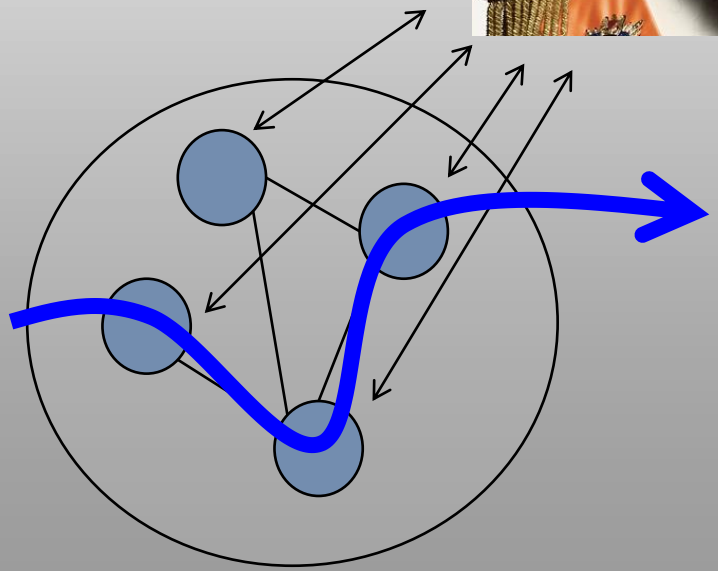
Implementing and Verifying Global SDN Policies with *Near-Sighted Controllers*

Stefan Schmid (TU Berlin & T-Labs)

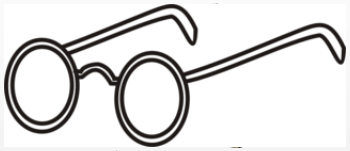
Centralized Control: I ❤(ed) SDN!

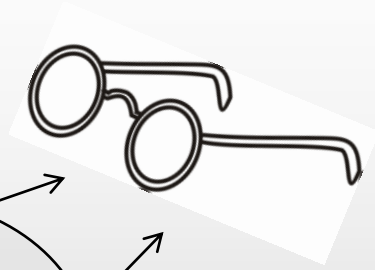
- Abstraction: “Simple” network management from central vantage point
- Global control of traffic
- Allows to express global network policies, e.g., load-balancing, adaptive monitoring / heavy hitter detection, …

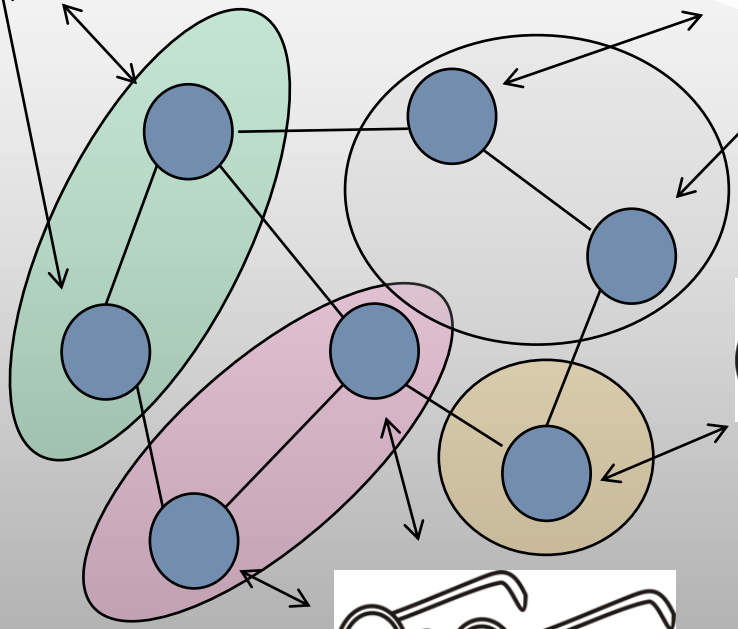


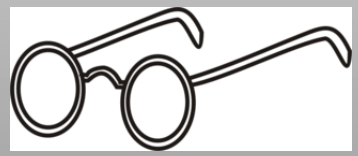
Distributed Control

Network Partition ("flat control"):

