Multi-Cloud Distribution of Virtual Functions and Dynamic Service Deployment: OpenADN Perspective

Deval Bhamare, Qatar University
Raj Jain, Washington University in St. Louis
Mohammed Samaka, Qatar University
Gabor Vaszkun, Washington University in St. Louis
Aiman Erbad, Qatar University

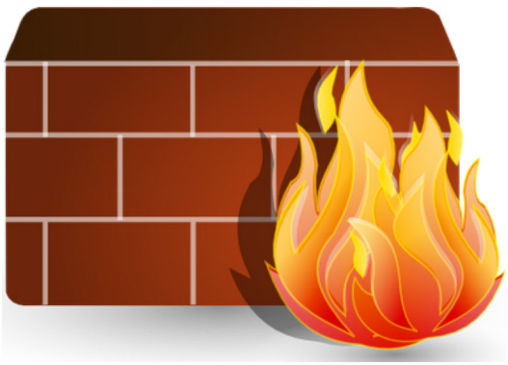
2nd IEEE International workshop on Software Defined Systems
Tempe, AZ, 9th March, 2015

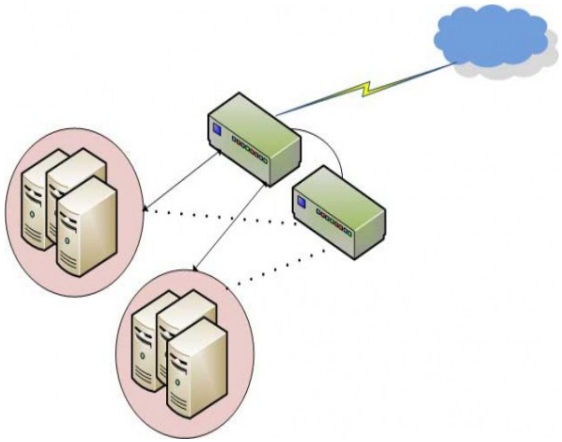
These slides and audio/video recordings of this talk are at:
http://www.cse.wustl.edu/~jain/talks/vm_distp.htm

1. Virtual Functions
2. Optimal and non-optimal location
3. Integer Linear Programming Formulation
4. Minimum Residue Algorithm
5. Comparison with Max-Min Algorithm

Virtual Functions

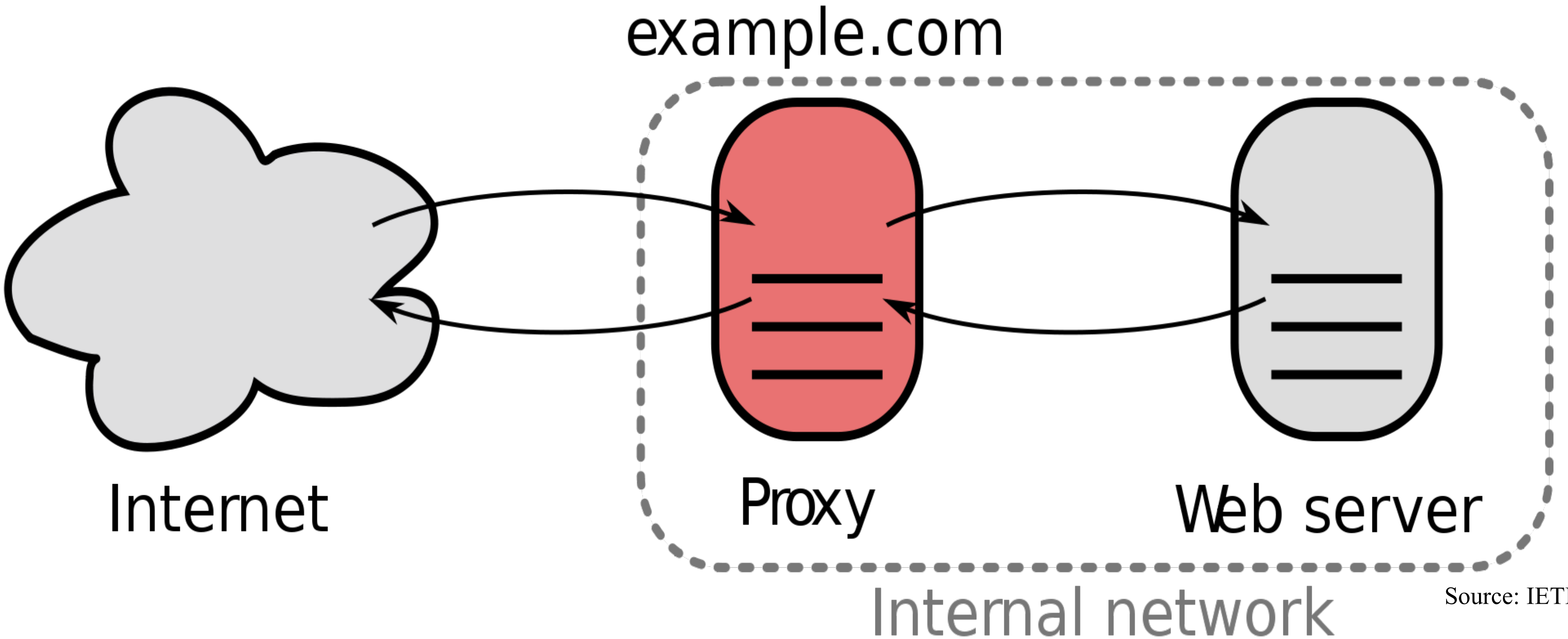
- **Virtual Functions:** Distributed entities: e.g., Web server, load balancers, data base servers, Proxy servers and many others





