Adaptive Congestion Control Mechanism in CoAP Application Protocol For Internet of Things (IoT)

Rosilah Hassan[1], Ahmed Mahdi Jubair[2],Khairul Azmi[3] ,Abu Bakar[4]
Network and Communication Technology Lab. Research Centre for Software Technology and Management (SOFTAM),
Faculty of Information Science and Technology (FTSM),
Universiti Kebangsaan Malaysia, 43600
rosilah@ukm.edu.my[1], crombo@yahoo.com[2], khairul.azmi@ukm.edu.my[3]

***Abstract*—The Internet of things (IoT) presents the future of internet by incorporating objects to communicate with themselves. Different protocols have been emerged to meet the requirements of limited resource objects of IoT. Constrained Application Protocol (CoAP) is the application protocol which is used in IoT communication stack is prone to performance degradation resulted from traffic congestion. Basic congestion control suffers from different shortages and problems which lead to bandwidth consumption, data loss and increased delay. In this paper, a new adaptive congestion control mechanism to enhance performance has been proposed to overcome basic congestion control issues.**

Keywords- Interent of Things, CoAP, Congestion Control, QoS, CoCoA

I. INTRODUCTION

The Internet of Things (IoT) has been emerged as a leading technology for smart object communication. IoT is defined as physical object networks, which involves embedded technology that allows these objects to sense communicate and interact with each other or with external devices to provide data for different purposes. IoT provides crucial enhancement and high impact for several everyday life aspects and users behavior. For a private user field, critical features for IoT can appear in domestic and working fields [1].

The wide range and diversity of IoT applications results in increasing demands for different types of data communication models to meet these application requirements. Application requirements can include both categories of communication, reliable and unreliable. Congestion control is considered a critical issue for reliable data communication. It is crucial to have a flexible and efficient congestion control to provide reliable IoT communication. various applications require a specific level of QoS or delivery guarantees like medical or financial information. Applying these QoS levels requires efficient network performance and data transfer [2]. Figure 1showan IoT device which can be found in everyday life. These devices can have different data priority. For example, data from medical sensors and surveillance devices require higher priority than data retrieved from some electric devices like TV, washer, andrefrigerators.Different protocols have been designed to work on application layer including HTTP, FTP, and TELNETetc. These protocols mainly depend on two main transport layers like User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). This transport layer protocols provide different services based on different working mechanism. The environment of IoT is considered a constrained environment where limited resources are available. Traditional application layer protocols required higher computation and processing resources. Constrained Application Protocol (CoAP) [3] is a web transfer protocol designed especially for constrained devices with limited processing power and memory that typically operate in low bit

Fig.1 IoT Environment with high and low priority

rate environments. CoAP follows the Representational State Transfer (REST) architecture and has only a 4-byte header with UDP as its default underlying transport protocol. CoAP provides optional reliability based on retransmission timeout (RTO) mechanism[4]. An exponential back off of the RTO is employed as the default congestion control mechanism that is simple and has minimal implementation requirements. Since CoAP operates on top of UDP, CoAP assumes optional end-to-end (e2e) reliability and basic Congestion control. Fig. 2 shows the protocol stack for IoT architecture.CoAP protocolsworks in the application layer which is responsible for data formatting and connection handshaking [5].

II. BASIC CONGESTION CONTROL

For data communication CoAP provides four messages types: i. Confirmable(CON), ii. Non-Confirmable(NON), iii. Acknowledgement (ACK), and iv. Reset (RST). CoAP works based on requests and reposes approach[6]. Confirmable

message is used for data which required ACK for correctly delivered packets where Non-confirmable messages are used with data doesn't require delivery acknowledgment. Reset message are sued to reset the connection[7].

