Concurrent Self-Adjusting Distributed Tree Networks

Bruna Peres

Olga Goussevskaia

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Stefan Schmid

AALBORG UNIVERSITY
DENMARK

Chen Avin

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

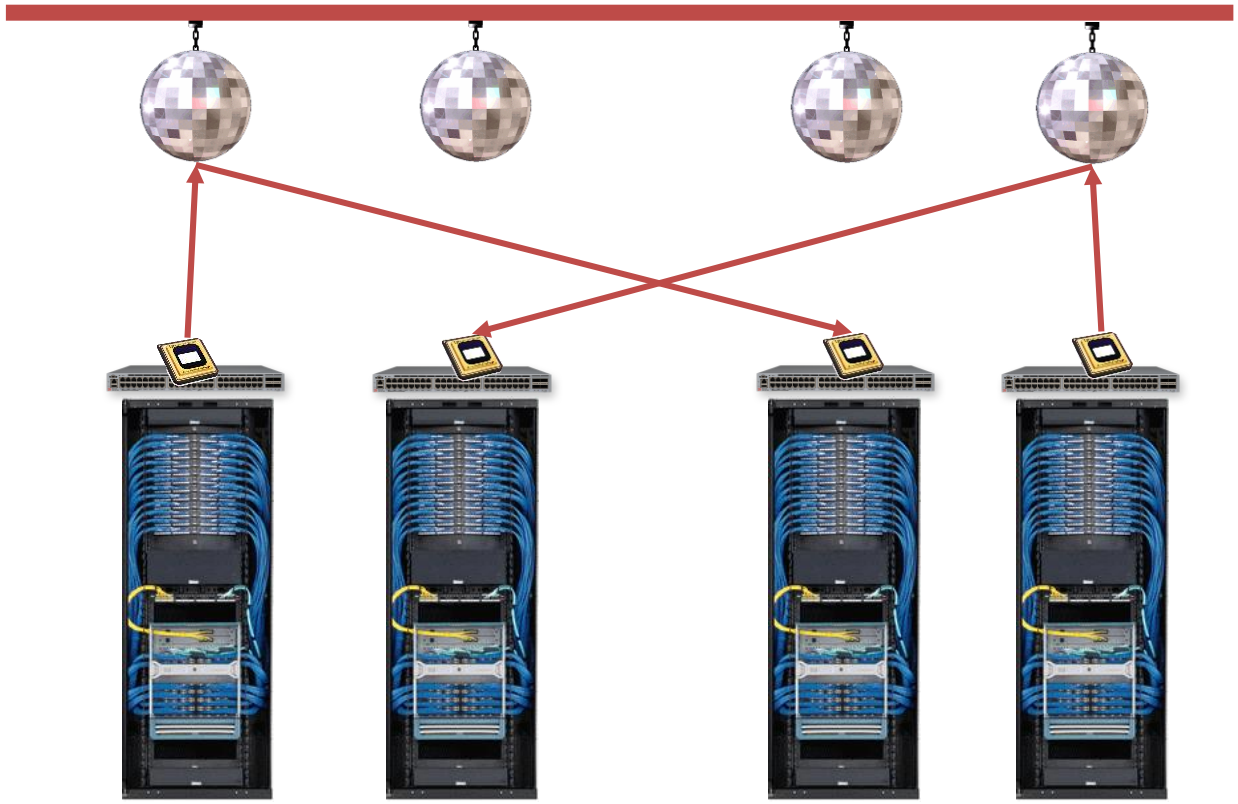
Motivation

- New technologies allow communication networks to be increasingly flexible and reconfigurable

- Traditional networks designs are still optimized toward static metrics

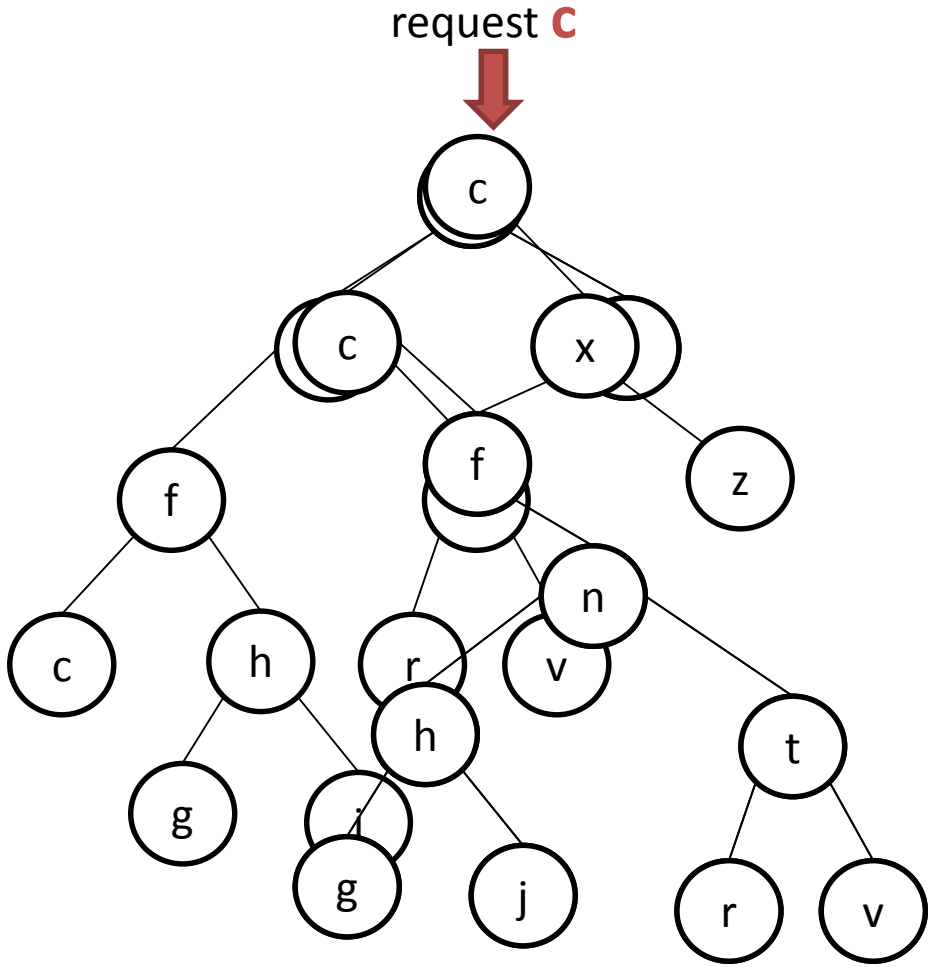
ProjecToR

- ProjecToR: Agile Reconfigurable Data Center Interconnect. Ghobadi et al., SIGCOMM'16



