Hamid M. Hasan
Computer Engineering, Control and systems Eng., University of Technology.
hamid20101net@gmail.com

Ahmed I. Ahmed
Computer Engineering, Control and systems Eng., University of Technology.
ahmed.ahmed@uokufa.edu.iq

Received on: 11/1/2018
Accepted on: 5/4/2018
Published online: 25/8/2018

A Comparative Analysis for Congestion Mechanism in COAP and COCOA

***Abstract**: Internet of things (IoT) is the paradigm for internetworking devices for data exchange and control. One of the important problems facing the IoT networks is the congestion; many researchers have discussed this problem. In this research, a comparison between two congestion control mechanisms used for internet of things the (CoAP and CoCoA) is presented. The cooja simulator was used for simulating different topologies and scenarios and the packets transmitted was captured using wireshak utility program. The captured packets were analyzed using MATLAB tools. The analysis showed that the CoCoA outperform CoAP in terms of goodput, number of dropped packets and bandwidth utilization.*

***Keywords**: CoAP, CoCoA, congestion control, Cooja,IoT,Wireshark.*

How to cite this article: H.M. Hasan, A.I. Ahmed, "A Comparative Analysis for Congestion Mechanism in COAP and COCOA," Engineering and Technology Journal, Vol. 36, Part A, No. 8, pp. 867-877, 2018.

1. Introduction

Congestion problem in the internet of things (IoT) influences their performance, and to improve the performance some control must exists. Many control mechanisms have been proposed. One of these mechanisms is use to add a congestion control to the CoAP which is the application protocol widely used for IoT. Many researchers investigated the performance of CoAP for different topologies of IoT networks. In this research, a comparative analysis is carried out between CoAP and CoCoA. Different topologies with many scenarios are simulated using Cooja simulator and with aid of Wireshark utility and MATLAB tools, the analysis was done. The results showed that CoCoA is more robust than CoAP due to its adaptive behavior.

2. The Internet of Things

The internet of things (IoT) is the networking module in which everyday objects can communicate with each other's and with other devices and services using the internet, each object is equipped with sensing, identification and networking capabilities [1]. There were many efforts in order to access computerized or embedded systems remotely via internet [2]. As well as M2M communications and WSN technologies, these efforts with advances in mobile technology and high electronic integration, low cost and high speed that lead to availability of embedded systems, are the main factors lead to IoT. Recently, the embedded systems is available and integrated into everyday objects, these provide the objects with capabilities of sensing and communicating. The smart embedded systems now used in control and automation in cars, homes, (HVAC), healthcare systems and industrials (M2M). In addition, it is expected that. There are 50 billion of such devices will be connected to the internet by 2020 [3], and 45% of the internet traffic will be for M2M traffic flow [2]. Hence, therefore the IoT is emerged to accommodate this growth in the interaction between objects.

I. The impact of IoT and its applications

The IoT have application in many fields like, personal applications, home automation, health care systems and industrial automation. Health care and industrial applications are the economic impact as (41% and 33% respectively) [2].

II. IoT architecture [2,4,5]

Many architectures have been proposed for the IoT but the yet not converged to a reference model. From the many proposed architecture the basic model is the 3-layered model, which is consists the application layer, network layer and perception layer. Recently, a 5-layer architecture is proposed which add more abstraction and known as service-oriented architecture (SOA), see Figure 1.

III. IoT enabled technology

There are six elements needed to bring up the functionality of the IoT that are (identification, sensing, communication, computation, services and semantics) [2]. The available technologies for each element are summarized in Figure 2.

IV. IoT *software stack. [6]*

The software stack or the communication suite

https://doi.org/10.30684/etj.36.8A.5
2412-0758/University of Technology-Iraq, Baghdad, Iraq

that used in the thing node is composed of five layers as shown in Figure 3 .The physical and data link layers are implemented as IEEE 802.15.4 standard. In the network, layer the IPV6 protocol used for communication protocol and the RPL as a routing protocol. Nevertheless, due to the header complexity of IPV6 the adaptation layer 6Lowpan is used for convert the complex header into the one used by IEEE802.15.4 the UDP and ICMPv6 are used for add controls over the IPV6 protocols. In the application layer many protocols can be used one of them is the CoAP, which is provided with simple congestion control mechanism.

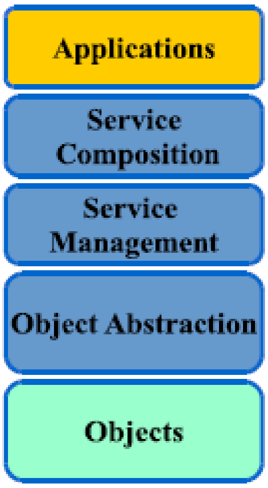

Figure1: IoT service oriented architecture (SOA) [2].

IoT Elements		Samples
Identification	Naming	EPC, uCode
	Addressing	IPv4, IPv6
Sensing		Smart Sensors, Wearable sensing devices, Embedded sensors, Actuators, RFID tag
Communication		RFID, NFC, UWB, Bluetooth, BLE, IEEE 802.15.4, Z-Wave, WiFi, WiFiDirect, , LTE-A
Computation	Hardware	SmartThings, Arduino, Phidgets, Intel Galileo, Raspberry Pi, Gadgeteer, BeagleBone, Cubieboard, Smart Phones
	Software	OS (Contiki, TinyOS, LiteOS, Riot OS, Android); Cloud (Nimbits, Hadoop, etc.)
Service		Identity-related (shipping), Information Aggregation (smart grid), Collaborative-Aware (smart home), Ubiquitous (smart city)
Semantic		RDF, OWL, EXI

Figure 2: Internet of Things: A Survey on Enabling Technologies [2].

Figure 3: IoT software stack

3. Congestion and Congestion Control

I. Congestion problem

In general, the congestion is occurred in network layer when the number of datagramsexceeds the network capacity [7]. This means that the traffic size is beyond the routers queue capacity or is large than to be consumed by the destination nodes .In this situation a lot of datagram will be dropped, which is lead to increase of the congestion due to the retransmission mechanism used by the upper layers. Then the congestion may cause the network to be collapsed and no datagrams will be delivered.

