
# Distributed simulation with MPI in ns-3

Josh Pelkey

Dr. George Riley

Georgia Institute of Technology

# Overview

- Parallel and distributed discrete event simulation [1]
  - Allows single simulation program to run on multiple interconnected processors
  - Reduced execution time! Larger topologies!
- Terminology
  - Logical process (LP)
  - Rank or system id

# Quick and Easy Example

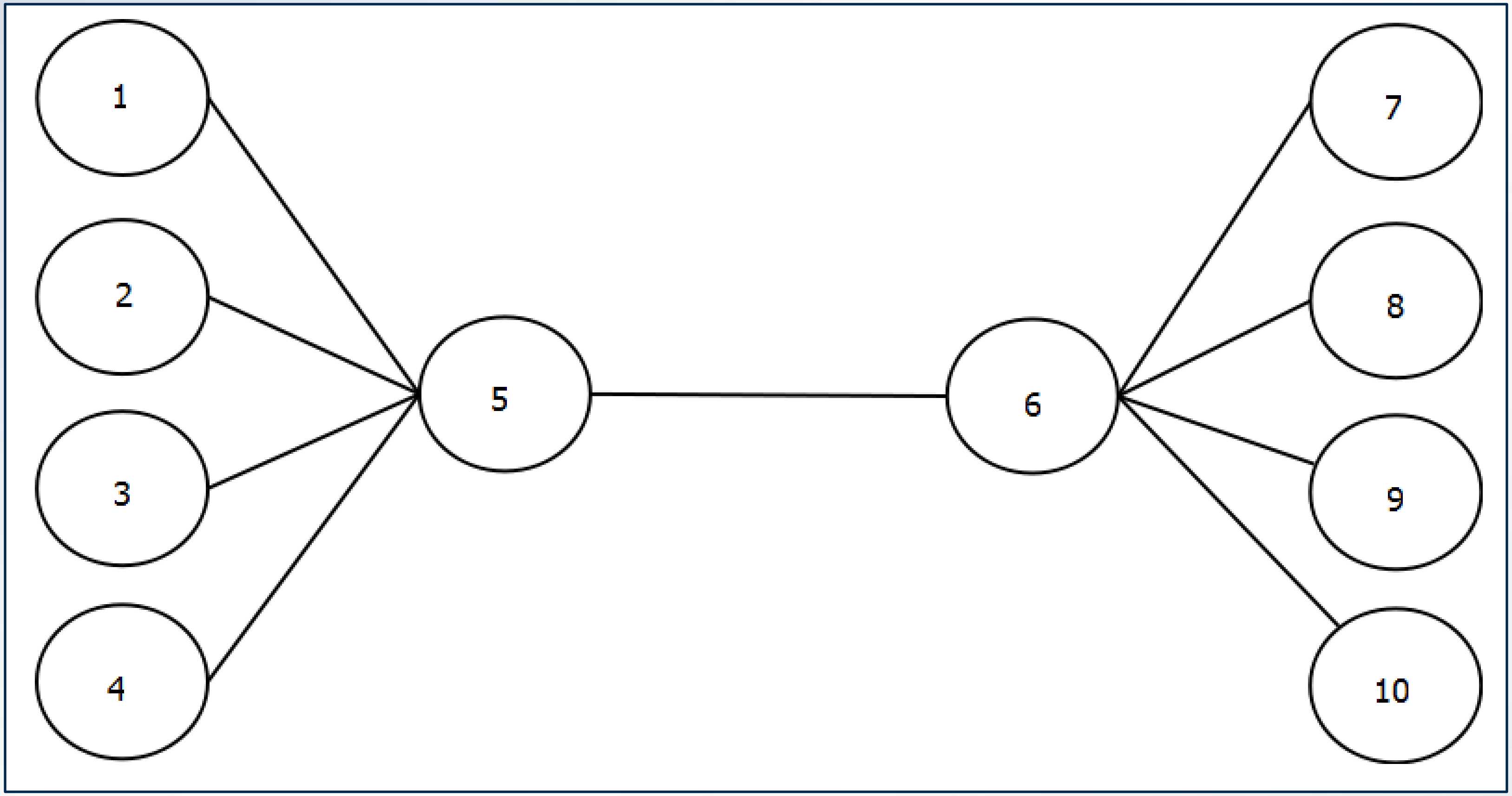