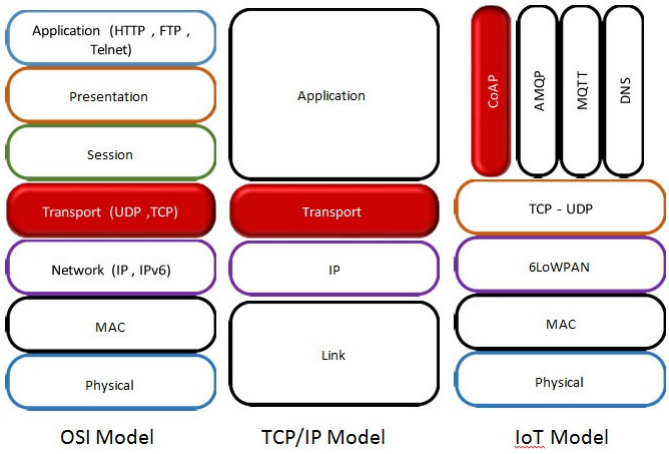

Fig 1: IoT model vs.TCP/IP and OSI Models

When a delivery guaranteed data needs to be sent, CoAP uses a confirmable message with random initial retransmission timeout (RTO) value. This chosen value is between 2 to 3 seconds for the first message sent. If the timer assigned to the chosen RTO value is expired and no ACK has been received for the sent message from the destination, then packet loss is assumed and the confirmable packet is retransmitted. For network congestion control, when a binary exponential back off protocol is implemented, this protocol increases the RTO value of the retransmitted packet in exponential manner. The mechanism for congestion control is related to any confirmable messages that CoAP can be sent whatever its destination. Another parameter applied which is referred to as NSTART, which specifies how many parallel connections CoAP can establish to the same destination at the same time. The specification states that NSTAT should start with the value of 1, which is sufficient for most applications running CoAP[8].

So Based on the CoAP specification, three main basic approaches can be used for the congestion control mechanism for CoAP including:

i. The calculation of RTO for the initial transmission of a confirmable CoAP message.

ii. The back off behavior applied to the RTO before retransmission of a confirmable CoAP message

iii. The state information stored about destinations of confirmable CoAP messages.

III. PROBLEM STATEMENT

The constrained resources characteristics of IoT devices are very crucial to be taken into consideration for both data communications between IoT devices or IoT device with control machine. CoAP as a light weight application protocol has been used to provide data communication. This protocol provides efficient, flexible and constrained resources suitable properties. Congestion is considered as one of the main performance degradation factors for data communication where packets loss is increased and results in high packet retransmitting attempts. This retransmitting attempts consume extra network bandwidth and processing resources which cause high performance degradation impact especially for IoT network environments[9].

Congestion control is the main solution to overcome congestion control issues and enhance data communication performance. CoAP provides a basic congestion control to enhance the network performance and minimize number of packet loss. Current proposed congestion control mechanisms suffer from different shortages including exponential back off timer and increased retransmitting timeout RTO. CoAP doesn't consider or collect any end to end communication details to optimize the value of RTO. So it behaves in the same manner with any network regardless of network state, number of connecting or transmitting devices and congestion level. CoAP application protocol requires a robust, adaptive and efficient congestion control mechanism to fit the changing environment of IoT wireless network and meet the requirement of increasing congestion and packet loss rates.

This research provides an adaptive congestion control mechanism for CoAP of IoT network, which is characterized as a resources constrained environment. Traditional CoAP congestion control mechanism doesn't fit with the congested behavior of IoT networks where performance is very critical and congestion can occur frequently. Adaptive congestion control is required to provide acceptable performance when increased number of packet loss happens.The objectives of this research are to enhance IoT data communication performance by enhancing the congestion control mechanism and to decrease the number of packet retransmission by applying adaptive congestion control mechanism.

IV. RELATED WORK

CoAP Simple Congestion Control/Advanced CoCoA [10] provides new updates to the simple congestion control of the CoAP by making the transmitting restrictions more flexible. The main idea for this approach is to change RTO value for any confirmable message by depending on the round trip time (RTT) of the sent packet for different destinations which is obtained using ACK packets. This approach is similar to TCP flow control mechanism. CoCoA record two main estimators for each destination node: strong RTO estimator and weak RTO estimator which are changed whenever a strong RTT or a weak RTT is occurred. The strong RTT is obtained when a first confirmable message is sent and its ACK is received. Where weak RTT is obtained when the ACK is received after single retransmitting attempt at least. Based on these two estimators, two average estimated values are calculated: strong RTO and weak RTO. Based on these two values the overall RTO is calculated based on the recent calculated RTO whether it is weak or strong as inequation 1.0.

