Resillient Communications though Mathematical Optimization

Jorge López and Charalampos Chatzinakis

Airbus

Gurobi EMEA Summit 2025
Vienna, October 28th, 2025

Before we start..

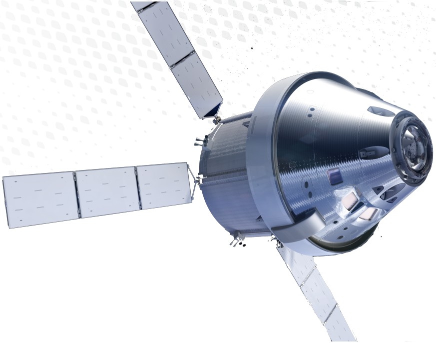
This is an Airbus. . .

Before we start..

And. . . this is an Airbus. . .

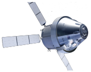
Before we start..

And. . . this is an Airbus. . .



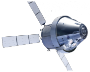
Before we start..

And. . . this is an Airbus. . .



Before we start..

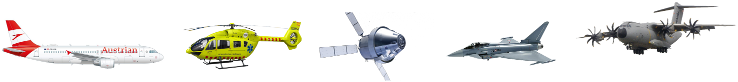
And. . . this is an Airbus. . .



Before we start..

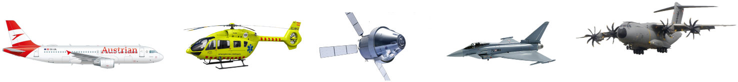
And. . .

We must manage all the complexity, design, and diverse functionality of our large portfolio. . .



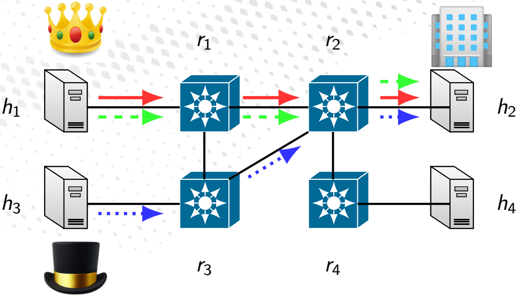
Before we start..

We must manage all the complexity, design, and diverse functionality of our large portfolio. . . Including what is the best position for the wireless router within the body of an A320 neo. . . or ensuring our assets are operational through resillient communication. . .



The main idea of what we want to achieve today

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

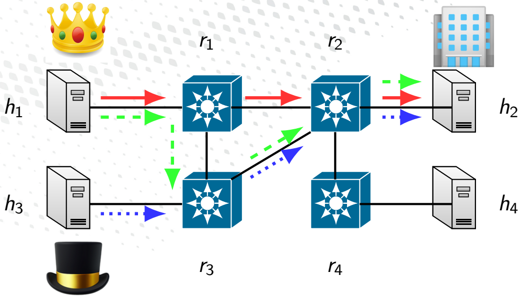


⚠ The link between r_1 and r_2 is degraded!

▶ We want to prioritize (and deliver) important traffic!

The main idea of what we want to achieve today

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

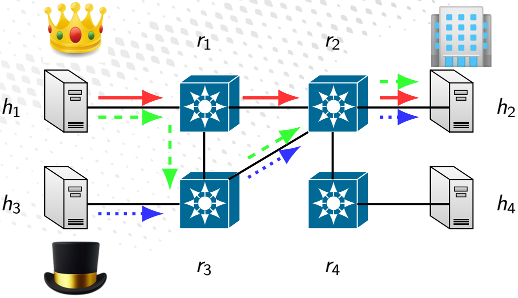


⚠ The link between r_1 and r_2 is degraded!

- ▶ We want to prioritize (and deliver) important traffic!
- ▶ If alternative paths are possible, we want to use them…

The main idea of what we want to achieve today

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

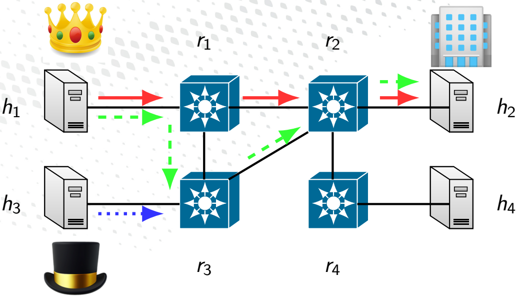


⚠⚠There is not enough space for all traffic between r_3 and r_2

- ▶ We want to prioritize (and deliver) important traffic!
- ▶ If alternative paths are possible, we want to use them...