In wireless sensor networks the congestion is a problem issue that happened when the load is being large, that is because the WSN is designed to working under light loads so their resources are constrained. As a consequence the increase in load will lead to increase in number of dropped packets .Another source of congestion in WSN is the collision that occurs in the data link layer which is lead to retransmission and rerouting.

IoT has been built on the WSN as one of the buildings blocks. So that the congestion problem which is happened in WSN will be faced by IoT. Another issue of congestion in IoT is the internet interface or gateway that is used in IoT architectures.

II. Congestion control

In order to control on the congestion problem, three actions are needed: (congestion, detection, congestion notification and overcoming the congestion). There were many different congestion control mechanisms used for different environments and implemented in different layers of the communication subsystem. In the internet environment, the congestion control method implemented in the TCP layer, which is known as (AIMD), and the different AQM techniques are used in the network layer of routers, these techniques largely lead to the stability of the internet today. For WSN many methods are proposed for congestion control some were implemented in the data link layer [8] such as (Self-organizing Medium Access Control (SMACS), on-demand TDMA extension of IEEE802.15.4 MAC, Hybrid TDMA/FDMA-based medium access, CSMA/CA). Another technique are implemented in network layer, for example (Beacon Order Based RED (BOB-RED). Techniques implemented in the transport layer like (Datagram Congestion Control Protocol (DCCP), Pump Slowly Fetch Quickly (PSFQ), Sensor Transmission Control Protocol (STCP), Light UDP, and Reliable UDP). Some techniques have the cross-layer nature, such as (Fusion and CODA) [9].

In IoT environment and due to its constraint nature, the UDP is used in the transport layer, so the TCP complex congestion techniques cannot be used. The congestion control in IoT is handled by the application protocols like COAP; this is to overcome the congestion caused by the node-constrained resources. The congestion caused by the link collision is handled by the data link MAC control like SCMA/CA.

4. Congestion Control Mechanisms for CoAP

CoAP is a lightweight REST protocol standardized by (IETF). CoAP is the most used application protocol for IoT and it makes the interaction between IP based devices like HTTP protocol [Co1]. Four types of messages are defined in CoAP, which are CON, NON, RST, and ACK, The CoAP specification consist of a simple congestion control mechanism. This mechanism has two folds, the first one is by imposing restriction on the number of concurrent transmissions to be the only one outstanding message between the sender and the receiver, the other one is by using RTO and BEB for the CON type of messages .An initial RTO is selected from the interval (2,3) seconds. If no ACK is received after RTO interval, a retransmission is conducted and a new RTO is calculated using BEB, which is the double of the last RTO. The number of retransmissions is restricted to be no more than four after that a disconnection is occurs. This simple CC is not sensitive to the network condition because RTO is not depends on the RTT.

5. CoAP Simple Congestion Control/ Advanced (CoCoA)

CoCoA differs from the default CoAP by the following additions:

I. Adaptive RTO calculations

In CoCoA the RTO is calculated using the estimated value of RTT. For CON messages, the ACK is used for estimating RTT value. There will be two types of RTT, strong RTT and weak RTT. The strong RTT is calculated using ACK when no retransmission is occurred, but weak RTT is calculated after two retransmission. When strong or weak RTT is estimated, the RTO is updated using the following formula: -

$$SRTT_x = (1-\alpha) \times SRTT_x + \alpha \times RTT_x. \quad (1)$$

$$RTTVAR_X = (1-\beta) \times RTTVAR_X + \beta \times |SRTT_x - RTT_x| . \quad (2)$$

$$RTO_X = SRTT_x + K_X * RTTAVR_X. \quad (3)$$

$$RTO_{overall} = 0.5 \times RTO_x \, 0.5 \times RTO_{overall}. \quad (4)$$

Where β equal to (1/8) and α to (1/4), x is either strong or weak, SRTT is the smoothed RTT, K is (4) for strong and (1) for weak, RTTAVR is the RTT variation.

II. Variable Backoff factor (VBF)

In VBF scheme the Backoff values are chosen according to the criteria summarized by the formula (5). In this formula when RTO_{init} is large the Backoff chooses to be less than 2 to avoid large idle time, but it should be no less than 1 to ensure safe congestion control .For small RTO_{init}Backoff value is selected to be more than 2 to avoid the fake retransmissions , but not too large to avoid long idle times. The value between 1.3 and 3 is verified to give best result[13].

$$VBF(RTO_{init}) = \begin{cases} 3\,, & RTO_{init} < 2S \\ 2, & RTO_{init} \le 3S \\ 1.3\,, & O_{init} > 3S \end{cases} \quad (5)$$

III. RTO aging.

If the value of $RTO_{overall}$ is larger than or less than 2, and not updated for more than 30s , the $RTO_{overall}$ is modified by equation (6).

$$RTO_{overall} = (2 + RTO_{overall})/2. \quad (6)$$

6. Related Works

Betzler et al. [11] have verified the performance of COAP protocol using three types of network topologies (grid, dumbbell, and chain), then compared COAP with CoCoA, they showed that the CoCoA in average perform the same or better than COAP, they also used a version of CoCoA which uses the strong RTO estimation only. The evaluation was carried out using Cooja simulator and Contiki platform. Same researchers evaluate CoAP and CoCoA performance compared to the

TCP congestion mechanisms [12]. Another verification to COAPwas presented in [13], they carried out experiments with cloud services that communicate with CoAP servers on real sensor nodes in a testbed and compare howthe different is well congestion control schemes for CoAP perform. The CoCoA+ was verified in [10] they showed that CoCoA+ wasoutperform the CoAP and CoCoA for the topologies used in [11]. They concluded that CoCoA+ provide high reliability and lower delay. (A PDR improvement of up to 19.8% and a reduction of average delays during bursts of notifications of up to 31.2% were observed in comparison to default CoAP). In [14] show experiments on IoT environment emulated using netem, they compare between CoAP and CoCoA and two TCP based algorithms, Linux RTO and Peak-Hopper. They showed thatCoAP is inefficient as compared to the other protocols. In [15], they were designed and implemented "CoCoA 4-state-Strong" which is an adaptation to CoCoA that uses a 4-state RTO-strong estimator. They showed that the designed algorithm achieved 35-40% higher throughput but 20% higher in number of retransmissions. In [16], they compare between CoAP and CoCoA+ for two IoT cases (continuous monitoring and global event detecting), they showed that CoCoA+ perform worst than CoAP in case of bursty traffic or small RTT. In [17] they present analysis of CoCoA+ in a simulated environment of 6 by 6-node grid with periodic traffic. They highlighted some shortcomings of CoCoA+, that is the many spurious retransmissions at some offered loads and they concluded that this is due to the weak estimator of RTO.

