Queue state globally updated only if the server node is chosen!

GLOBAL DEV SHEET

(soft queue spot reservation & hard queue spot reservation logic) ①

① (TX/RX available)

At the BS when capable

For a new request (r,θ):

LAGRANGIAN = $\frac{L}{\bar{R}_{GB}(r)}$ — BS-specific routine (NO FREQ REUSE)

SERVICE TIME = $\frac{L}{\bar{R}_{GB}(r)}$

WAIT TIME = $\min_{k\in\{1,2,\ldots,N_c\}}$ find_wait_time_BS (current queue state of channel k).

Share COST = LAGRANGIAN + WAIT TIME. Shimmering

Share SERVICE TIME + WAIT TIME as well

At an available UAV (obviously capable)

LAGRANGIAN = $\hat{f}(\underline{p}^*, \underline{v}^*)$ (eq. in page 21)

For new request (r,θ):

② WAIT TIME = $\arg\min_{k\in\{1,2,\ldots,N_c\}}$ find_wait_time_UAV (current queue state of channel k) — UAV-specific routine (DECODE REUSE ONLY)

(NOTE: Both decode & forward wait times are encapsulated in WAIT TIME)

SERVICE TIME = $\left(\sum_{m=0}^{M-1} \|v_m\|_2 / v_m\right) + (\hat{t}_{p,1} + \hat{t}_{p,2})$.

Share COST = LAGRANGIAN + WAIT TIME. Shimmering

Share SERVICE TIME + WAIT TIME as well

At an unavailable (already serving) **and capable** (transceiver available) **UAV**

For a new request (r,θ): ③

endpoint for current request = $\underline{x}'_M$

remaining trajectory = $(\underline{p}'^*, \underline{v}'^*)$ — includes current traj + traj of piggybacked requests (previously) (if LAG ≠ 0 for that request)

optimal trajectory for $(r,\theta) = (\underline{p}^*, \underline{v}^*)$ } collinearate — for freq reuse IR analysis

(curr ... cvr-1 ... cvr-3 ... next)

WAIT TIME = $\min_{k\in\{1,2,\ldots,N_c\}}$ find_wait_time_UAV (current queue state of channel k)

forward & decode channels available (for piggybacking) after WAIT TIME.

position after WAIT TIME = $\underline{x}'_m$.

If $\underline{x}'_m = \underline{x}'_M$: $L' = 0$, $L'' = 0$, LAGRANGIAN = $\hat{f}(\underline{p}^*, \underline{v}^*)$ (eq. in page 21)

If $\underline{x}'_m \neq \underline{x}'_M$: still serving current request

↳ $L' = \sum_{m=m'}^{M'} F_m$, m'' in the position where $L' = L$.

↳ if $L' - L \leq 0$: $L'' = 0$

LAGRANGIAN = $\hat{f}(\underline{p}^*, \underline{v}^*)$ with $h_1(\underline{p}^*, \underline{v}^*)$ modified to consider $L - L'$

$h_1(p^*, v^*) = L - L' - \sum_{m=0}^{\frac{M}{2}-1} F_m$

$h_2(p^*, v^*) = L - L'' - \sum_{m=\frac{M}{2}}^{M} F_m$

↳ if $L' - L > 0$: $L'' = \sum_{m=m''}^{M'} F_m$

→ If $L'' - L \geq 0$: LAGRANGIAN = 0

→ If $L'' - L < 0$: LAGRANGIAN = $\hat{f}(\underline{p}^*, \underline{v}^*)$ (with h_2 modified)

SERVICE TIME:
If LAGRANGIAN = 0: SERVICE TIME = 0.
Else: SERVICE TIME = $\sum_{m=0}^{M-1} \frac{\|v_m\|_2}{v_m} + \hat{t}_{p,1} + \hat{t}_{p,2}$

