CSCI 8873 - 3 week overview

Anas Alhadi

B00895875

February 11, 2025

1 Main Goal

To asses different modes of dynamic rule updating on the SCHC compression scheme. And their impact on device and bandwidth usage.

2 Network Structure

We assume a single connection between a SCHC device and a router/core device. All update decisions and calculations are made at the core and communicated back to the device.

3 Modes

3.1 Static Rules

3.1.1 Description

Rules are kept as generic as possible, with fields either being known by both device and core or explicitly sent (the only compression actions used are sent/not-sent)

3.1.2 Details

The core maintains a priority queue for some fields in it's existing set of rules. When sending/recieving a packet we keep note of the number of times a field takes a specific value. Once the counter crosses a specified cutoff value, we add a new rule with that value set.
We need to consider rule merging, and different cutoff values before adding new rules. This raises an important issue of having too many rules, in which we will need to include a lifetime for rules.

3.2 Map Updating

3.2.1 Description

No new rules are added, instead, variable fields always use the “Matching” compression action, in which we maintain a map of possible values and only transmit the index of the target value.

3.2.2 Details

A simple way of maintaing the “map” is by having it be a 2 level fixed size priority queue. Target values are ordered in the queue based on the number of times they are encountered. Values in the top queue are added to the map, while values in the second level queue remain monitored but not compressed. Every time an element is swapped in the top level queue it triggers a rule update (which is communicated to the device). The top level queue's size is fixed as it dictates the number of bits needed to represent the index of the element in the map. We test the mode's performance of different map sizes.

3.3 Field updating

3.3.1 Description

For field in which the set of target values follow a certain pattern or slightly differes (think of the options field in coap in which we access resources in the same path ”dir/file_1”) in this case (depending on the encoding) we can avoid the transmission of a set of the most significant bits, and instead only transmit the parts the vary.

3.3.2 Details

Similar to the previous mode, we keep a 2 level, fixed size, priority queues holding the most used target values. We then find the LCP of the values (so their most significant bits) and only transmit the LSB. (I need to verify that this is possible the openSCHC as i'm not sure if we have bit control on compression)

4 Short Term Steps:

4.1 Step 1

By the end of this week we need to have the network (device and core) setup, and prepare the sample set of random coap packets for the emulation.

4.2 Step 2

By the end of reading week, I need to have finished implementing mode 1 (Static rules) and start running the emulation on the sample packets. We record the number of times a rule change occurs and the size of the packet as a measure of performance

4.3 Step 3

By the end of the week after reading week, I need to have a working implementation of mode 2 (Map Updating)

4.4 Future steps

I still need to verify the feasibility of mode 3. further a combination of mode 1 and 2 in which we make use of the 2 level priority queue with 3 levels where level 1 implements Static rules, level 2 implements Mapping, and level 3 is uncompressed.