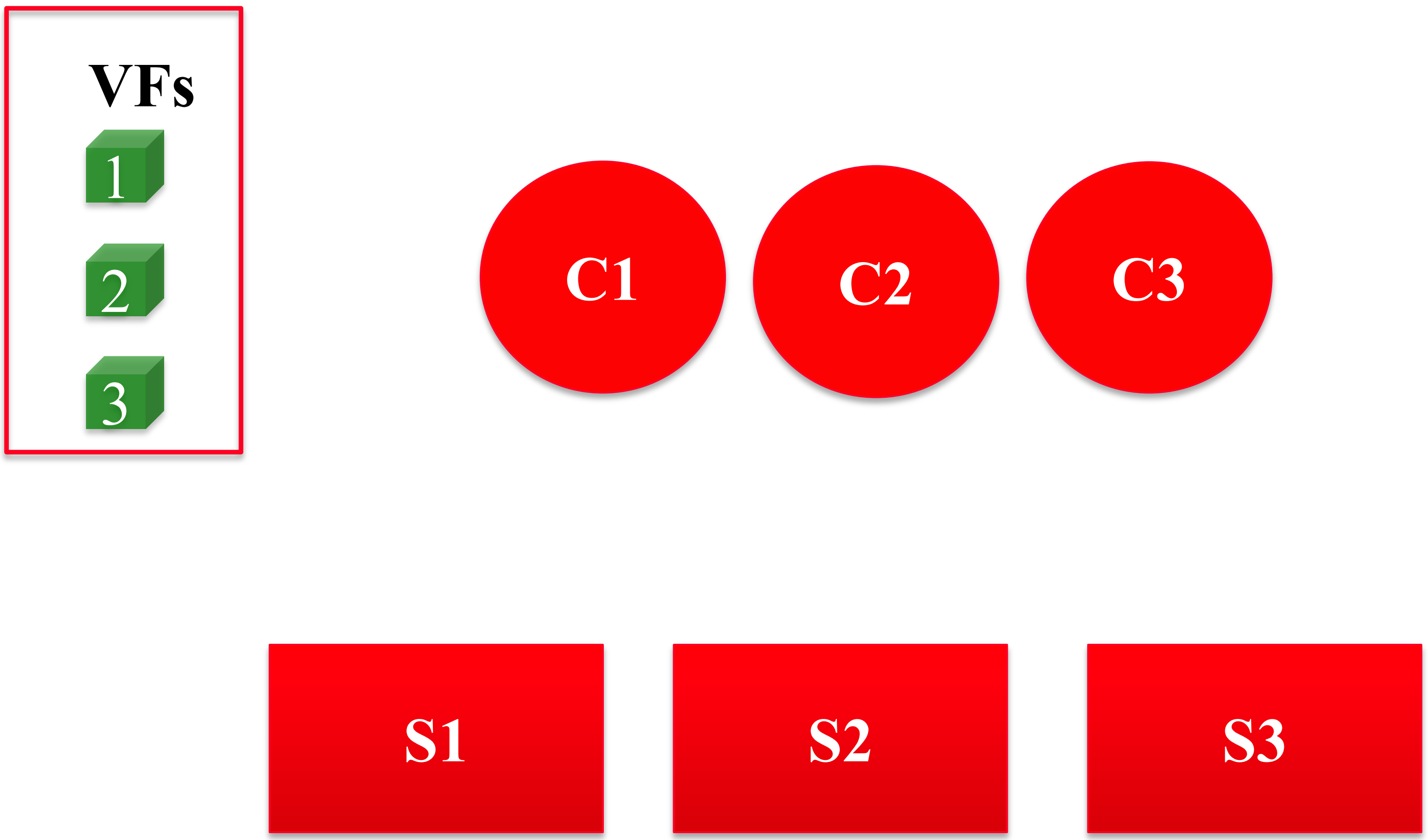
Service Chaining

- Individual Virtual Functions communicate with each other in defined ways
- Traffic between components is required (by policy) to flow through specialized network services (e.g., firewalls, IDS, etc.)

