Dual Load Balancing Strategy for Virtual Network Embedding in SDN-Enabled Distributed Cloud

Chinmay Sharma T
Aparna Singh
Guruprasad A

WHAT IS VNE?

VNE is the process of mapping and placing virtual networks onto the underlying physical network infrastructure.

Request: VNE typically starts with a request for a virtual network, specifying its requirements (like bandwidth, latency, etc.).

Mapping: The SDN (Software-Defined Networking) controller or a network management system analyzes the physical network's current state and attempts to map the requested virtual network onto the available physical resources.

ADVANTAGES

RESOURCE OPTIMIZATION

Efficiently allocate and utilize physical network resources to meet virtual network demands.

LATENCY MINIMIZATION

Prioritize reducing data transmission delays for responsive virtual network communication.

QOS COMPLIANCE

Meet defined Quality of Service (QoS) criteria for reliability, bandwidth, and performance metrics.

FORMULATION

APPLICATION PLANE

Receives and processes Virtual Network Requests (VNRs) through the SDN broker, determining acceptance based on policies and network conditions

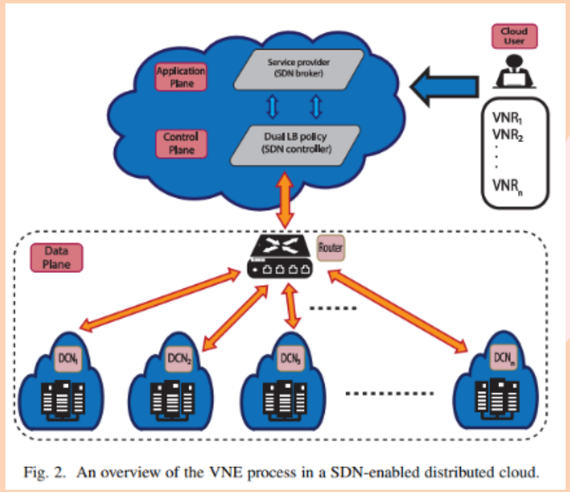


Fig. 2. An overview of the VNE process in a SDN-enabled distributed cloud.

CONTROL PLANE

Central to SDN, manages network intelligence and policy-based management via the SDN controller, automating issue resolution.

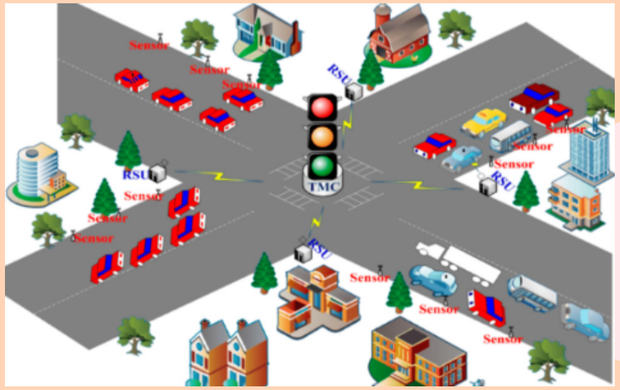
DATA PLANE:

The infrastructure layer housing physical network equipment, responsible for data forwarding guided by decisions from the SDN controller.

REAL WORLD ANALOGY: TRAFFIC MANAGEMENT

APPLICATION PLANE (TRAFFIC CONTROL)

Like a Traffic Control Center, it processes Virtual Network Requests (VNRs) analogous to managing traffic conditions



CONTROL PLANE (TRAFFIC OFFICERS)

Acts as Traffic Officers implementing decisions, akin to how the SDN Controller configures network devices

DATA PLANE (VEHICLES AND ROADS)

Analogous to actual vehicles and roads, it's the infrastructure executing decisions from the Traffic Control and actions of Traffic Officers.

The authors of this paper used a dual load balancing strategy to, that dual load policy is divided into two parts:

A.VNR ASSIGNMENT

- The **SDN Broker** analyzes all the VNRs received by comparing total DCN capacity (which is calculated in terms of total SNODs and DCN bandwidth) with the total VNR requirement (which is calculated in terms of total VNODs and VNR bandwidth) it should be **VNR <= max(All DCNs capacity) to VNR to get accepted**

$$DCN^c = \sum_{i=1}^{n} SNod^c + DCN^{bw}$$

$$VNR^r = \sum_{i=1}^{n} VNod^r + VNR^{bw}$$

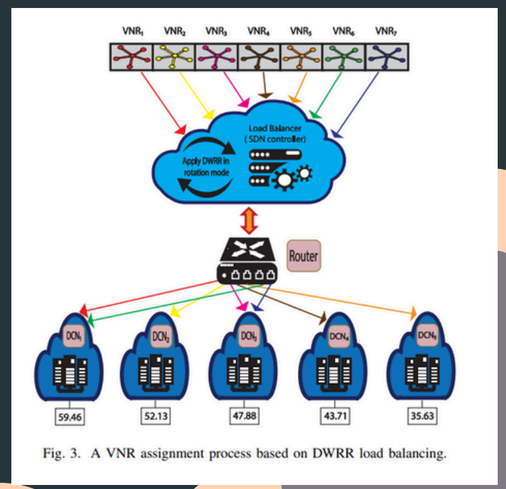


Fig. 3. A VNR assignment process based on DWRR load balancing.

