IPIC 2023

9th International Physical Internet Conference

June 13-15, 2023
Athens, Greece

MINES PARIS | PSL | Centre de Gestion Scientifique

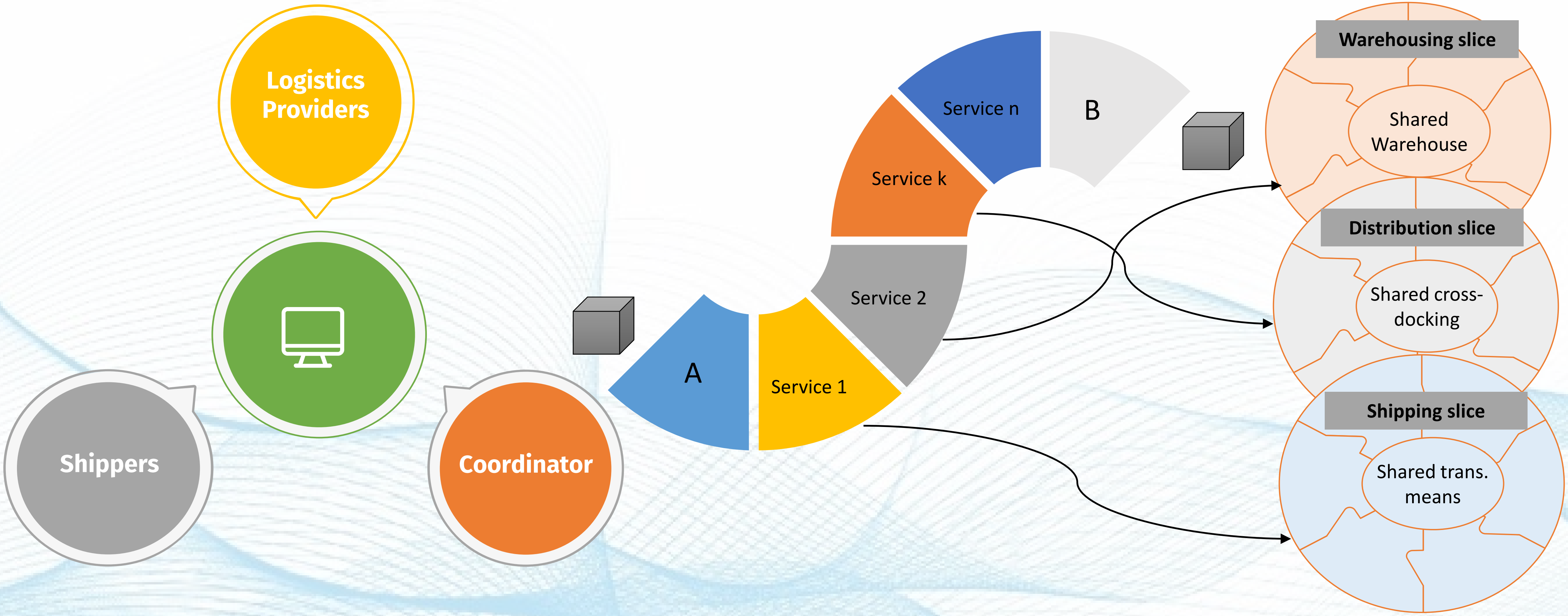
CHAIRE INTERNET PHYSIQUE

Routing and network-slicing-based protocols for the Physical Internet network

By Fatima-Ezzahra Achamrah, Mariam Lafkihi & Eric Ballot

June 14, 2023

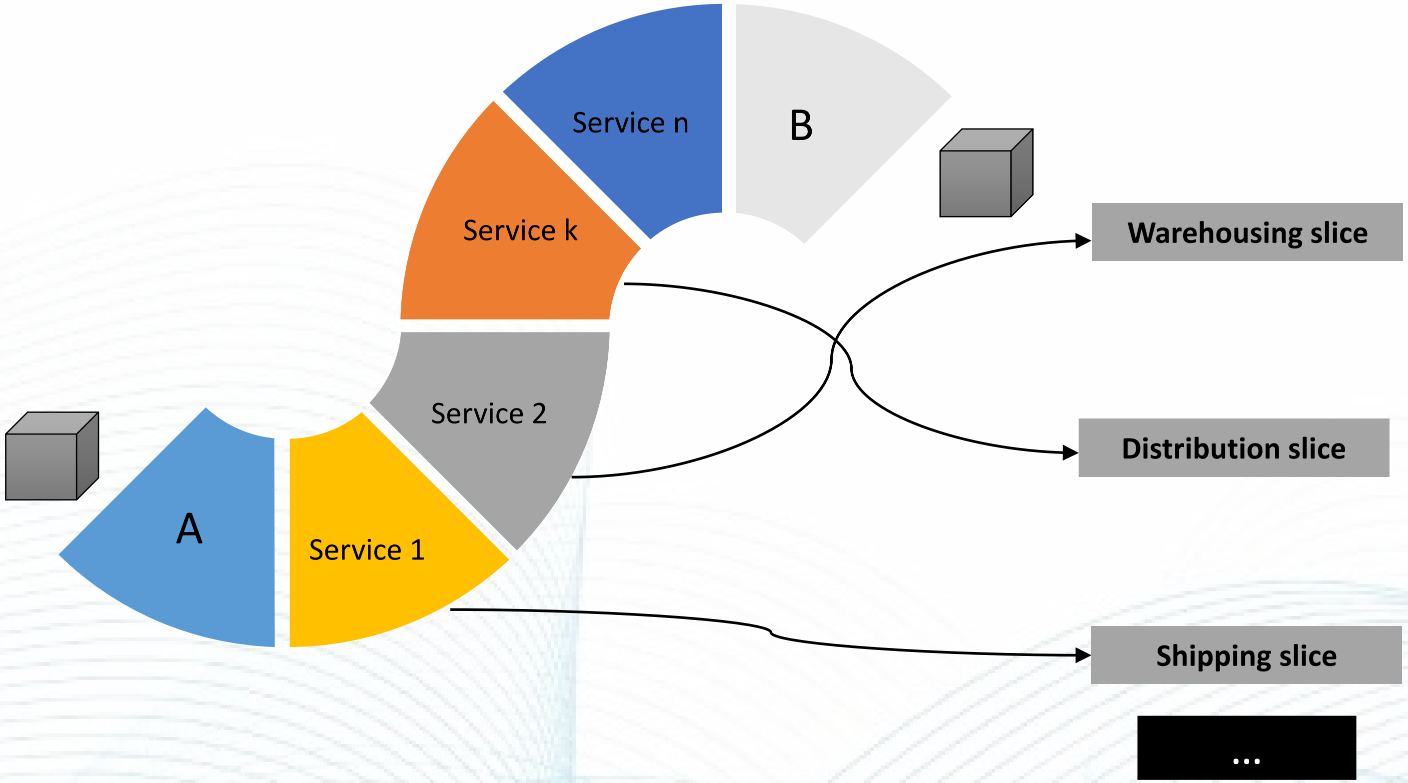
Context



…

IPIC 2023

Context

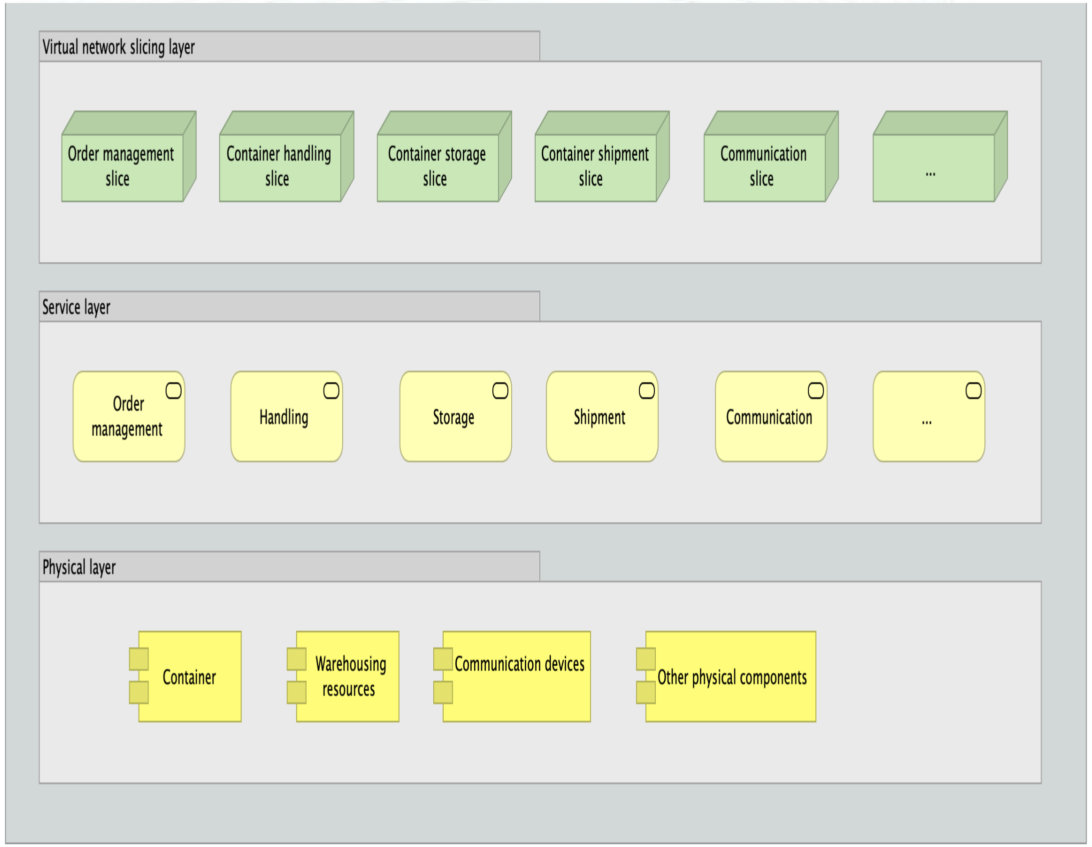


<u>Containers</u> or shipments have to be **<u>allocated</u>** to resources and delivered to customers while guaranteeing an end-to-end services, considering route-constraints (e.g., capacity, costs, lead time), and ever **changing** environments, i.e., **<u>disruptions</u>** (e.g., emergency or cancelled orders, hub unserviceability, transportation means unavailability).

<u>Resources</u> and **<u>shipments</u>** are considered to be able to compute decisions within a legal framework: they are supposed to **coordinate** with other entities in the network **<u>about routes and loads allocation</u>** based on **<u>actual context</u>** and **<u>local knowledge</u>** (known state of the system); **centralized** decision-making **can no longer** be **applied.**

Network-Slicing-Based Protocol

Network slicing allows the creation of **multiple virtual networks** with a **common physical infrastructure**, each optimized for **specific services** or user groups, aiming at providing customized and more efficient **end-to-end services** and **allocating the network cost** to the different deployed slices.



It is well-suited for use cases that require customized and more efficient **end-to-end services**

It involves the allocation of **dedicated resources** and the deployment of a wide range of services with **varying requirements**

It provides a high level of granularity, allowing coordinators to define and manage multiple network slices with distinct characteristics.

It requires a comprehensive management and orchestration framework to handle the lifecycle of network slices, including slice creation, configuration, monitoring, and scaling.

