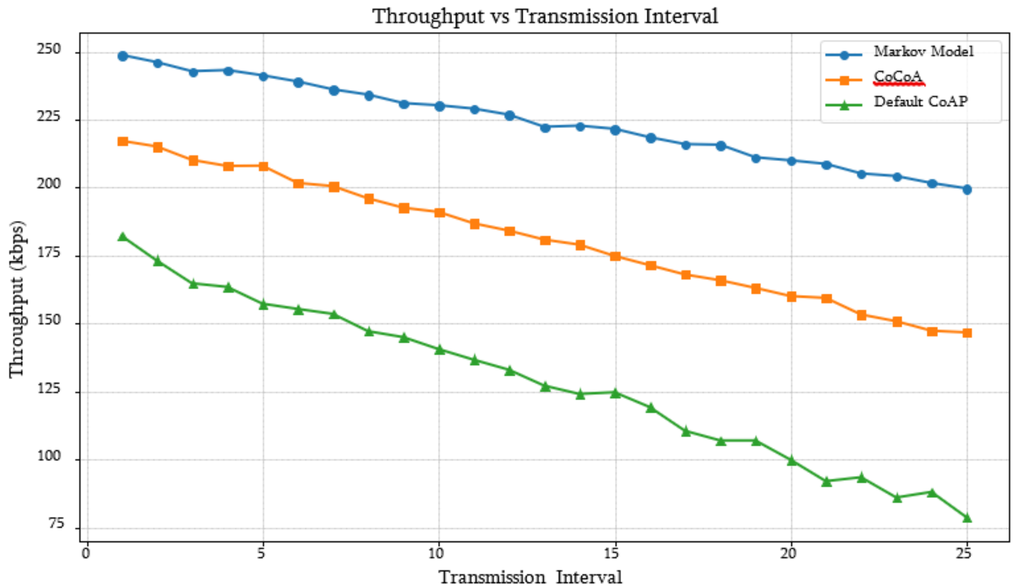

Figure 6.
Performance evaluation based on data transmission interval for throughput.

Table 2.
Summary of generated results for Throughput (kbps) at different transmission intervals.

Intervals	MM (T_p)	CoCoA (T_p)	CoAP (T_p)
1	247.254	217.925	176.054
5	241.053	207.944	162.598
10	230.741	187.230	138.435
15	218.313	175.990	119.073
20	210.815	158.610	101.825
25	199.237	145.065	78.714

Throughput, measured in kilobits per second (kbps), indicates the efficiency of the protocols in successfully transmitting packets across the network. Fig. 6 illustrates that the throughput of all protocols decreases as the transmission interval increases, corresponding to higher congestion levels.

However, the Markov model maintains a consistently higher throughput across all intervals compared to both CoCoA and Default CoAP. The Markov model maintains throughput levels near 248 kbps at low transmission intervals and gradually decreases to about 200 kbps during severe congestion. In contrast, CoCoA initiates at approximately 218 kbps and deteriorates more rapidly, and default CoAP demonstrates the most pronounced decline, dropping below 80 kbps at high intervals. This performance demonstrates that the Markov model may more effectively adapt its transmission strategy in advance to maintain network efficiency despite unfavourable conditions.

The throughput advantage of the Markov model is due to its ability to predict network congestion trends through transition probabilities between states, thereby optimising packet generation rates without relying on packet loss to initiate reactive actions, as demonstrated in CoCoA and Default CoAP. Table 2 presents the throughputs at different intervals for the three protocols evaluated.