$$RTO\_overall = 0.5 * RTO\_recent + 0.5 * RTO\_overall \quad \text{.....eq (1.0)}$$

Betzler et. al in [11] provides an enhancement on the working mechanism of CoCoA approach. CoCoA optimization has been applied by modifying CoCoA mechanism and implementing new steps. This optimization include RTO estimator aging for high values RTOs where if the estimator is not updated for about 60 seconds and its value is over than two seconds then RTT and RTT variation is decreased. This optimization has been performed because of constant values of RTO after specific time may not consider a real indicator about network real state. Another updates include ignoring RTT values for the first retransmission and reducing the dependability on weak RTO in overall RTO calculation as shown in the equation 2.0 below:

RTO_overall= 0.25 * RTO_weak recent + 0.75 * RTO_overall……eq (2.0)

Equations 1.0, provides better avoidance against ACK ambiguity problem and minimize its effects.

Back pressure congestion control for CoAP/6LoWPAN networks[12]has addressed network architectures design of for the IoT. Different algorithms has been presented to fit the protocol stack of IETF CoAP/6LoWPAN with congestion control functionalities[13]. Proposed congestion control design mainly depends on pressure routing. Three different cross-layer and fully decentralized congestion control a scheme has been proposed to overcome congestion. These schemes are compared with ideal back pressure and current UDP-based protocol stacks. Proposed lightweight congestion control algorithms can meet the requirements constrained resource devices. Two main scenarios have been investigated unidirectional flows and bidirectional flows.

A Channel Trust Based Approach for Congestion Control in loT[14] is a proposed congestion control mechanism which mainly depends on a heterogeneous node. It main goals is to check whether the devices channels are reliable to start sending data from source node to destination node based on two main metrics: the congestion level of nodes and the trust of the channel. In the proposed approach the MAC layer and network layer are collaborates to provide congestion control functionality. Application layer handled the source data or forwarded data to network layer for congestion avoidance. The proposed approach has three main stages: Priority setter, node congestion measurement and channel trust investigation. CoCoA+[15] is an improved version of CoCoA. Proposed schema has three main advances over the CoCoA. The first advance is an update to the calculation of the weak estimator to minimize the impact of weak RTT changes and its affect to the overall RTO. The second advance include back off timer replacement which has been used for retransmitting process to a variable backoff. The final updates are related to the age of overall RTO which result from leaving overall RTO for a certain time without changes.

V. PROPOSED METHOD

The proposed solution includes an adaptive mechanism for node to send data this adaptive mechanism mainly depends on the traffic priority class and the loss rate. The proposed solution has three main stages: traffic priority assignment, adaptive RTO and adaptive backoff timer as shown in Figure 3.

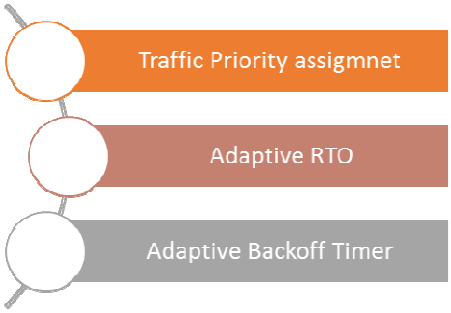

Fig. 3. Proposed mechanism phases

IoT networks can include different types of equipment with different priority levels. Some of this equipment can be critical like medical sensors and fire alarming system. The adaptive mechanism depends on the loss rate as follows:

- Loss Rate $_{recent}$< Loss Rate $_{avg}$, better network performance and less congestion assumed. RTOoverall and backoff timer should be decreased.
- Loss Rate $_{recent}$> Loss Rate$_{avg}$, bad network performance and higher congestion assumed. RTO$_{overall}$and backoff timer should be increased.