The main idea of what we want to achieve today

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

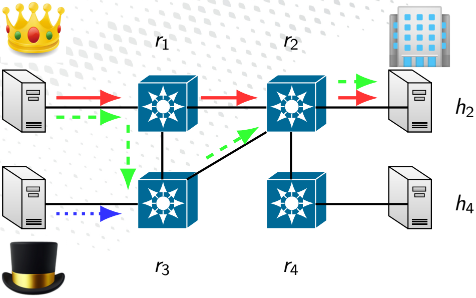


⚠️⚠️There is not enough space for all traffic between r_3 and r_2

- We want to prioritize (and deliver) important traffic!
- If alternative paths are possible, we want to use them...
- If passing all traffic is impossible, we take the **bold** decision to drop traffic, including local traffic!

The main idea of what we want to achieve today

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

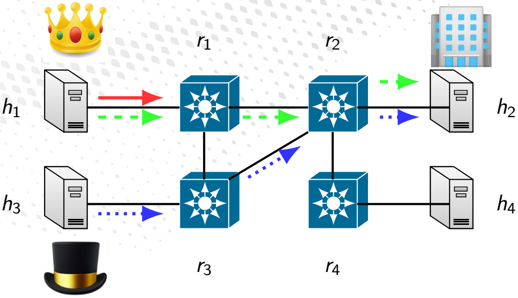


⚠⚠⚠Operational context changes $\implies$ change in priorities!

- ▶ We want to prioritize (and deliver) important traffic!
- ▶ If alternative paths are possible, we want to use them...
- ▶ If passing all traffic is impossible, we take the **bold** decision to drop traffic, including local traffic!

The main idea of what we want to achieve today

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network



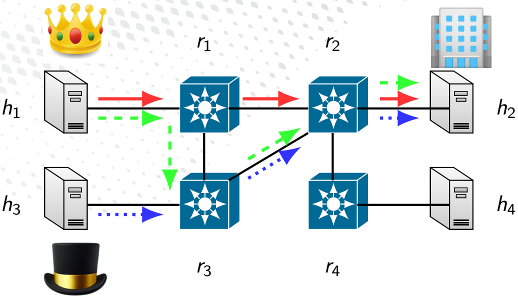
⚠️⚠️⚠️Operational context changes $\Longrightarrow$ change in priorities!

- We want to prioritize (and deliver) important traffic!
- If alternative paths are possible, we want to use them...
- If passing all traffic is impossible, we take the **bold** decision to drop traffic, including local traffic!
- We want to arbitrarily assign priority to traffic

But, with some more industrial requirements. . .

Latency constraints

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

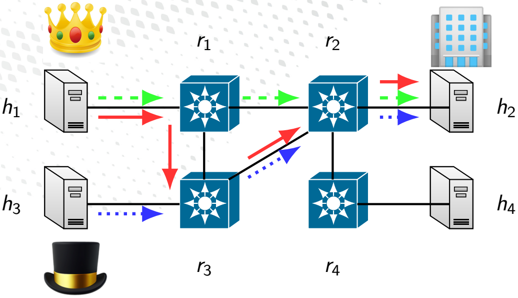


⚠ Assume that the link between r_1 and r_2 can route only one flow and that each link has a delay of 50ms and the green flow - - → has a maximal tolerance to delay of 75ms!

But, with some more industrial requirements. . .

Latency constraints

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network



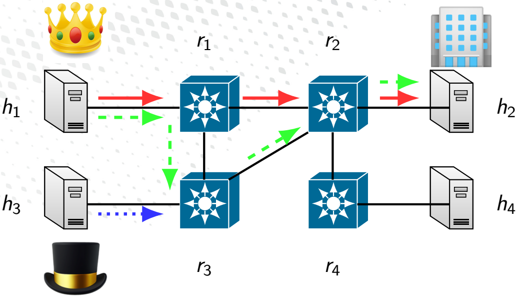
⚠️ Assume that the link between r_1 and r_2 can route only one flow and that each link has a delay of 50ms and the green flow - - → has a maximal tolerance to delay of 75ms!

▶ We want to respect the Quality of Experience (QoE) constraints w.r.t delay!

But, with some more industrial requirements. . .