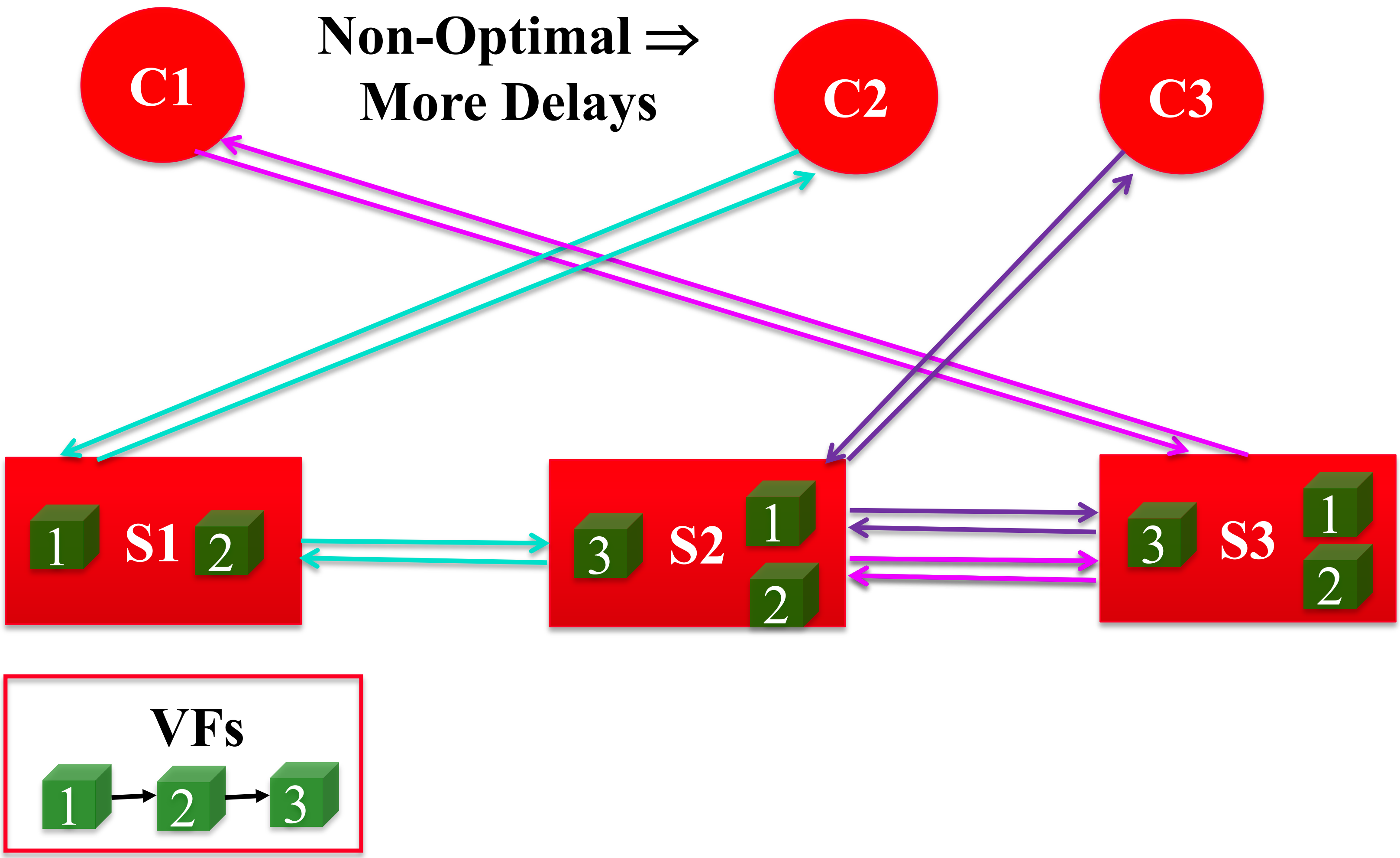


Source: IETF-87

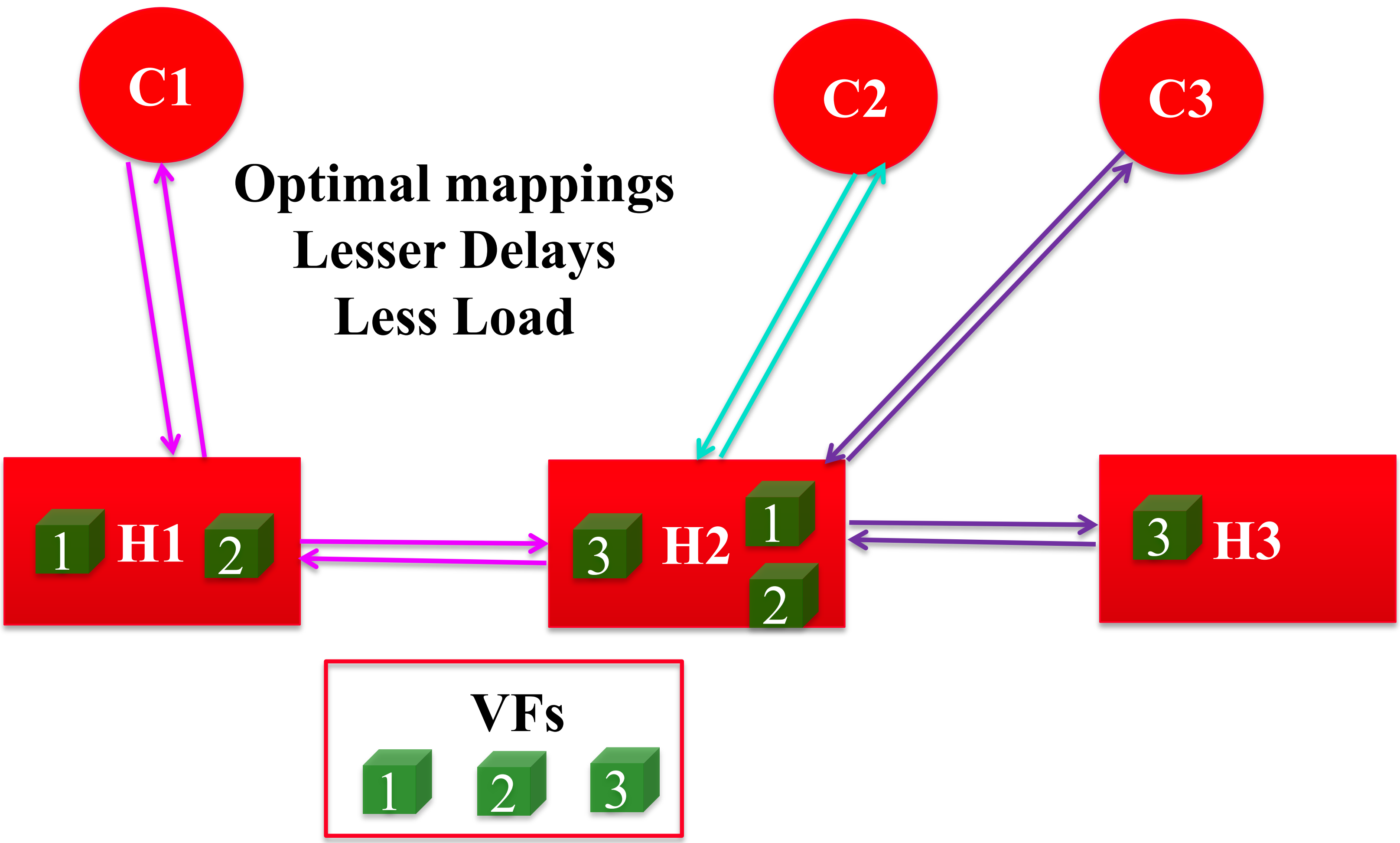
VFs, Clients, and Locations



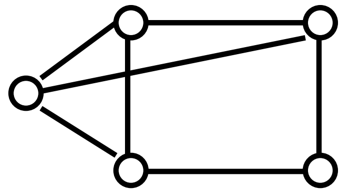
Mappings: Non-Optimal



Mappings: Optimal



Integer Linear Programming Formulation



- $G = \{V, E\}$ ➔ Network
- $E \subseteq V \times V$ ➔ Edges or Links
- λ ➔ Total VF instances ($\lambda_{min} \geq \lambda \geq \lambda_{max}$)
- C ➔ Capacity Matrix
  - $(C^1{}_1 + C^2{}_1 + C^3{}_1)$: Capacity of node 1 (CPU, Storage, NW)
- P ➔ Transmission Delay Matrix
- W ➔ Traffic Matrix
- **D** ➔ Demand Matrix
  - $(D^1{}_1 + D^2{}_1 + D^3{}_1)$: Demand of VF 1 (CPU, Storage, NW)
- A ➔ Allocation Matrix
- **T** ➔ Instance Matrix (How many instances of VF i are installed at server j = T_{ij})

