VNF Placement Algorithm

April 3, 2025

Algorithm 1: Improved VNF Placement Algorithm

```
Input: List of worker nodes and optional weights
Input: Prometheus endpoint, thresholds $T_{CPU}, T_{MEM}$
Output: Best node $n^*$ or nil if no suitable nodes found
Function GetBestNode():
    $bestScore \leftarrow -1$;
    $n^* \leftarrow$ nil;
    $validNodes \leftarrow \emptyset$;
    forall nodes $n$ in cluster do
        Query $CPU_n$, $MEM_n$ from Prometheus;
        if $CPU_n > T_{CPU}$ or $MEM_n > T_{MEM}$ then
            continue;
            ;                                   /* Skip overloaded node */
        Add $n$ to $validNodes$;
        Query from Prometheus: $TX_n$, $RX_n$, $TXDrop_n$, $RXErr_n$, $OVSUtil_n$;
        Compute:
            $\text{TXDropRate}_n = \frac{TXDrop_n}{TX_n}$
            $\text{RXErrorRate}_n = \frac{RXErr_n}{RX_n}$
            $\text{TrafficUtil}_n = \frac{TX_n + RX_n}{\text{MaxCapacity}}$
            $\text{OVSUtil}_n$ from OvS byte rate;
        Compute $score_n$ using Equation 1;
        if $score_n > bestScore$ then
            $bestScore \leftarrow score_n$;
            $n^* \leftarrow n$;
    if $validNodes = \emptyset$ then
        return null;
        ;                                       /* No valid nodes available */
    else
        return $n^*$;
```

$$\text{score}_n = w_1 \cdot (1 - \text{TXDropRate}_n) + w_2 \cdot (1 - \text{RXErrorRate}_n) + w_3 \cdot (1 - \text{TrafficUtil}_n) + w_4 \cdot (1 - \text{OVSUtil}_n) \quad (1)$$