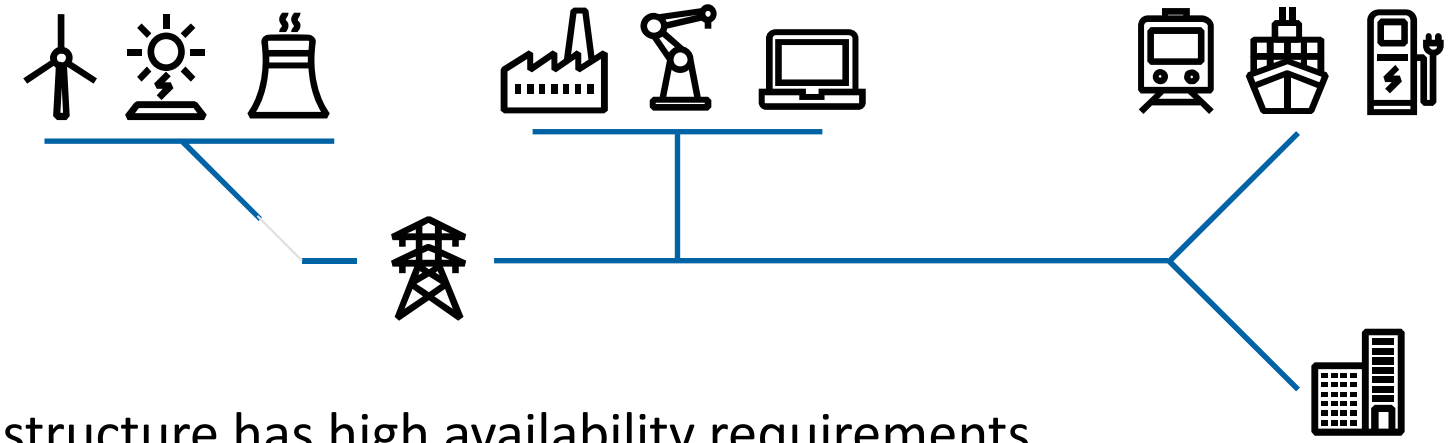
universität wien

BA: What Can(not) Be Perfectly Rerouted Locally



- Critical network infrastructure has high availability requirements
- Hard real-time requirements on packet routing

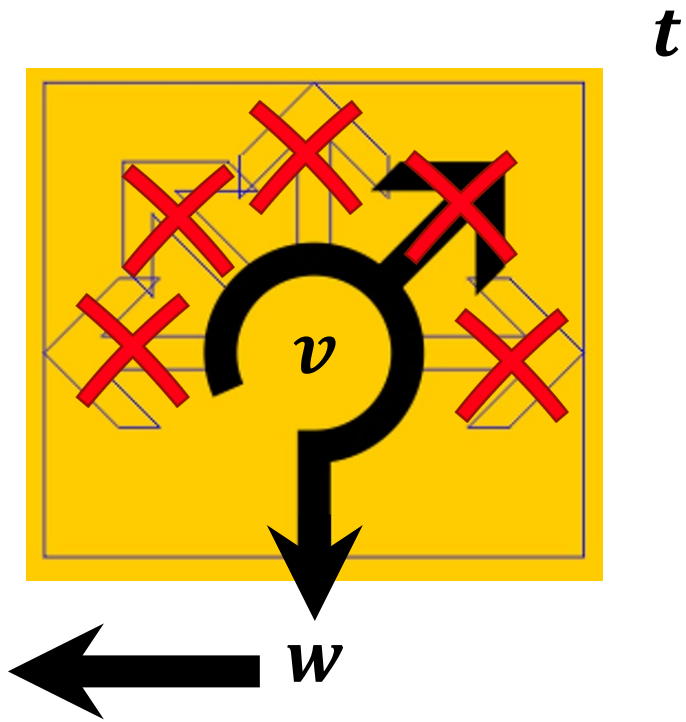
⇒ How to provide dependability guarantee despite edge failures in networks?
⇒ Possible without communication between nodes?

Prior work: Impossible in general [PODC 2012, Feigenbaum et al.]

universität wien

Model

- Network is a connected undirected graph $G = (V, E)$
- Forwarding at node v may only match on:
  1. Destination t
  2. Incident edge failures $F \cap E(v)$
  3. Incoming port from $E(v) \cup \{\perp\}$
- No packet (header) changes allowed, no communication
- Static routing tables, deterministic behavior
- Goal: Install rules ***ahead of time*** to reach destination t under any edge failures F (if connected)



universität wien

Perfect Resiliency on Non-Planar Graphs? Impossible!

- Perfect resiliency on Graph *G* -> **Any subgraph** *G'* of *G* also allows for perfect resiliency
  - Idea: Take routing on *G*, fail edges to create *G'*, routing must still work ☺

- **Contracting** works as well, by a simulation argument

  - Essentially: Take perfectly resilient routing function, show that we can make it work after contraction

- Combined: Perfect resiliency on Graph *G* -> **Any minor** *G'* of *G* as well
  - Definition: G' is minor of G if can be obtained by contracting/subgraphing

- We show $K_5, K_{3,3}$ no perfect resilience -> **non-planar graphs do not allow for perfect resilience**

universität wien

BA: What Can(not) Be Perfectly Rerouted Locally

- Perfect resiliency **impossible**:
  - On **some planar** graphs [already with just 7 nodes] and on **all non-planar** graphs

- Perfect resiliency **possible**:
  - On **some planar** graphs and on **all outerplanar** graphs

- In the extended version (link in paper):
  - Results on **more powerful routing** models, on **rule space size**, further open questions

Full version also appears at SIAM Symposium on Algorithmic Principles of Computer Systems [APOCS'21]