Two different values are identified to control the increasing and decreasing factor with different values for different traffic priority ***Fmax*** and ***Fmin***. These values are shown in Table 1:

TABLE I. F_{MAX} AND F_{MIN} VALUES FOR LOW ANF HIGH PRIORTY

	Fmax	**Fmin**
Low priority	1.5	0.5
High priority	1.2	0.2

Phase 1: Prioritize Classes For Different Traffic Types

IoT networks can include different types of equipment with different priority levels. Some of these equipments can be critical like medical sensors and fire alarming system. In this proposal, data traffic is divided into two main categories as shown in figure 4: high priority traffic and low priority traffic. High priority traffic (HP) class is assigned to traffic which requires special QoS requirements like critical or important traffic. On the other hand, low priority traffic (LP) class can be assigned to other types of traffic.

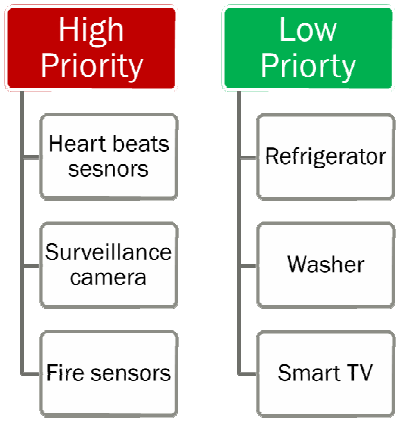

Fig. 4. Categories of IoT Devices

Phase 2: Adaptive Retransmission Time RTO

As we have illustrated earlier in CoCoA the value of RTO can be calculated using two estimators and the value of estimator can affect the overall RTO. In this research, the value of considering the weak estimator is related to traffic class as follows:

For High priority traffic the value of weak estimator has decreased effect on the calculation of the overall RTO where;

- $RTO_{new} = (0.25 * RTO_{weak} + 0.75 * RTO_{overall}) * Fmax$ where LossRate recent >LossRateavg
- $RTO_{new} = (0.25 * RTO_{weak} + 0.75 * RTO_{overall}) * Fmin$ where LossRate recent <LossRateavg

The value of ***NSTAT*** which specify the allowed number of parallel communication to a single destination is set to the maximum value which equals to 4. On the other hand, for the low priority, weak estimator can affect the overall RTO to provide higher throughput and sending opportunities for high priority traffic

- $RTO_{new} = (0.75 * RTO_{weak} + 0.25 * RTO_{overall}) * F_{max}$ where LossRate recent >LossRateavg
- $RTO_{new} = (0.75 * RTO_{weak} + 0.25 * RTO_{overall}) * Fmin$ where LossRate recent <LossRateavg

The value of ***NSTAT*** which specify the allowed number of parallel communication to a single destination is set to the minimum value which equals to 1.

Phase 3: Adaptive Backoff TIMER

Backoff timer used for default CoAP was mainly increased on exponential level. On next proposed mechanism the value of backoff time is mainly depends on the measured value of RTO for any type of traffic as follows:

$$Backoff\ Timer\ (BFT) = \begin{cases} 2, & 1 \le RTO_{init} \le 3s \\ 1.3, & RTO_{init} > 3s \end{cases}$$

The proposed mechanism has developed an adaptive back off time which mainly depends on data loss rate and traffic priority where shorted backoff timer has been assigned to high priority traffic and the backoff time is increased when the data loss is increased to overcome congestion.

The backoff time for high priority and low priority traffic can be calculated as follows with different values of increasing and decreasing factor:

- $BFT = BFT_{old} * Fmax$ where LossRate recent >LossRateavg
- $BFT = BFT_{old} * Fmin$ where LossRate recent <LossRateavg

Figure 5 illustrates the CoAP congestion adaptive mechanism for healthcare usage. The illustrated examples include a monitoring network for patients and patient's rooms. Sensors are divided into two main categories. High priority sensors including blood pressure sensor, heart beats sensor and body temperature sensor. Low priority sensors include room temperature sensor, room humidity sensor and room movement sensors. All data are collected by monitoring server through room gateway. High priority data has better chances to send data when congestion is occurred over low priority data.