IPIC 2023

Network-Slicing-Based Protocol

1

Slice creation

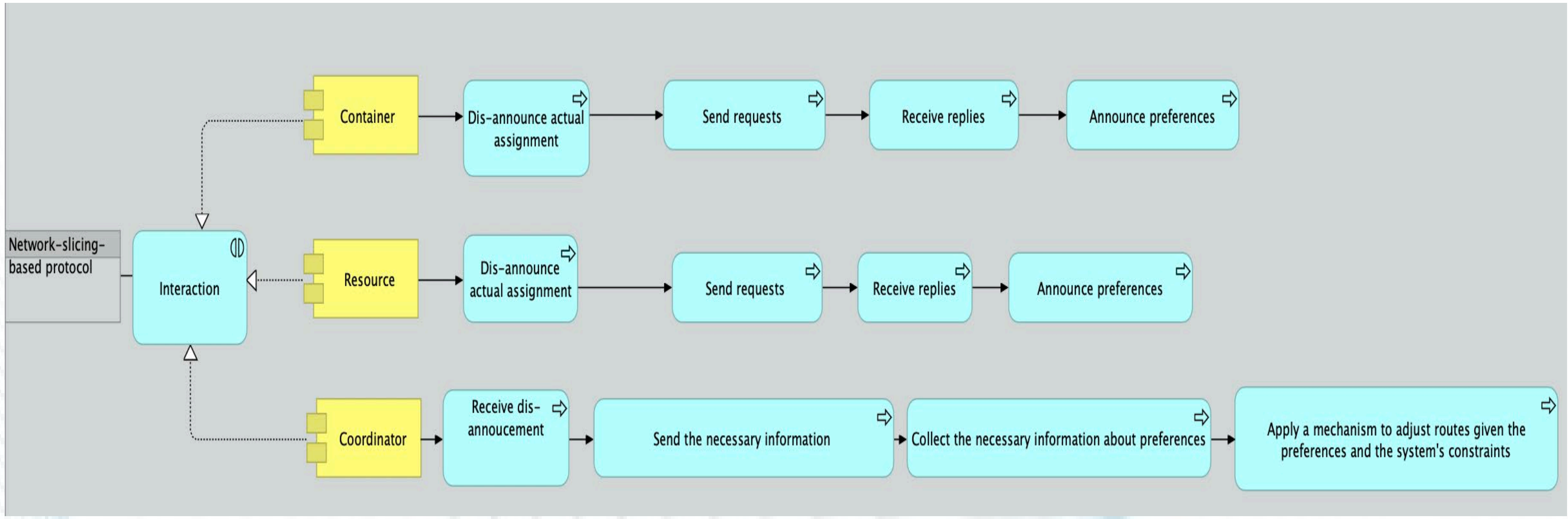
Based on the service layer requirements

2

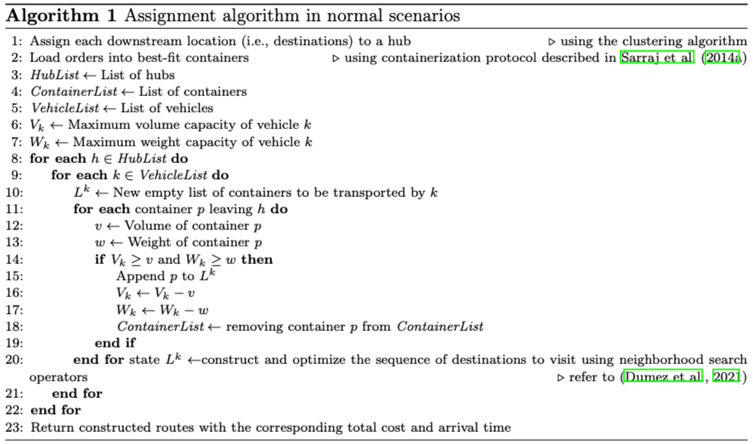
Allocating resources in a slice

MCDM and assignment algorithms while minimizing logistics costs and CO2 emission

Disruption management & preference integration

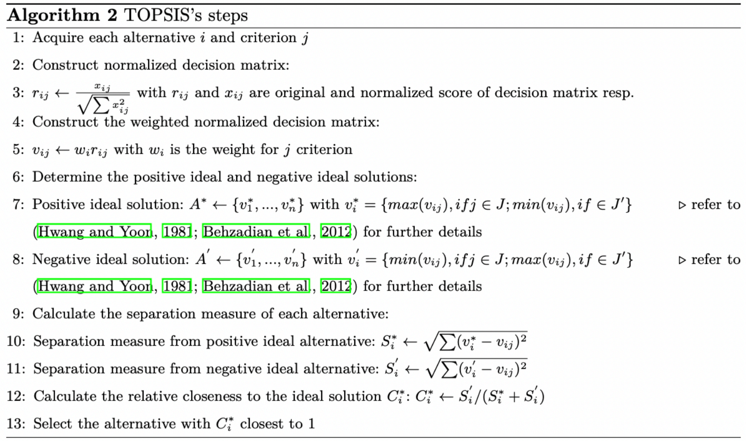


Examples of reactive assignment algorithms designed to offer good-quality solutions with simple computational capabilities.