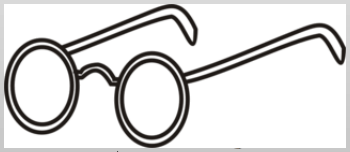


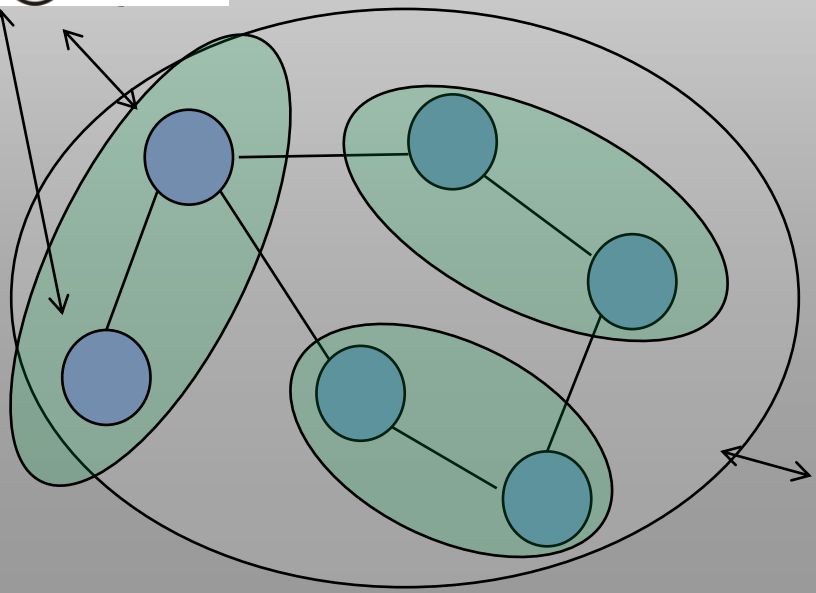


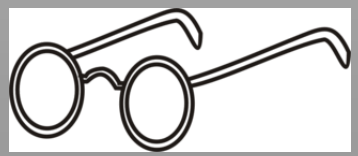




Function Partition ("hierarchical control"):

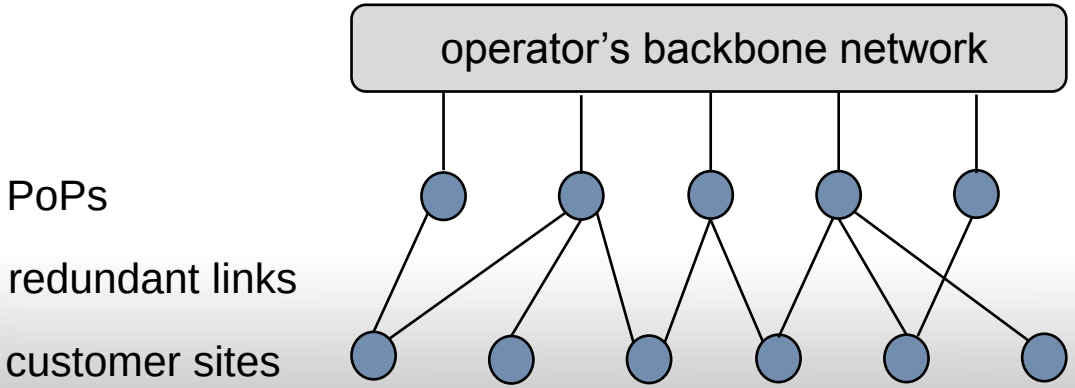




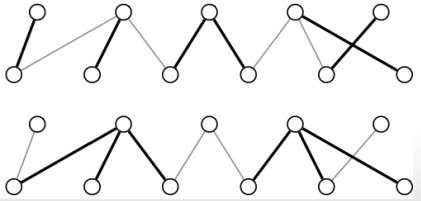


Solving Global Problems Locally: Two Use Cases

SDN Task 1: Resource Allocation / Load Balancing

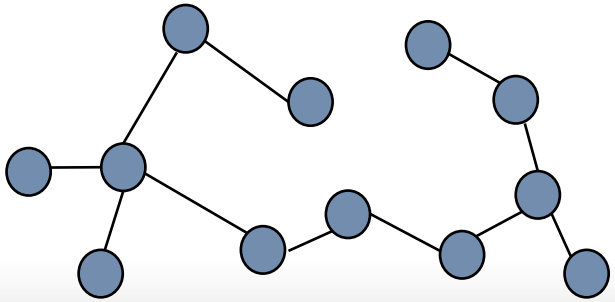


- Bipartite: customer to access routers
- How to assign?