VI. CONCLUSION

The main goal for IoT technology is to connect large number of constrained resources smart devices to each other and to the internet. These connected devices require a reliable communication mechanism, which can effectively handle data transfer and address the limitation of devices resources. CoAP protocol has been used as efficient protocol for handling application data but current proposed congestion mechanisms for this protocol have different issues, which degraded the data transfer performance. In this proposal, a new congestion

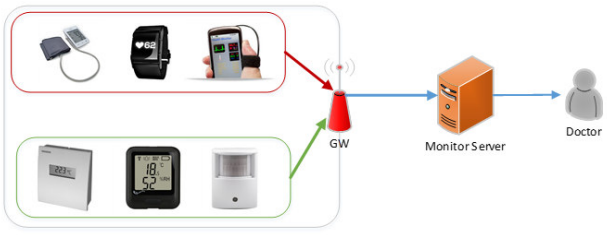

Fig. 5. Proposed Mechanism at healthcare environment

mechanism is proposed for CoAP protocol which can address the limitation of existing congestion mechanism by applying an adaptive mechanism which can act based on the condition of the network and the priority of the traffic being sent. IoT traffic includes different types of traffic which require

different handling and priority mechanism.Providing multiple priority classes for different IoT traffic types can be done as a future works. Optimized values of retransmission and backoff times can be calculated for multiple types of IoT traffic to enhance data transferring mechanism for all types of traffic.

ACKNOWLEDGMENT

The authors would like to acknowledge the assistance provided by the Network and Communication Technology Research Group, FTSM, UKM in providing facilities throughout the research. This project is supported under the Malaysian government -Fundamental Research Grant Scheme FRGS/1/2015/ICT03/UKM/02/2.

REFERENCES

[1] L. Da Xu, W. He, and S. Li, "Internet of things in industries: a survey," *Industrial Informatics, IEEE Transactions on,* vol. 10, pp. 2233-2243, 2014.

[2] I. Awan, M. Younas, and W. Naveed, "Modelling QoS in IoT Applications," in *Network-Based Information Systems (NBiS), 2014 17th International Conference on*, 2014, pp. 99-105.

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

[4] I. Jarvinen, L. Daniel, and M. Kojo, "Experimental evaluation of alternative congestion control algorithms for Constrained Application Protocol (CoAP)," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, 2015, pp. 453-458.

[5] N. H. A. Ismail and R. Hassan, "6LoWPAN local repair using bio inspired artificial bee colony routing protocol," *Procedia Technology,* vol. 11, pp. 281-287, 2013.

[6] R. Mietz, P. Abraham, and K. Römer, "High-level states with CoAP: Giving meaning to raw sensor values to support IoT applications," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, 2014, pp. 1-6.

[7] K. Hartke, "Observing resources in coap," *IETF - https://tools.ietf.org/html/draft-ietf-core-observe-05,* 2014.

[8] K. Hartke and C. Bormann, "Congestion Control Principles for CoAP," *IETF - https://tools.ietf.org/html/draft-bormann-core-congestion-control-02,* 2012.

[9] 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.

[10] C. Bormann, A. Betzler, C. Gomez, and I. Demirkol, "CoAP simple congestion control/advanced," *ID: draft-bormann-core-cocoa-02,* 2014.

[11] A. Betzler, C. Gomez, I. Demirkol, and M. Kovatsch, "Congestion Control for CoAP cloud services," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, 2014, pp. 1-6.

[12] A. P. Castellani, M. Rossi, and M. Zorzi, "Back pressure congestion control for CoAP/6LoWPAN networks," *Ad Hoc Networks,* vol. 18, pp. 71-84, 2014.

[13] N. H. A. Ismail, R. Hassan, and K. Wan Mohd Ghazali, "A study on protocol stack in 6lowpan model," *Journal of Theoretical and Applied Information Technology,* vol. 41, pp. 220-229, 2012.

[14] M. Poddar, R. Chaki, and D. Pal, "A channel trust based approach for congestion control in IoT," in *Application of Information and Communication Technologies (AICT), 2015 9th International Conference on*, 2015, pp. 319-324.

[15] 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.