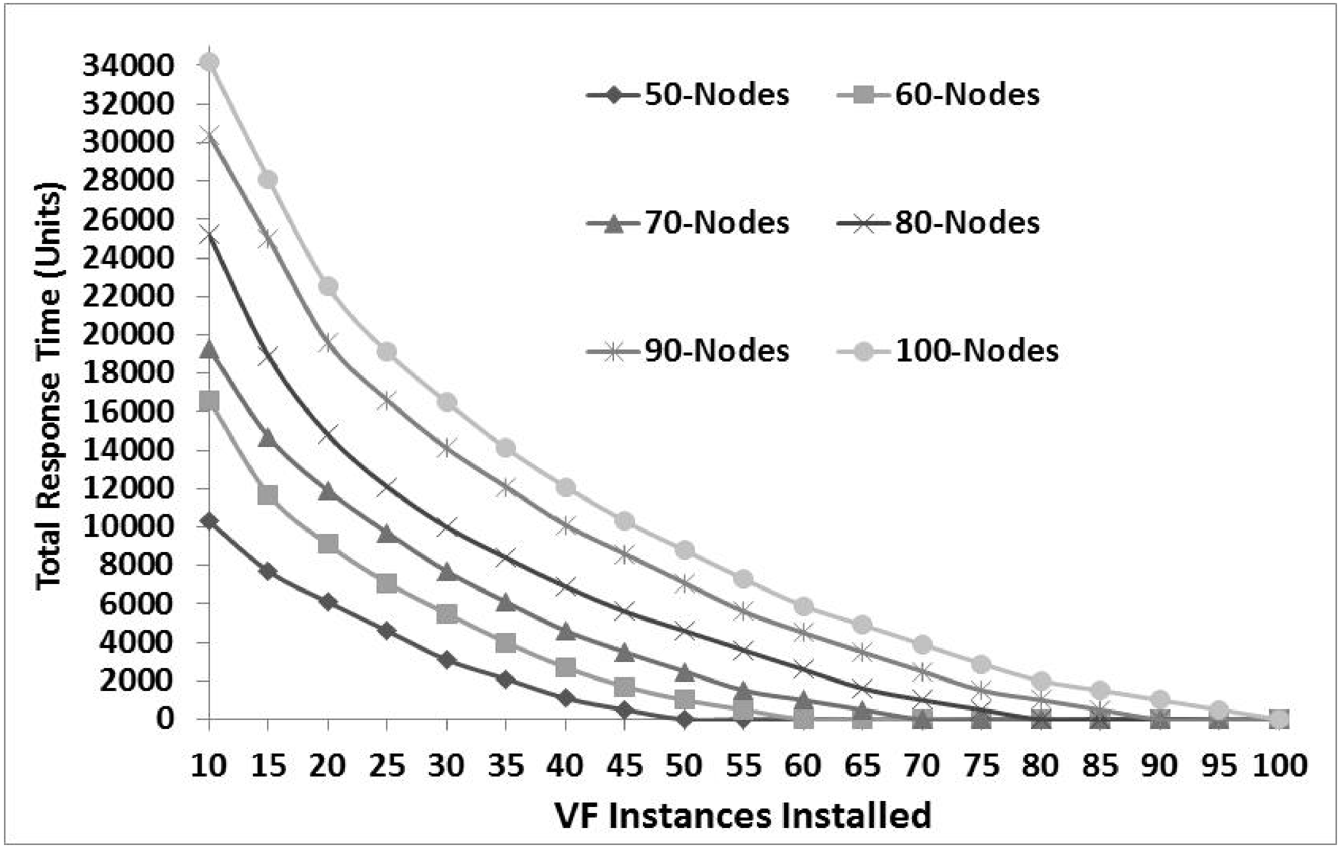
Minimization Function

- Minimize total delay without exceeding the server capacity

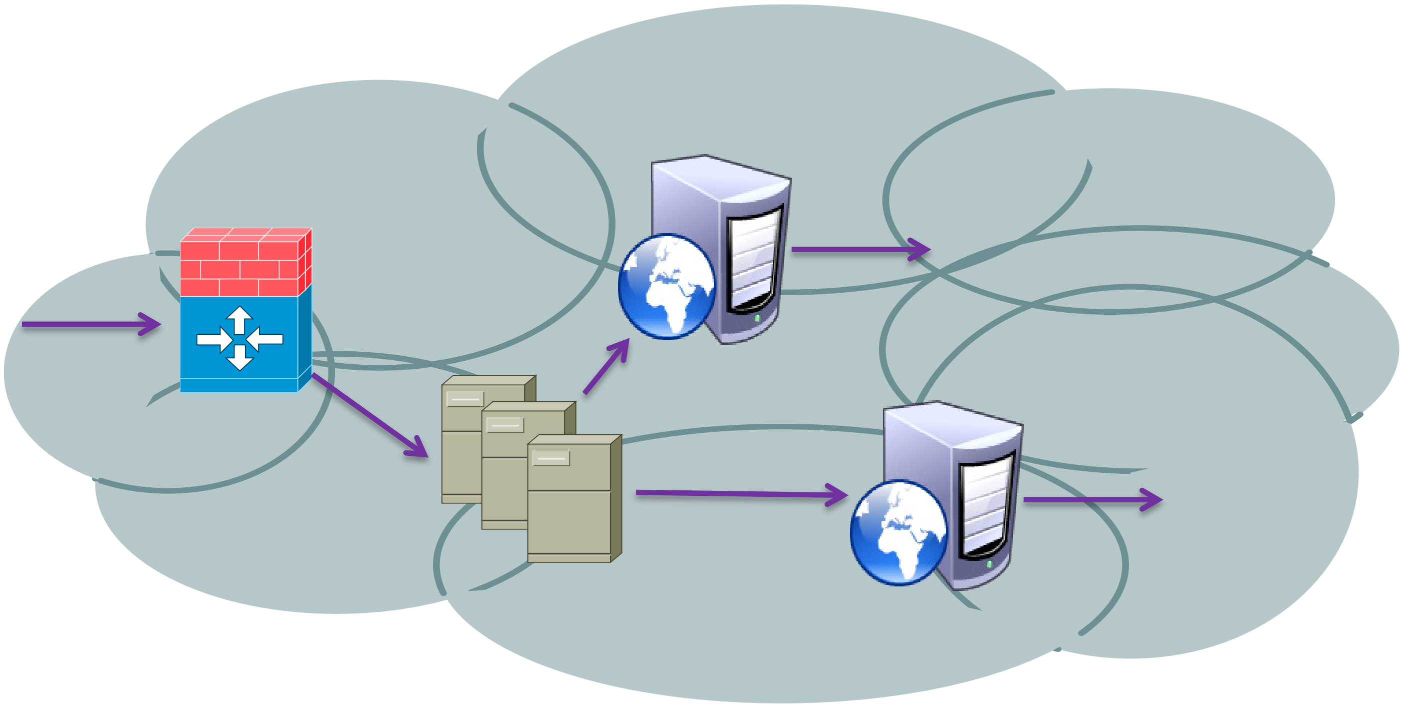
$$\sum_{i \in |V|} \sum_{j \in |V|} (w_i \times A_{ij}) / p_{ij}$$