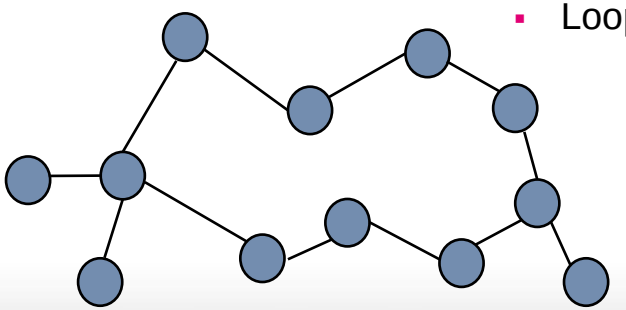


- Quick and balanced?

SDN Task 2: Loop-free Forwarding Verification



OK

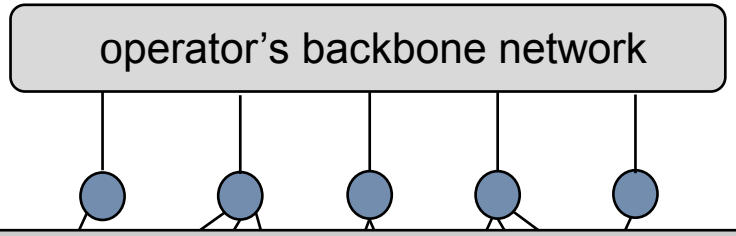


not OK

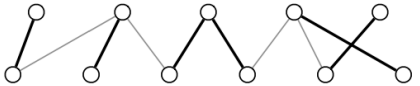
- Loop-free forwarding set?

Solving Global Problems Locally: Two Use Cases

SDN Task 1: Resource Allocation / Load Balancing



- Bipartite: customer to access routers
- How to assign?



PoPs

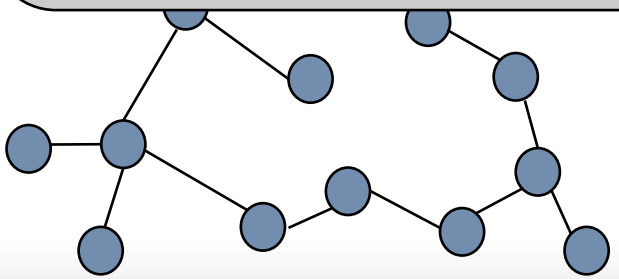
re

cu

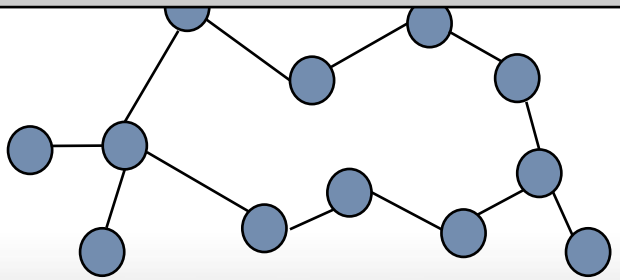
Given an up-to-date network view:

Trivial under a single controller!

g set?



