Multipath Routing for Network Functions Virtualization

minhpt@tlu.edu.vn...

Algorithm 1 AlphaBeta Multipath

Input: NFVI $G=(V,E,B,C)$, D, α, β, n_p
Output: Optimal rate allocation and routing paths

Initialization: $G^r=(V,E,B^r,C^r)$;
    $B^r=\{b_e^r : b_e^r=b_e, \forall e\}$; $C^r=\{c_v^r : c_v^r=c_v, \forall v\}$;
    $y_d \leftarrow y_d^l$, $\forall d$; $U^* \leftarrow 0$;
    $lb \leftarrow \min_d y_d$; $ub \leftarrow f_{\max}/|D|$;
while $lb \le ub$ **do**
    Sort D in descending order of y_d
    for all $d \in D$ **do**
        $(Q_d^n, Q_d^m, \delta_d^n, \delta_d^m) = \mathrm{PATHRATE}(G^r, d, n_p)$
        $N_d^n = \mathrm{NODE}(Q_d^n, \delta_d^n)$
        $N_d^m = \mathrm{NODE}(Q_d^m, \delta_d^m)$
    end for
    Compute $F_n = F\left(Q_d^n, \delta_d^n, N_d^n\right)$ and
        $F_m = F\left(Q_d^m, \delta_d^m, N_d^m\right)$ according to the
        objective function
    if $F^* \le F_n$ **or** $F^* \le F_m$ **then**
        Update F^* and the better solution
        $lb \leftarrow \min_{d\in D} y_d$
        $y_d \leftarrow \max\{y_{dl}, (ub+lb)/2\}$, $\forall d$
    else
        $ub \leftarrow \min_{d\in D} y_d$
        $y_d \leftarrow \min\{y_d^u, (ub+lb)/2\}$, $\forall d$
    end if
end while

Algorithm 2 Find the best candidates for path and rate allocation to a demand

function PATHRATE(G^r, d, n_p)
    Initialization: Let Q denote the set of all shortest
        paths from s_d to t_d; Path $q \in Q$ consists of a
        sequence of nodes V_q, and a sequence of links
        E_q; $b_q^r \leftarrow \min_{e\in E_q} b_e^r$; $c_q^r \leftarrow \min_{v\in V_q} c_v^r$;
        $Q_d^m \leftarrow \emptyset$;
    $a_q^r \leftarrow \min\left\{b_q^r, c_q^r\right\}$ ▷ Path capacity
    $Q_d^n \leftarrow n_p$ largest paths in Q
    $\pi = y_d \Big/ \sum_{q\in Q_d^n} a_q^r$
    if $\pi \le 1$ **then**
        $\delta_n(q) \leftarrow \pi a_q^r$, $\forall q \in Q_d^n$
    else
        $Q_d^n \leftarrow \emptyset$
    end if
    while $|Q_d^m| \le n_p$ **and** $y_d \ge 0$ **and** $Q \ne \emptyset$ **do**
        Scan Q for smallest path q with $a_q^r \ge y_d$
        if <path not found> **then**
            Find the largest path q in Q
            Update path and rate allocation:
                $Q_d^m \leftarrow Q_d^m \cup \{q\}$; $\delta_d^m(q) \leftarrow a_q^r$;
            Update remaining resources and demand:
                $b_e^r \leftarrow b_e^r - a_q^r, \forall e \in E_q$; $Q \leftarrow Q \setminus \{q\}$;
                $y_d \leftarrow y_d - a_q^r$
        else
            Update path and rate allocation:
                $Q_d^m \leftarrow Q_d^m \cup \{q\}$; $\delta_d^m(q) \leftarrow y_d$;
            Update remaining resources and demand:
                $b_e^r \leftarrow b_e^r - y_d, \forall e \in E_q$; $y_d \leftarrow 0$;
        end if
    end while
    return Q_d^n, Q_d^m, $\delta_d^n(q)$, $\delta_d^m(q)$
end function

***Abstract*—Multipath Routing for Network Functions Virtualization...**

I. Introduction

II. Problem Formulation

III. Heuristic

Graph:

AA

IV. Evaluation

A. Dataset and setup

Generated topologies: use the Georgia Tech [**?**] and BRITE [**?**] topology generators

Actual ISP topology: Abilene (Internet2) dataset [**?**] and Geant dataset [**?**]

B. Convergence speed

C. Throughput

D. End-to-end latency

E. Routing overhead

V. Discussion

A. Implementation

VI. Conclusion

Algorithm 3 Find the assignment of nodes for a SFC on a path allocated to demand

function NODE(G^r, d, q, δ)
  for all $f_i \in F_d$, $i = 1 \dots |F_d|$ **do**
    $v_i \leftarrow$ the first node v on path q with $c_v^r \geq r_{vf_i}(\delta)$
    $N_d^n \leftarrow N_d^n \cup \{v_i\}$
  end for
  Return failure if solution not found
  $v_{|F_d|+1} \leftarrow t_d$; $N_d^n \leftarrow v_{|F_d|+1}$;
  while node found **do**
    for all $f_i \in F_d$, $i = 1 \dots |F_d|$ **do**
      Find the smallest node v with $c_v^r \geq r_{vf_i}(\delta)$ along path q from v_i to v_{i+1}
      Update v_i if node found
    end for
  end while
  return N_d^n
end function