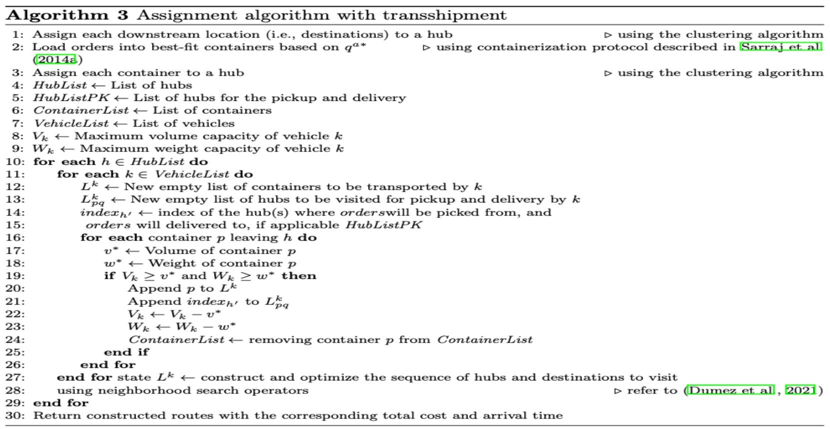
Algorithm 1 Assignment algorithm in normal scenarios

```
Assign each downstream location (i.e., destinations) to a hub            ▷ using the clustering algorithm
Load orders into best-fit containers      ▷ using containerization protocol described in Sarraj et al. (2014a)
$HubList$ ← List of hubs
$ContainerList$ ← List of containers
$VehicleList$ ← List of vehicles
$V_k$ ← Maximum volume capacity of vehicle $k$
$W_k$ ← Maximum weight capacity of vehicle $k$
for each $h \in HubList$ do
    for each $k \in VehicleList$ do
        $L^k$ ← New empty list of containers to be transported by $k$
        for each container $p$ leaving $h$ do
            $v$ ← Volume of container $p$
            $w$ ← Weight of container $p$
            if $V_k \geq v$ and $W_k \geq w$ then
                Append $p$ to $L^k$
                $V_k \leftarrow V_k - v$
                $W_k \leftarrow W_k - w$
                $ContainerList$ ← removing container $p$ from $ContainerList$
            end if
        end for state $L^k$ ←construct and optimize the sequence of destinations to visit using neighborhood search
operators                                                                    ▷ refer to (Dumez et al., 2021)
    end for
end for
Return constructed routes with the corresponding total cost and arrival time
```



Algorithm 2 TOPSIS's steps

```
Acquire each alternative $i$ and criterion $j$
Construct normalized decision matrix:
$r_{ij} \leftarrow \frac{x_{ij}}{\sqrt{\sum x_{ij}^2}}$ with $r_{ij}$ and $x_{ij}$ are original and normalized score of decision matrix resp.
Construct the weighted normalized decision matrix:
$v_{ij} \leftarrow w_i r_{ij}$ with $w_i$ is the weight for $j$ criterion
Determine the positive ideal and negative ideal solutions:
Positive ideal solution: $A^* \leftarrow \{v_1^*, ..., v_n^*\}$ with $v_i^* = \{max(v_{ij}), if j \in J; min(v_{ij}), if \in J'\}$     ▷ refer to
    (Hwang and Yoon, 1981; Behzadian et al., 2012) for further details
Negative ideal solution: $A^{'} \leftarrow \{v_1^{'}, ..., v_n^{'}\}$ with $v_i^{'} = \{min(v_{ij}), if j \in J; max(v_{ij}), if \in J'\}$     ▷ refer to
    (Hwang and Yoon, 1981; Behzadian et al., 2012) for further details
Calculate the separation measure of each alternative:
Separation measure from positive ideal alternative: $S_i^* \leftarrow \sqrt{\sum(v_i^* - v_{ij})^2}$
Separation measure from negative ideal alternative: $S_i^{'} \leftarrow \sqrt{\sum(v_i^{'} - v_{ij})^2}$
Calculate the relative closeness to the ideal solution $C_i^*$: $C_i^* \leftarrow S_i^{'}/(S_i^* + S_i^{'})$
Select the alternative with $C_i^*$ closest to 1
```