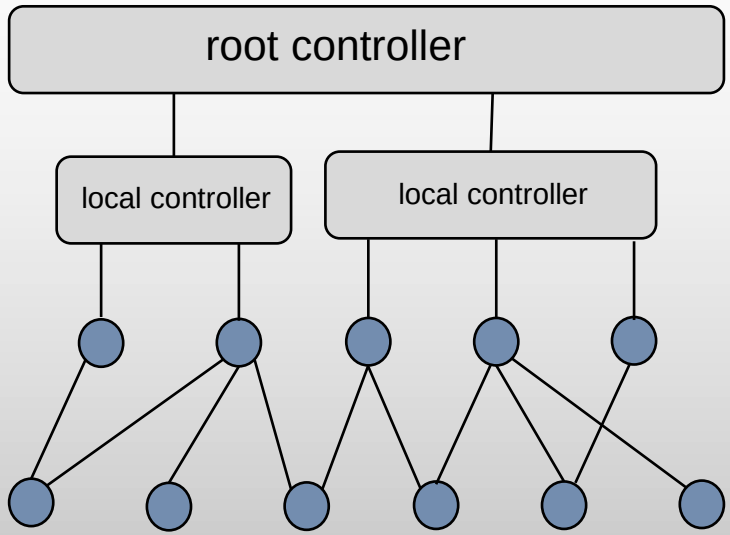
OK



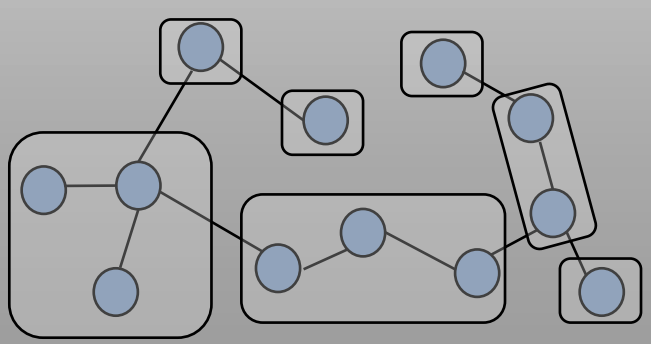
not OK

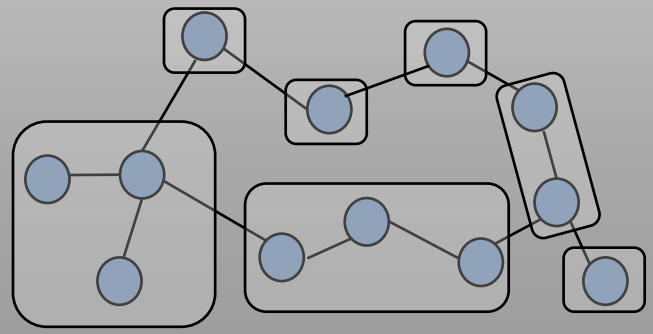
Under a Distributed Control Plane

- Hierarchical control:



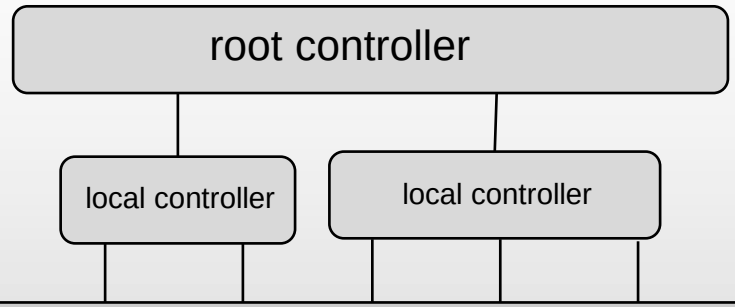
- Flat control:





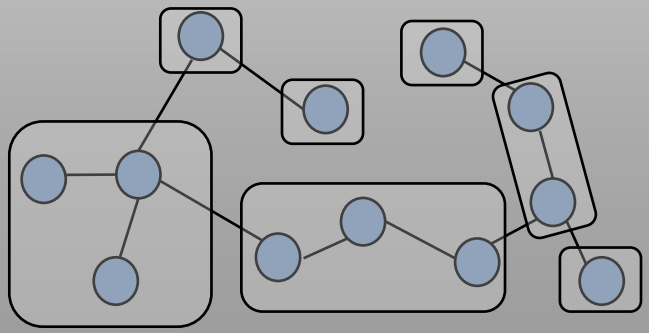
Under a Distributed Control Plane

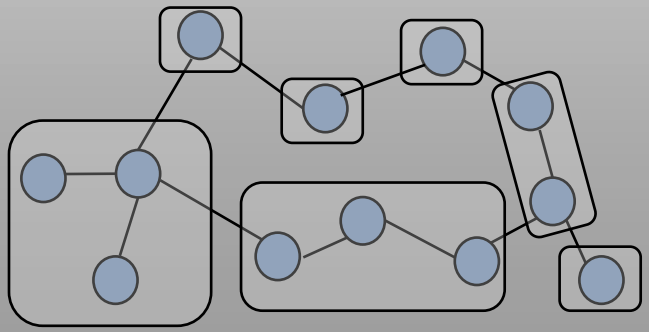
- Hierarchical control:



How to compute matchings here? And how to verify loop-free forwarding?