- Now, SDN Broker establishes a DWRR policy for VNR assignment, which is known as the importance value for each DCN, Here **SDN controller** acts as a Load Balancer where VNR are assigned to the DCNs in rotation mode considering DWRR value

B.VNR MAPPING

A greedy technique for load balancing in a DCN with a focus on **minimizing stress on SNods compared to the overall hosted load.**

The authors have put up a **Resource-Constrained (RC)** policy that classifies SNods into **underloaded(>0.7), balanced(>0.4 & <0.7) , and overloaded(<0.4) categories based on a threshold ratio(0.7 & 0.4).** The embedding process selects servers with the highest **residual capacity ratio (θ)** for VNod placement, following to constraints on SNod and link capacities.

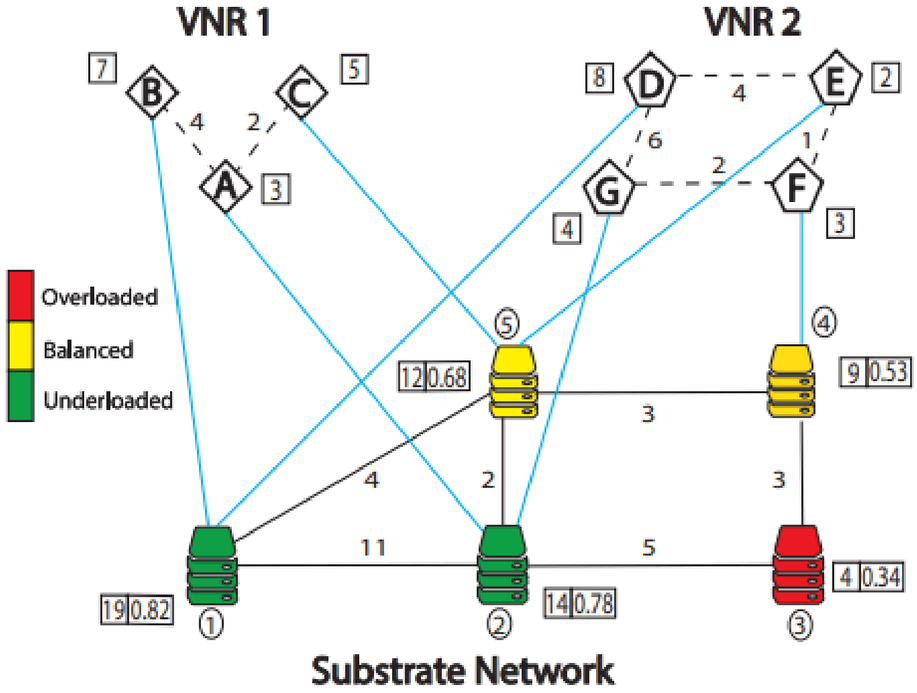
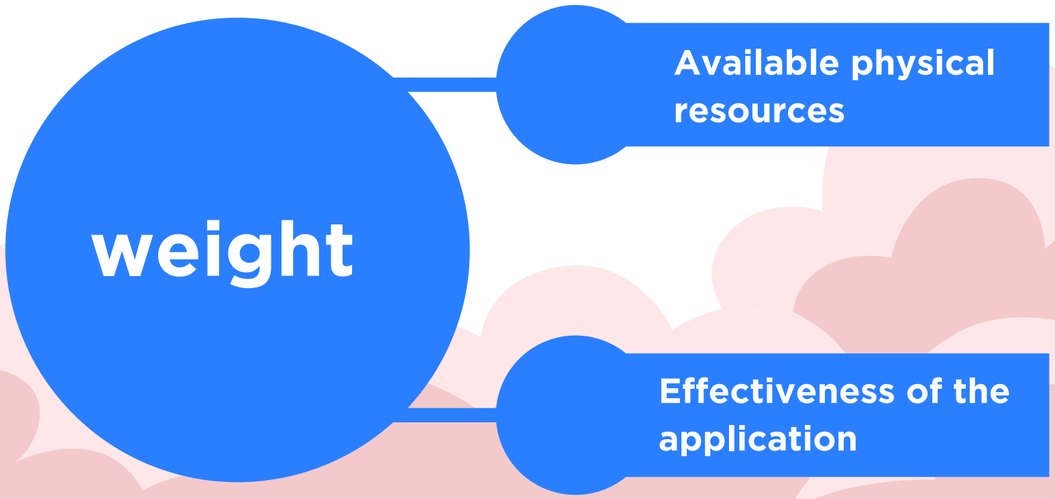