Algorithm 3 Assignment algorithm with transshipment

```
Assign each downstream location (i.e., destinations) to a hub            ▷ using the clustering algorithm
Load orders into best-fit containers based on $q^{a*}$     ▷ using containerization protocol described in Sarraj et al.
(2014a)
Assign each container to a hub                                           ▷ using the clustering algorithm
$HubList$ ← List of hubs
$HubListPK$ ← List of hubs for the pickup and delivery
$ContainerList$ ← List of containers
$VehicleList$ ← List of vehicles
$V_k$ ← Maximum volume capacity of vehicle $k$
$W_k$ ← Maximum weight capacity of vehicle $k$
for each $h \in HubList$ do
    for each $k \in VehicleList$ do
        $L^k$ ← New empty list of containers to be transported by $k$
        $L_{pq}^k$ ← New empty list of hubs to be visited for pickup and delivery by $k$
        $index_{h'}$ ← index of the hub(s) where $orders$will be picked from, and
         $orders$ will delivered to, if applicable $HubListPK$
        for each container $p$ leaving $h$ do
            $v^*$ ← Volume of container $p$
            $w^*$ ← Weight of container $p$
            if $V_k \geq v^*$ and $W_k \geq w^*$ then
                Append $p$ to $L^k$
                Append $index_{h'}$ to $L_{pq}^k$
                $V_k \leftarrow V_k - v^*$
                $W_k \leftarrow W_k - w^*$
                $ContainerList$ ← removing container $p$ from $ContainerList$
            end if
        end for
    end for state $L^k$ ← construct and optimize the sequence of hubs and destinations to visit
    using neighborhood search operators                                  ▷ refer to (Dumez et al., 2021)
end for
Return constructed routes with the corresponding total cost and arrival time
```

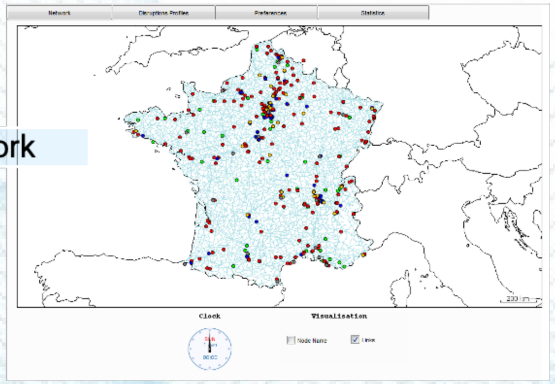
Experiments

Using a muli agent simulation, and data from major French retail chains Carrefour and Casino and their 106 largest suppliers,

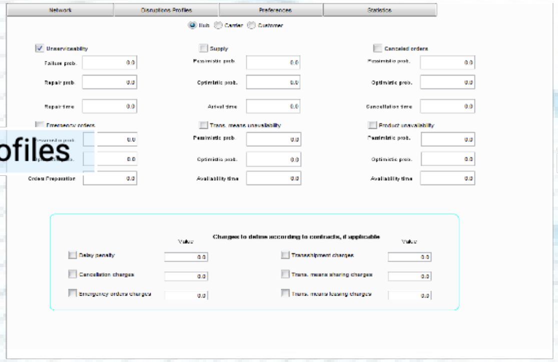
- 303 plants, 57 warehouses, and 58 distribution centers across France.
- Three products: liquids, groceries, and personal and home care.
- 2,582,692 full pallets are routed.
- 211,167 orders of 702 different products, accounting for approximately 20% of French FMCG market share for the considered product families.