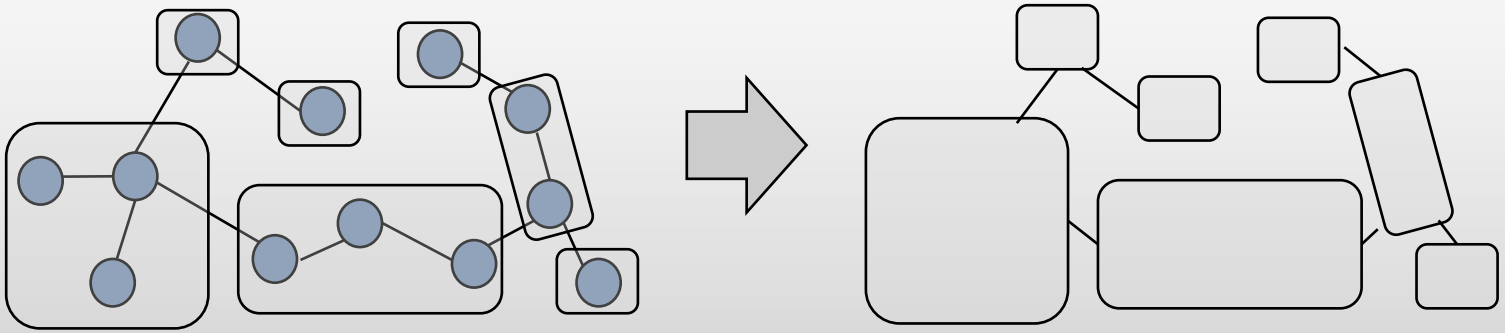
- Flat





Controller Graph and the Need for Communication

Controllers need to communicate to discover loops!



How much communication is needed?

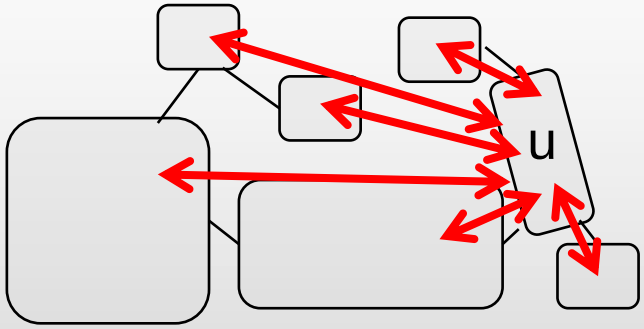
How to bound the cost?

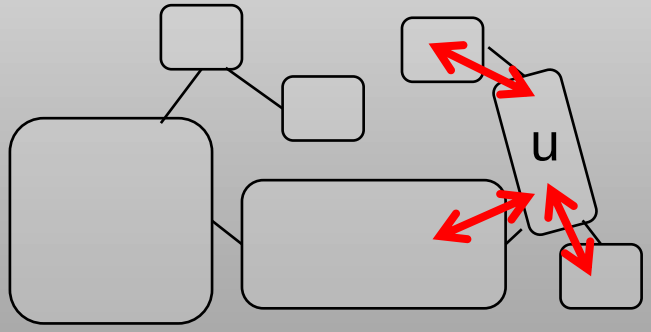
Some tasks are inherently global! E.g.: loop verification.

Others can be solved locally. E.g.: link assignment.

Local vs Global

Global task: inherently need to respond to events occurring at all devices.