Figure 1. Simple point-to-point topology

# Quick and Easy Example

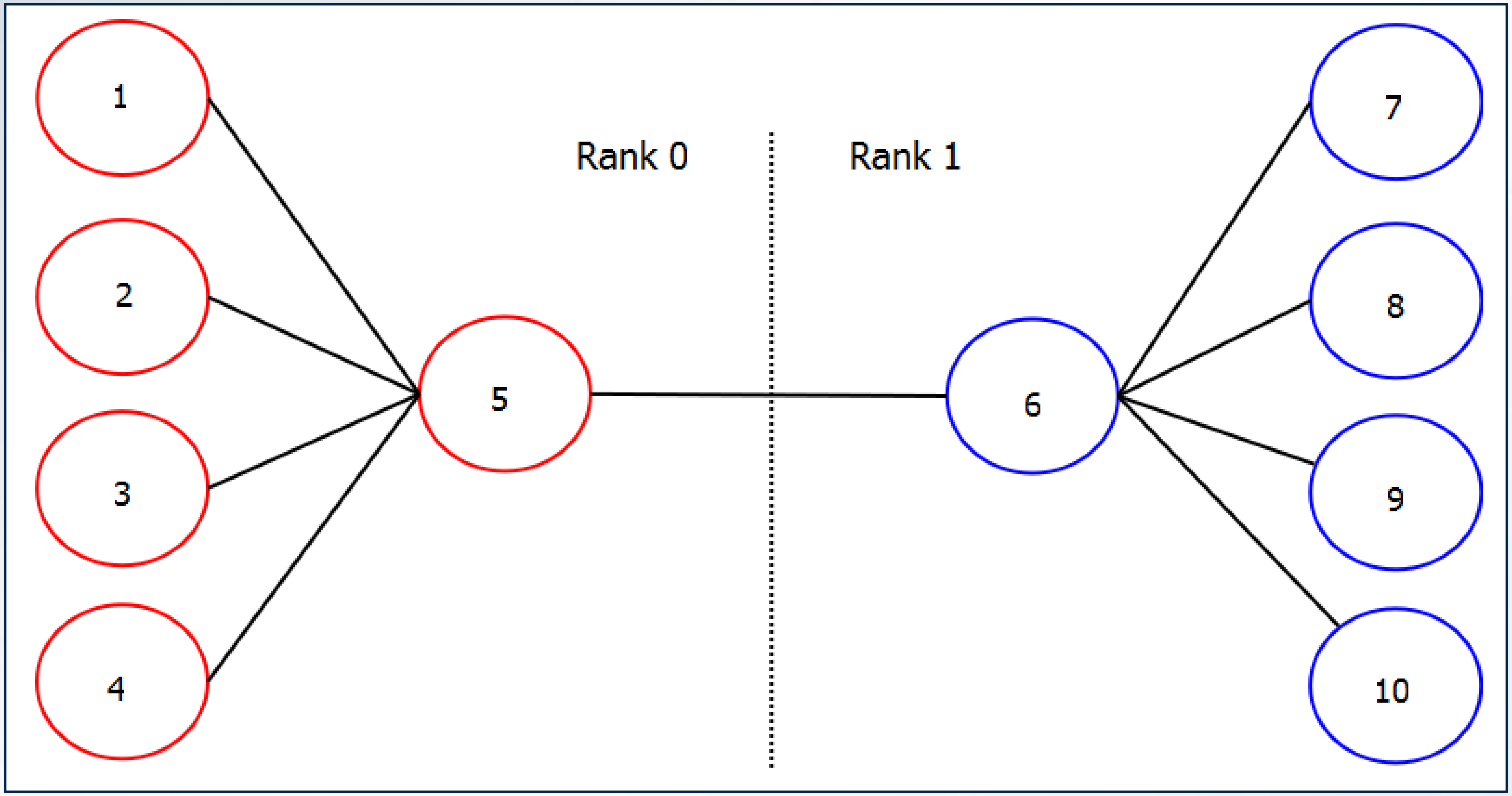


Figure 2. Simple point-to-point topology, distributed

# Implementation Details

- LP communication
  - Message Passing Interface (MPI) standard
  - Send/Receive time-stamped messages
  - MpiInterface in ns-3
- Synchronization
  - Conservative algorithm using lookahead
  - DistributedSimulator in ns-3

