4D-04

Window based congestion control for CoAP burst traffic

Archana K Rajan† Masaki Bandai†

†Department of Information and Communication Sciences, Sophia University

1. Introduction

Burst traffic is generated in Internet of Things (IoT) networks when sensors detect events. Sensors transmit event based data stream to a server for further analysis. Efficiency of the analysis improves with more packets reaching the server with less delay using a proper congestion control technique.

When constrained devices communicate in IoT networks, confirmable (CON) message type of Constrained Application Protocol (CoAP) [1] is generally used. However, for burst traffic, CON message ensures reliability at the cost of delay and overhead of acknowledgements. Non-confirmable (NON) message type of CoAP is a promising alternative. However, CoAP does not define any congestion control for NON messages, which we refer to as the default method.

We propose a congestion control method for NON messages to use in burst traffic to result in more packet delivery with less delay.

2. Proposed Method

We propose a window based congestion control technique inheriting the well-known additive increase multiplicative decrease (AIMD) policy. Window size defines the number of packets that the client can transmit at any given time. Congestion in the network is indicated by packet losses. Therefore, window size is modified with respect to the packet loss.

We generate sequentially increasing message IDs (MID) for each new packet in the client. Algorithm 1 explains the steps that each client executes. Window size, monitored by *window_Size,* starts with an initial value of 2 packets. After sending packets up to *window_Size,* the client checks if any response is received from the server. Absence of response indicates that all packets are received at the server and thus triggers the client to send more packets. In the constrained network, a careful increase of window size is preferred to prevent congestion. Therefore *window_Size* increases by 1.

Algorithm 1: Client Side

1: **Initialize**
2: *window_Size=2, MID= random ()*
3: **for** i=1 to *window_Size* **do**
4: *Send NON packet with MID*
5: *MID++*
8: **end for**
14: Check for response packet from the server
15: **if** no response packet, **then**
16: *window_Size+=1*
17: **else**
18: *window_Size/=2*
19: **end if**
20: **Go to** step 3

Response packet is an indication that the server identified a packet loss. Therefore, a traffic reduction from the client is favored to reduce congestion. As a result, *window_Size* is reduced by half. A multiplicative reduction of window size helps to recover from congestion quickly.

Algorithm 2: Server Side

1: **Initialize**
2: prev*_MID= 0*
3: **for** each packet received **do**
4: *Extract MID*
5: **if** *prev_MID+1 is not equal to MID*//Ignore 1st packet
6: Send response packet
7: **end if**
8: **end for**

Algorithm 2 shows the steps to be followed by the server for each client. Server saves the MID of the previous packet, *prev_MID,* to compare if the current packet is received in order. If MID is found to be missing, a response packet is sent to the client to notify the loss.

Window based congestion control for CoAP burst traffic
Archana K Rajan†, and Masaki Bandai†
†Department of Information and Communication Sciences, Sophia University

3. Evaluation Environment

We evaluated the performance of the proposed method against the default method and COCOA [2] using Contiki Cooja simulator. The default method permits unrestricted entry of NON packets into the network. COCOA introduces one CON packet after every eight NON packets. COCOA computes retransmission time out (RTO) based on RTT of CON packet. RTO from CON is used as traffic rate for NON.

We integrated COCOA into the Cooja simulator and modified it for NON burst traffic. A 3X3 grid topology with 1 border router (BR), 1 server(S) and 7 clients(C), is used for experiments, as shown in Fig. 1.

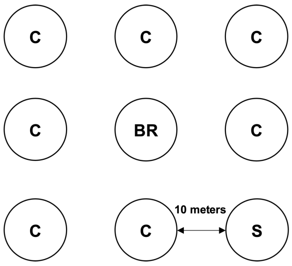

Figure 1. Grid Topology

Table 1 shows the simulation parameters that we used in the experiment. Radio duty cycling (RDC) of the MAC layer is turned off by using Null RDC driver.

Table 1. Simulation Parameters

Parameter	Value
Radio Medium	Unit Disk Graph
Radio Duty Cycling	Null RDC
MAC Driver	CSMA
Mote Type	T-mote Sky Zolerita Z1
Node transmission range	10m
Node Interference range	20 m
Routing protocol	RPL
Transmission/Reception ratio	100%

Border router initiates the RPL and sets up the destination oriented directed acyclic graph across the network. Thereafter, the client starts sending a sequence of CoAP packets to the server. Border router acts as a relay agent for CoAP packets.

Each client follows an ON-OFF pattern of packet transmission. A burst of 50 NON packets is sent during an ON period of 1 minute followed by an OFF period. Length of OFF period is randomly set up to 2 minutes. We conducted experiments for 15 minutes and repeated using 3 random seeds.

4. Performance Evaluation

We use packet delivery ratio (PDR), delay and MAC layer packet drops for performance evaluation. PDR is computed as the ratio of total messages received at server over total messages sent by the client. PDR represents network reliability as well as efficiency of data analysis at the server. Delay is the time taken for a message to reach from client to server. Delay is important when “freshness” of information is important. The packet drops at MAC layer can adversely affect data packets and control packets.

Table 2. Results

Metric	Default Method	COCOA	Proposed Method
PDR (in %)	62.0	80.2	91.3
Delay	1152 ms	720 ms	392 ms
MAC layer drops	519	112	202

Table 2 presents the average result we obtained. The default method transmits messages without traffic control. Consequently, congestion is experienced by the data packets and therefore exhibits poor performance in all metrics. Congestion control of COCOA is impacted by CON packet. RTO computation of CON packet considers retransmitted packet also and thereby deriving an imprecise RTO. Therefore, COCOA makes all metrics better than the default method. However, mandating CON packet in NON packet stream is not possible for all applications. Referring to Table 2, the proposed method exhibits more PDR with the least delay. The window size is adapted appropriately according to the loss identified by the server and displays desired performance for burst traffic. The proposed method overshadows the minor benefit of MAC layer drops, an effect of imprecise RTO, in COCOA with better PDR.

5. Conclusion

The proposed method unleashes the capacity of NON messages for burst traffic with a basic congestion control technique. The proposed method is the first of its kind, defining congestion control for NON messages by its response. A method to compute dynamic scaling factor for window is to be carried out as our future work.

References

[1] Shelby, Zach, Klaus Hartke, and Carsten Bormann. "RFC 7252: The constrained application protocol (CoAP)." (2014).

[2] Betzler, August, et al. "Experimental evaluation of congestion control for CoAP communications without end-to-end reliability." *Ad Hoc Networks* 52 (2016): 183-194.