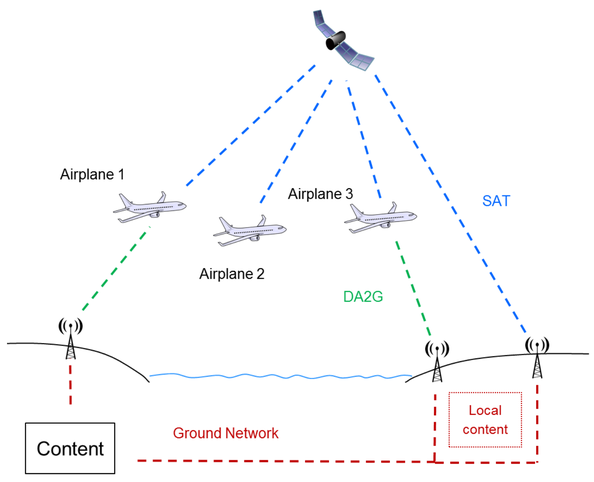
Mobility-Aware Service Placement

Amir Varasteh, Carmen Mas Machuca, Wolfgang Kellerer
TU München, Lehrstuhl für Kommunikationsnetze (TUM-LKN)

Scenario

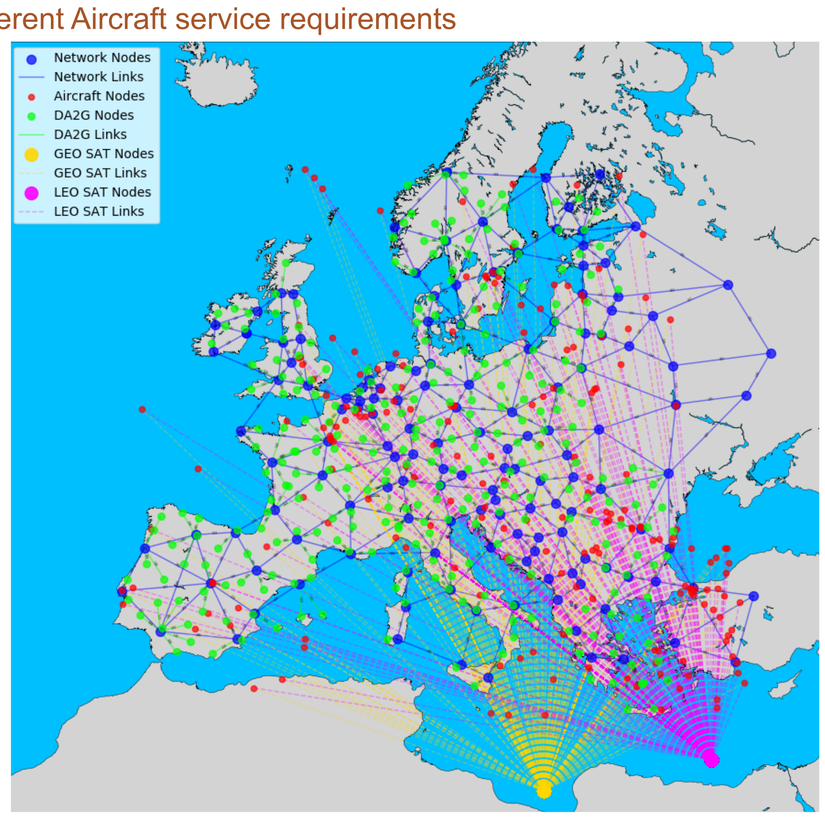
- Space-ground integrated network



- Aircrafts flying over Europe

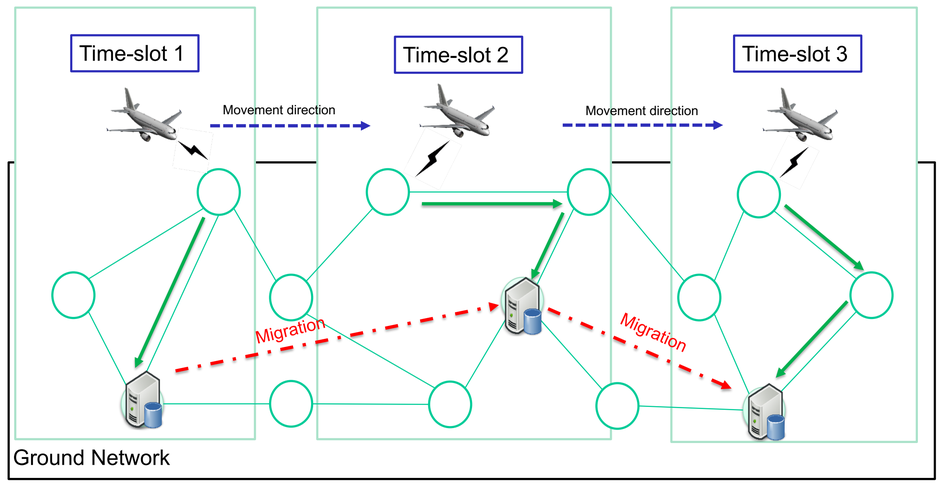
Problem Definition

- Dynamic environment
  - Aircrafts positioning
  - Different links, coverage, and properties (Delay, cost, bandwidth)
    - Direct Air to Ground Link (DA2G)
    - Satellites
    - Ground Network
- Different Aircraft service requirements



Research Questions

- How to map service instances to datacenters considering Aircrafts' mobility?
- How to guarantee QoS (Delay/Availability) for services required by Aircrafts?
- How to route traffic?



