Algorithm Description for the Paper Published in Proc. IEEE WCNC 2020

Bangzhao Zhai, Mengxin Yu, Aimin Tang, and Xudong Wang
Shanghai Jiao Tong University, Shanghai 200240, China
Email: {bangzhao, catherineyu7, tangaiming, wxudong}@sjtu.edu.cn

This report shows the details of algorithms used in [1].

Algorithm for mesh IAB

```
Input the traffic demand $d_i$ of each bi-directional link $l_i^{(b)}$, $i = 1, \dots, Blink\_num$;
Input the node set $N$, the neighbor list $N(j)$ of each node $n_j, j = 1, \dots, node\_num$;
Initialize $t = 0$, the number of slots in a central scheduling period $Nslot\_Per\_Period$
while $t < Nslot\_Per\_Period$
    $t = t + 1$;
    Set the link set $L^t = \emptyset$, master node set $M^t = \emptyset$, slave node set $S^t = \emptyset$, and $k = 0$;
    Sort all links in a descend order of  $d_i$, and put the sorted links into a set $H^{(b)}$;
    while $k < Blink\_num$
        $k = k + 1$;
        Get the $k^{th}$ link $l_k^{(b)}$ in $H^{(b)}$;
        For  nodes in both sides of the link, denote the node with a larger traffic demand as
        $V_k^{(L)}$, and the other one $V_k^{(S)}$;
        if ($\forall n_i \in N(V_k^{(L)}) - \{V_k^{(S)}\}, n_i \notin S^t$) & ($\forall n_j \in N(V_k^{(S)}) - \{V_k^{(L)}\}, n_j \notin M^t$ ) then
            $L^t = L^t \cup l_k^{(b)}$; $M^t = M^t \cup V_k^{(L)}$; $S^t = S^t \cup V_k^{(S)}$;
        elseif ($\forall n_i \in N(V_k^{(S)}) - \{V_k^{(L)}\}, n_i \notin S^t$)  & ($\forall n_j \in N(V_k^{(L)}) - \{V_k^{(S)}\}, n_j \notin M^t$ ) then
            $L^t = L^t \cup l_k^{(b)}$; $M^t = M^t \cup V_k^{(S)}$; $S^t = S^t \cup V_k^{(L)}$;
        endif
    end while
    Calculate the non-available node set $E^t = N - M^t - S^t$;
    Update the estimated traffic demand $d_i$, $i = 1, \dots, Blink\_num$;
end while
Output the role pattern (M,S,NA) of each node
```

Algorithm for DAG IAB

```
Input the traffic demand $d_i$ and conflict link set $C_i$ of each directional link $l_i^{(d)}, i = 1, \ldots, Dlink\_num$;
Input the node set $N$;
Initialize $t = 0$, the number of slots in a central scheduling period $Nslot\_Per\_Period$
while $t < Nslot\_Per\_Period$
    $t = t + 1$;
    Set the link set $L^t = \emptyset$, and $k = 0$;
    Sort all links in a descend order of $d_i$, and put the sorted links into a set $H^{(d)}$;
    while $k < Dlink\_num$
        $k = k + 1$;
        Get the $k^{th}$ link $l_k^{(d)}$ in $H^{(d)}$;
        if ($\forall l_i^{(d)} \in L^t, l_k^{(d)} \notin C_i$) then
            $L^t = L^t \cup l_k^{(d)}$;
        endif
    end while
    Update the estimated traffic demand $d_i, i = 1, \ldots, Blink\_num$;
end while
Output the type pattern (H-D, H-U, NA) of each link
```

[1] B. Zhai, M. Yu, A. Tang, and X. Wang, "Mesh Architecture for Efficient Integrated Access and Backhaul Networking," in Proc. IEEE WCNC 2020.