7. Network Model

The model of the network used for congestion analysis is composed of multiple of lossly-coupled CoAP clients, which can access multiple of lossly coupled CoAP servers. The communication is done via Boarder Router. This model can be illustrated in Figure 4.

8. Simulation Environment

Each node (server or client) has a Contiki operating system. Contiki is open source operating system for IoT, the Contiki has the implementation of the IoT protocol stack for communication, see Figure 5. Contiki is used for many motes with different hardware .We used the wisemote for the servers and client nodes. Cooja simulator is a simulator and emulator for Contiki, and can simulate a wireless network of motes running Contiki. The Border Router is implemented in Cooja, which enables the simulated servers to be communicated with client from outside the Cooja simulator.

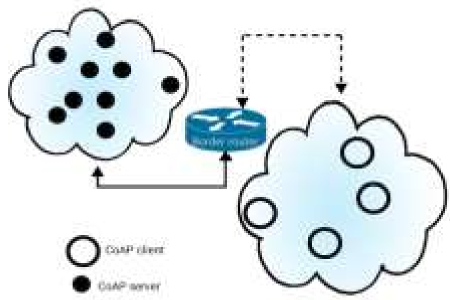

Figure 4:Network Model.

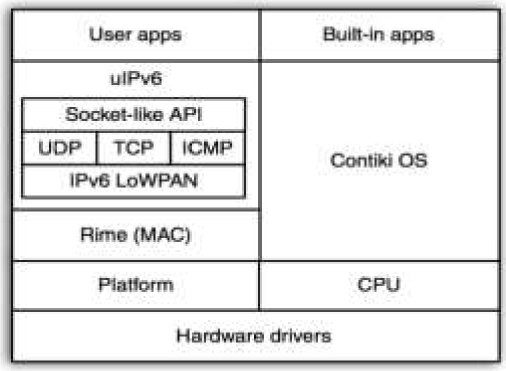

Figure 5: Contiki software stack [6].

9. Analysis Method and the Results

In order to carry out the comparative analysis between CoAP and CoCoA. Many scenarios of the network model were simulated using Cooja simulator. The simulation parameter is shown in Table 1. The packet's traffic was captured by Wireshark program. The filtered data from the Wireshark are feed to MATLAB program where the underlying network's parameters are computed, using MID no. that got from Wireshark with its relevant time. We compute average delay, the number of packets dropped, goodput, ACK-CON using MATLAB. The analysis method is summarized in Figure 6 show the work flow diagram above. The scenarios conducted are:

I. Single client-single server

Here only one client is communicating with one server through a boarder router.

II. Multi clients – single server.

The tests were carried out on multi clients arranged like dumbbell with single server. The cases used were, six clients to one server, eight clients to one server and twelve clients to one server See Figures 7-9.

III. Multi clients – multi server.

In this case, many clients communicate with many servers in one to one manner. Using dumbbell topology for six clients and servers. In addition, the chain topology for twelve clients and servers. See Figures 10, 11.

IV. E-health scenario.

In this scenario a real hospital's section is used as the environment (we used the paracticl section from AL-ALMANI hospital AN-NAJAF Governorate -IRAQ). The clients and servers as well as the border router are distributed as shown in Figure 12.Using Cooja simulator , each scenario runs for one hour and for three times for each protocols (CoAP and CoCoA). The packets were captured by the Wireshark utility. The captured packets analyzed using MATLAB tool and the evaluation data was obtained. The evaluation data is summarized in Figures 13 – 20.

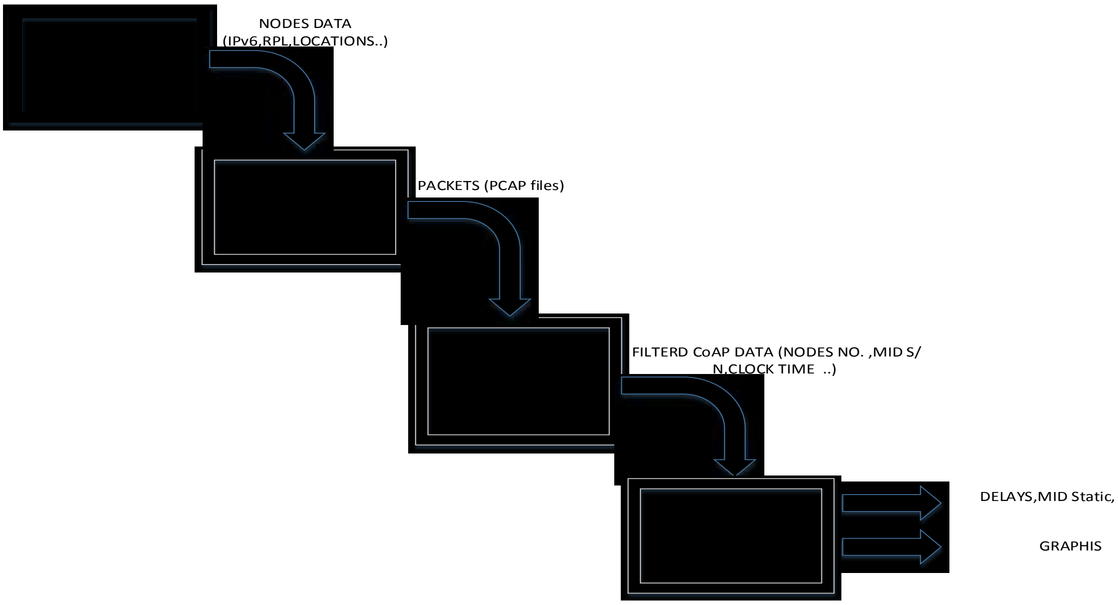

Figure 6: Work flow diagram

Table 1: simulation parameters

Parameter name	Value
Radio medium	Unit Disk Graph Medium
Mote Type	Wismote
MAC Layer	CSMA/CA
Bit Rate	250 kps
Radio duty cycling	Null RDC
Node Transmission range	50 m
Node Carrier Sense	100m
Tx/Rx Ratio	100%