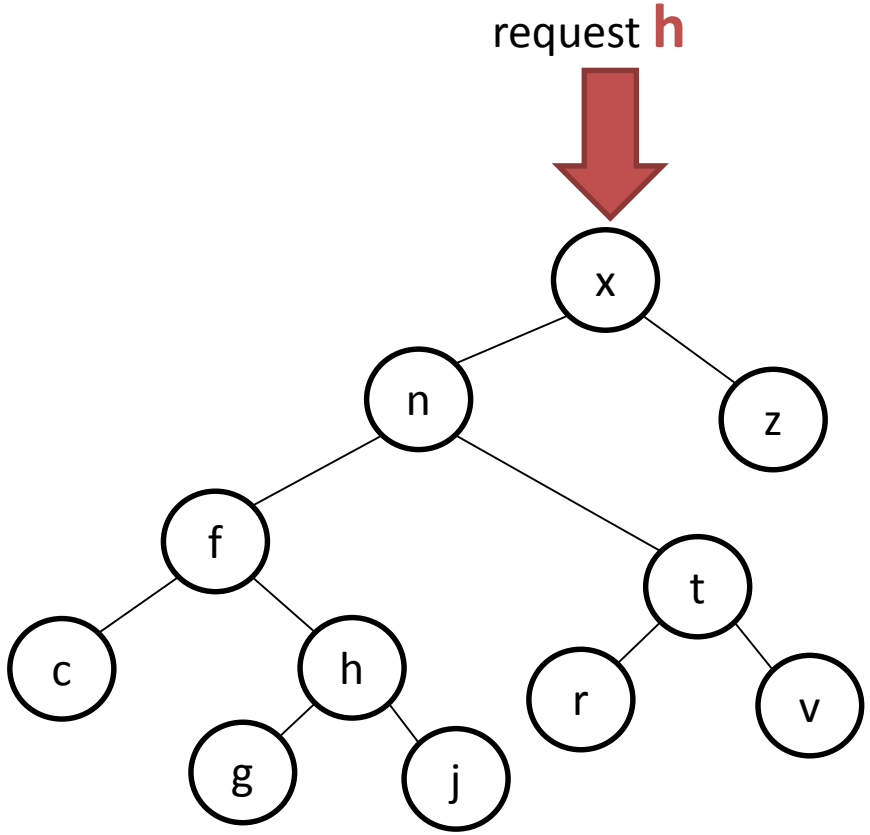
Self-Adjusting Data Structures

- Self-adjusting networks ↔ self-adjusting data structures
- Splay Trees

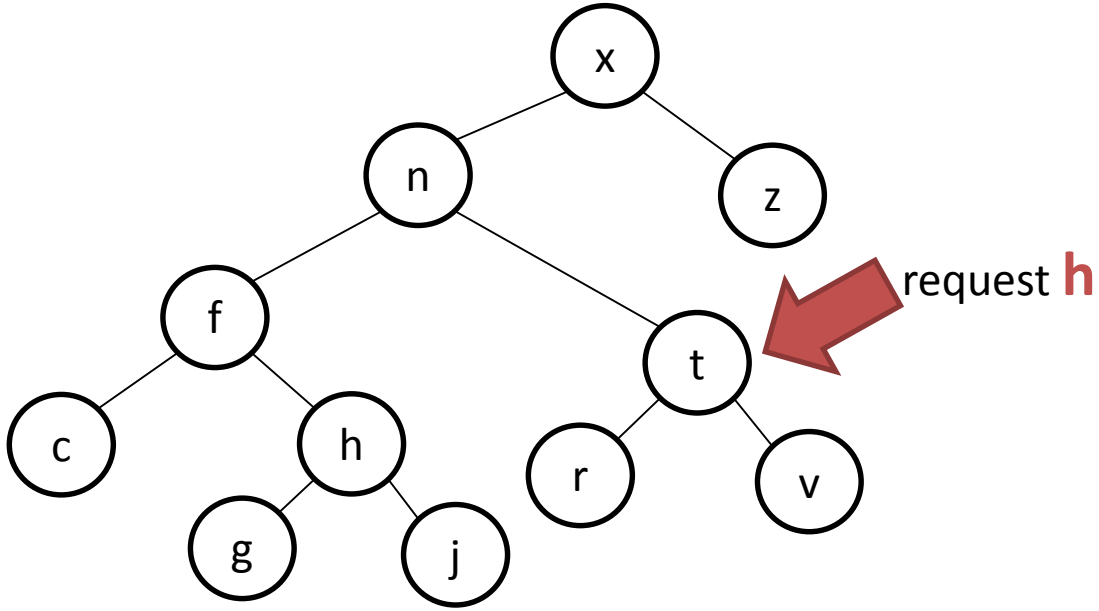


SplayNets

S. Schmid, et al., SplayNet: Towards Locally Self-Adjusting Networks
IEEE/ACM Transactions on Networking, 2016.



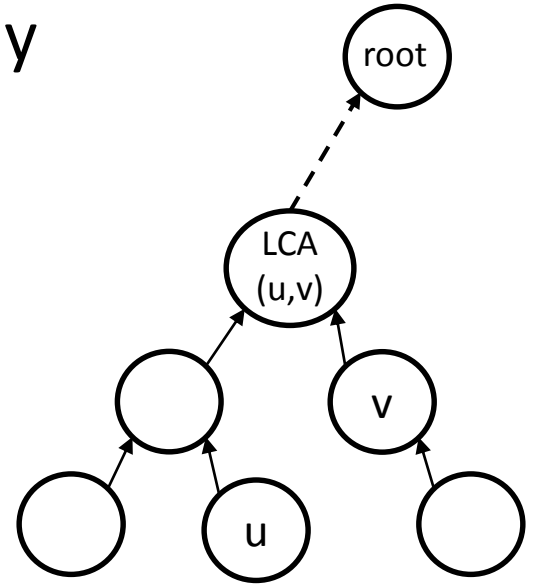
SplayTree



SplayNet

SplayNets

- Distributed tree network
- Improves the communication cost between two nodes in a self-adjusting manner
- Nodes communicating more frequently become topologically closer to each other over time
- *Lowest common ancestor LCA(u,v):* locality is preserved



Our Contributions

- While SplayNets are inherently intended to distributed applications, so far, only sequential algorithms are known to maintain SplayNets

- We present DiSplayNets, the first distributed and concurrent implementation of SplayNets

Model

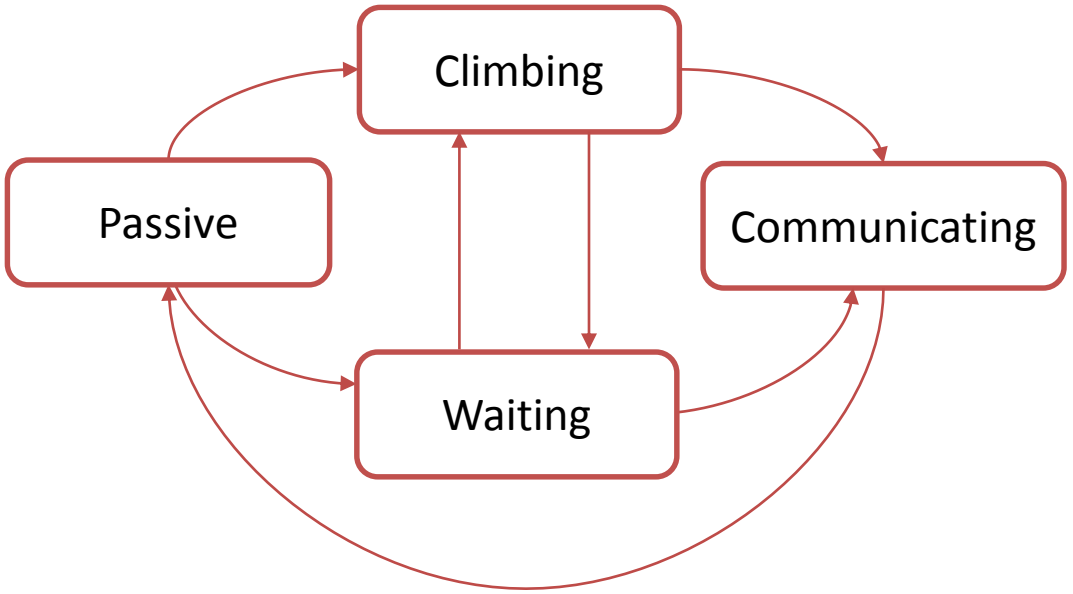
- Network model:
  - Binary tree T comprised of a set of n communication nodes

