

Consider the case where all vehicles request for the same data item, and the item can be transferred in one go from an RSU. The request deadlines are all  $T$  (end time).

We maintain a list of new caches, that is updated each time a vehicle is able to satisfy its request using the current allotment.

---

**Algorithm 1** Contact based ranking

---

```

1: for  $t = 0$  to  $T$  do
2:   for each RSU do
3:     for Each vehicle in RSU transfer list do
4:       if transfer complete then
5:         Remove vehicle from transfer list
6:         Decrement RSU load
7:       end if
8:     end for
9:     while RSU load  $\leq$  capacity  $\wedge$  more vehicles in range do
10:      Calculate score based on potential data transfer
11:      Rank in-range vehicles by scores
12:      Add highest ranked vehicle (say  $V_{best}$ ) to list of RSU transfers
13:      Increment RSU load
14:      Update the future storage of vehicles contacted by  $V_{best}$ 
15:      for each new cache  $V_{new}$  from V2V contacts do
16:        Update the future storage of vehicles contacted by  $V_{new}$ 
17:        if contacted vehicle  $V_{contact}$  becomes a cache then
18:          Add  $V_{contact}$  to list of new caches
19:        end if
20:      end for
21:    end while
22:  end for
23: end for

```

---



---

**Algorithm 2** Score Calculation

---

```

1: Given vehicle  $V$ , set score to 0
2: for each vehicle  $V_{contact}$  contacted by  $V$  do
3:   if  $V_{contact}$  can satisfy its request then
4:     Add  $w_{sat} * (Data\ transferred)$  to score
5:   else
6:     Add  $w_{nsat} * (Data\ transferred)$  to score
7:   end if
8: end for
9: return score

```

---