Figure 7: Six clients-to-one server scenario

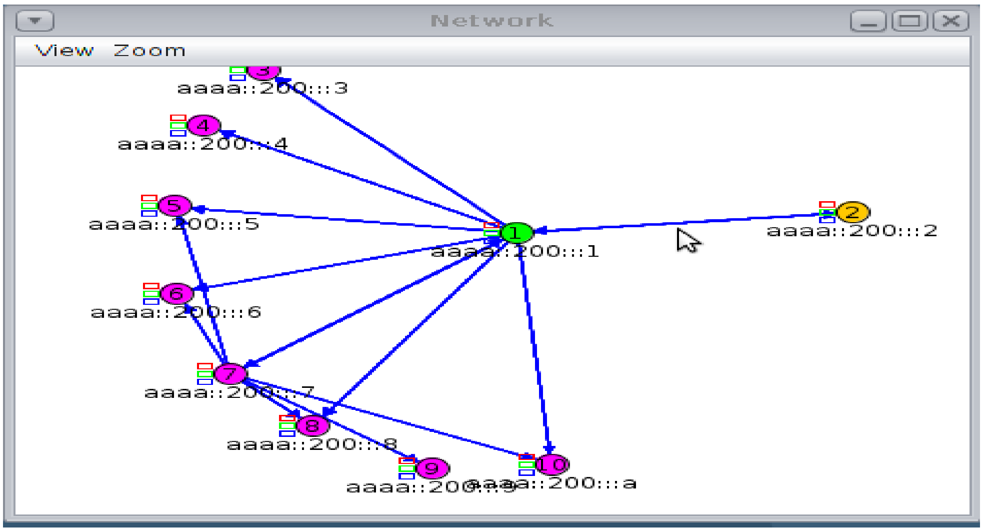

Figure 8: Eight clients-to-one server scenario

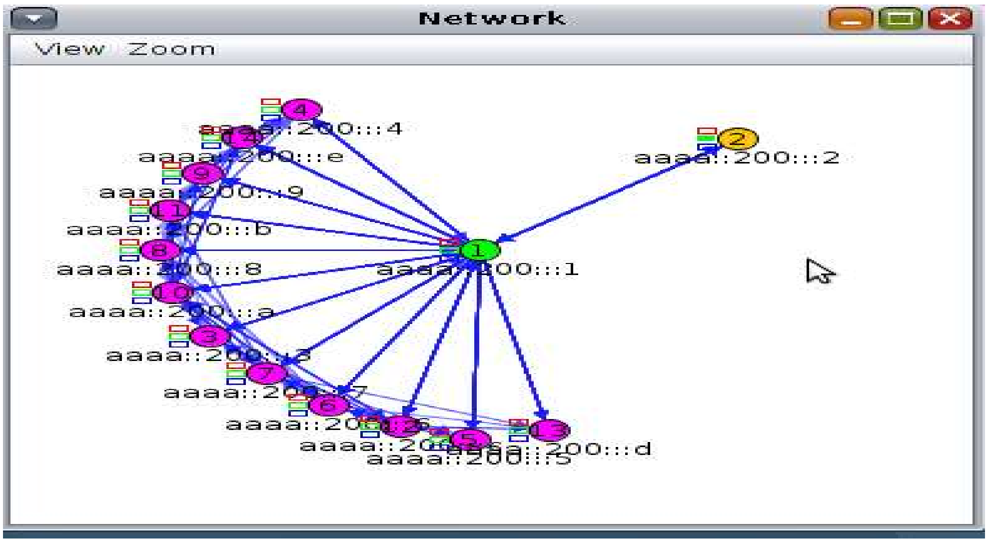

Figure 9: Twelve clients-to-one server scenario

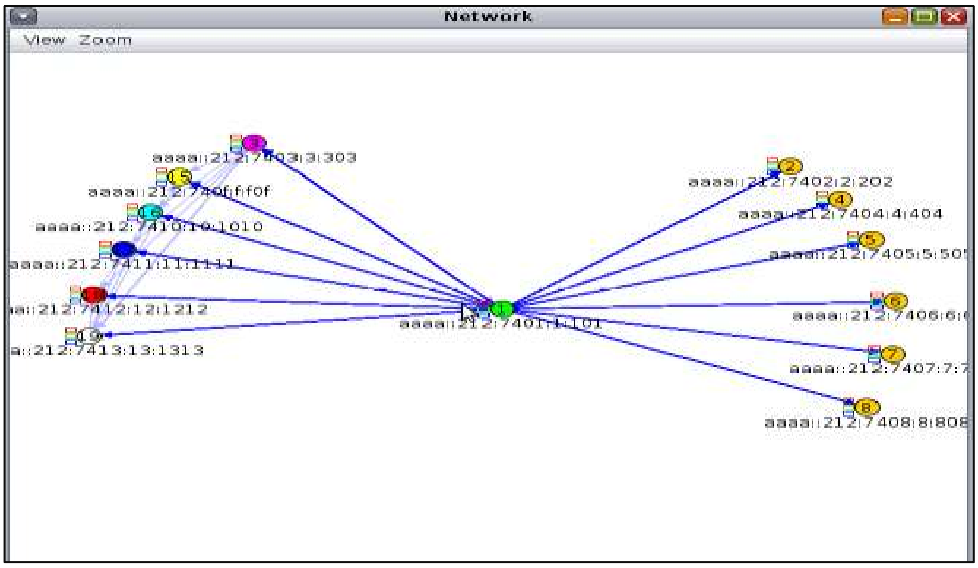

Figure 10: Six clients-to-Six server scenario