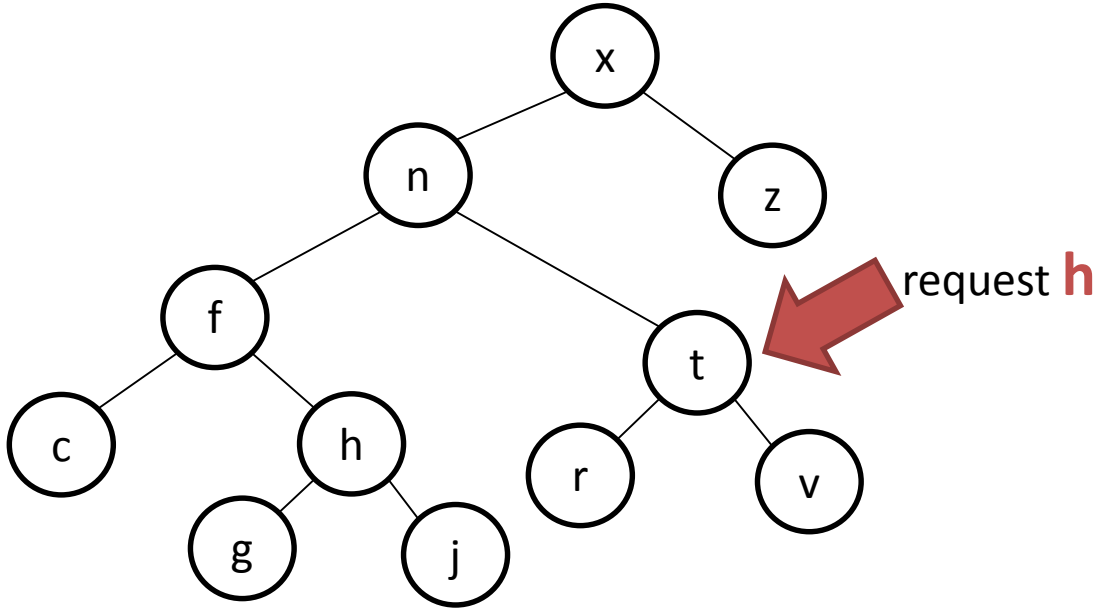
- Sequence of communication requests $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_m)$:
  - $\sigma_i = (s, d)$
  - $t_b(\sigma_i)$ and $t_e(\sigma_i)$

- Given $\sigma_i(s, d)$, s and d rotate in parallel towards the LCA(s,d)
  - LCA might change over time

DiSplayNet

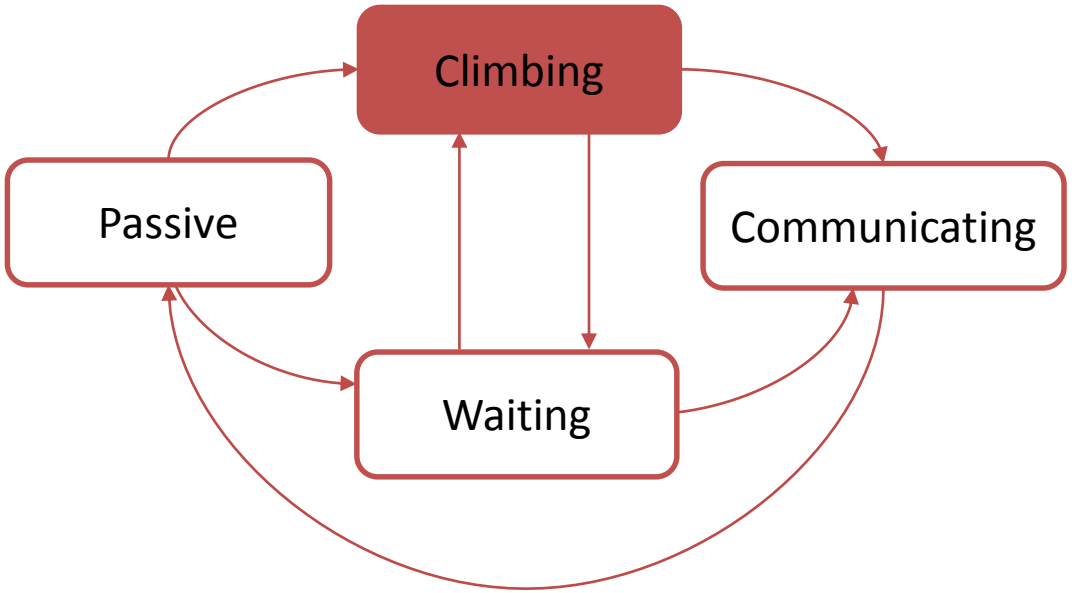
- State machine executed by each node in parallel

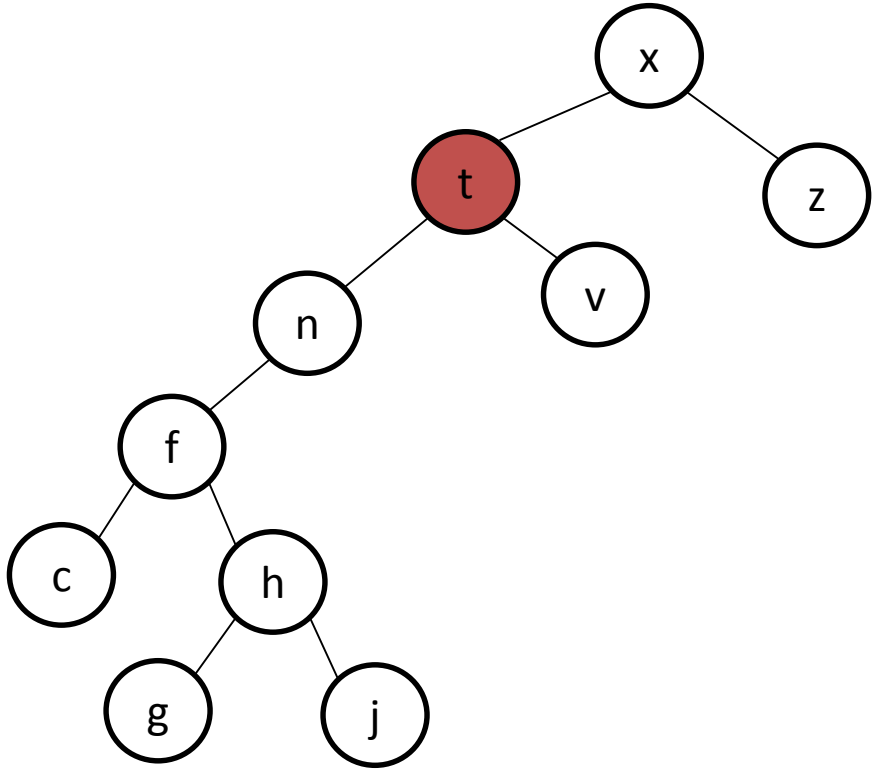




DiSplayNet

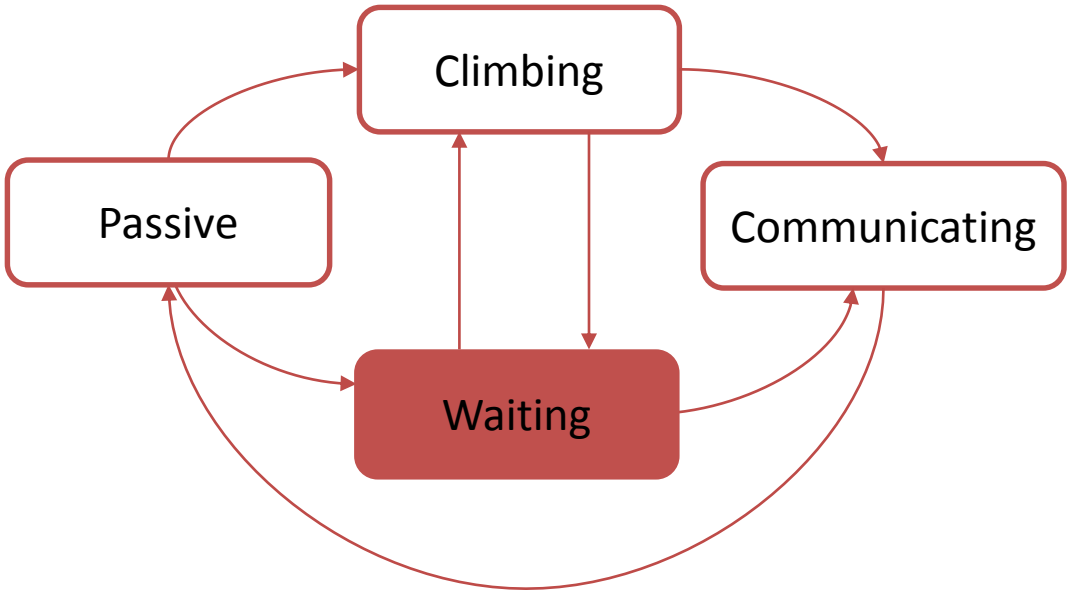
- State machine executed by each node in parallel

