# Broadcast Bully: A New Algorithm for Leader Election in Distributed Systems

Angelica Escobar, University of Southern California
Research Advisor: Bhaskar Krishnamachari

USC Viterbi School of Engineering
Ming Hsieh Department of Electrical and Computer Engineering

## Introduction

- A **distributed system** is a collection of 2+ computers on a shared network
- Work concurrently to complete tasks more efficiently as a broader system
- Tasks in a distributed system may be assigned and coordinated by a leader
- The **Bully algorithm** is a classic leader election algorithm[1]
- The Bully algorithm uses several “immediate procedures” that guarantee a response will be received within some interval
- Leads to several different types of messages sent and received
- We developed and implemented **Broadcast Bully** as an alternative to simplify and reduce the amount of messages needed
- Uses a broadcast communication protocol with periodic beacons
- Grants every node the ability to compare all process IDs with their own
- Eliminates the multi-stage election process described in the classic Bully algorithm

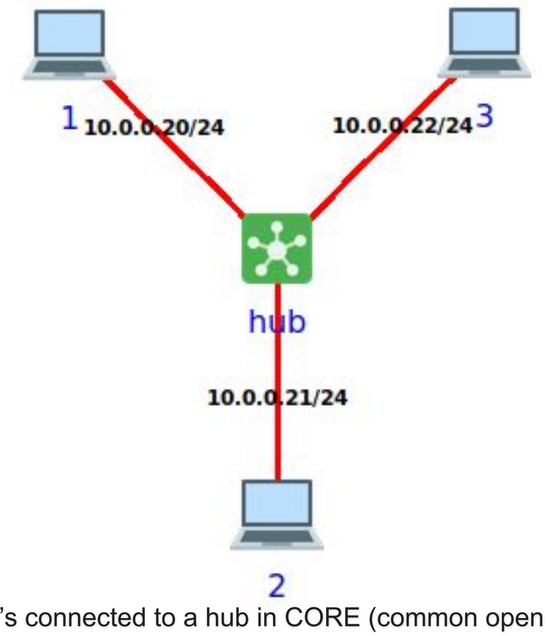
**Acknowledgements:**
I would like to thank my professor, advisor, and mentor, Bhaskar Krishnamachari, for his continued help and support. I’d also like to thank the University of Southern California and Sigma Xi for this opportunity.

## Hypothesis

- By implementing a broadcast communication protocol and eliminating multi-step election process, the types of messages sent and received are reduced thus simplifying the original algorithm while maintaining correctness.

## Methods

- Implement Broadcast Bully as a distributed program in Python
- Incorporated the asyncio library to write concurrent code
- Used IP addresses for comparison as max process IDs
- Run algorithm on a network emulation tool, CORE (Common Open Research Emulator)




Three pc’s connected to a hub in CORE (common open research emulator) each with IP addresses.

**References:**
[1]Garcia-Molina, "Elections in a Distributed Computing System," in IEEE Transactions on Computers, vol. C-31, no. 1, pp. 48-59, Jan. 1982, doi: 10.1109/TC.1982.1675885.

## Outcome

```
root@n1:/tmp/pycore.1/n1.conf# runudp 70
Time period: 70

My IP is: 10.0.0.20          Status message

Received: Broadcast  10.0.0.20
My new coordinator is:  None
Received: Broadcast  10.0.0.21
My new coordinator is:  None
Received: Broadcast  10.0.0.22
My new coordinator is:  None
Broadcast state:  True
Received: Broadcast  10.0.0.20
My new coordinator is:  None
Received: Broadcast  Victory 10.0.0.22
My new coordinator is:  10.0.0.22
```

Terminal window of node 1 in CORE displaying messages received from other nodes in the shared network with new coordinator message

- Message types include:
  - Broadcast beacon message from all nodes on the same network
  - Broadcast Victory message from new coordinator
- Shown to be capable of electing a new coordinator under dynamic failures of nodes

## Future Work

- Implement classic Bully algorithm and evaluate on CORE
- Comparatively evaluate Bully and Broadcast bully algorithms under different network conditions
- Test link limitations, delays, and robustness with newly-added nodes and failure detection