4.2.2. Packet Loss vs Transmission Interval

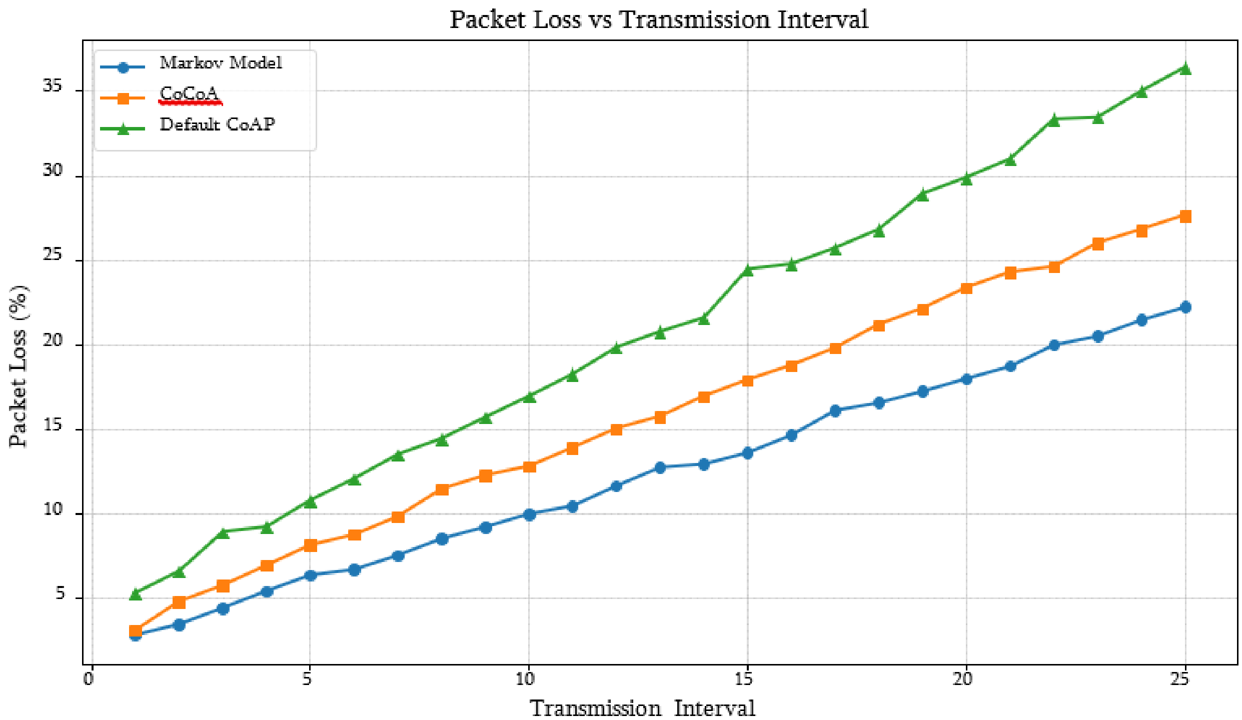

Figure 7.
Performance evaluation based on data transmission interval for packet loss.

Table 3.
Summary of generated results for Packet Loss (%) at different transmission intervals.

Intervals	MM (P_l)	CoCoA (P_l)	CoAP (P_l)
1	2.906	4.302	5.312
5	5.977	7.994	10.771
10	9.825	13.195	16.778
15	14.01	18.611	23.769
20	18.104	23.169	29.928
25	22.174	28.243	37.441

Packet loss serves as a vital metric for assessing network reliability and the extent of congestion. Figure 7 shows that packet loss increases with the transmission interval across all three protocols. The Markov model consistently demonstrates the lowest packet loss rates throughout all intervals. When there is little traffic, the Markov model keeps packet loss rates around 3%, while CoCoA shows about 4% and Default CoAP goes over 5%. As traffic increases, the Markov model's packet loss stays much lower, not going over 23%, while CoCoA gets close to 29% and Default CoAP goes over 37% at the highest transmission times.

This performance reflects the predictive adjustment capacity of the Markov model, in which the protocol transitions early into moderate or high congestion states based on probabilistic modeling. By managing traffic levels early on to avoid major losses, the Markov model stops packet drops better than the reactive timeout and backoff methods used in CoCoA and Default CoAP. Table 3 presents the packet loss rate at different intervals for the three protocols evaluated.