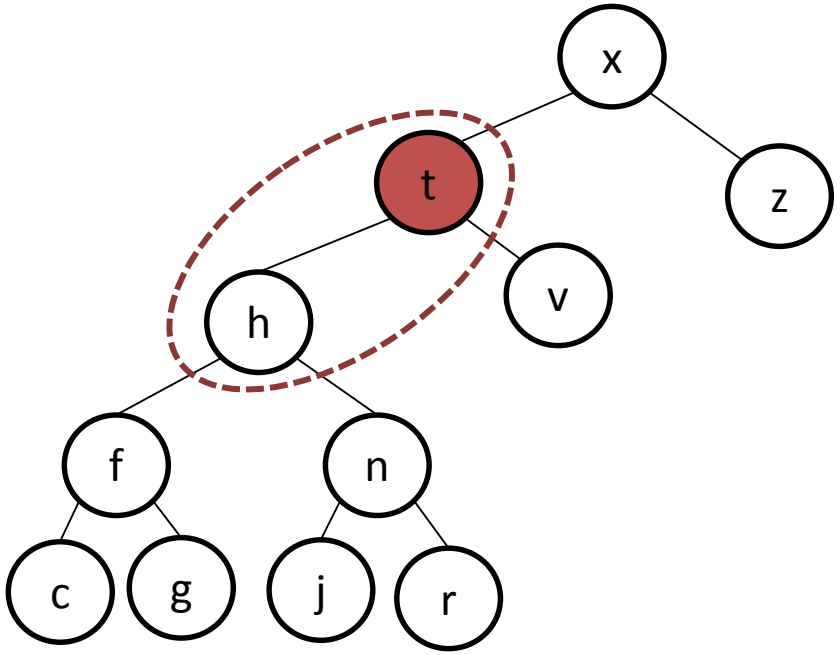




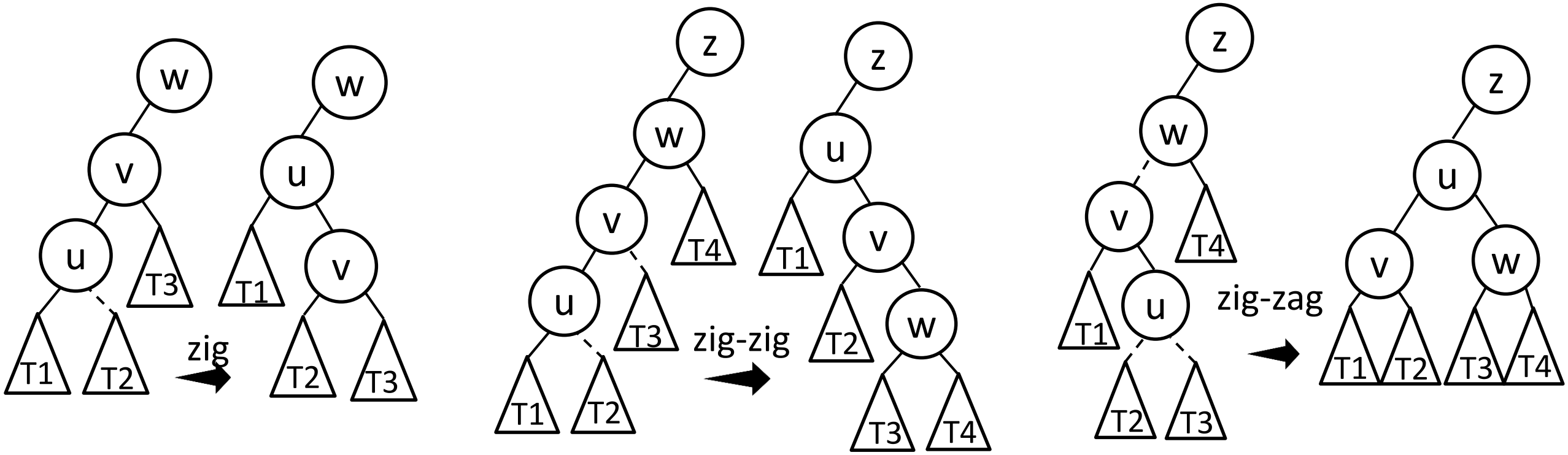
DiSplayNet

- State machine executed by each node in parallel

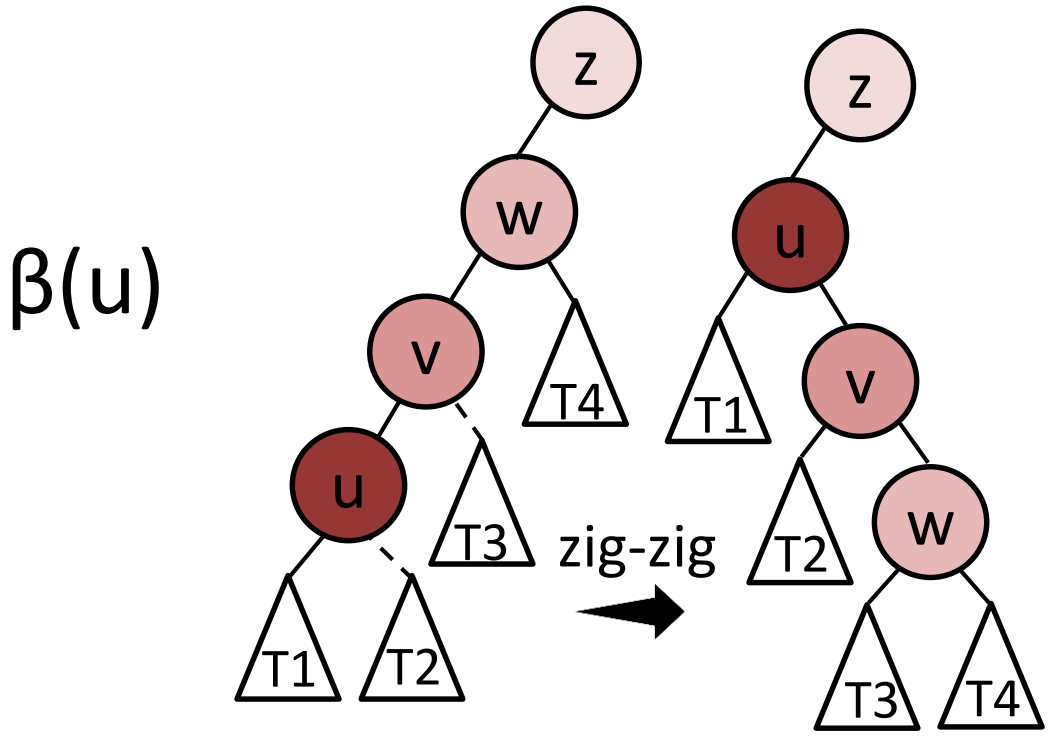




Local reconfigurations



Local reconfigurations



DiSplayNet

- In order to ensure deadlock and starvation freedom, concurrent operations are performed according to a priority