- **P** is a Transmission Delay Matrix,
- **W** is a Traffic Matrix,
- ***A*** is an Allocation Matrix.

Response Time (10-100 Nodes)



Dynamic Mappings



- In the Cloud, a single ASP may have a set of services to be deployed.
- Each server may host a specific set of VFs.
- We propose a scheme for Dynamic Allocation of VFs to the servers.

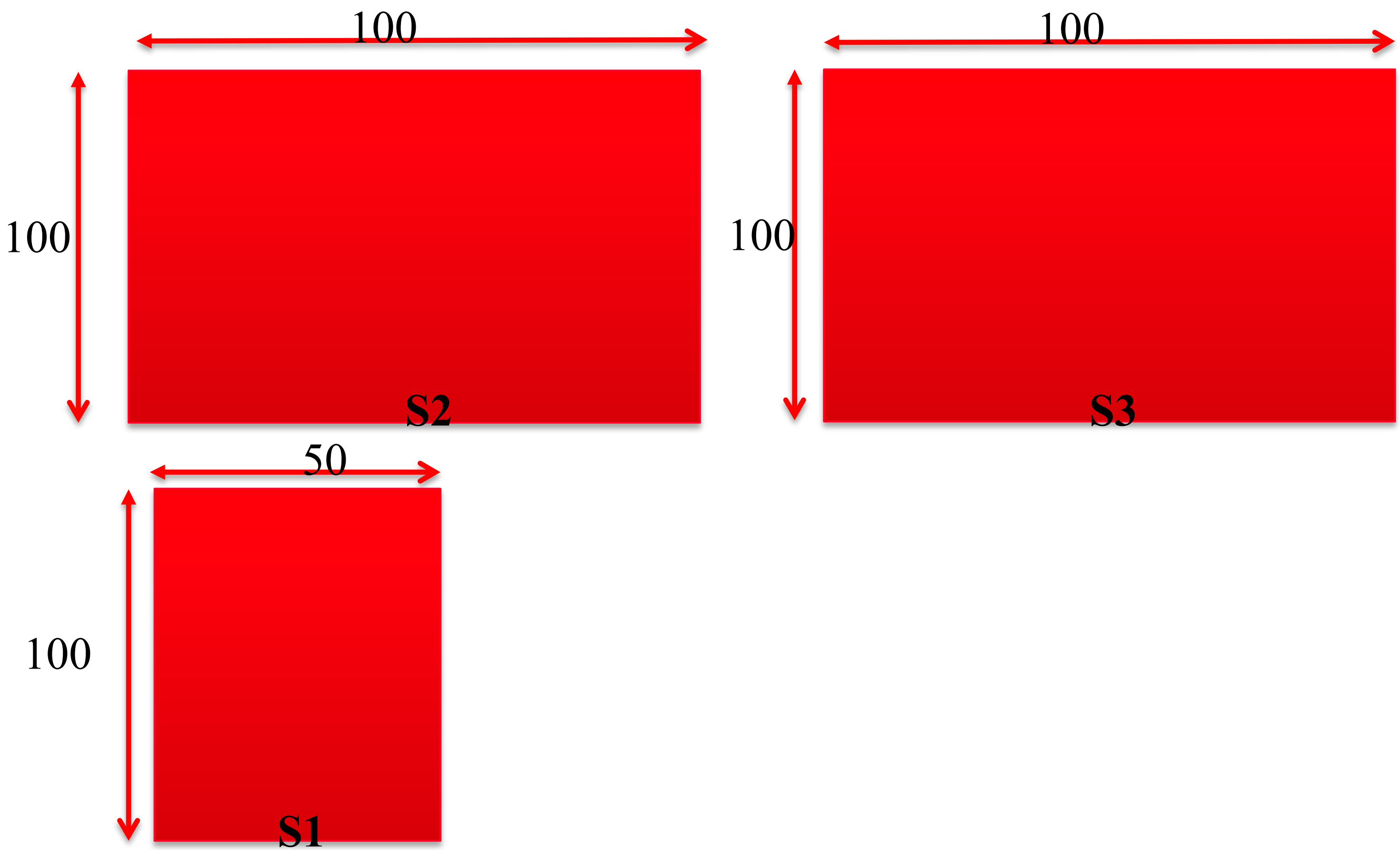
Simple Example

- 5 VFs and 3 servers. Each with 2 resources (2-dimension)

