Algorithms

1 On-Board Unit Function

Algorithm 1 presents the steps performed by the OBU. As the paper shows, the OBU is responsible for (1) identifying and connecting to RSU (lines 1 to 6); (2) detecting the moment when an RSU is passed (line 9); (3) calculating, in real time, the mean speed on the road segment (line 10); (4) calculating the TTL for the current road segment (lines 11 to 17); (5) updating the local SCT (line 19); (6) sending the local SCT to the nearest RSU (line 20); (7) receiving the SCT from the RSU and updating all information about the road segments in the local SCT (lines 21 to 31); and (8) disconnecting (line 32). Finally, another function of the OBUs is to register existing obstacles on the road segment (line 29) in the local SCT.

More explanations about OBU functions can be found in the paper, in the Section III – C, named *On-Board Unit Function*.

2 Roadside Unit Function

Algorithm 2 presents the steps executed in the RSU. Again, as shown in the paper, when the RSU receives the SCT from the OBU (line 5), it needs to compare each line of the SCT received with the local SCT and, if necessary, update the information about other road segments (lines 7 to 44).

As mentioned in the paper, information about the road segments in the opposite direction must be known so that the TTL value can be estimated. Therefore, the TTL value is zero until the intersection of at least two OBUs (in opposite directions). At this point, the TTL can be calculated in at least half of the road segments. Before that, the vehicle assigns zero to the TTL value in all traveled road segments. In this case, when the TTL is zero, we consider updating the SCT entries of the RSU in all the road segments traveled by the OBU, including the parallel road segments. This mechanism is presented in Algorithm 2, on lines 8 to 26.

More explanations about RSU functions can be found in the paper, in the Section III – D, named *Roadside Unit Function*.

3 TTL Mechanism

At specific moments, due to events such as accidents, road maintenance, traffic of slow vehicles, among others, the mean speed on some road segments may become much lower. To prevent such situations, DOCTraMS records the highest TTL value already calculated for each of the road segments. If the new TTL value is lower than the maximum TTL registred for the road segment, the maximum TTL is maintained and sent to the OBU. In this case, only the mean speed of

Algorithm 1 Algorithm executed by an OBU.

```
Input: ID from RSU.
Output: Local SCT from the OBU.
1: total_RS ← ((total_RSU − 1) ∗ total_D)
2: while true do
3:     Search for known RSU_id
4:     if RSU is known then
5:         currentRSU ← RSU_id
6:         Try to connect
7:         if it is connected then
8:             while connected do
9:                 if OBU passed by RSU then
10:                    Calculate the mean speed
11:                    Analyze the SCT to calculate the variance and TTL
12:                    iteratorLines_SCT ← 1
13:                    while iteratorLines_SCT < total_RS do
14:                        if RS_id >= PRS_id and RS_id <= lastRS_opDir_id then
15:                            if currentCond < previousCond then
16:                                Calculates the variance of speed on the road segment
17:                            Calculates the TTL, adding the value of the variance
18:                        iteratorLines_SCT ← iteratorLines_SCT + 1
19:                    Update local SCT
20:                    Send the updated SCT to the associated RSU
21:                    Wait for an updated SCT
22:                    Receive the updated SCT from the RSU
23:                    iteratorLines_SCT ← 1
24:                    while iteratorLines_SCT < total_RSU do
25:                        currentCond_Local ← currentCond_Received
26:                        previousCond_Local ← previousCond_Received
27:                        TTL_Local ← TTL_Received
28:                        maxTTL_Local ← maxTTL_Received
29:                        obstLane_Local ← obstLane_Received
30:                        timer_Local ← timer_Received
31:                        iteratorLines_SCT ← iteratorLines_SCT + 1
32:                    Disconnect
33:                    previousAP ← currentAP
34:                    currentAP ← {}
35:            else
36:                Try to connect with the RSU
37:    else
38:        Search for known RSU_id
```

the road segment is updated. Otherwise, the new TTL value is maintained and the maximum TTL receives the new TTL value Algorithm. 2, lines 31 to 44).

More explanations about this mechanism can be found in the paper, in the Section III – B, named *TTL Mechanism*.

Algorithm 2 Algorithm executed in the RSU.

Input: SCT sent by the OBU.
Output: The overall traffic condition of all segments.

$total_{RS} \leftarrow ((total_{RSU} - 1) * total_{D})$
while true **do**
    Connect
    Wait for an updated SCT
    Receive the SCT from the OBU
    $iteratorLine_{SCT} \leftarrow 1$
    while $iteratorLine_{SCT} < total_{RS}$ **do**
        if $TTL_{Local} = 0$ **then**
            Updates all road segments traveled by the OBU (including parallel)
            if RS_{id} was traveled by the OBU **then**
                if $previousCond_{Local} = zero$ **then**
                    $previousCond_{temporary} = currentCond_{Received}$
                else
                    $previousCond_{temporary} = currentCond_{Local}$
                $currentCond_{Local} \leftarrow currentCond_{Received}$
                $previousCond_{Local} \leftarrow previousCond_{temporary}$
                $TTL_{Local} \leftarrow TTL_{Received}$
                if $RS_{id} = RS_{id_{curTraveledOBU}}$ **then**
                    $maxTTL_{Local} \leftarrow TTL_{Received}$
                else
                    $maxTTL_{Local} \leftarrow maxTTL_{Received}$
                $obstLane_{Local} \leftarrow obstLane_{Received}$
                $timer_{Local} \leftarrow timer_{Received}$
            else
                if RS_{id} is parallel to a road segment already traveled by the OBU **then**
                    Updates all entries in the local SCT of the RSU;
        else
            if $TTL_{Local} < TTL_{Receiv}$ **or** $RS_{id} = RS_{id_{curTraveledOBU}}$ **then**
                $previousCond_{temporary} = currentCond_{Local}$
                if $RS_{id} = RS_{id_{curTraveledOBU}}$ **then**
                    $currentCond_{Local} \leftarrow harmonicMean_{speedRS}$
                    $previousCond_{Local} \leftarrow previousCond_{temporary}$
                    if $TTL_{Received} < maxTTL_{Local}$ **then**
                        $TTL_{Local} \leftarrow maxTTL_{Local}$
                        $maxTTL_{Local} \leftarrow maxTTL_{Local}$
                    else
                        $TTL_{Local} \leftarrow TTL_{Received}$
                        $maxTTL_{Local} \leftarrow TTL_{Received}$
                    $obstLane_{Local} \leftarrow obstLane_{Received}$
                    $timer_{Local} \leftarrow timer_{Received}$
                else
                    Updates all entries in the local SCT of the RSU, assigning as previous condition
                    the current condition of each road segment ($previousCond_{temporary}$)
        $iteratorLine_{SCT} \leftarrow iteratorLine_{SCT} + 1$
    Send updated SCT to the OBU