4.2.3. Latency vs Transmission Interval

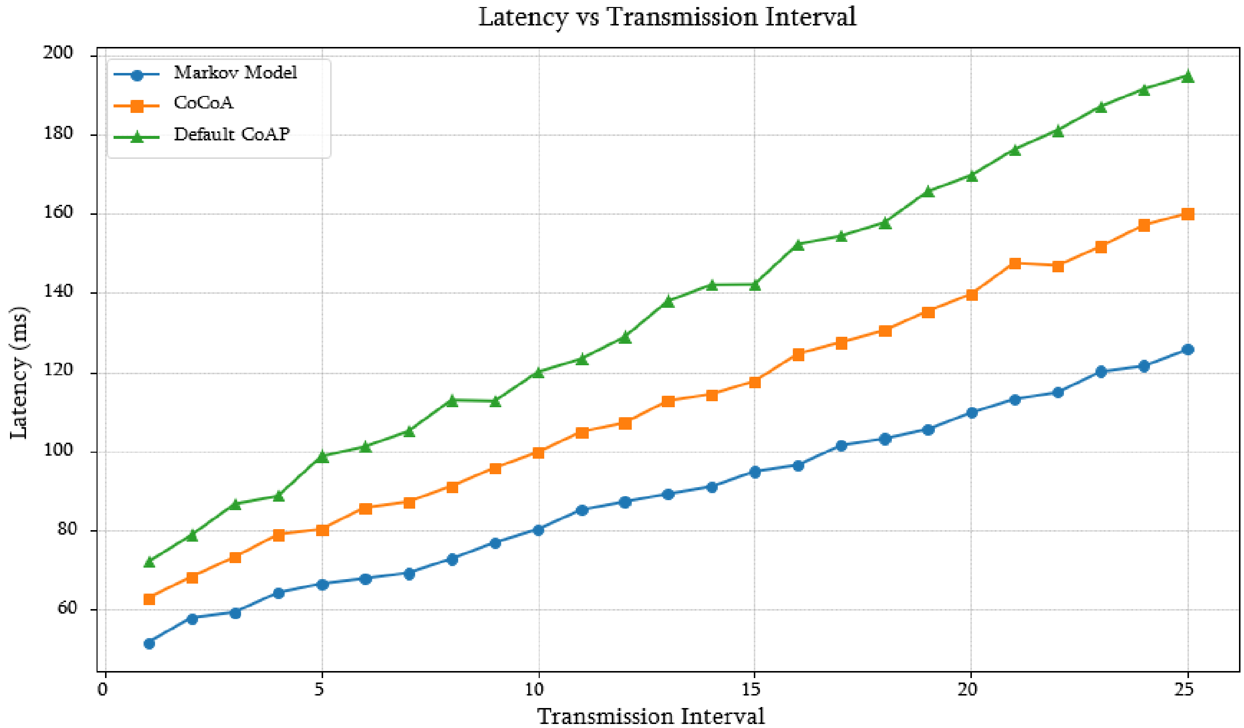

Figure 8.
Performance evaluation based on data transmission interval for latency.

The latency, measured in milliseconds (ms), indicates how long it takes for packets to travel through the network. Fig. 8 shows that the delay increases steadily with longer transmission intervals for all protocols; however, the Markov model always achieves the lowest latency. The Markov model initially maintains latencies of approximately 52 ms, whereas CoCoA begins at 63 ms and Default CoAP at approximately 77 ms. As network congestion escalates, Markov's latency rises considerably to around 127 ms, whereas CoCoA and Default CoAP have more pronounced spikes, exceeding 157 ms and 198 ms, respectively.

Table 4.
Summary of generated results for Latency at different transmission intervals.

Intervals	MM (L)	CoCoA (L)	CoAP (L)
1	51.667	63.446	76.876
5	65.538	76.465	95.691
10	77.706	100.289	121.912
15	95.871	120.016	141.986
20	111.166	139.136	171.140
25	126.603	157.875	198.993

The Markov model's superiority in latency stems from its prompt congestion detection and traffic adaptation techniques. The Markov Model reduces the time packets spend in the network by quickly changing states, which lowers packet queuing and retransmissions; in contrast, CoCoA and Default CoAP, which rely on reacting to changes, face longer queuing delays when there is a lot of congestion. Table 4 presents the latency at different intervals for the three protocols evaluated. Overall, the proposed Markov model outperforms both CoCoA and Default CoAP across all evaluated metrics. Its proactive, probability-based adaptation to congestion yields the following results:

- Enhanced sustained throughput with escalating transmission intervals.
- Have reduced packet loss rates even under significant congestion.
- Minimise end-to-end latency under all network situations.

These findings highlight the effectiveness of integrating stochastic modelling techniques, such as Markov state transitions, into congestion control protocols for IoT and resourceconstrained environments. The Markov model enhances communication reliability, efficiency, and responsiveness by dynamically predicting and responding to network behaviours.
CDF of Latency