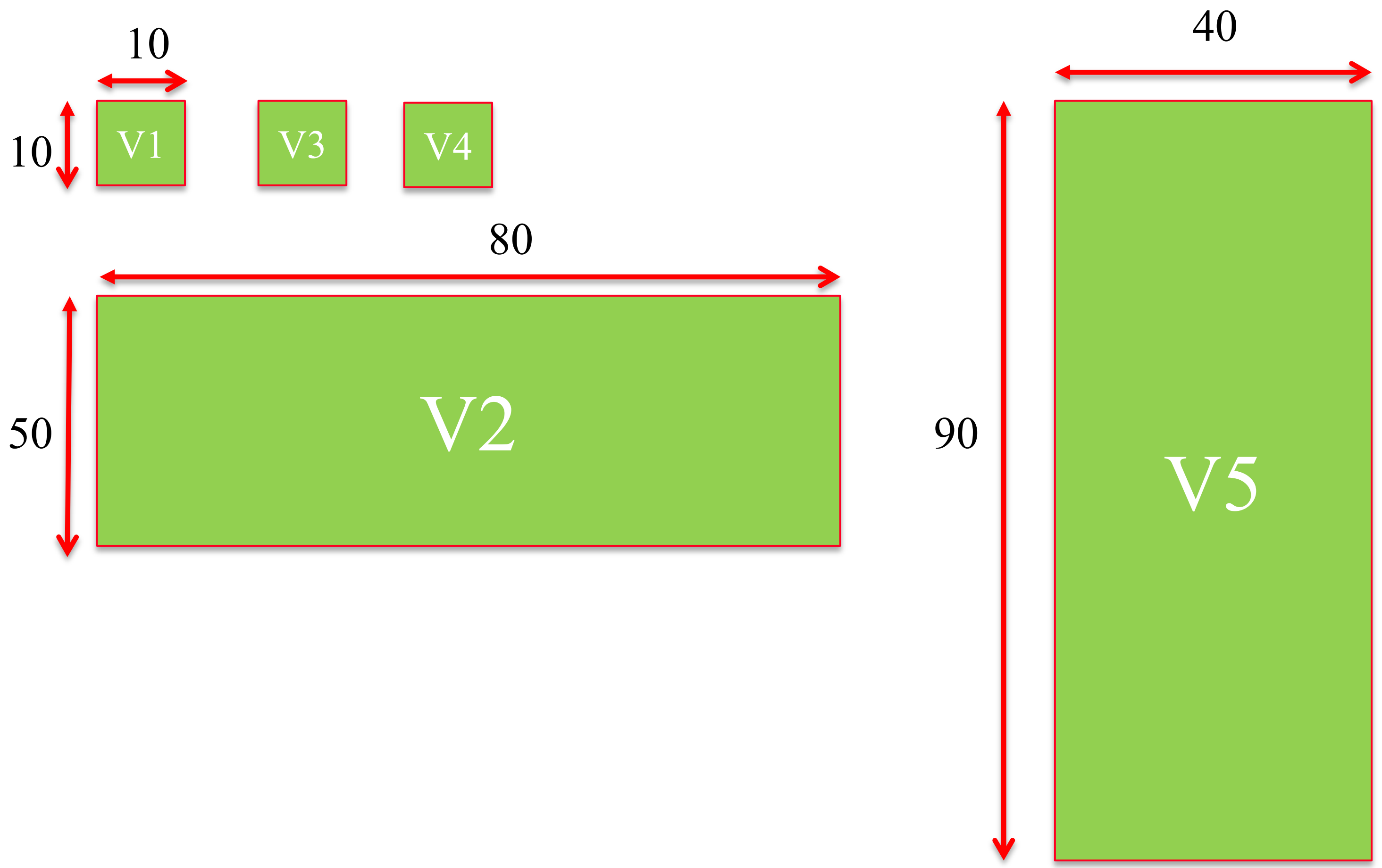
VFs	V1	V2	V3	V4	V5
r_1	10	80	10	10	40
r_2	10	50	10	10	90