Partial admission

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

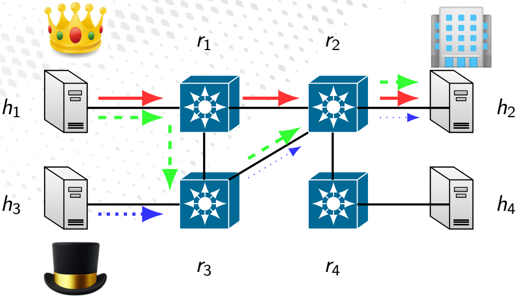


⚠️ Assume there's not enough bandwidth for both the - - → and the ·····→ flows, the - - → flow should get dropped, but, what if we could cut say 50% of the blue flow, and both could be admitted? 🤔

But, with some more industrial requirements. . .

Partial admission

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network



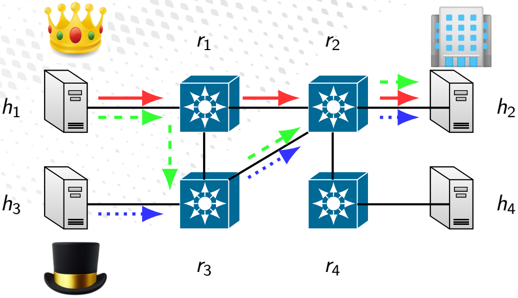
⚠ Assume there's not enough bandwidth for both the - - → and the ······→ flows, the - - → flow should get dropped, but, what if we could cut say 50% of the blue flow, and both could be admitted? 🤔

▶ We want to respect the Quality of Experience (QoE) constraints w.r.t maximal loss!
  – At the same time, try to admit as much traffic as possible. . .

But, with some more industrial requirements. . .

Shortest possible paths

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network

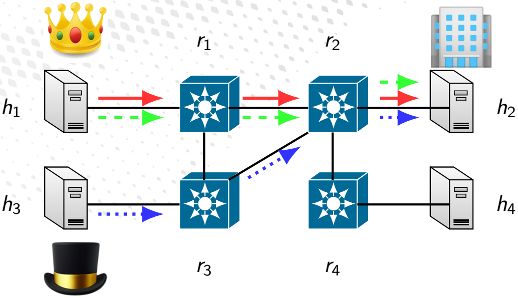


⚠️ Assume there is no bandwidth nor latency limitation, this configuration is perfectly valid! But, it wastes resources (the - - → flow using two hops) 😱

But, with some more industrial requirements. . .

Shortest possible paths

- h_1 is used by a VIP user
- h_3 is used by a regular user
- h_2 hosts many services
- There are flows going through the network



⚠️ Assume there is no bandwidth nor latency limitation, this configuration is perfectly valid! But, it wastes resources (the - - → flow using two hops) 😱

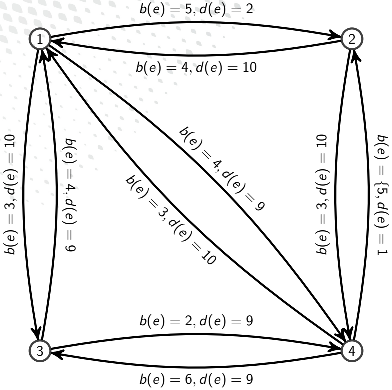
- We want to use the shortest paths whenever available. . .
  - Evidently, if it comes to a choice, we want to favor priority flows!

Notations for modeling the problem

A (transport) network

Represents a state of resources, a directed and and multi-weighted graph (V, E, c, d), where:

- V is a set of nodes;
- $E \subseteq V \times V$ is a set of edges (ordered pairs of nodes);
- $c : E \to \mathbb{Z}_+$ is a bandwidth capacity function;
- $d : E \to \mathbb{Z}_+$ is a delay function.



A path p is a sequence of edges ($p \in E^*$; the empty path is denoted as ϵ)

- We denote p_i the i-th edge of the path and $|p|$ its length
- p is simple if all nodes in the path are distinct
- p is valid if $\forall i \in \{1, \ldots, |p| - 1\}\ dst(p_i) = src(p_{i+1})$

Notations for modeling the problem (2)

A (network) flow

Following RFC 3917 [QZCZ04], a flow f is a four-tuple $f = (s, d, b, h)$, where $s, d \in V$, are source (s) and destination (d) nodes for the flow, $b \in \mathbb{Z}_+$ is the required bandwidth, and finally $h \in \{0,1\}^k$ is the packet header (of k bits) of the packets belonging to the flow

Flow policy

A policy P for the header of the packets belonging to a flow $h(f)$ is the tuple $P = (\mathcal{P}, \mathcal{D}, \mathcal{L})$, where:

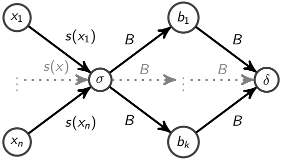
- $\mathcal{P}$ is a priority function $\mathcal{P} : \{0,1\}^k \to \mathbb{Z}_+$;
- $\mathcal{D}$ is a maximal delay function $\mathcal{D} : \{0,1\}^k \to \mathbb{Z}_+$;
- $\mathcal{L}$ is a maximal loss function (in percentage) $\mathcal{L} : \{0,1\}^k \to [0,1)$.