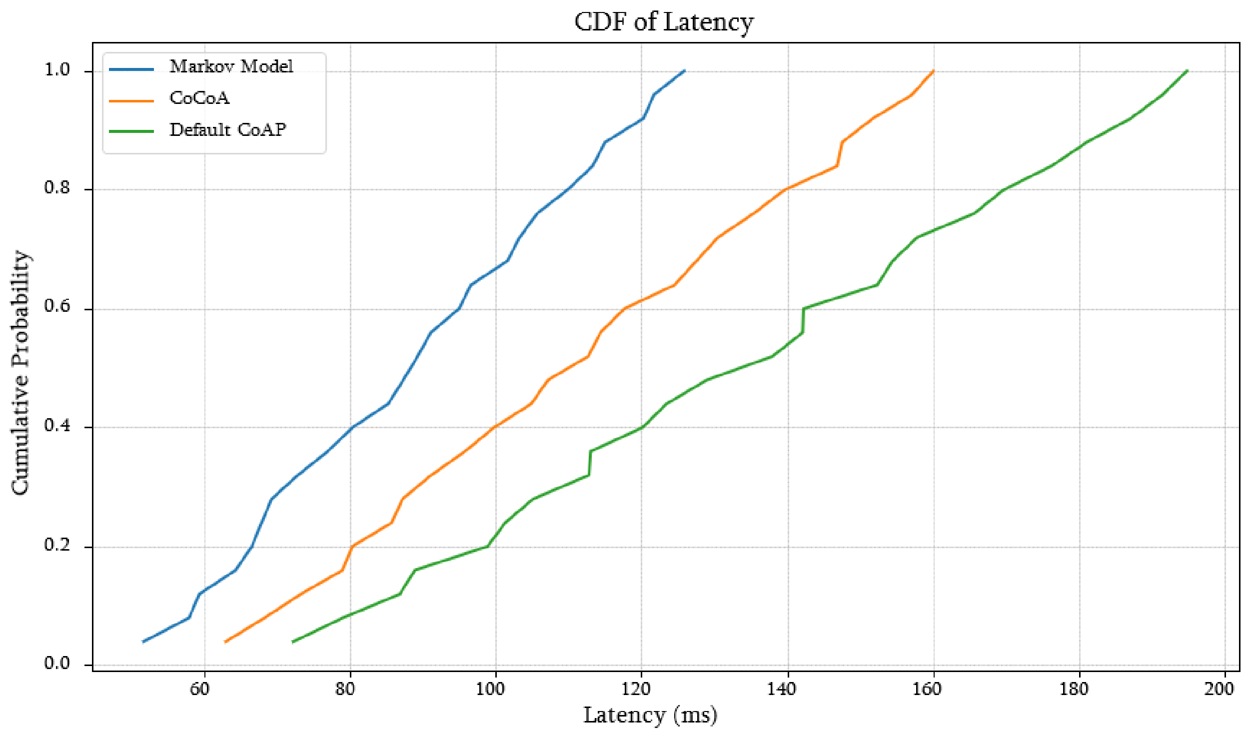

Figure 9.
Performance evaluation based on cumulative distribution function (CDF) of latency.

The cumulative distribution function (CDF) of latency for the three protocols, shown in Fig. 9, provides further insight into delay performance under different network conditions. The Markov Model curve consistently shifts to the left in comparison to CoCoA and Default CoAP, indicating a higher frequency of achieving lower latency values.

At a cumulative probability of 0.8, the Markov Model shows much lower latency values compared to the other two protocols, demonstrating its ability to lower end-to-end delay even when congestion increases. This outcome further substantiates the efficacy of the Markov Model's predictive, statedriven congestion adaptation method in diminishing queuing delays and retransmission overheads, in contrast to the reactive mechanisms used by CoCoA and Default CoAP.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 7: 1187-1204, 2025
DOI: 10.55214/25768484.v9i7.8874