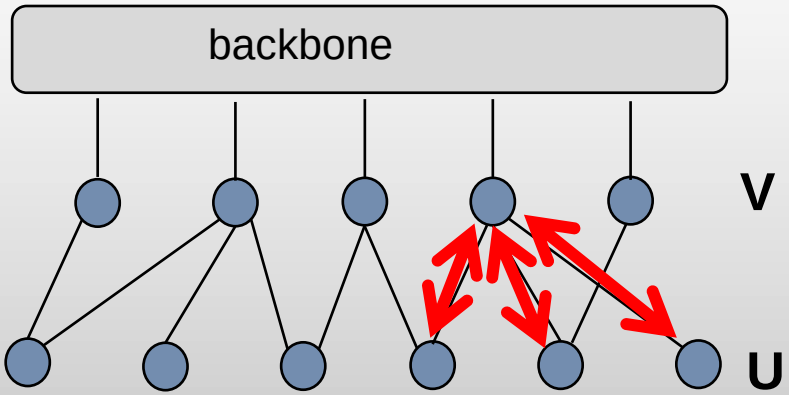




Local task: sufficient to respond to events occurring in vicinity!

From Local Algorithms to SDN: Link Assignment

A semi-matching problem:

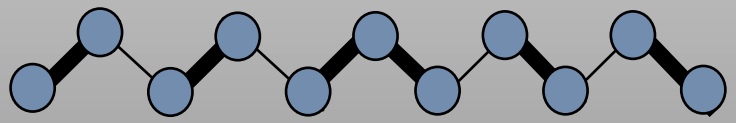


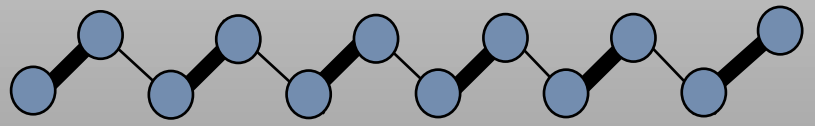
Semi-matching

If a customer u connects to a POP with c clients connected to it, the customer u costs c.

Minimize the average cost of customers!

The bad news: Generally the problem is inherently global e.g.,





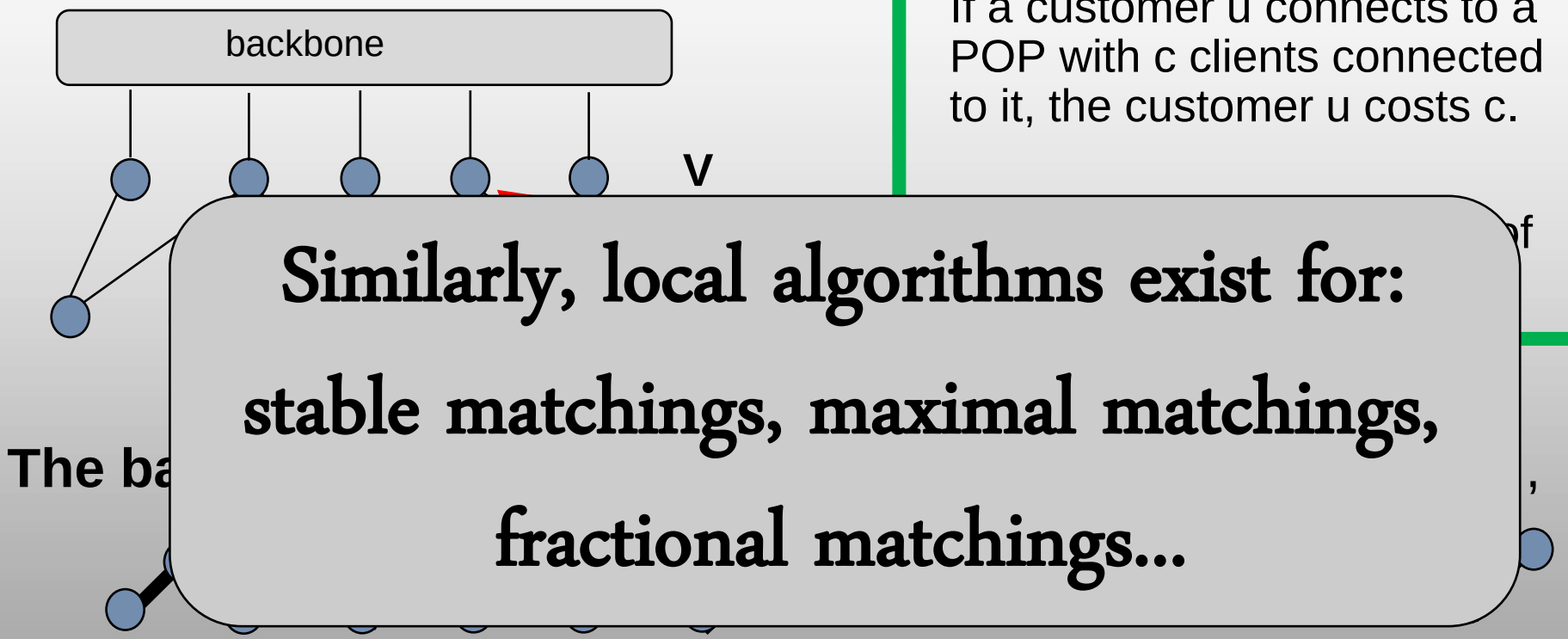
The good news: Near-optimal solutions can be found efficiently and locally! Runtime independent of graph size and local communication only.