⚠ Note that, for convenience if $f = (s', d', b', h')$, we denote $s(f) = s'$, $d(f) = d'$, $b(f) = b'$, and $h(f) = h'$

Policy-based Admission and Shortest Paths Routing (PASPR) problem

Given a transport network $N = (V, E, c, d)$, a finite set of flows F, and a policy $P = (\mathcal{P}, \mathcal{D}, \mathcal{L})$, find a path for each flow (mapping solution) $\mathcal{S} : F \to E^*$, such that: i) the paths are simple and valid for the flows and network; ii) the bandwidth of the allocated flows does not exceed the bandwidth capacity of each link; iii) the paths do not violate the maximal accepted delays; iv) the value of the priority of the allocated flows is maximal; and finally v) the admitted bandwidth demand is as high as possible (considering it does not affect the acceptance of other priority flows).

The PASPR problem is **NP**-hard (in the strong sense [GJ78]): proof by a reduction from the BIN-PACKING problem [Vaz13]



Notations. . . to solve the PASPR problem

We take the **bold** decision to solve such a hard problem. . .

Symbol	**Meaning**
f_i	The i-th flow in lexicographical order of F
α_i	A variable denoting if the i-th flow should be admitted ($\alpha_i = 1$) or dropped ($\alpha_i = 0$)
$\mathbf{paths}(f_i)$	A set of (simple and valid) paths for flow f_i
$\mathbf{paths}(f_i)_m$	The m-th path in lexicographical order for the set of paths for flow f_i
$\rho_{i,m}$	A variable denoting if the i-th flow should be routed via the m-th path for f_i (in lexicographical order of $\mathbf{paths}(f_i)$)
$\beta_{i,m}$	A variable denoting the accepted percentage of the i-th flow demand if routed via the m-th path for f_i (in lexicographical order of $\mathbf{paths}(f_i)$)
$\varepsilon_{i,j,l}$	A variable denoting the percentage of the i-th flow demand if it goes through the edge (j, l)
$\mathbf{1_{i,m}}(j, l)$	A characteristic function, indicating if edge (j, l) belongs to the path m for the i-th flow
$\mathbf{max\_prio}(F)$	The maximal priority for the set of flows F

Aim for the moon. . . seeking optimality

$$\textbf{maximize}: \sum_{i=1}^{|F|} \mathcal{P}(h(f_i)) * \alpha_i + \sum_{i=1}^{|F|} \sum_{m=1}^{|\textbf{paths}(f_i)|} \frac{\mathcal{P}(h(f_i))}{\textbf{max\_prio}(F)} \cdot \beta_{i,m}$$

$$- \frac{1}{|V| \cdot \textbf{max\_prio}(F)} \sum_{i=1}^{|F|} \sum_{m=1}^{|\textbf{paths}(f_i)|} \mathcal{P}(h(f_i)) \cdot |\textbf{paths}(f_i)_m| \cdot \rho_{i,m}$$

subject to:

$$\sum_{m=1}^{|\textbf{paths}(f_i)|} \rho_{i,m} = \alpha_i; \qquad \forall i \in \{1, \ldots, |F|\} \text{ (i)}$$

$$\beta_{i,m} \geq (1 - \mathcal{L}(h(f_i))) \cdot \rho_{i,m}; \qquad \forall i \in \{1, \ldots, |F|\} \forall m \in \{1, \ldots, |\textbf{paths}(f_i)|\} \text{ (ii)}$$

$$\beta_{i,m} \leq \rho_{i,m}; \qquad \forall i \in \{1, \ldots, |F|\} \forall m \in \{1, \ldots, |\textbf{paths}(f_i)|\} \text{ (iii)}$$

$$\varepsilon_{i,j,l} = \sum_{m=1}^{|\textbf{paths}(f_i)|} \mathbf{1_{i,m}}(j, l) \cdot \beta_{i,m}; \qquad \forall i \in \{1, \ldots, |F|\} \forall (l, j) \in E \text{ (iv)}$$