Assessing the performance of the protocols

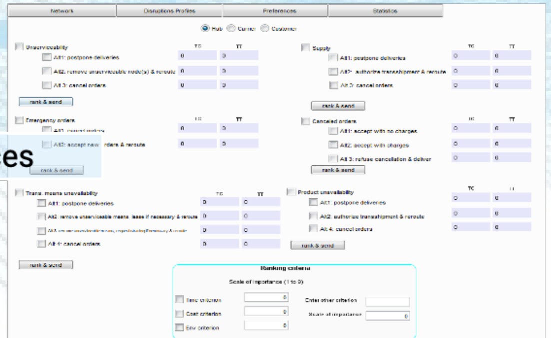
Network



Disruption profiles



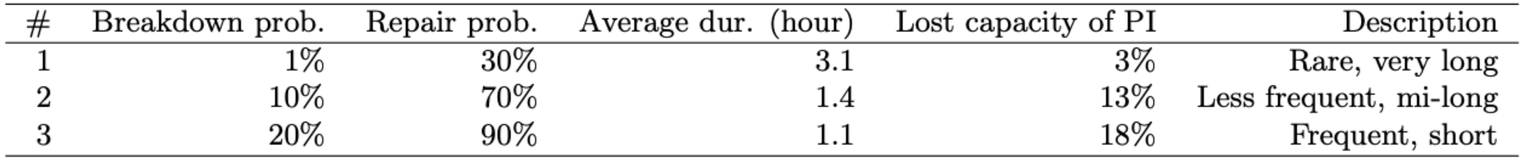
Preferences



Statistics

Experiments : Simplified Example

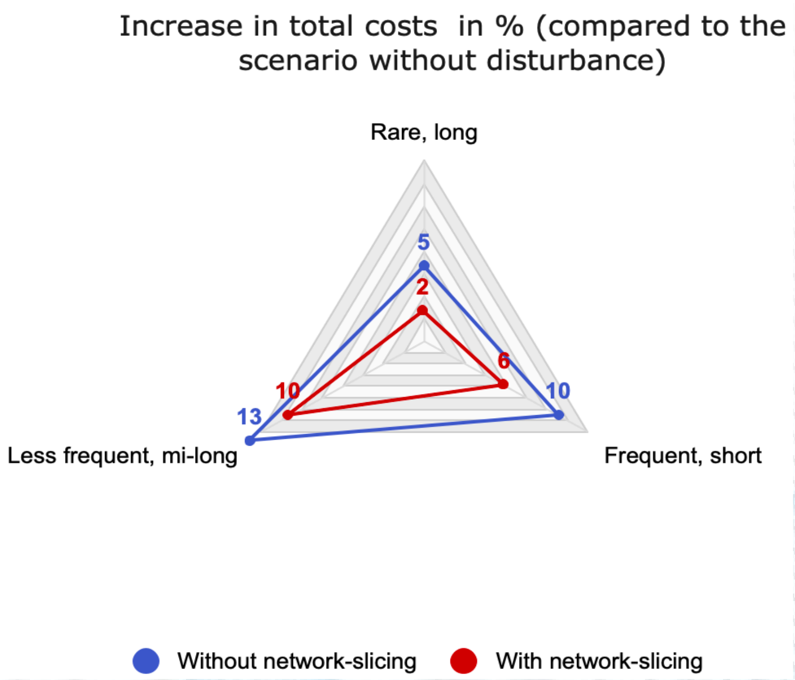
- A 7-day simulation, 10 plants, 10 warehouses, 10 carriers (trains and trucks), 1000 orders of 5 different products to ship, and only storing and shipping services.
- We evaluate the performance of the proposed protocols in absence and presence of disturbance, namely, hub resource breakdown.
- We evaluate the performance of the proposed protocols in presence of disturbance, namely, hub breakdown. We consider long, rare ; less frequent, mi-long and frequent, short disturbance profiles.

