FDPS Reproduction Guide

I. FD Simultaneous Broadcast and Sensing

Algorithm 1 FD Simultaneous Broadcast and Sensing

1: **procedure** FD Simultaneous Broadcast and Sensing
2: **Input:** Environmental parameters; Eq. (1); Eq. (7) of [23]; parameter settings
3: **Output:** Sensing results (M_s); received signal energy E_1, E_2
4: s**tart:**
Initialise variables for relevant parameters
5: **loop:**
if CAV is not broadcasting
Substitute relevant parameters into Eq. (7) of [23].
Simulator calculates the inverse Q function, and returns the threshold, ε_1.
Simulator calculates the received signal energy, E_1.
if $E_1 > \varepsilon_1$
$M_s \leftarrow 1$ (RB is occupied)
else
$M_s \leftarrow 0$ (RB is free)
end if
else (CAV is broadcasting)
Substitute relevant parameters into Eq. (1).
Simulator calculates the inverse Q function, and returns the threshold, ε_2.
Simulator calculates the received signal energy, E_2.
if $E_2 > \varepsilon_2$
$M_s \leftarrow 1$ (Collision)
else
$M_s \leftarrow 0$ (Successful Transmission)
end if
end if
6: **end loop**
7: **return:** Sensing results (M_s); received signal energy E_1, E_2
8: **end procedure**

II. Internal Collision Resolution

FCFS algorithm is used for internal collision resolution.

Algorithm 2 Internal Collision Resolution (FCFS Algorithm)

1: **procedure** Internal Collision Resolution

2: **Input:** Broadcast Request B_{t_i}; Generation Instant G_{t_i}

3: **Output:** Broadcast List L_b

4: **start:**

Initialise variables for relevant parameters

5: **loop:**

for scheduling instant t_i

$L_b \leftarrow sort(B_{t_i}, G_{t_i})$

end for

end loop

6: **return:** Broadcast List L_b

7: **end procedure**

III. Resource Allocation/Reservation and Scheduling

Algorithm 3 Resource Allocation/Reservation and Scheduling

1: **procedure** Resource Allocation/Reservation and Scheduling
2: **Input:** Sensing results (M_s); received signal energy E_1, E_2
3: **Output:** Resource allocation/reservation and scheduling L_3
4: **start:**
Initialise variables to receive and store sensing results and energy
Pass sensing related arguments to corresponding variables
Initialise L_1 to tabulate available resources
5: **loop:**
for resource exclusion processes
switch
case 1: $M_s == 1$
$L_1 \leftarrow 0$ (not available resources)
case 2: $M_s == 0$ or $RSRP > \varepsilon$
$L_1 \leftarrow 1$ (candidate resources)
end switch
if $size(L_1) \geq 20\% \cdot size\,(M_s)$
$L_2 \leftarrow sort(L_1, RSSI)$
else
$\varepsilon \leftarrow \varepsilon + 3$
end if
end for
$L_3 \leftarrow$ random resources selection from L_2
end loop
6: **return:** Resource allocation/reservation and scheduling L_3
7: **end procedure**

IV. External Collision Resolution

Algorithm 4 External Collision Resolution

1: **procedure** External Resource Resolution
2: **Input:** Sensing results (M_s)
3: **Output:** Resource re-selection decision D_{RRD}
4: **start:**
Initialise variables to receive and store sensing results and energy
Pass sensing related arguments to corresponding variables
Initialise D_{RRD} to store resource re-selection decision
5: **loop:**
if $M_s == 1$
Initialise $RAC \leftarrow 0$
Transmission Abortion
if $RAC < RAC_{max}$
Re-broadcast
$D_{RRD} \leftarrow 0$
$RAC \leftarrow RAC + 1$
else
$D_{RRD} \leftarrow 1$
end if
else
$M_s == 0$ (Successful Transmission)
6: **return:** Resource re-selection decision D_{RRD}
7: **end procedure**

V. Average PDR/Latency/Collision Duration Analyser

Algorithm 5 Average PDR/Latency/Collision Duration Analyser

1: **procedure** Average PDR/Latency/Collision Duration Analyser
2: **Input:** Sensing results (M_s)
3: **Output:** PDR PDR; Latency Lat; Collision duration CD
4: **start:**
Initialise variables for relevant parameters
5: **loop:**
if succesful transmission
$PDR \leftarrow PDR + 1$
else
$PDR \leftarrow PDR$
end if
$Lat = t_2 - t_1$
$CD =$ transmission start instant t_{st} − transmission abortion instant t_{ab}
6: **return:** PDR PDR; Latency Lat; Collision duration CD
7: **end procedure**