Algorithm 1: Description of the SCL MAC decoder

Input: Received vector z^N, path vector b^{2N}, list L_{max}
Output: Decoded user bits $(\hat{u}^N, \hat{v}^N)$

$\mathcal{L} \leftarrow \{0\}, PM_0^{(0)} \leftarrow 0, i \leftarrow 0, j \leftarrow 0$;
for $k = 1, ..., 2N$ **do**
    calculate $P[\tilde{s}_k] \leftarrow W_N^{(b_k,i,j)}(z^N, \tilde{s}^{k-1}|\tilde{s}_k)$ from (11) and obtain $L_{2n}^{(k)}[l]$ from (19) for $\forall l \in \mathcal{L}$;
    calculate $PM_l^{(k)}$ from (20) according to $L_{2n}^{(k)}[l], \hat{\tilde{s}}_{k-1}[l], PM_l^{(k-1)}$ and duplicatePath(l) for $\forall l \in \mathcal{L}$;
    if $\tilde{s}_k$ *is frozen* **then**
        foreach $l \in \mathcal{L}$ **do**
            if $\hat{\tilde{s}}_k[l]$ *is not a frozen value* **then**
                kill the thread l and set $\mathcal{L} \leftarrow \mathcal{L} \setminus \{l\}$
    sorted by PM value in descending order for $\forall l \in \mathcal{L}$;
    if $|\mathcal{L}| > L_{max}$ **then**
        kill the thread l whose PM value is in the first $|\mathcal{L}| - L_{max}$ and set $\mathcal{L} \leftarrow \mathcal{L} \setminus \{l\}$ for $\forall l \in \mathcal{L}$;
$l^* \leftarrow \arg\min_{l \in \mathcal{L}} PM_l^{(2N)}$;
return $(\hat{u}^N[l^*], \hat{v}^N[l^*])$;
subroutine duplicatePath(l);
Copy the thread l into a new thread $l' \notin \mathcal{L}$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{l'\}$;
$(\hat{\tilde{s}}_k[l], PM_l^{(k)}) \leftarrow (0, P[0])$;
$(\hat{\tilde{s}}_k[l'], PM_{l'}^{(k)}) \leftarrow (1, P[1])$;