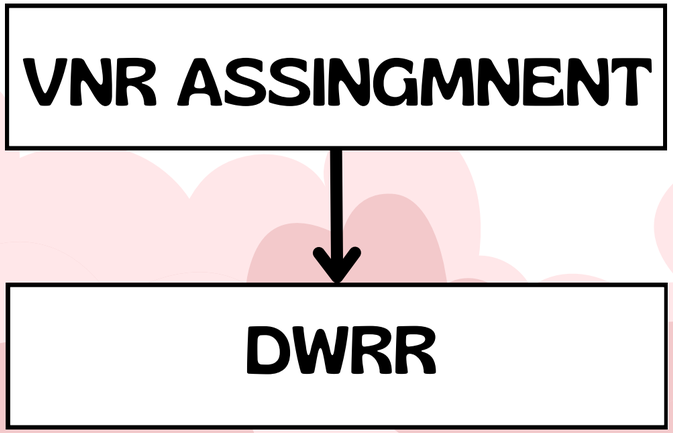
Fig. 4. A VNR mapping process based on a RC load balancing.

DYNAMIC WEIGHTED ROUND ROBIN (DWRR) LOAD BALANCING ALGORITHM

- Each DCN’s physical resource has an assigned weight.

- Based on the weights assigned to the physical resources, the DWRR algorithm chooses one for the VN.

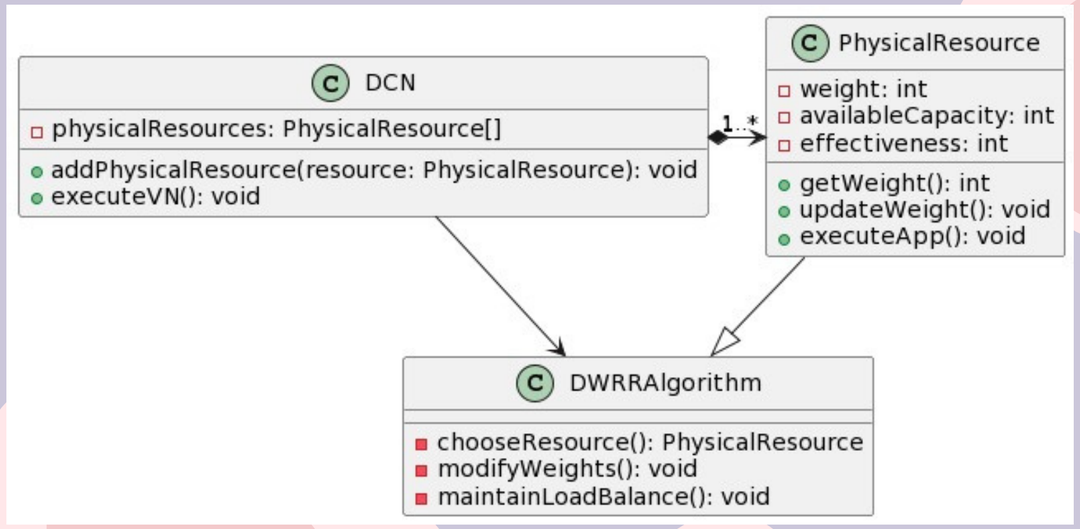




Firstly choice is on the basis of physical resources having the largest weight.

DWRR then modifies the physical resource weight according to the load that is applied to each one at any given time.

In order to maintain load balance and an equal distribution of VNs among the DCN's physical resources, the DWRR algorithm is employed



IMPLEMENTATION

CloudSim SDN represents an enhanced iteration of CloudSim, a Java-based toolkit utilized for simulating network configurations within cloud data centers enabled with Software-Defined Networking (SDN).

VNRs : queue(FIFO)
DCNs' physical weight: assigned on basis of DWRR and kept in priority queue (largest weight is always on the top of priority queue).

PSEUDO CODE

```
class WeightComparator {
public:
 bool operator()(const PhysicalResource& a, const PhysicalResource& b) {
 return a.weight < b.weight;
 }
};

void assignWeightsToVNR(VNRRequest& vnr, std::priority_queue<PhysicalResource, std::vector<PhysicalResource>, WeightComparator>&
physicalResources) {
 // Implementation of assigning weights based on DWRR algorithm.
 // Update the state of the physical resource after every iteration.
}

void processVNRRequests(std::queue<VNRRequest>& vnrQueue, std::priority_queue<PhysicalResource, std::vector<PhysicalResource>,
WeightComparator>& physicalResourcesQueue) {
 while (!vnrQueue.empty()) {
 VNRRequest vnr = vnrQueue.front(); // Dequeue VNR request from the FIFO queue
 vnrQueue.pop();
 assignWeightsToVNR(vnr, physicalResourcesQueue);
 // Process the VNR request and assign weights based on DWRR algorithm.
 // Update the physical resource's weight through DWRR, based on the VNRRequest VNR.
 }
//After all the VNR Request have been fulfilled, the final state of the substrate network would be returned.
}
```

THANKYOU