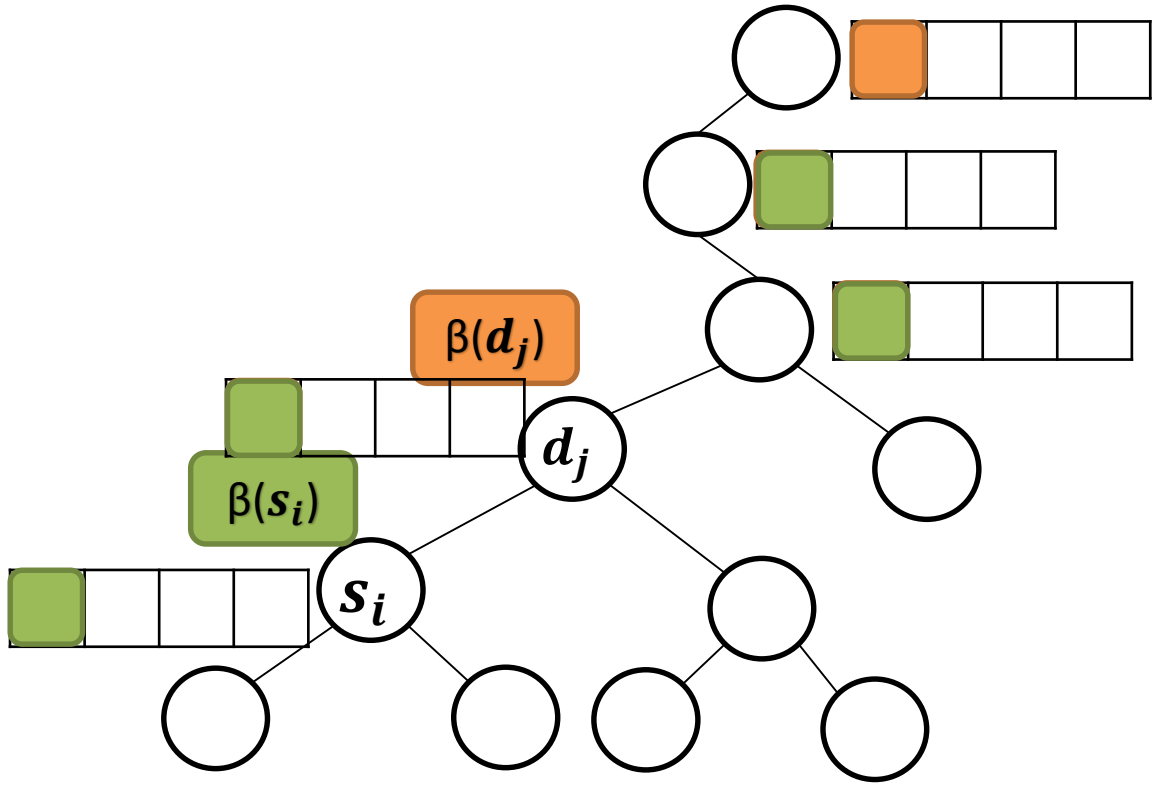
$$t_b(\sigma_i(s_i, d_i)) < t_b\big(\sigma_j(s_j, d_j)\big)$$