4.2.4. Throughput vs Packet Loss

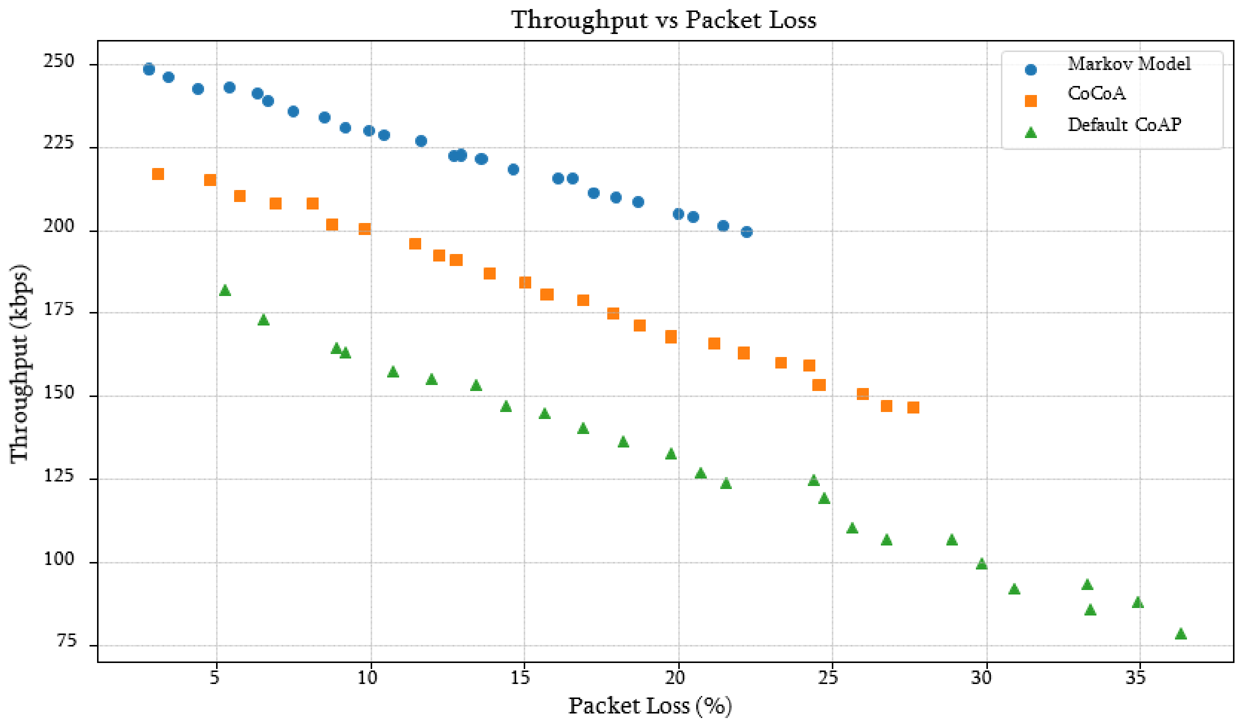

Figure 10.
Performance evaluation based on throughput vs packet loss analysis.

Figure 10 illustrates an examination of throughput in relation to packet loss, emphasising the robustness of the Markov model in congested network environments. The Markov Model sustains higher throughput levels despite rising packet loss, whereas CoCoA demonstrates a modest reduction, and Default CoAP experiences the most significant throughput deterioration.

This behaviour demonstrates that the proactive modifications facilitated by the Markov model's probabilistic state transitions maintain efficient data transmissions despite increasing network impairments. By preemptively identifying congestion and adjusting transmission rates, the Markov Model prevents the severe throughput decline seen with Default CoAP and alleviates the performance degradation apparent in CoCoA.

5. Conclusion and Future Work

CoAP delineates a conservative and non-adaptive mechanism for congestion control. CoCoA, a sophisticated congestion control mechanism for CoAP, offers a flexible and adaptive solution by integrating an adaptive RTO computation, weak RTTs, a VBF, and an aging mechanism to enhance performance.

This paper extends the congestion control framework to a three-state Markov model for CoAP-based IoT networks. This study conducted a comparative evaluation of the default CoAP, CoCoA, and a new Markov model for congestion control in constrained network environments. Through extensive simulations across a variety of transmission intervals, the Markov model has demonstrated improved performance in throughput, packet loss, and latency measures. Its proactive statebased adaptation mechanism, based on probabilistic modeling of network behaviour, allowed it to anticipate and alleviate congestion more efficiently than the reactive strategies used by CoCoA and Default CoAP.

Although CoCoA may be better suited for active congestion management, Markov excels at providing predictable and stable behaviour over time. It can be considered a foundational model for congestion control, with the possibility of future enhancements and optimisation. As IoT networks continue to evolve, the Markov model offers an easy-to-implement, scalable, and adaptive framework that ensures long-term efficiency in network performance. Moreover, it serves as a complement to CoCoA, particularly in systems that require both immediate congestion control and sustainable network management.

The results highlight how the use ofse of Markovian dynamics in protocol design can improve communication efficiency and reliability in Internet of Things (IoT) applications. Future work may aim to improve state transition methods and by integrating the arrival-service rate dependency in multi-hop networks with heterogeneous traffic patterns, reduce computational costs, and test the model in large real-world settings to further solidify its practical applicability.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Acknowledgment:

The author acknowledges the financial support from the Postgraduate Research and Innovation Program of the Tshwane University of Technology, Pretoria, South Africa.