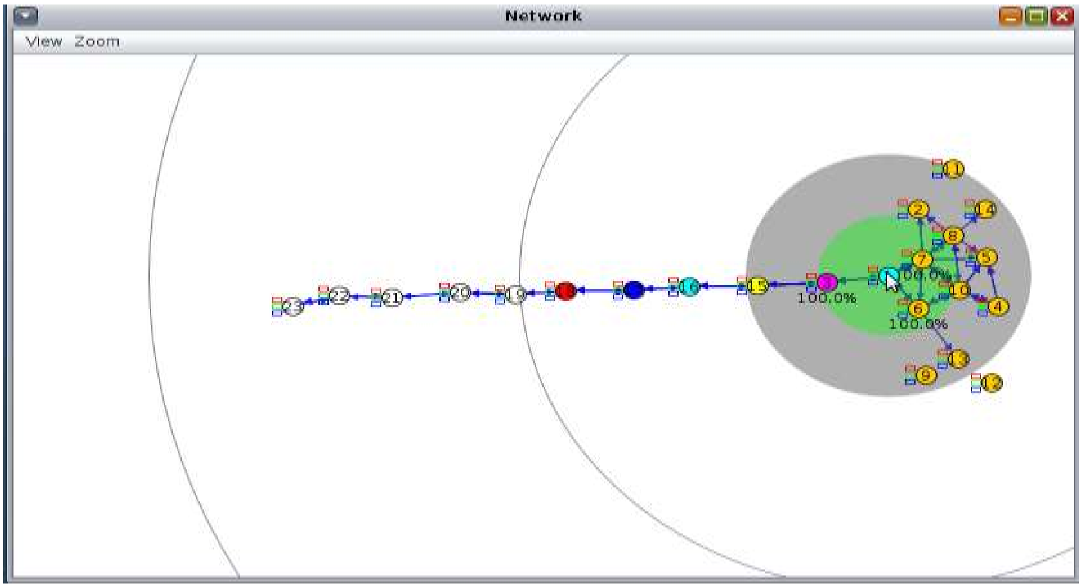

Figure 11: Twelve clients -to-Twelve-server dumbbell scenario

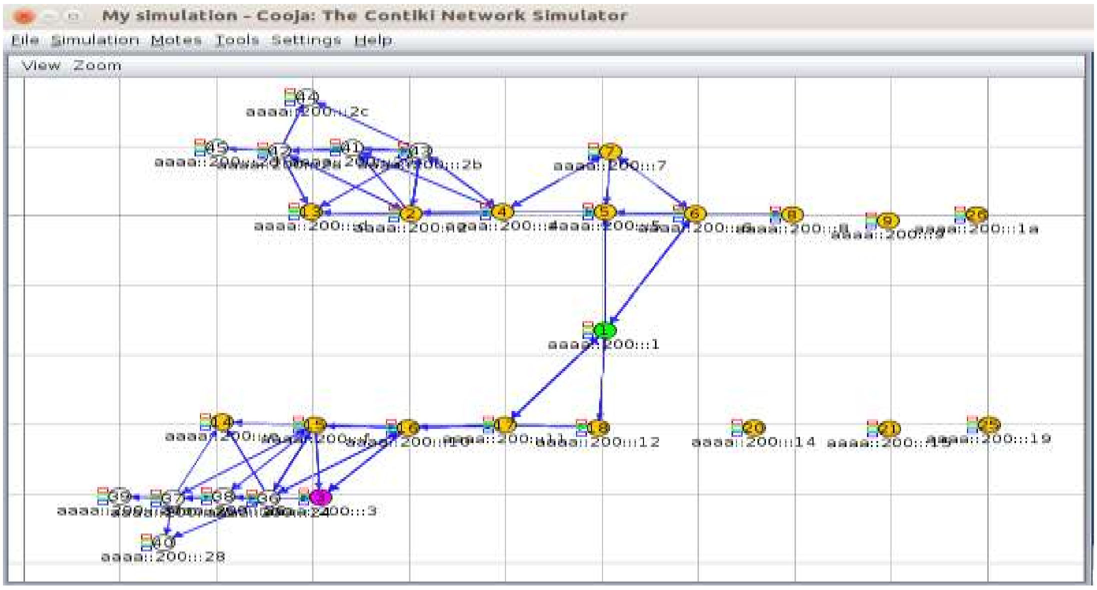

Figure 12: e-Health scenario

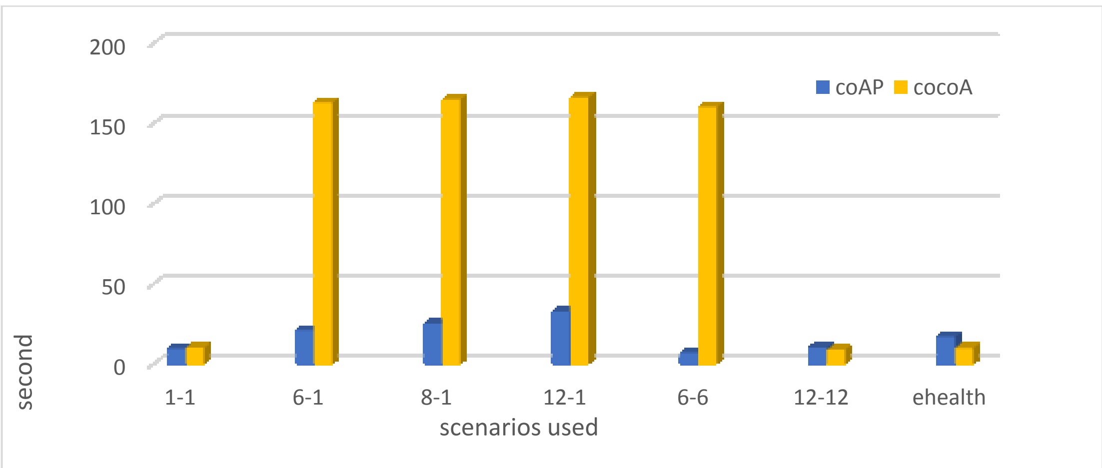

Figure 13: Average Delay

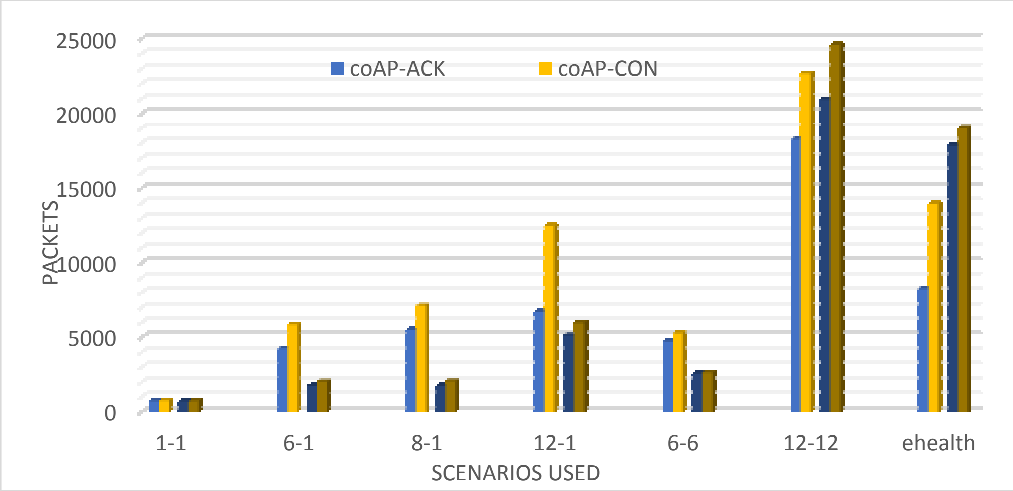

Figure 14: Total CON-ACK