Algorithm

- The algorithm is executed in rounds

$$t_b(\sigma_i(s_i, d_i)) < t_b\big(\sigma_j(s_j, d_j)\big)$$

Phase 1

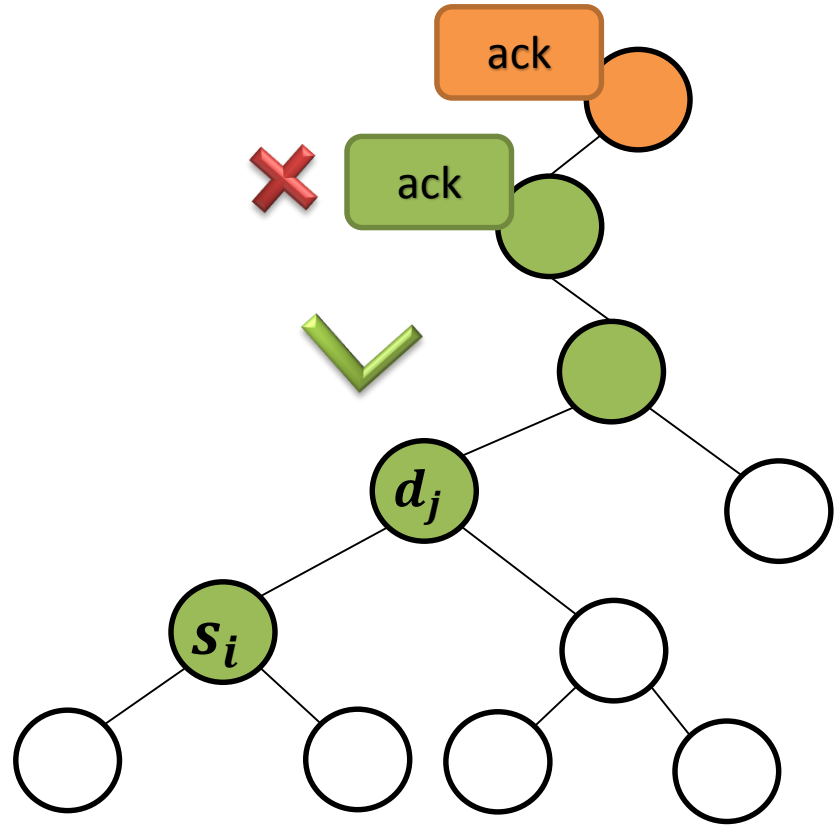


Algorithm

- The algorithm is executed in rounds

$$t_b(\sigma_i(s_i, d_i)) < t_b(\sigma_j(s_j, d_j))$$

Phase 2

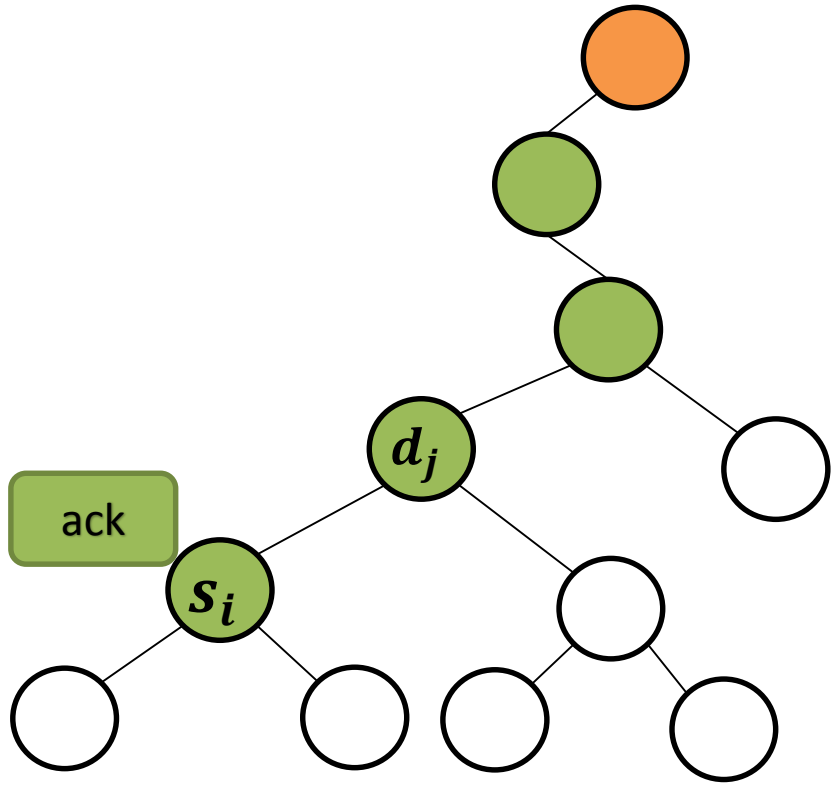


Algorithm

- The algorithm is executed in rounds

$$t_b(\sigma_i(s_i, d_i)) < t_b\big(\sigma_j(s_j, d_j)\big)$$

Phase 3

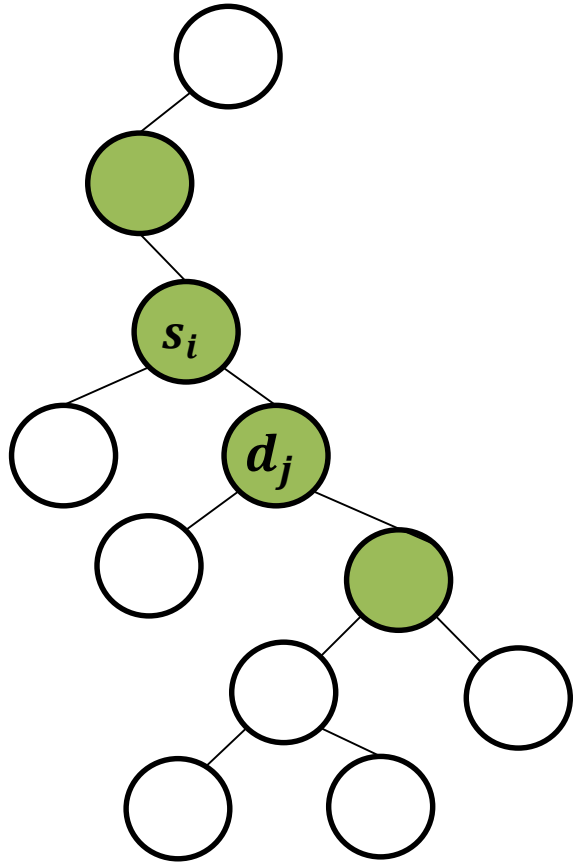


Algorithm

- The algorithm is executed in rounds

$$t_b(\sigma_i(s_i, d_i)) < t_b(\sigma_j(s_j, d_j))$$

Phase 4

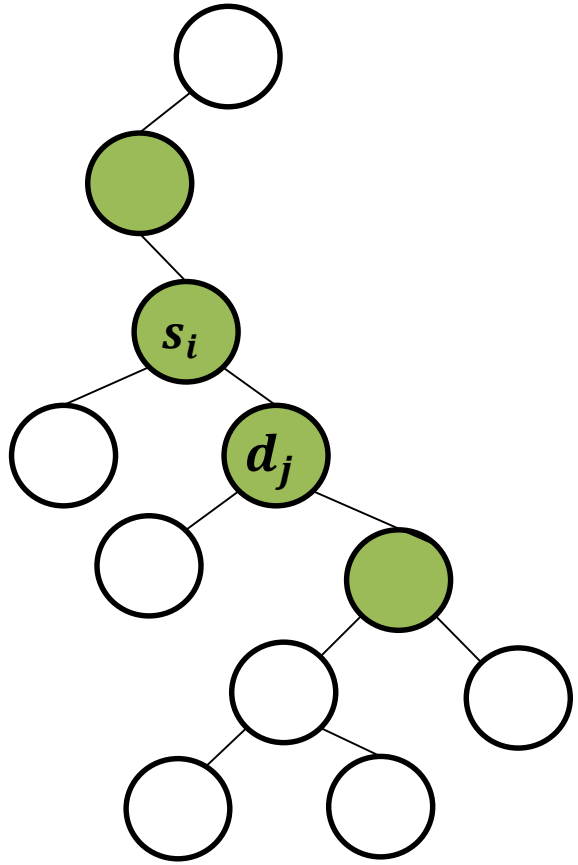


Algorithm

- The algorithm is executed in rounds

$$t_b(\sigma_i(s_i, d_i)) < t_b(\sigma_j(s_j, d_j))$$

Phase 5



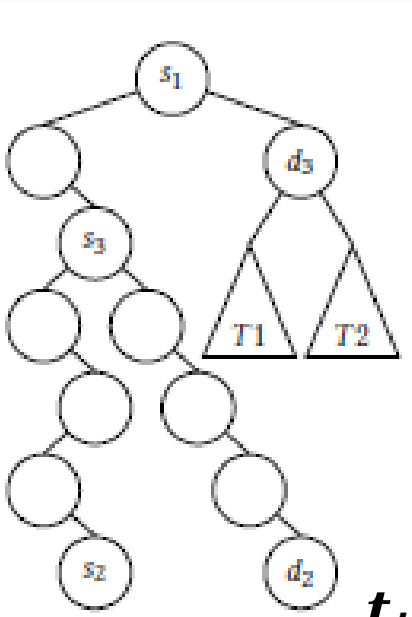
DiSplayNets

- Self-adjust to the communication pattern in a fully-decentralized manner
  - Starvation free
  - Deadlock free

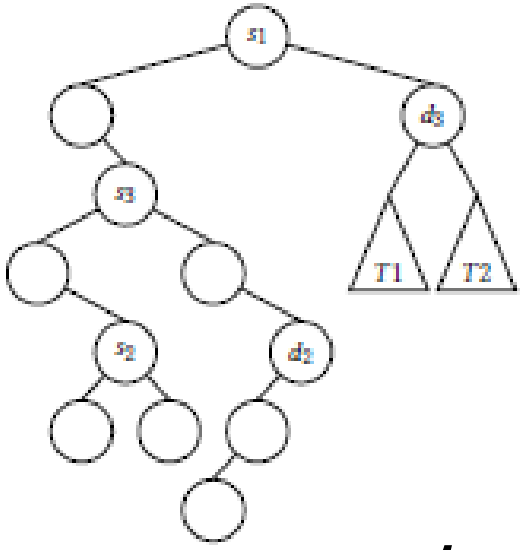
Future Work

- Analyze the efficiency
  - Work cost: $W(DiSplayNet, T_0, \sigma) = \sum_{\sigma_i \epsilon \sigma} w(\sigma_i)$
  - Time cost:
    - Request delay: $t_d(\sigma_i) = t_e(\sigma_i) - t_b(\sigma_i)$
    - Makespan: $T(T_0, \sigma) = \max_{\sigma_i \epsilon \sigma} t_e(\sigma_i) - \min_{\sigma_i \epsilon \sigma} t_b(\sigma_i)$

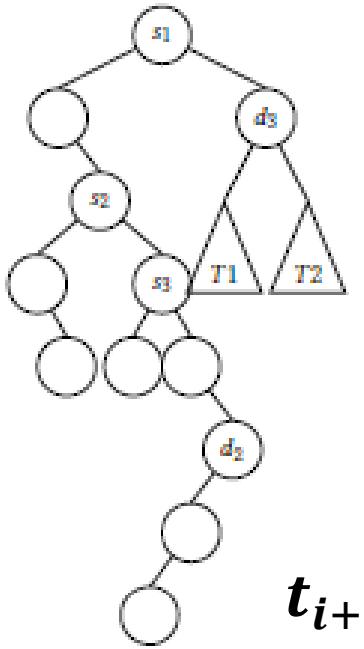
Progress Matrix



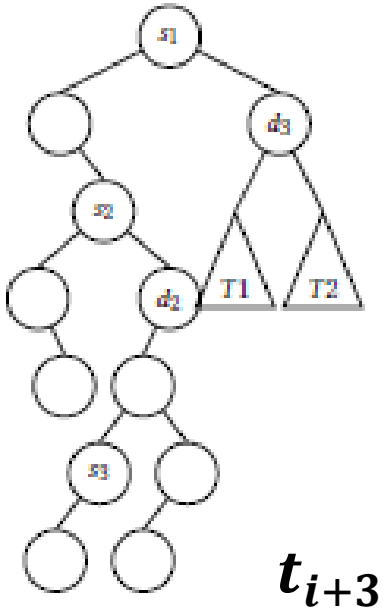
t_i



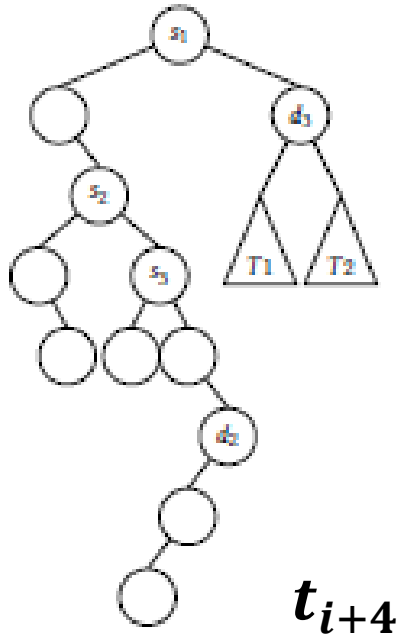
t_{i+1}



t_{i+2}



t_{i+3}



t_{i+4}

	t_1	t_2	...	t_i	t_{i+1}	t_{i+2}	t_{i+3}	t_{i+4}	t_{i+5}	...	t_j	...	t_k
s_1	✓	✓	...	✓	✓	✓	✓	✓	✓	...	-	...	-
d_1	✓	✓	...	✓	✓	✓	✓	✓	✓	...	-	...	-
s_2	✓	✓	...	✓	✓	✓	✓	-	-	...	-	...	-
d_2	✓	✓	...	✓	✓	✗	✓	-	-	...	-	...	-
s_3	✓	✓	...	✗	✗	✗	✗	✓	✗	...	✓	...	-
d_3	✗	✗	...	✗	✗	✗	✗	✗	✗	...	✓	...	-