# Implementation Details (cont.)

- Assigning rank
  - Currently handled manually in simulation script
  - Next step, MpiHelper for easier node/rank mapping
- Remote point-to-point links
  - Created automatically between nodes with different ranks through point-to-point helper
  - Packet sent across using MpiInterface

# Implementation Details (cont.)

- Distributing the topology
  - All nodes created on all LPs, regardless of rank
  - Applications are only installed on LPs with target node

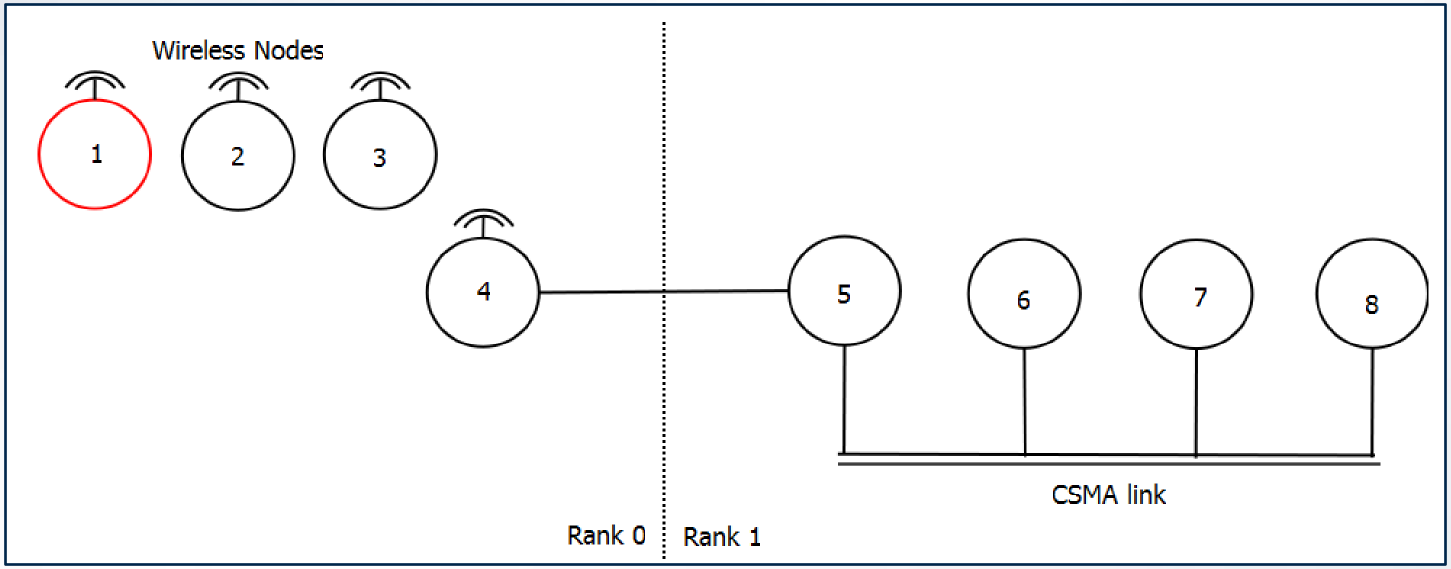


Figure 3. Mixed topology, distributed

# Performance Test

- DARPA NMS campus network simulation
  - Allows creation of very large topologies
  - Any number of campus networks are created and connected together
  - Different campus networks can be placed on different LPs
  - Tested with 2 CNs, 4 CNs, and 6 CNs