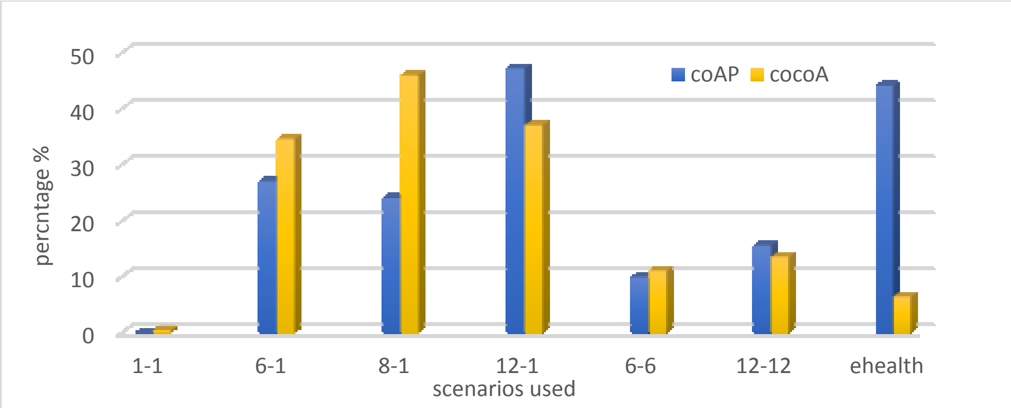

Figure 15: Number of packets dropped

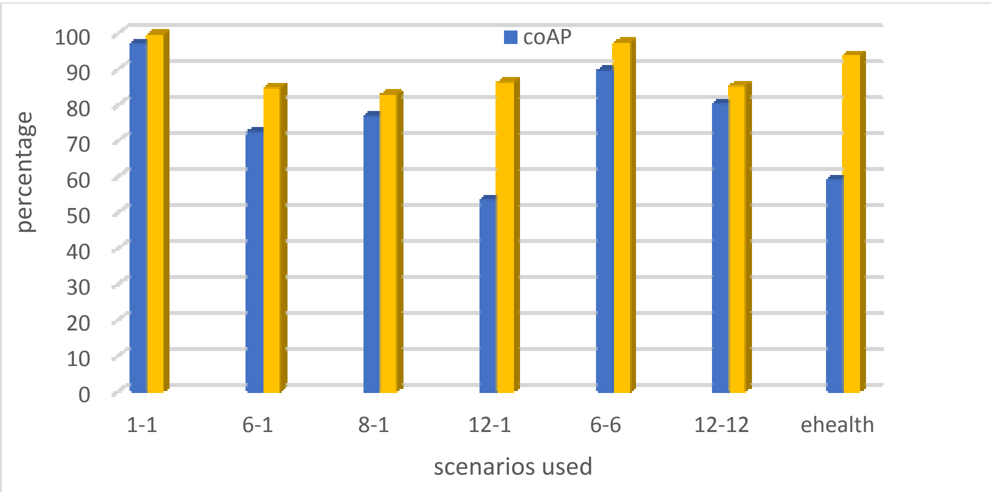

Figure 16: Goodput

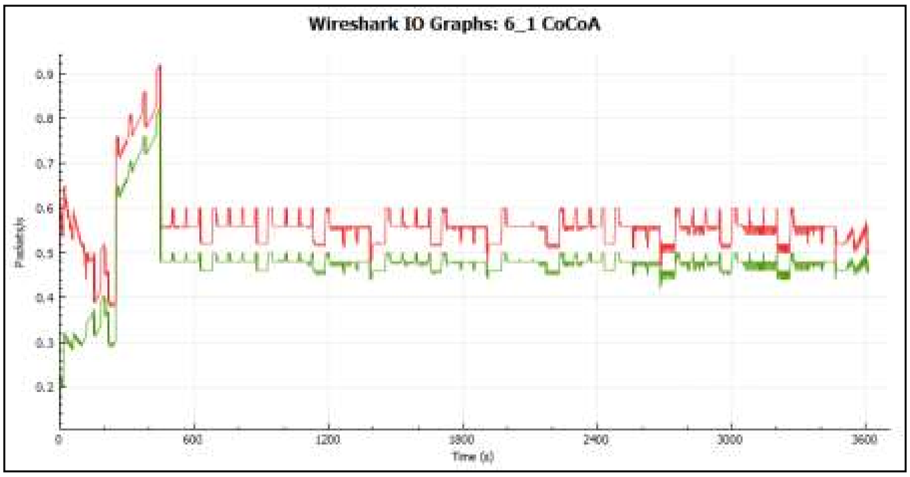

Figure 17: Generated CONs (red upper graph) and Generated ACKs, (Green lower graph) with time for six to one CoCoA.