$$\sum_{i}^{|F|} b(f_i) * \varepsilon_{i,j,l} \leq c((j, l)); \qquad \forall (j, l) \in E \text{ (v)}$$

$$\rho_{i,m} \cdot \sum_{(j,l) \in \textbf{paths}(f_i)_m} d(j, l) \leq \mathcal{D}(h(f_i)); \qquad \forall i \in \{1, \ldots, |F|\} \forall m \in \{1, \ldots, \textbf{paths}(f_i) \text{ (vi)}$$

$$\alpha_i \in \{0, 1\}; \qquad \forall i \in \{1, \ldots, |F|\} \text{ (vii)}$$

$$\rho_{i,m} \in \{0, 1\}; \qquad \forall i \in \{1, \ldots, |F|\} \forall m \in \{1, \ldots, \textbf{paths}(f_i)\} \text{ (viii)}$$

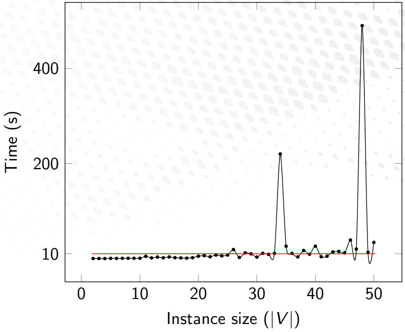
$$\varepsilon_{i,j,l} \in [0, 1]; \qquad \forall i \in \{1, \ldots, |F|\} \forall (i, j) \in E \text{ (ix)}$$

$$\beta i, m \in [0, 1]; \qquad \forall i \in \{1, \ldots, |F|\} \forall m \in \{1, \ldots, \textbf{paths}(f_i)\} \text{ (x)}$$

We note that an equivalent formulation with flow conservation constraints of the multi-comidity flow problem is possible. . . when explicit path enumeration is not possible

Gurobi empowering our **bold** decisions 😄

Can we really provide an answer in an acceptable time to reconfigure?



Running time[1]

- This experiments were performed with an out of the box Gurobi 9 (2020) 😱
  - No NoRel considering all the binary variables
  - No Specific cuts (NetworkCuts, FlowPathCuts, FlowCoverCuts, etc.)
  - No warm start...
  - No nothing...

- Of course an optimal solution $\neq$ optimaly proven...
  - In this setting, agility is more important than optimality proofs
  - We can guarantee the performance to be better than any heurstic solution easily trough warm starts sessions! 😉

[1] [LLPB20] Jorge López, Maxime Labonne, Claude Poletti, Dallal Belabed: Priority Flow Admission and Routing in SDN: Exact and Heuristic Approaches. IEEE NCA 2020: 1-10

Video Demonstration: worth more than 1000 words. . .

If the video doesn't play, click **here**!

Some key takeaways. . .

Our solution 😉:

- Continously tacks the network and traffic state to react to to pertinent decisions
- We compute a global optimal solution for the network that finds the relevant accepted bandwidth ratio and paths which are the shortest possible but, respect all SLA (policy) requirements
- Speed is essential, and a close-to-optimal solutions are better than no solution if instances are too big or too dynamic
- Important, we provide real time adaptation capablities and fine control ⚠️

We are working to implmeent the solution in different parts of the business, this can be lifechaning when asset operations rely on reliable communications, which is more and more the case 😱

Thank you! 😄

Questions, collaboration proposals, or remarks are welcome!

You can contact me at:

jorge.lopez-c@airbus.com

Michael R Garey and David S Johnson.
"strong" np-completeness results: Motivation, examples, and implications.
Journal of the ACM (JACM), 25(3):499–508, 1978.

Jorge López, Maxime Labonne, Claude Poletti, and Dallal Belabed.
Priority flow admission and routing in SDN: exact and heuristic approaches.
In *19th IEEE International Symposium on Network Computing and Applications, NCA 2020, Cambridge, MA, USA, November 24-27, 2020*, pages 1–10. IEEE, 2020.

Juergen Quittek, Tanja Zseby, Benoit Claise, and Sebastian Zander.
RFC 3917: requirements for ip flow information export: Ipfix.
Published by Internet Engineering Task Force (IETF). Internet Society (ISOC) RFC Editor. USA. out, 2004.

Vijay V Vazirani.
Approximation algorithms.
Springer Science & Business Media, 2013.