# Campus Network Topology

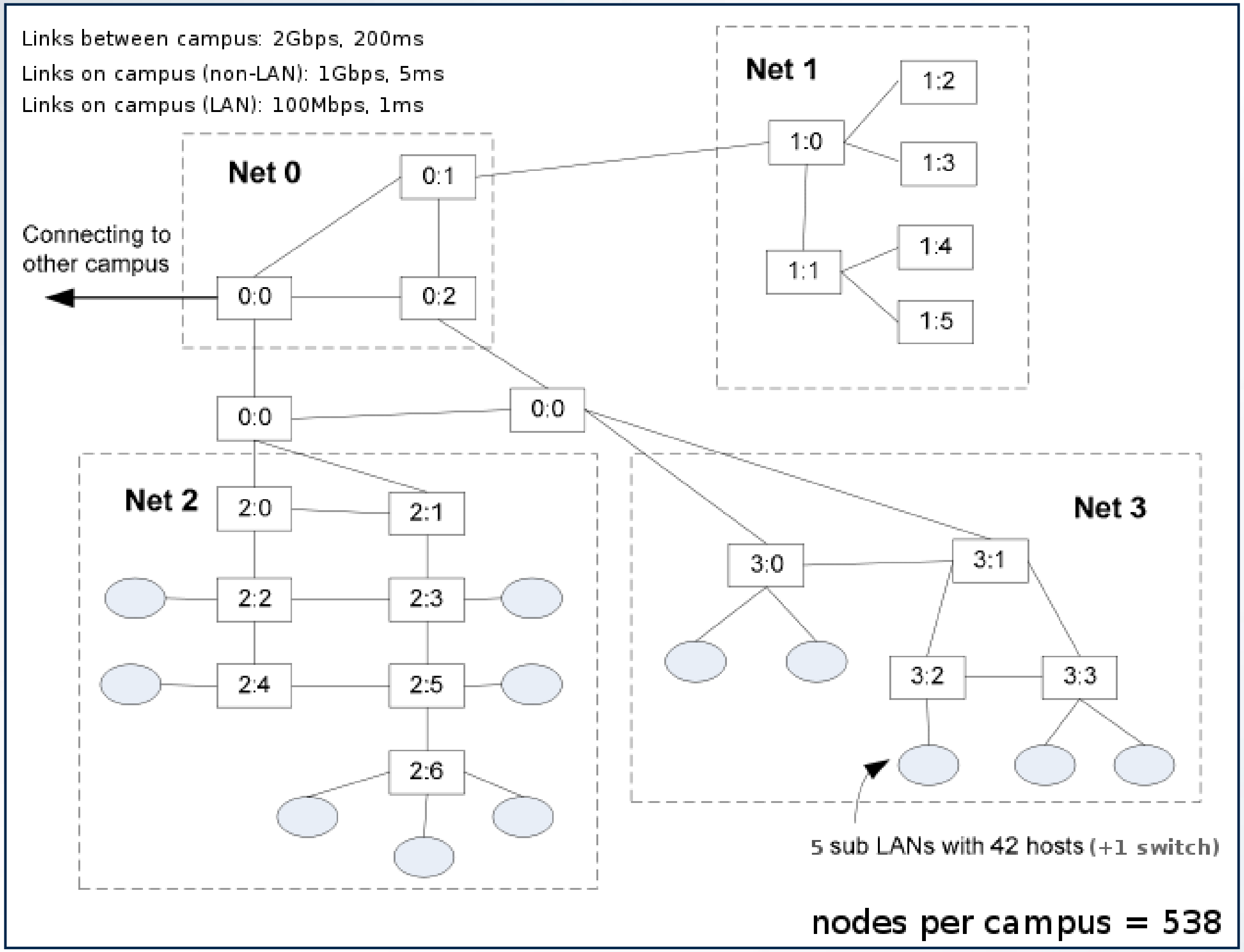


Figure 4. Single campus network

# 2 Campus Networks

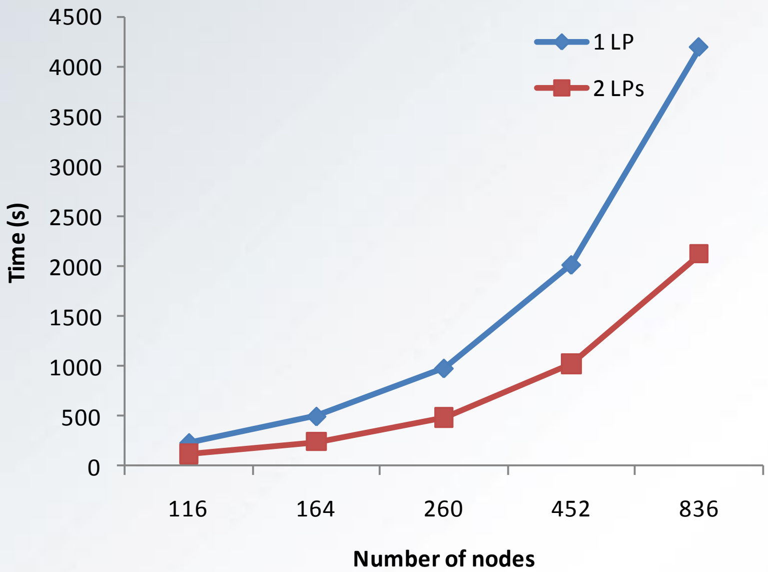


Figure 5. Execution time with 2 campus networks

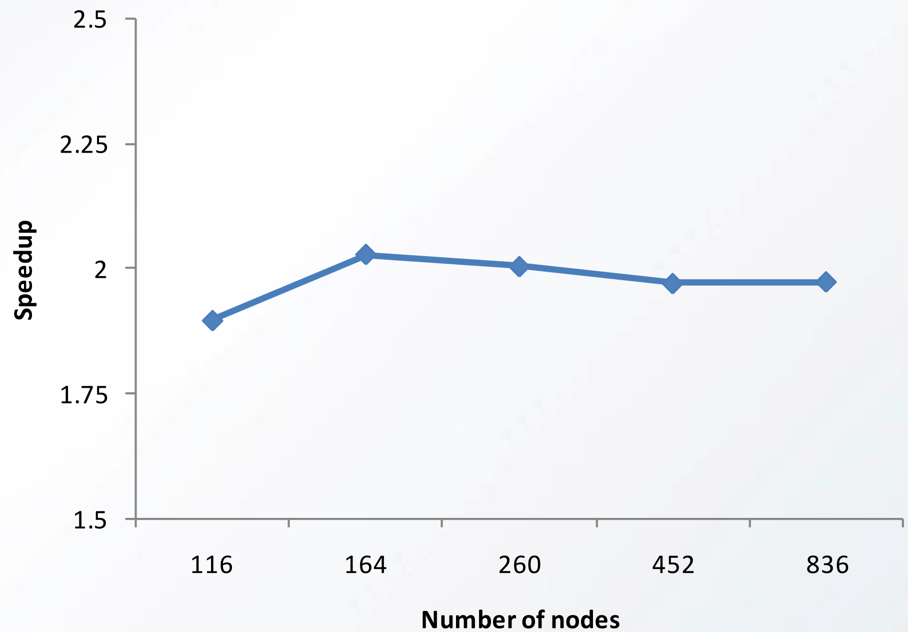


Figure 6. Speedup with 2 LPs

# 4 Campus Networks

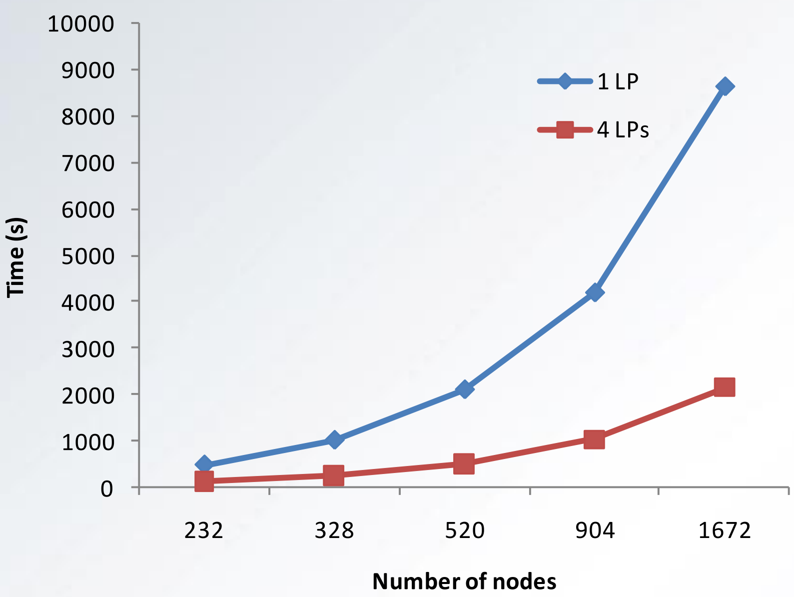