Share COST = LAGRANGIAN + WAIT TIME

Share SERVICE TIME + WAIT TIME Shimmering

(already serving a req. case) (no transceivers available) ④ current trajs → coordinate optimal routing → traj. reuse & R analyses

At an unavailable and incapable UAV:

$$\text{WAIT-TIME-1} = \min_{k \in \{1, 2, \ldots, N_c\}} \text{find-wait-time}_{UAV}(\text{current queue state of channel } k)$$

After WAIT TIME-1,

forward & decode channel available,
$$\text{WAIT TIME-2} = \min_{x \in \{1, 2, \ldots, N_x\}} \text{find-wait-time}_{UAV\ TxRx}(\text{current queue state of TxRx } x),$$

After WAIT TIME-2,

TxRx available too, same logic as case ③, find LAGRANGIAN.

Then, share COST = LAGRANGIAN + WAIT-TIME-1 + WAIT-TIME-2

also, have SERVICE TIME + WAIT-TIME-1 + WAIT-TIME-2 as well

find-wait-time$_{BS}$ (q_k):

NOREUSE

time-server = max (current req's being served)

(+)

time-qwait = ~~① get all reqs with one ahead of me and the zero in front of them~~ ~~• max serv time~~ • of all these concurrent ~~reqs~~

~~② if all in front of me are zero: add all their service times~~

③ [0][0][0 1][0][0 1 1 1 1][0][0][▨]
+ + max + max + + dist

combine

At the BS when not capable (no TxRx available): SERVICE TIME = LAGRANGIAN = $\frac{L}{\bar{R}_{QB}(r)}$, $\text{WAIT-TIME-1} = \min_{k \in \{1, 2, \ldots, N_c\}} \text{find-wait-time}(\text{queue state of channel } k)$

After WAIT-TIME-1: $\text{WAIT-TIME-2} = \min_{x \in \{1, 2, \ldots, N_{xBS}\}} \text{find-wait-time}(\text{current queue state of TxRx } x \text{ at the BS})$

Share LAGRANGIAN + WAIT-TIME-1 + WAIT-TIME-2

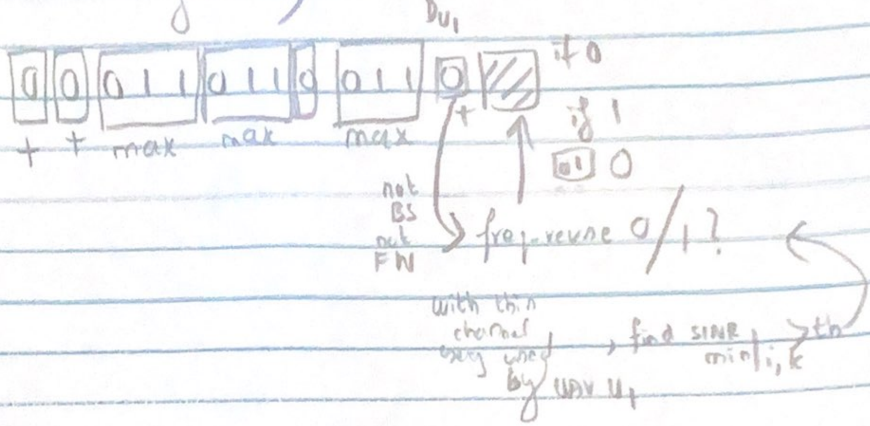
Share SERVICE TIME + WAIT-TIME-1 + WAIT-TIME-2

find_wait_time (q_k) :
UAV

time_server : max (current reqs being served)

return (+)

time_qwait :



---

find_wait_time (UAV TxRx x) :

~~Is this TxRx waiting for something ? Yes? add rem wait time~~
~~No? 0~~

Is this TxRx serving something ? Yes ? add rem SERVICE TIME + WAITTIME
(+) No ? 0
↑ include the wait times for that req as well

Are req ahead of me?
Yes? What is their SERVICE TIME + WAIT TIME ?
add all

No ? 0