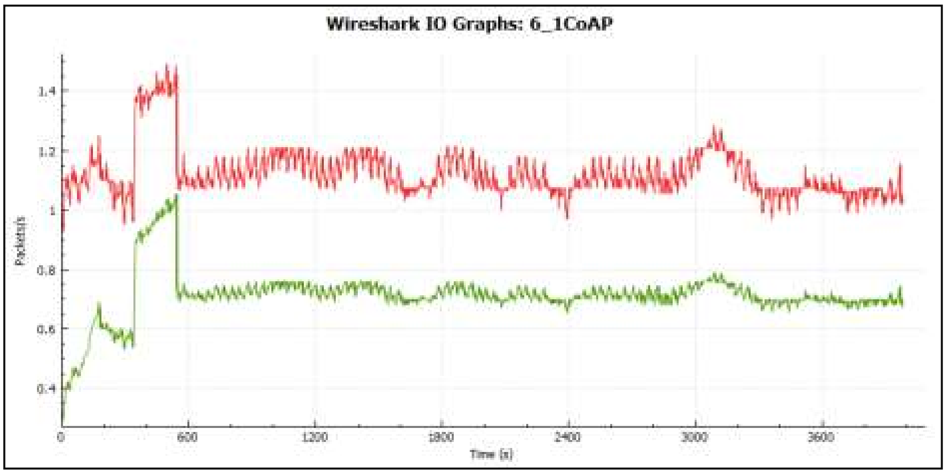

Figure 18: Generated CONs (red upper graph) and generated ACKs (Green lower graph) with time for six to one CoAP.

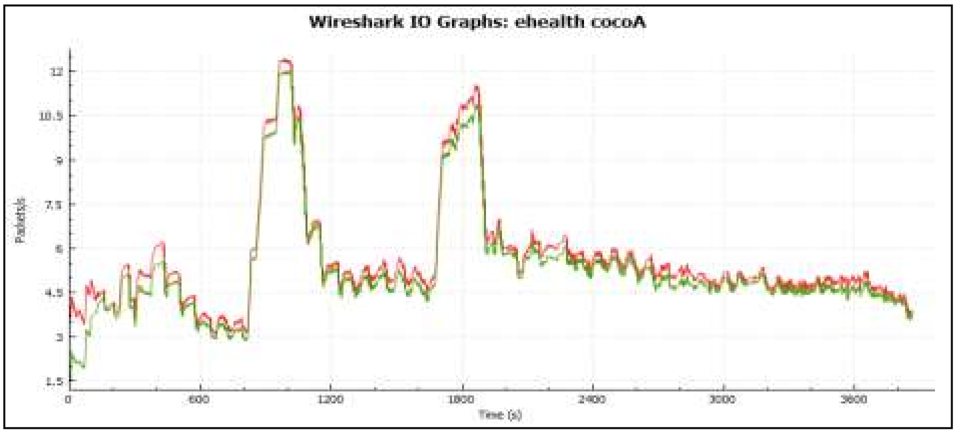

Figure 19: Generated CONs (red upper graph) and generated ACKs (Green lower graph) with time in ehealth using CoCoA.

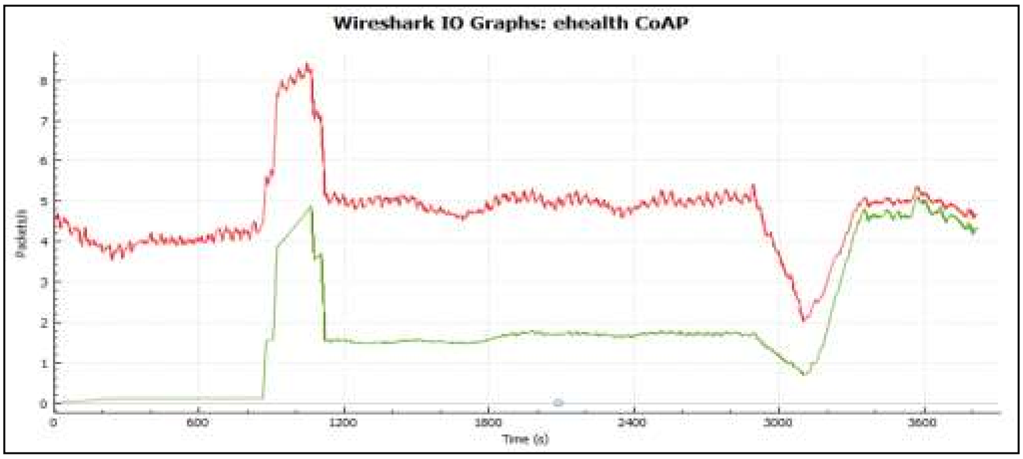

Figure 20: Generated CONs (red upper graph,) and generated ACKs (green lower graph) with time in ehealth using CoAP.

