# Implementation of a Distributed Computation Framework

Matej Pavlovič

École Polytechnique Fédérale de Lausanne
School of Computer and Communication Sciences
Distributed Programming Laboratory
Supervisors: Prof. Rachid Guerraoui (EPF Lausanne)
Univ.Prof. Dipl.-Ing. Dr.techn. Ulrich Schmid (TU wien)

## Motivation and Task

- Large-scale distributed applications, thousands of nodes
- Nodes crash / misbehave
- Many applications require node addition / removal / replacement at run-time
- Need for robust and scalable algorithms

⇓

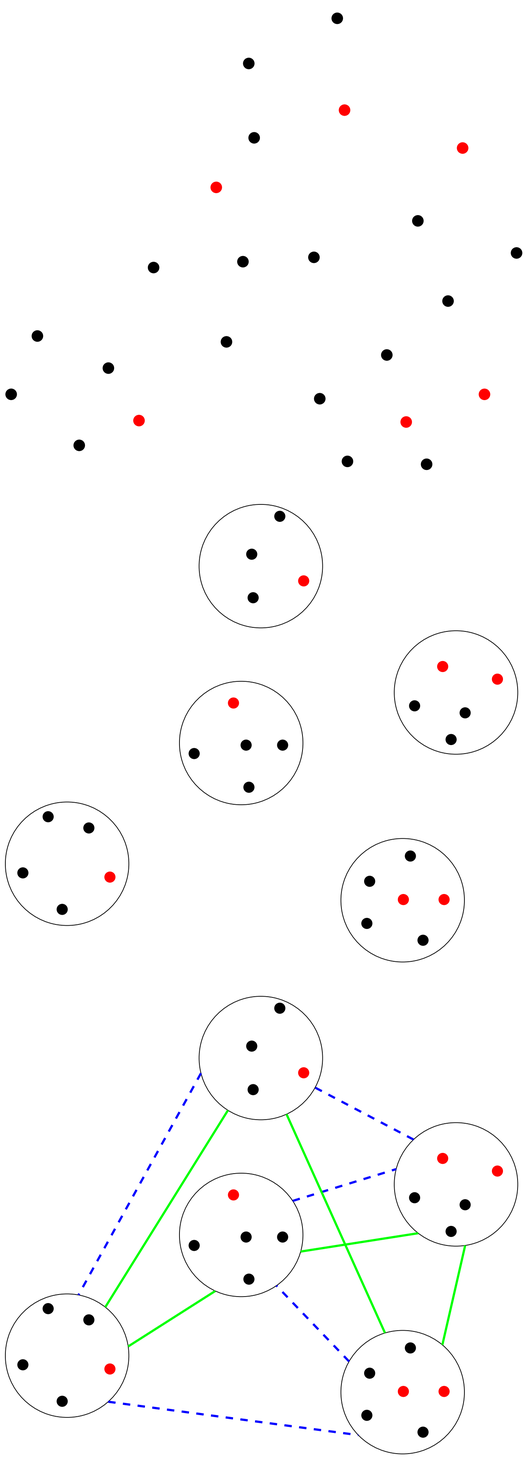
**Task:**
**implement a framework for fault-tolerant scalable distributed computation in a dynamic environment**

## Context

- Fully connected network of nodes
- Communication by message passing
- Direct communication channels
- Synchronized time
- Nodes join / leave at run-time
- Byzantine adversary

## Algorithm in a Nutshell

The core algorithm was proposed by [1].



### Initial state assumption

Start with a network of interconnected nodes with synchronized clocks. A certain fraction of the nodes may be Byzantine-faulty (i.e. fail arbitrarily).

### Clusterization

Partition the network into clusters of small size (polylogarithmic in terms of network size). With high probability, every cluster contains a majority of honest nodes.

### Cluster interconnection

Interconnect the clusters by logical links. The links and the clusters form an expander graph (so-called $\mathbb{H}$-Graph [2]) consisting of multiple random Hamilton cycles.

### Node exchange

Exchange all nodes of a cluster for nodes picked uniformly at random from the whole network whenever a node joins or leaves this cluster. The exchange operation ensures an honest majority in the cluster with high probability.

## Fault-Tolerant Communication Scheme

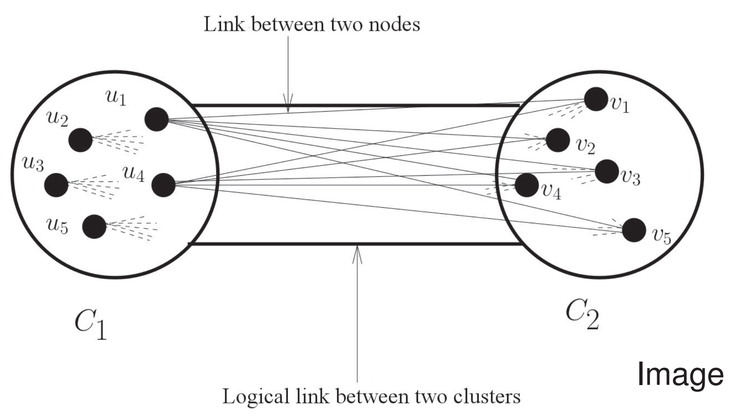



Image source: [1]

A message logically sent by one cluster to another cluster is actually sent by every node of the sending cluster to every node of the the receiving cluster. A node only accepts messages received from the majority of the sending cluster's nodes. This ensures that only correct messages are accepted, given that every cluster contains a majority of honest nodes.

## Results and Conclusion

Starting from scratch, we implemented a working prototype of the distributed computation framework that is able to handle a dynamic set of nodes and on top of which problems such as broadcast or node sampling can be efficiently solved.
The functionality is somewhat limited and additional restrictive assumptions had to be made on the nodes as well as on the adversary for the implementation to work correctly.
The implementation should be seen as a proof-of-concept rather than a fully functional ready-to-use solution. It can provide a basis for further extensions, many of which are needed to to be able to drop the above-mentioned restrictive assumptions on the nodes and the adversary.

## References

[1] Sébastien Gambs, Rachid Guerraoui, Florian Huc, and Anne-Marie Kermarrec. On dynamic distributed computing. CoRR, abs/1202.3084, 2012.
[2] Ching Law and Kai-Yeung Siu. Distributed construction of random expander networks. In INFOCOM, 2003.

Kontakt: matej.pavlovic@epfl.ch