Servers	S1	S2	S3
r_1	50	100	100
r_2	100	100	100

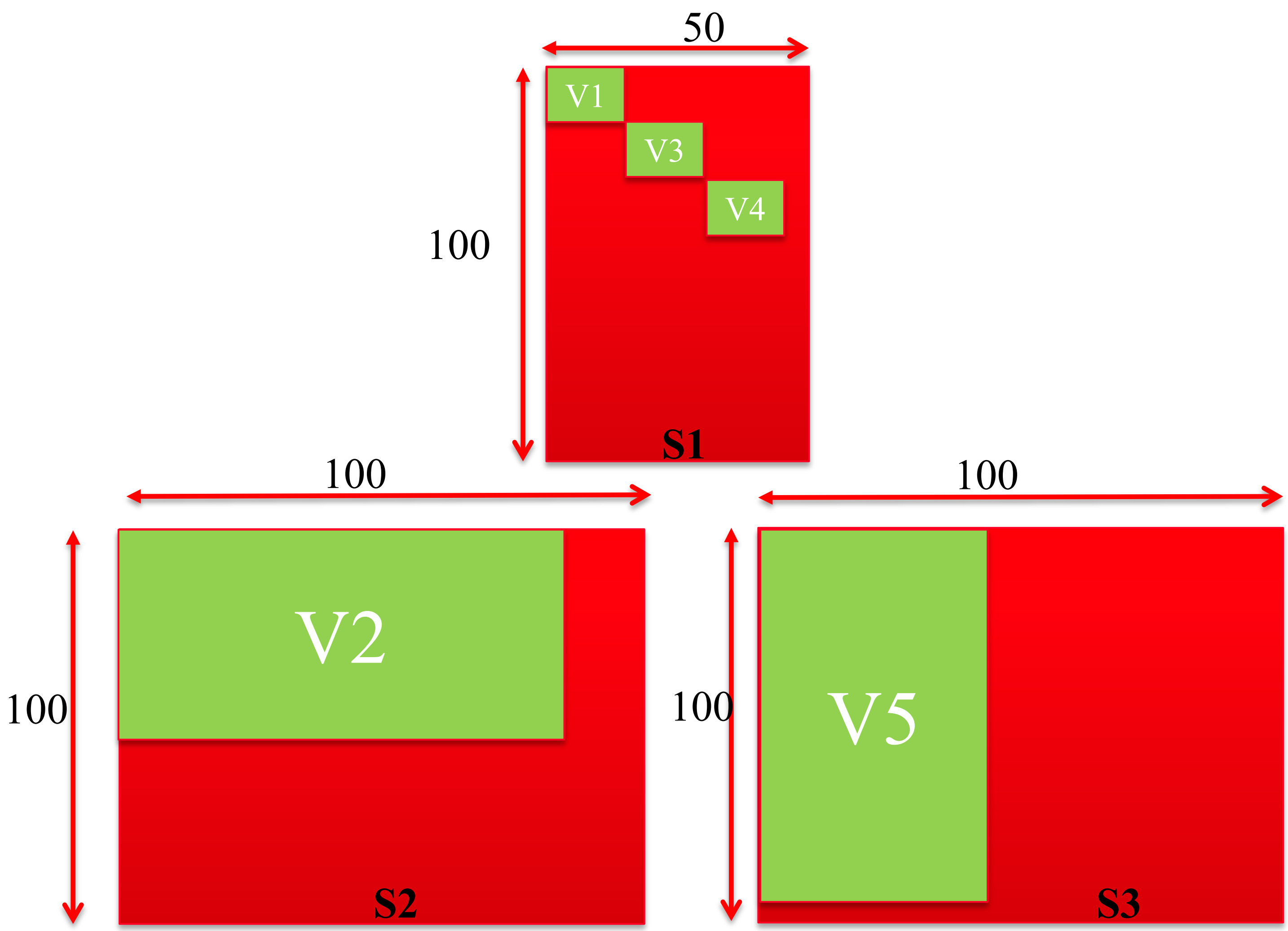
Resources



Demands

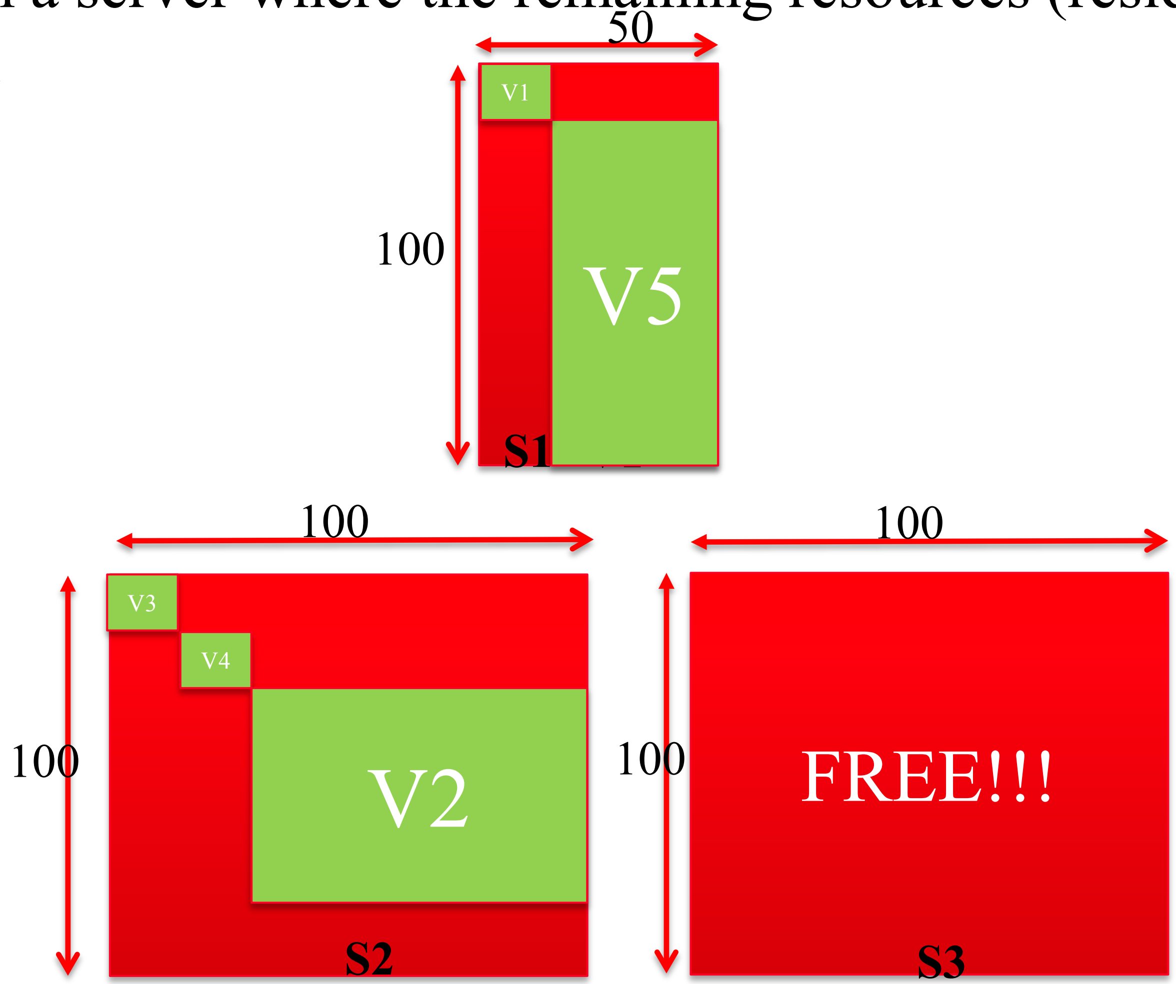


Max-Min Allocation

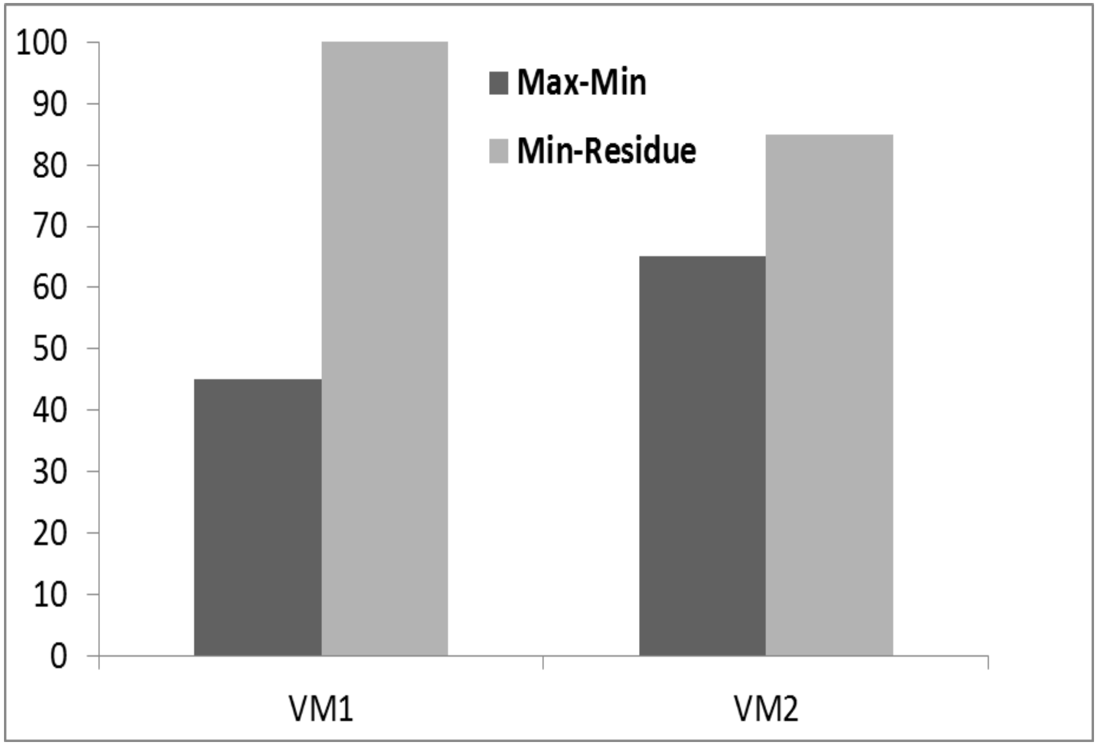


Minimum Residue Allocation

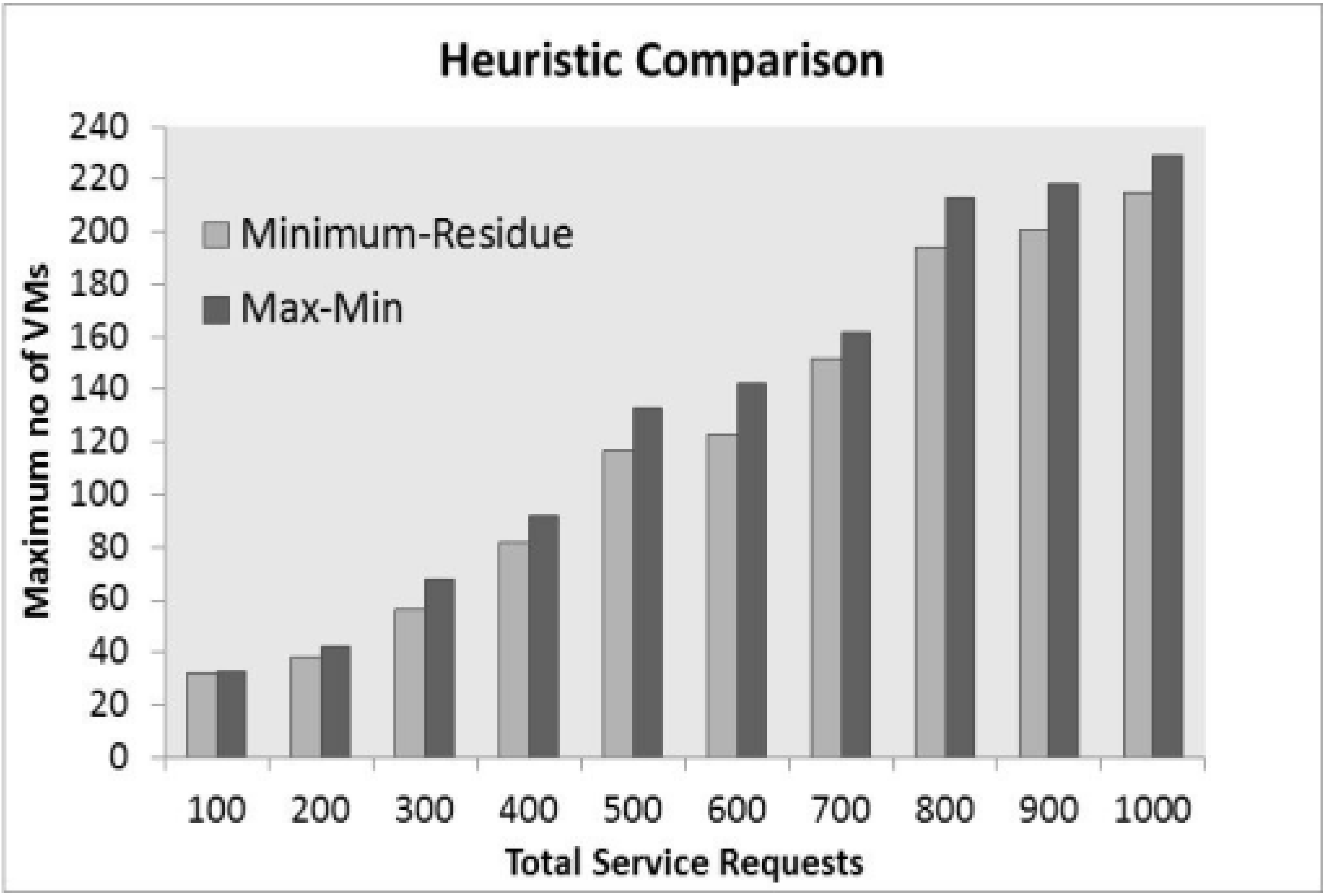
- Place a VF in a server where the remaining resources (residue) is minimized



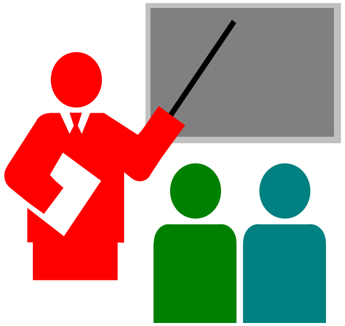
Average Utilization



Average Utilization of the VMs



Summary



1. We propose an Optimization model to ***Locate*** the Virtual Functions in the given network and ***Allocate*** the clients accordingly.

2. We propose a Heuristic scheme "***Minimum Residue***" for dynamic VF deployment.

3. The proposed scheme is compared against the standard ***Max-Min*** approach and improvements are showcased.