Objective

- Minimize total cost, while meeting delay/capacity constraints
  - VM costs
  - Routing costs (Through DA2G/SAT/Ground links)
  - VM migration (Dynamic case)

Methodology

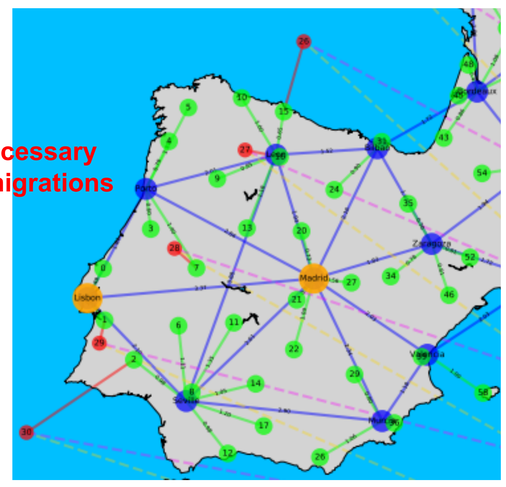
- Static optimization formulation
- Dynamic optimization formulation
- Heuristics for improving scalability

Static Optimization

- Objective function: $\boldsymbol{minimize}\ (\boldsymbol{\alpha Cost_{vm}} + \boldsymbol{\beta Cost_{Routing}})$
- Minimizing placement and routing costs
- Solve **per time-slot**

Time-slot	Selected DC	Path	Runtime (ms)
30	Madrid	30-2-Seville-Madrid	539
29	Lisbon	29-1-Lisbon	530
28	Madrid	28-7-Porto-Madrid	554
27	Madrid	27-16-Leon-Madrid	562
26	Madrid	26-15-Leon-Madrid	545

Unnecessary VM migrations

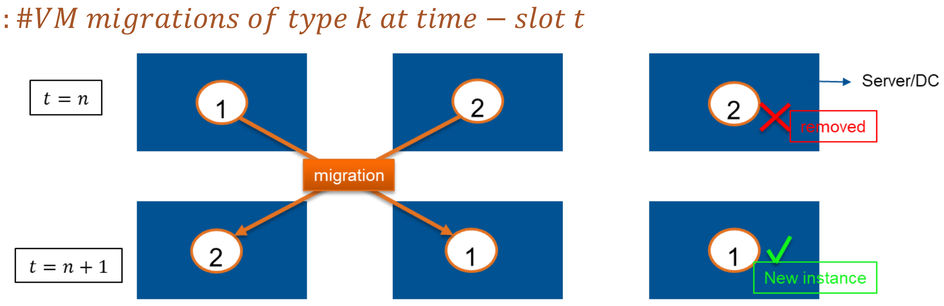


Dynamic Optimization

- Objective function: $\boldsymbol{minimize}\ (\boldsymbol{\alpha Cost_{vm}} + \boldsymbol{\beta Cost_{Routing}} + \boldsymbol{\gamma Cost_{Mig}})$
- Solve for the **whole time horizon**
- Trade-off between routing/VM/migration costs
- Adding VM migration cost

VM Migration Model

- $m_k^t = \begin{cases} \sum_{j\in J}\left(n_{jk}^t - n_{jk}^{t-1}\right)^+ - \left(\sum_{j\in J} n_{jk}^t - \sum_{j\in J} n_{jk}^{t-1}\right)^+ ; \forall k \in K, \forall\, t \in T^+ \\ 0 \qquad\qquad ; if\ t = 0, \forall k \in k \end{cases}$
- $m_{total}^t = \sum_{k\in K} m_k^t\ ; \forall t \in T$
- n_{jk}^t: $VM\ instances\ for\ service\ type\ k\ in\ DC\ j\ at\ time - slot\ t$
- m_k^t: $\#VM\ migrations\ of\ type\ k\ at\ time - slot\ t$

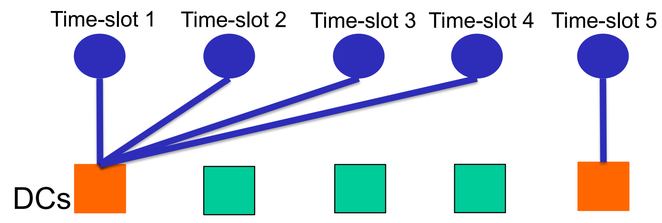


$$\#mig(service1) = [(0-1)^+ + (1-0)^+ + (1-0)^+] - [(2-1)^+] = 1$$
$$\#mig(service2) = [(1-0)^+ + (0-1)^+ + (0-1)^+] - [(1-2)^+] = 1$$
$$Total\ \#mig = 1 + 1 = 2$$

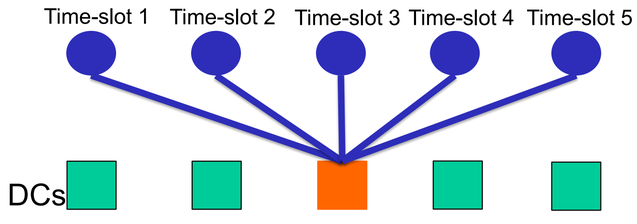
Heuristics

- Per Aircraft, per service approaches

1) Greedy: Assign aircrafts to the first available DC, while it meets end-to-end QoS requirements



2) Merge: Assign aircrafts to the DC, which minimizes the total cost over time horizon while meeting QoS requirements.



Next Steps

- Performance evaluation
- Prepare and submit a paper
- Study of a case with unknown future (e.g., autonomous vehicles)

In collaboration with AIRBUS