References

[1] G. Akpakwu, A. M. Abu-Mahfouz, G. P. Hancke, and B. J. Silva, "An efficient congestion control scheme for a lightweight constrained application protocol in the Internet of Things," Ph.D. Dissertation. University of Pretoria, 2020.

[2] L. Mhamdi and H. Abdul Khalek, "Congestion control in constrained internet of things networks," *IET Wireless Sensor Systems*, vol. 13, no. 6, pp. 247-255, 2023.

[3] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431-440, 2015. https://doi.org/10.1016/j.bushor.2015.03.008

[4] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619-3647, 2018. https://doi.org/10.1109/ACCESS.2017.2779844

[5] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2014: IEEE, pp. 1-6.

[6] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "CoCoA+: An advanced congestion control mechanism for CoAP," *Ad Hoc Networks*, vol. 33, pp. 126-139, 2015.

[7] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," 2070-1721, 2014.

[8] C. Cirstea, M. Cernaianu, and A. Gontean, "Packet loss analysis in wireless sensor networks routing protocols," in *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*, 2012: IEEE, pp. 37-41.

[9] M. H. Asghar and N. Mohammadzadeh, "Design and simulation of energy efficiency in node based on MQTT protocol in Internet of Things," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015: IEEE, pp. 1413-1417.

[10] S. Bendel, T. Springer, D. Schuster, A. Schill, R. Ackermann, and M. Ameling, "A service infrastructure for the Internet of Things based on XMPP," in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2013: IEEE, pp. 385-388.

[11] G. A. Akpakwu, G. P. Hancke, and A. M. Abu-Mahfouz, "CACC: Context-aware congestion control approach for lightweight CoAP/UDP-based Internet of Things traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, p. e3822, 2020. https://doi.org/10.1002/ett.3822

[12] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Computing*, vol. 10, no. 6, pp. 87-89, 2006.
[13] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power CoAP for Contiki," in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011: IEEE, pp. 855-860.
[14] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1-31, 2014. https://doi.org/10.1016/j.comcom.2014.09.008
[15] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in IoT networking via TCP/IP architecture," Technical Report NDN-0038. NDN Project, 2016.
[16] L. Eggert, "Congestion control for the constrained application protocol (CoAP)," *draft-eggert-core-congestion-control-01 (work in progress)*, 2011.
[17] A. Betzler, C. Gomez, I. Demirkol, and 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*, 2013, pp. 365-372.
[18] S. Thombre, R. U. Islam, K. Andersson, and M. S. Hossain, "Performance analysis of an IP based protocol stack for WSNs," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016: IEEE, pp. 360-365.
[19] Y. Chen and T. Kunz, "Performance evaluation of IoT protocols under a constrained wireless access network," in *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, 2016: IEEE, pp. 1-7.
[20] M. Collina, M. Bartolucci, A. Vanelli-Coralli, and G. E. Corazza, "Internet of Things application layer protocol analysis over error and delay prone links," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, 2014: IEEE, pp. 398-404.
[21] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "CoAP congestion control for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 154-160, 2016.
[22] G. A. Akpakwu, T. E. Mathonsi, T. M. Tshilongamulenze, S. P. Maswikaneng, and T. Muchenje, "Congestion control in constrained application protocol for the internet of things: State-of-the-art, challenges and future directions," *IEEE Access*, vol. 13, pp. 33733-33767, 2025.
[23] R. Bhalerao, S. S. Subramanian, and J. Pasquale, "An analysis and improvement of congestion control in the CoAP Internet-of-Things protocol," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2016: IEEE, pp. 889-894.
[24] R. Herrero, "Dynamic CoAP mode control in real time wireless IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 801-807, 2018.
[25] K. Hartke, "Observing resources in the constrained application protocol (CoAP)," 2070-1721, 2015.
[26] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, 2006: IEEE, pp. 641-648.
[27] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *29th Annual IEEE International Conference on Local Computer Networks*, 2004: IEEE, pp. 455-462.
[28] Texas Instruments, "Single-Chip 2.4 GHz IEEE 802.15.4 compliant and ZigBee™ Ready RF Transceiver," 2013. https://www.ti.com/lit/ds/symlink/cc2420.pdf. [Accessed Jul. 14, 2025]
[29] M. Corporation, "Tmote Sky: Ultra low power IEEE 802.15.4 compliant wireless sensor Module, Datasheet," 2006. https://www.sentilla.com/. [Accessed Jul. 14, 2025]
[30] T. Winter *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," IETF RFC 6550, 2012.