## 1 Node Resources

The current load of substrate node $i \in N^S$ is calculated as:

$$\Delta(n_i^S) = \sum_{\forall n_j^V \in N^V} \gamma_{ij} \cdot \varsigma_j \tag{1}$$

The available processing capacity $r_i^{proc}$ of substrate node $i \in N^S$ is calculated as:

$$r_i^{proc} = T_i^{proc} - \Delta(n_i^S) \tag{2}$$

such that: $\gamma_{ij} = \begin{cases} 1, & \text{if node } n_i^S | n_i^S \in N^S \text{ is allocated to virtual node } n_j^V | n_j^V \in N^V \\ 0, & \text{otherwise} \end{cases}$

$\varsigma$ is the processing quota allocated to virtual node $j$

$T_i^{proc}$ is the maximum capacity of node $n_i^S$

(total active processing time)

## 2 Radio Resources

The current load of substrate link $i \in E^S$ is calculated as:

$$\Delta(e_i^S) = \sum_{\forall e_l^V \in E^V} \beta_{il} \cdot \upsilon_l \tag{3}$$

The remaining link capacity $r_i^{radio}$ of substrate link $i \in E^S$ is calculated as:

$$r_i^{radio} = T_i^{radio} - \Delta(e_i^S) \tag{4}$$

such that: $\beta_{il} = \begin{cases} 1, & \text{if link } e_i^S | e_i^S \in E^S \text{ is allocated to virtual link } e_l^V | e_j^V \in E^V \\ 0, & \text{otherwise} \end{cases}$

$\upsilon$ is the link quota allocated to virtual link $e_l^V | e_l^V \in E^V$

$T_i^{radio}$ is the maximum capacity of link $e_i^S$

(total active transceiving time)

## 3 Embedding Cost Function

The cost of embedding can be calculated as the weighted sum of the assigned physical resources:

$$C(G^V) = \sum_{\forall n_i^S \in N^S} \sum_{\forall n_k^V \in N^V} \beta_{ik} \cdot \pi_k + \sum_{\forall e_j^S \in E^S} \sum_{\forall e_m^V \in E^V} \gamma_{jm} \cdot \mu_m \tag{5}$$

The objective function is :

$$minimize \;\; C(G^V) \tag{6}$$

Such that:

$$\gamma jm = \begin{cases} 1, & \text{if } e_j^S \text{ is allocated to virtual edge } \ e_m^V \\ 0, & \text{otherwise} \end{cases}$$

$\mu_m$ is the weight of embedding onto $e_j^S$

and:

$$\beta ik = \begin{cases} 1, & \text{if } n_i^S \text{ is allocated to virtual node } \ n_k^V \\ 0, & \text{otherwise} \end{cases}$$

$\pi_k$ is the weight of embedding onto $n_i^S$

## 4 Link Weight

$\pi_g$ is the weight of an edge $e^S \in G^S$ is defined as a composite metric and calculated using EIGRP's formula:

$$W(e^S) = 256 \cdot ((K1 \cdot BW) + (K2{\cdot}BW)/(256{-}Load){+}(K3{\cdot}Delay){\cdot}(K5/(Reliability{+}K4))) \tag{7}$$

Such that:

- K1(Bandwidth)={1,0}  calculated as $10^7/BW$
- K2(Load)={0,1}
- K3(Delay)={1,0}
- K4(Reliability)={0,1}
- K5(Reliability)={0,1}

## 5 Preprocessing:

- *Generate the Augmented Substrate Graph (ASG) by infusing the network model (NIB), the static location information and the resource state information (RIB)*
- *Generate conflict graphs based on interference*
- *Generate shortest path / minimal spanning tree*

## 6 VNE Algorithm

**Algorithm 1** Mapping Algorithm

**Input:** AUGMENTED SUBSTRATE GRAPH; CONFLICT GRAPH; VNRs
**Output:** MAPPING RELATIONS $\forall$ ACCEPTED VNRs

```
EVENT: VNRs Submitted

Initialize:
$G^S \leftarrow$ Augmented Substrate Graph;
$VNR \leftarrow$ Set of Virtual Network Requests;
$M \leftarrow$ null;

while $\exists$ unprocessed $VNR$ and sufficient resources do
  for all $VNR[i]$ do
    for all $n_j^V \in VNR[i]$ do
      find candidate based on location constraint;
      if $n_j^V.location = n_j^S.location$ then
        if $n_j^V.resource = n_j^S.resource$ then
          map $n_j^S \rightarrow n_j^V$ and add $n_j^V$ to $N_i^V$;
        else
          reject $VNR[i]$;
          break
        end if
      else
        reject $VNR[i]$;
        break
      end if
    end for
    add $N_i^V$ to $G_i^V$
    find all shortest paths $P[]$, $\forall n_\alpha^V \in G_i^V \iff n_\Omega^S \in G^S$
    Sort $P[]$
    $P_l = min(P)$
    for all $n_j^V \in N_i^V$ do
      if $\exists$ an edge $e_k^S \in G^S$ in $P_l$ between $(n_\Omega^S,\ n_{\Omega+1}^S) \Rightarrow n_{\alpha l}^S \in G_i^V =$ true, and $e_k^S.resource \geq required.resource$ and $(n_\Omega^S,\ n_{\Omega+1}^S)$ not interfering then
        map $e_k^S \in G^S \rightarrow e_k^V$ and add $e_k^V$ to $E_i^V$;
        traverse on path
      else
        backtrack;
      end if
    end for
  end for
  add $E_i^V$ to $G_i^V$
  if all constraints are satisfied then
    commit embedding
    populate $M$ with mapping related information
    update $G^S$
  end if
end while
```

### 6.1 Constraints

The goal is to satisfy the resource demands of VNRs while not violating the capacity constraints of the substrate network.

#### 6.1.1 Capacity Constraints

#### 6.1.2 Demand Satisfaction Constraints

#### 6.1.3 Flow Conservation Constraints

#### 6.1.4 Precedence Constraints

Task/step $i$ has to be performed before task/step $j$. (sorting of resources, finding connectivity paths before mapping/assignment)

#### 6.1.5 Conditional Constraints

## 7 Objective

To overcome the limitations of dedicated single user WSN deployments by enabling physical infrastructures to be shared among several concurrently coexisting users. By means of virtualisation technologies a common physical substrate can be used for multiple purposes beyond the scope of the original deployment, leading to reduced deployment and maintenance cost as well as to more efficient overall resource utilization.

Such networks allow the logical separation of geographically overlapping virtual networks that may share subsets of substrate resources.

An optimal solution is preferred but finding a feasible assignment is the primary goal.

## 8 Challenges

The primary challenge is to efficiently allocate substrate resources to VNs.

### 8.1 Node Level Virtualisation

An abstraction layer is required to ensure the independent execution of virtual instances.

- Virtual Machines
- Middleware

### 8.2 Network Level Virtualisation

Typical approaches utilize multiple access mechanisms (SDMA, TDMA, FDMA)

- Multiple radios
- Overlay Networks
- Non deterministic broadcast nature of the wireless medium

- Isolation - (prevent the different VWSNs from adversely impact one another)
- Interference
- Multiway separator problem
- Minimum cut linear arrangement problem
- Minimum degree graph partitioning problem
- Multi commodity flow problem