Table 2: Evolution parameters compression with pervious works

Source	Topology	Test time	Evaluation parameters	RDC, NSTRART	Analysis tools	CoCoA outperform
[10]	Chain 17 node Dumbbell 21 node Grid 49 node	10-5 min.	end-to-end delay, RTO, offered load	RDC, 1	COOJA	YES
[11]	Same 10	360 seconds	Dropped packet, throughput	RDC 1 and 4	COOJA	YES
[12]	12,20,30,40 nodes	Not mentioned	Average settling time, average retransmission, fairness	RDC, 1	COOJA	YES
[13]	2 one -one, many-to many 4, cross traffic burst 4,25 node	15 min	RTO	RDC and Null RDC,1and 4	COOJA	YES
[14]	Same 12	50 CON-CK completion 4-8 sec	Clients completion time	RDC, 1	COOJA	YES
[15]	20 node many-to-one	0-60 sec	RTO, good put	RDC, 1	COOJA	YES
[16]	Grid of 9,16,25,36	3 hours	PLR, retransmission /message	RDC, 1	COOJA	NO
[17]	Grid 36 nodes	800 seconds 12 times	Buffer overflow, carried/offered load, average ACK/ successful transaction, protocols operation	RDC, 1	COOJA	YES, with modification
This paper	1-1,6-1,8-1,12-1,6-6,12-12 chain, health (26nodes)	3600 sec	Average delay , number of dropped packet ,total CON-ACK, good put, throughput	RDC,1	COOJA, Wire shark, MATLAB	YES

10. Conclusions

In this paper, a comparative analysis for CoAP and CoCoA protocols is conducted by means of simulation of different topologies using cooja simulator. Many scenarios have been executed and the generated packets was captured using WireShark utility and analyzed using MATLAB tools. Table 2 show various evolution parameters for pervious work compared with this paper. CoCoA exhibits very good "good puts "for all scenarios and topologies which was be between (82% to 99%), compared to CoAP which was between (53% to 92%). The number of dropped packets was small in CoCoA as compared to CoAP. Although CoCoA experience more average delay in some scenarios its average delay is less in chain and ehealth topologies used in this evaluation. In addition, the generated CON in the CoAP is more than CON generated in CoCoA, which is, deduce efficient using of network bandwidth. So we can conclude that CoCoA is scalable than CoAP, that is due its adaptive behavior.

References

[1] A. Whitmore, A. Agarwal, L. Da Xu, "The Internet of Things—A survey of topics and trends," InfSyst Front vol. 17, pp. 261–274, 2015.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communication Surveys & Tutorials, Vol. 17, No. 4, Fourth Quarter 2015.

[3] D. Evans," The Internet of Things How the Next Evolution of the Internet Is Changing Everything," Cisco Internet Business Solutions Group (IBSG), 2011.

[4] R. Khan, S. Ullah Khan, R. Zaheer, S. Khan," Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges ", 10th International Conference on Frontiers of Information Technology, 2012.

[5] P. Sethi, S.R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," Journal of Electrical and Computer Engineering, Volume 2017, Article ID 9324035, 2017.

[6] S. Thombre, R.Ul Islam, K. Andersson, M.S. Hossain, "IP based Wireless Sensor Networks: Performance Analysis using Simulations and Experiments," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, Vol.7 Issue 3, pp. 53-76 , September 2016.

[7] B.A. Forouzan, "Data Communications and Networking," ISBN 978-0-07-337622-6, McGraw-Hill Copyright 2013.

[8] K. Kharb, B. Sharma, T.C. Aseri, "Reliable and Congestion Control Protocols for Wireless Sensor Networks," International Journal of Engineering and Technology Innovation, vol. 6, no. 1, pp. 68 - 78, 2016.

[9] G.W. Lee, S.Y. Lee and E.N. Huh, "Congestion Prediction Modeling for Quality of Service Improvement in Wireless Sensor Networks," Sensors Vol.14, Issue 5, pp.7857-7880, 2014.

[10] A. Betzler, C. Gomez, I. Demirkol, J. Paradells, "CoCoA+: An advanced congestion con-trol mechanism for CoAP," Ad Hoc Networks, 33, Pages 126-139, 2015,

[11] A. Betzler, C. Gomez, I. Demirkol, J. Paradells, "Congestion control in reliable CoAP communication," in: Proceedings of the 16th ACM International Conference on Modeling, Analysis & Simulation of Wireless and Mobile Systems, MSWiM '13, ACM, New York, NY, USA, pp. 365-372, 2013.

[12] A. Betzler, c. Gomez, I. Demirkol, J. Paradells, "CoAP Congestion Control for the Internet of Things," IEEE Communications Magazine, vol. 54, no. 7, pp. 154–160, July 2016.

[13] A. Betzler et al. "Congestion control for CoAP cloud services," Emerging Technology and Factory Automation (ETFA), 2014 IEEE.

[14] I. Järvinen, D. Laila K. Markku, "Experimental evaluation of alternative congestion control algorithms for Constrained Application Protocol (CoAP)," Internet of Things (WF-IoT). IEEE, 2015.

[15] R.I. Bhalerao, S.S. Sridhar, P. Joseph, "An analysis and improvement of congestion control in the CoAP Internet-of-Things protocol," Consumer Communications & Networking Con-ference (CCNC), IEEE, 2016.

[16] Emilio Ancillotti, Raffaele Bruno "Comparison of CoAP and CoCoA+ CongestionControl Mechanisms for Different IoT Application Scenarios," 22nd IEEE Symposium on Computers and Communications, Heraklion, Crete, Greece, 2017.

[17] Simone Bolettieri, Carlo Vallati, Giacomo Tanganelli, Enzo Mingozzi, "Highlighting Some Shortcomings of the CoCoA+ Congestion Control Algorithm," 16th International Conference on Ad Hoc Networks and Wireless, ADHOC-NOW 2017, Messina, Italy, September 20-22, 2017, Proceedings.

Author's biography

Dr. Hamid M. Hasan. Received B.Sc. in control and systems engineering, 1986, and M.Sc. in computer engineering, 1990, from university of technology, Iraq. He received Ph.D. in computer engineering, 2012, from Basra University, Iraq. He currently working as Assistant professor at the faculty of control and systems engineering, university of technology, Iraq. He is interested in computer networks studies and researches, wireless sensor networks realization, mobile robot navigation, IoT development and investigation, pattern recognition, Face recognition, image processing, stochastic DSP applications.

Ahmed I Ahmed. Received B. Sc in control and system engineering in 1994 and High diploma in information technology from Iraqi Commission for Computer &

Informatic , He currently working in university of Kufa . He interesting on network development, VSAT, WSN, IoT Development, Digital communication.