Progress Matrix

Makespan

	t_1	t_2	...	t_i	t_{i+1}	t_{i+2}	t_{i+3}	t_{i+4}	t_{i+5}	...	t_j	...	t_k
s_1	✓	✓	...	✓	✓	✓	✓	✓	✓	...	-	...	-
d_1	✓	✓	...	✓	✓	✓	✓	✓	✓	...	-	...	-
s_2	✓	✓	...	✓	✓	✓	✓	-	-	...	-	...	-
d_2	✓	✓	...	✓	✓	✗	✓	-	-	...	-	...	-
s_3	✓	✓	...	✗	✗	✗	✗	✓	✗	...	✓	...	-
d_3	✗	✗	...	✗	✗	✗	✗	✗	✗	...	✓	...	-

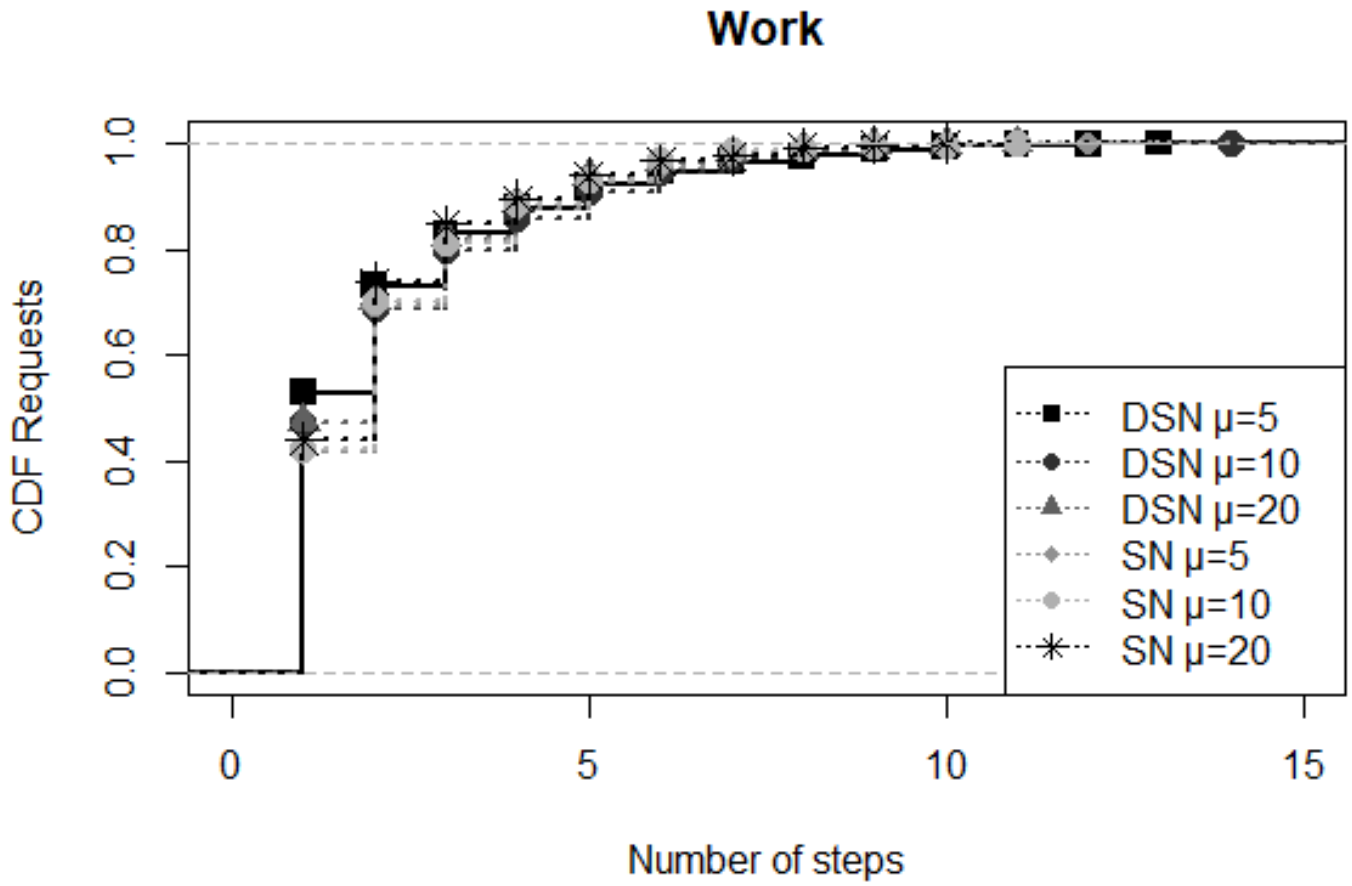
Work

Future Work

- Our simulations show first promising results
  - ProjecToR data
    - 128 node randomly selected from 2 production clusters (running a mix of workloads, including MapReduce-type jobs, index builders, and database and storage systems)
  - 1000 requests
  - Poisson process