Figure 7. Execution time with 4 campus networks

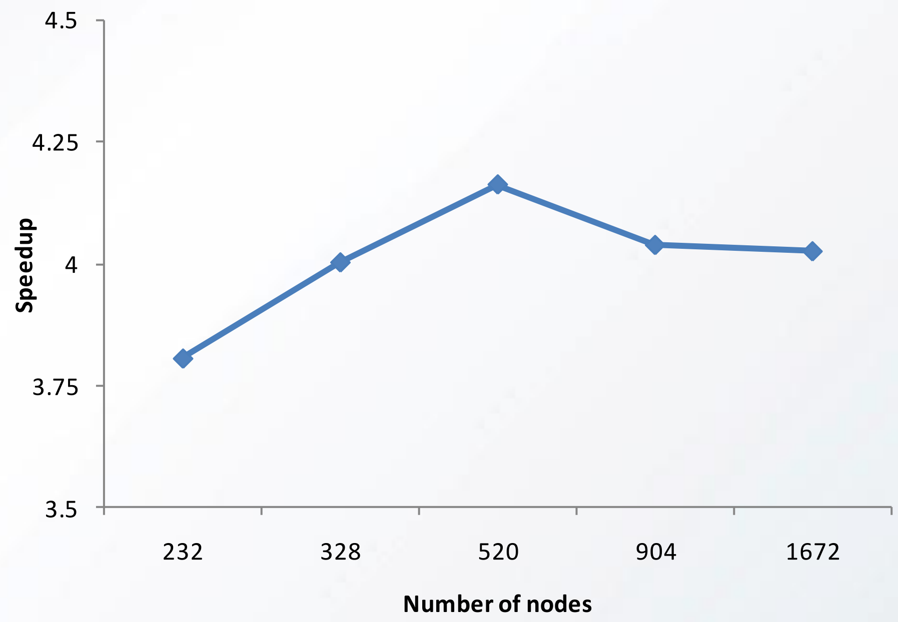


Figure 8. Speedup with 4 LPs

# 6 Campus Networks

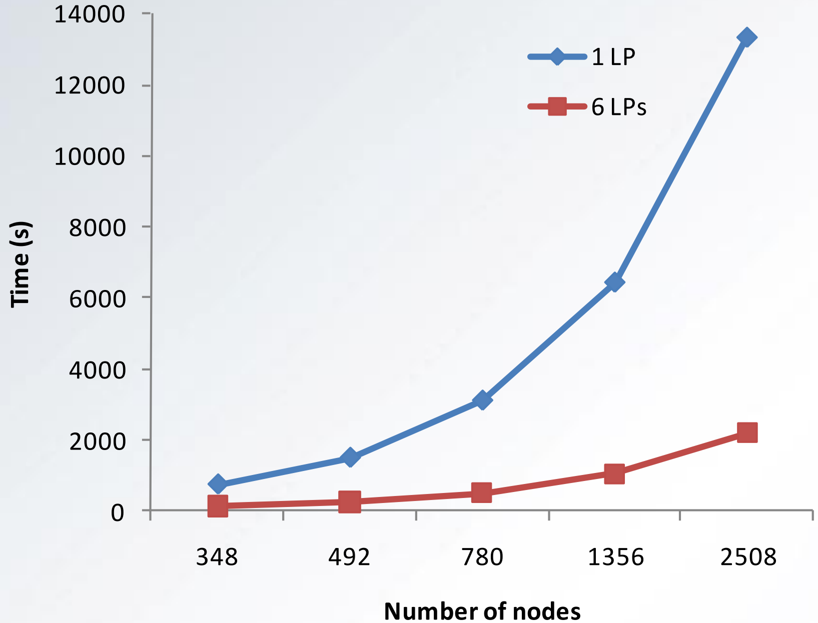


Figure 9. Execution time with 6 campus networks

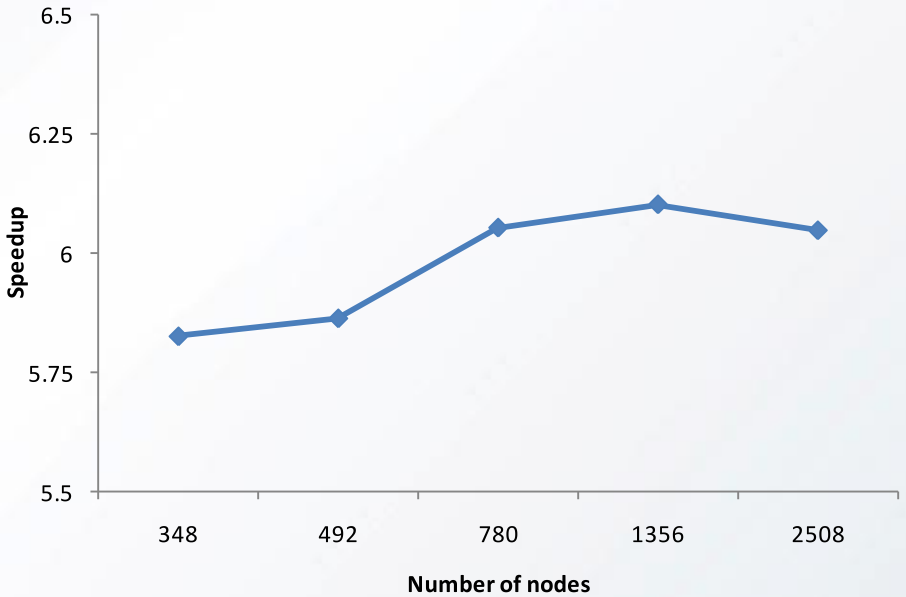


Figure 10. Speedup with 6 LPs

# Summary

- Distributed simulation in ns-3 allows a user to run a single simulation in parallel on multiple processors