From Local Algorithms to SDN: Link Assignment

A semi-matching problem:

Semi-matching

If a customer u connects to a POP with c clients connected to it, the customer u costs c.

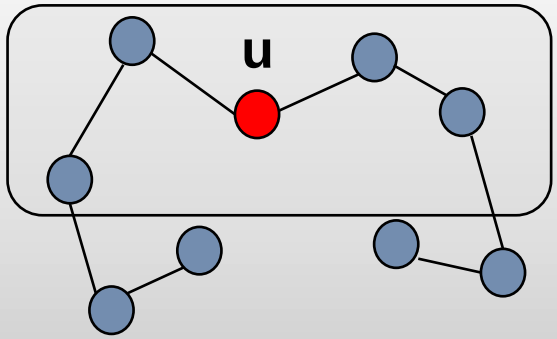


Similarly, local algorithms exist for: stable matchings, maximal matchings, fractional matchings...

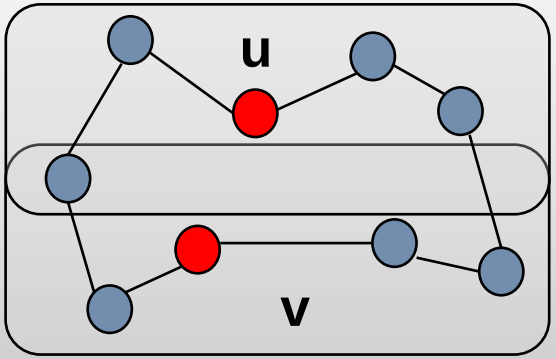
The good news: Near-optimal solutions can be found efficiently and locally! Runtime independent of graph size and local communication only.

From Local Algorithms to SDN: Spanning Tree

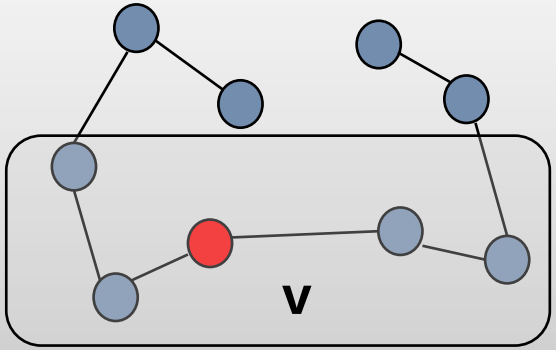
Bad news: Spanning tree verification is an inherently global task.



OK



not OK



OK

2-hop local views of contrullers u and v: in the three examples, at least one node cannot distinguish the local view of a good instance from the local view of the bad instance.