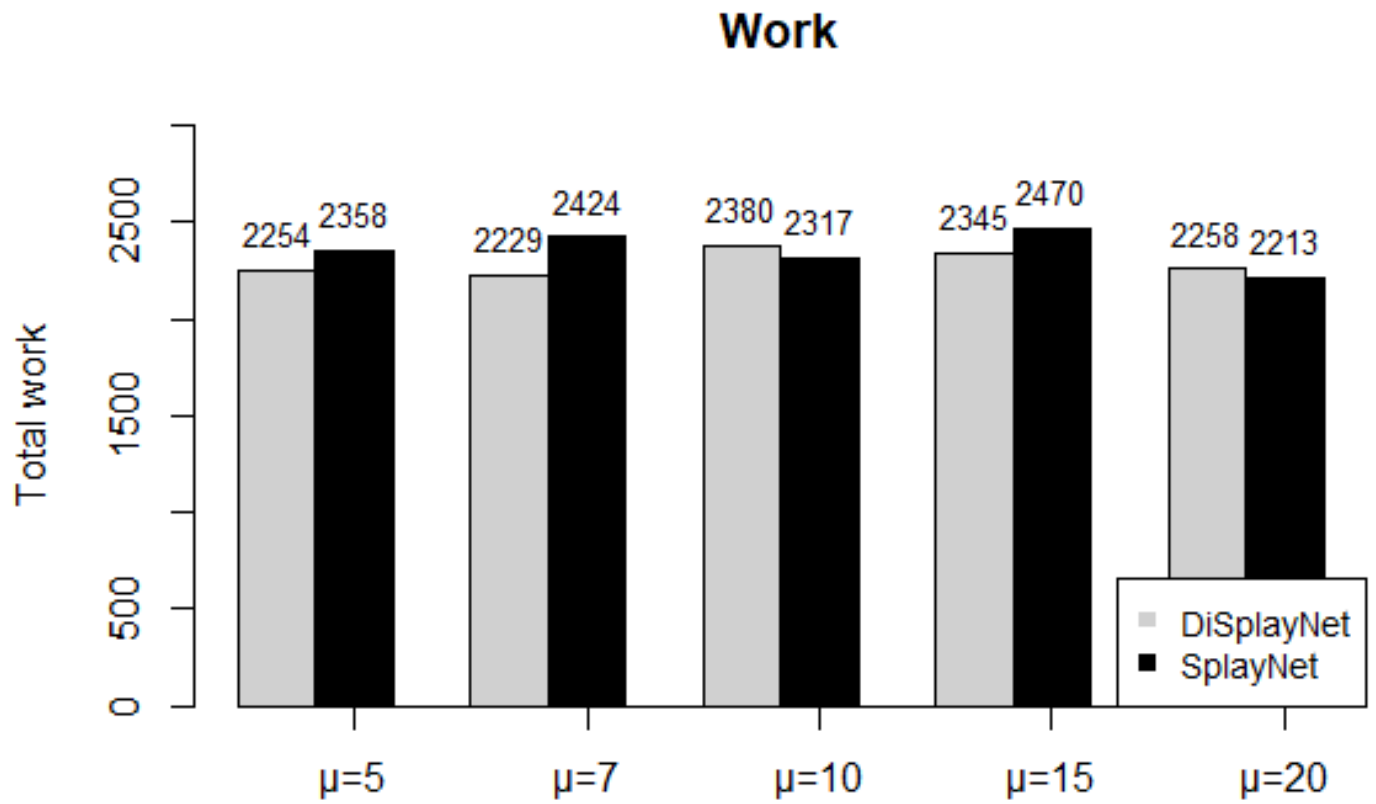
Future Work

- Our simulations show first promising results
  - Individual work CDF



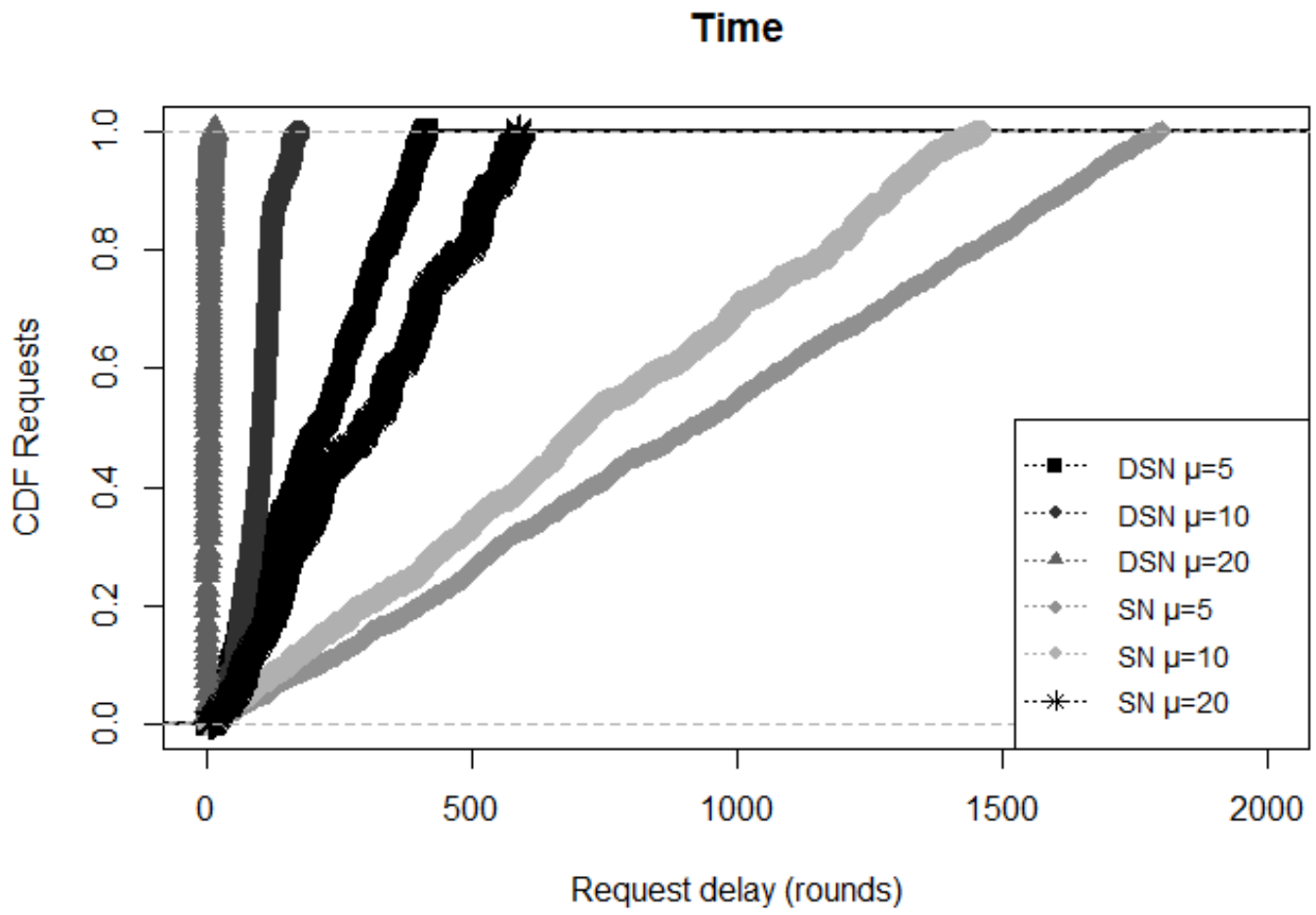
Future Work

- Our simulations show first promising results
  - Total work



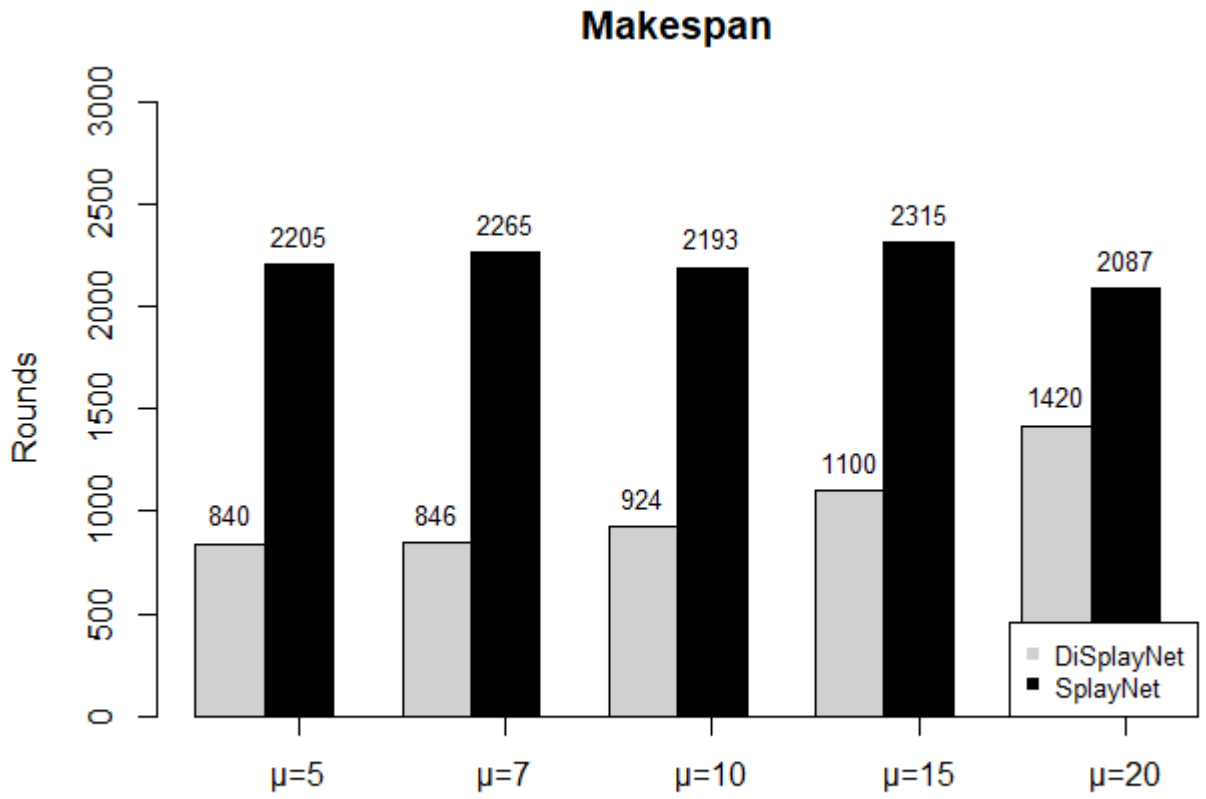
Future Work

- Our simulations show first promising results
  - Request delay



Future Work

- Our simulations show first promising results
  - Makespan



Thank you

Bruna Peres

bperes@dcc.ufmg.br