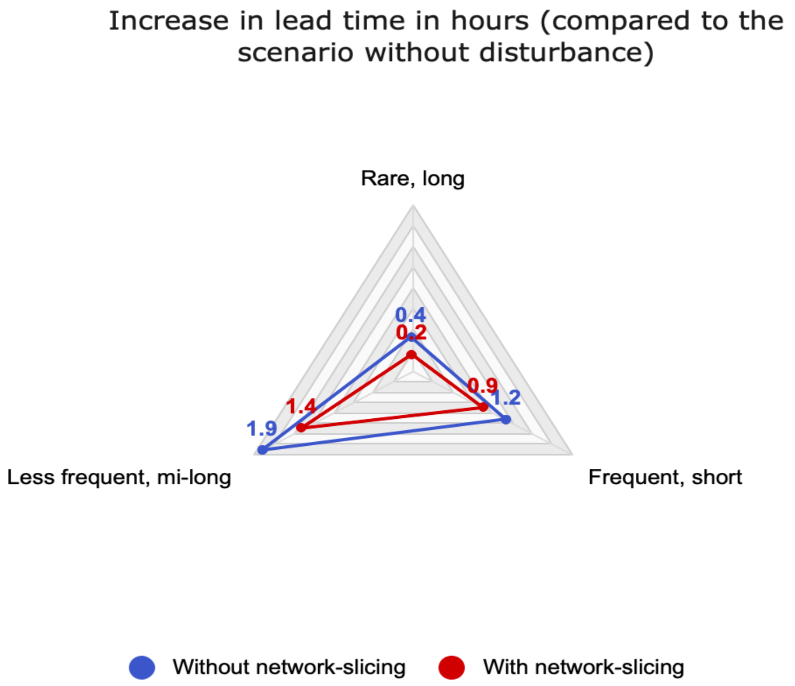


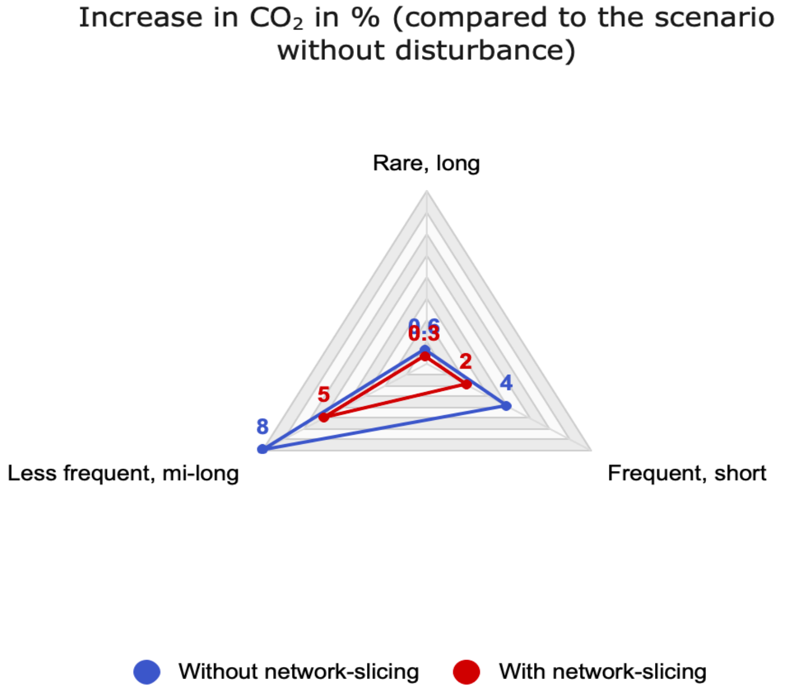
#	Breakdown prob.	Repair prob.	Average dur. (hour)	Lost capacity of PI	Description
1	1%	30%	3.1	3%	Rare, very long
2	10%	70%	1.4	13%	Less frequent, mi-long
3	20%	90%	1.1	18%	Frequent, short

- The performance is evaluated using three key performance indicators, namely, increase in total cost (i.e., transportation and storage), in lead time, and CO_2 emissions.
- Two scenarios are evaluated:
  - Without network-slicing: locally optimizing resources allocation.
  - With network-slicing: optimizing resources allocation for all slices.

Experiments


Increase in total costs in % (compared to the scenario without disturbance)


Increase in lead time in hours (compared to the scenario without disturbance)


Increase in CO_2 in % (compared to the scenario without disturbance)

Conclusions & Perspectives

- *A network slicing-based protocol for more efficient resource allocation and management and to guide individual self-interested decisions toward a system-wide common goal.*
- *A dynamic and reactive protocol to consider various contexts, such as disturbing events, route constraints, and PI actors' preferences and local knowledge of the system's state.*

- Future research should investigate the trade-offs in implementing logistics network slicing while considering decentralized protocol and managing disruptions.
- Investigate other classes of operational disruptions, but also tactical and strategic ones, along with reactive alternatives that have yet to be investigated.
- Generalize the scope to the case where multiple coordinators (i.e., platforms) provide communication and container delivery in and between several PI sub-networks.
- Generalize the scope to cover and compare on-demand and scheduled services;
- Capitalize on data collected, using the routing tables, to assist the disturbance management by using similarities and accurate predictions (e.g., reinforcement learning, transfer learning, or case and rule-based reasoning).