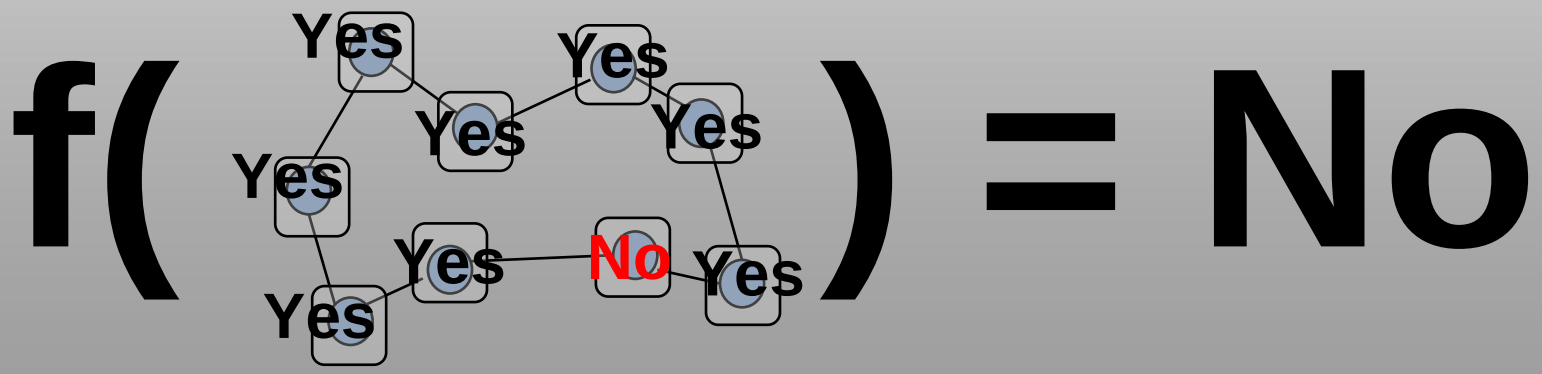
Good news: Proof labeling systems may allow for local verification of spanning tree property.

Proof Labeling Scheme

Idea: Often it is sufficient if at least one controller notices inconsistency: it can then trigger global re-computation

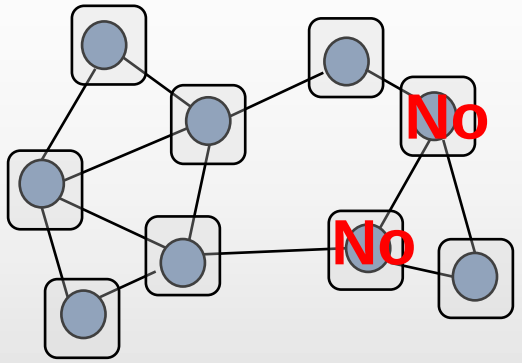
Requirements:

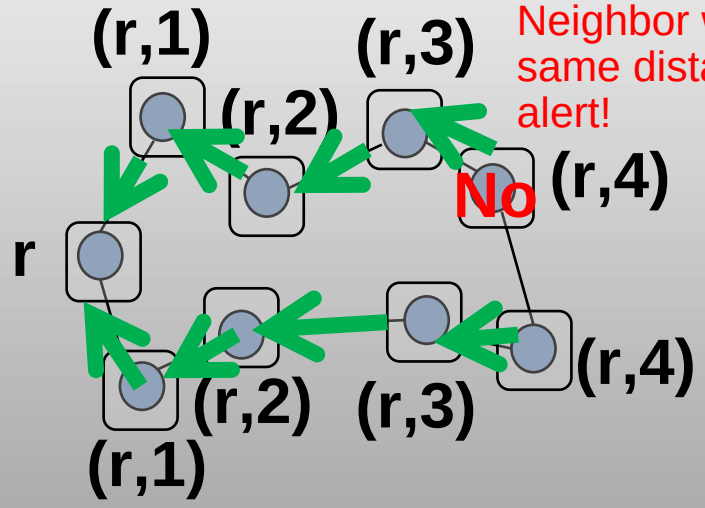
- Controllers exchange minimal amount of information ("proofs labels")
- Proof labels are small
- Communicate only with controllers with incident domains
- Verification: if property not true, *at least one* controller will notice…
- … and raise alarm (re-compute labels)



Verification is Easier than Computation!

Euler Property: Hard to compute Euler tour ("each edge exactly once"), but easy to verify! 0-bits (= no communication) : output whether degree is even.





Spanning Tree Property: Label encodes root node plus distance & direction to root. At least one node notices that root/distance not consistent! Requires O(log n) bits.

Any (Topological) Property: $O(n^2)$ bits.

And……..: concurrency and consistency