Self-* Breakout Session
Origins (?): Self-Stabilization

Self-stabilizing algorithms pioneered by Dijkstra (1973): for example self-stabilizing mutual exclusion.

“I regard this as Dijkstra’s most brilliant work. Self-stabilization is a very important concept in fault tolerance.”

Leslie Lamport (PODC 1983)
The Idea

Put simple: Recover from any possible state!
Formally

A self-stabilizing system can be started in any arbitrary configuration and will eventually exhibit a desired “legal” behavior.

That is: convergence and closure (once legal, stay legal)
Examples

Radia Perlman: Vision of “Self-Stabilizing Internet”

E.g., self-stabilizing spanning tree protocols

E.g., self-stabilizing SDN control plane

Convergence != stop: *token ring*

---

**Renaissance: A Self-Stabilizing Distributed SDN Control Plane**
Marco Canini, Iosif Salem, Liron Schiff, Elad Michael Schiller, and Stefan Schmid.
38th IEEE International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, July 2018.

**SKIP+: A Self-Stabilizing Skip Graph**
Riko Jacob, Andrea Richa, Christian Scheideler, Stefan Schmid, and Hanjo Täubig.
Journal of the ACM (JACM), Volume 61, Number 6, Article 36, November 2014.
Differences in Self-Driving Networks?

- Networks *measure*, control and optimize themselves
- Reconfiguration *triggered by workload/demand* (e.g., traffic): *data / AI* plays an important role
- More *intent/task driven*: high-level goals
- Goal is „*optimization“, not reaching legal state“
The Ideal Self-Driving Network?

• E.g., perfect prediction:
  – Automatically installs a new firewall rule 1sec before it is needed
Issues (1)

- Properties *during convergence* / moving from one configuration (local optimum) to know one?
  - How long and deep can the valley be?
  - How long are we willing to drive?
  - Maybe even illegal / not policy compliant?

- Concept of *empowerment* from robotics: sometimes suboptimal configuration is better than optimal if it comes with *flexibilities to react* efficiently
  - Are there „robust equilibria“, analogous to game theory?
Issues (2)

• Interesting question: can the network notice itself *when it needs more data*? E.g., sensor data?

• Can the network note itself when decision goes *beyond its capabilities*? When is human decision needed? Like in self-driving cars? Or do *crowdsourcing*?
Issues (3)

• At which scale do we envision self-* networks?

• *Internet-wide*: but then what if there are conflicting goals? Can the network at least detect conflict?

• Self-* systems should be modular and composable
Issues (4)

• Which parts of a network can we make self-* first („let go“)? E.g., *queue management*? Something non-critical? Can we have *formal guarantees as well as AI*?

• What is the cost of migrating to such networks? *Incremental deployment*?
Issues (5)

• Use cases:
  – E.g., *community networks*: naturally distributed resource management
  – E.g., self-optimizing *wifi networks* in houses (across tenants): lots of potential
  – Revisit *self-* storage systems*