- By assigning a different rank to nodes and connecting these nodes with point-to-point links, simulator boundaries are created
- Simulator boundaries divide LPs, and each LP can be executed by a different processor
- Distributed simulation in ns-3 offers solid performance gains in time of execution for large topologies

# Distributed wireless simulation

- Popular feature request
  - Wireless technology is everywhere
  - Wireless simulation is complex
- Introduces new issues
  - Partitioning (We have mobility!)
  - Small propagation delay, small lookahead
  - Very large number of events

# Sample Topology

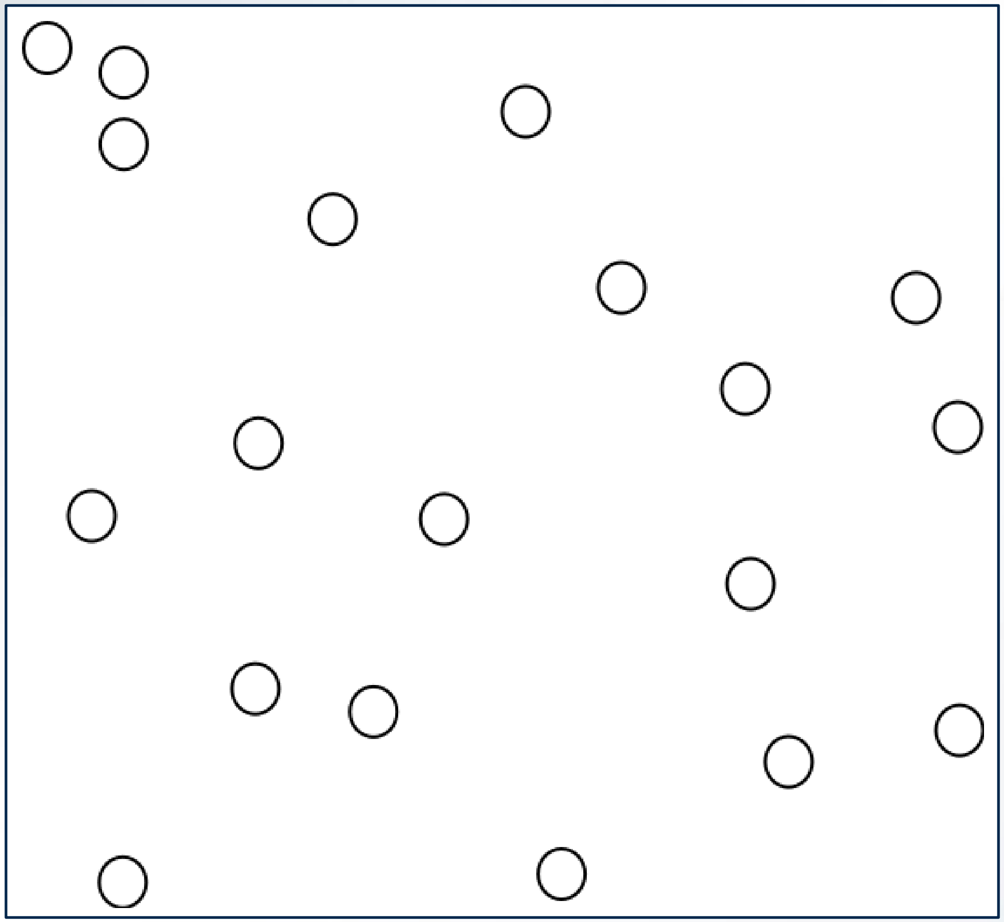

Figure 11. Wireless network topology

# Geographic Partitioning

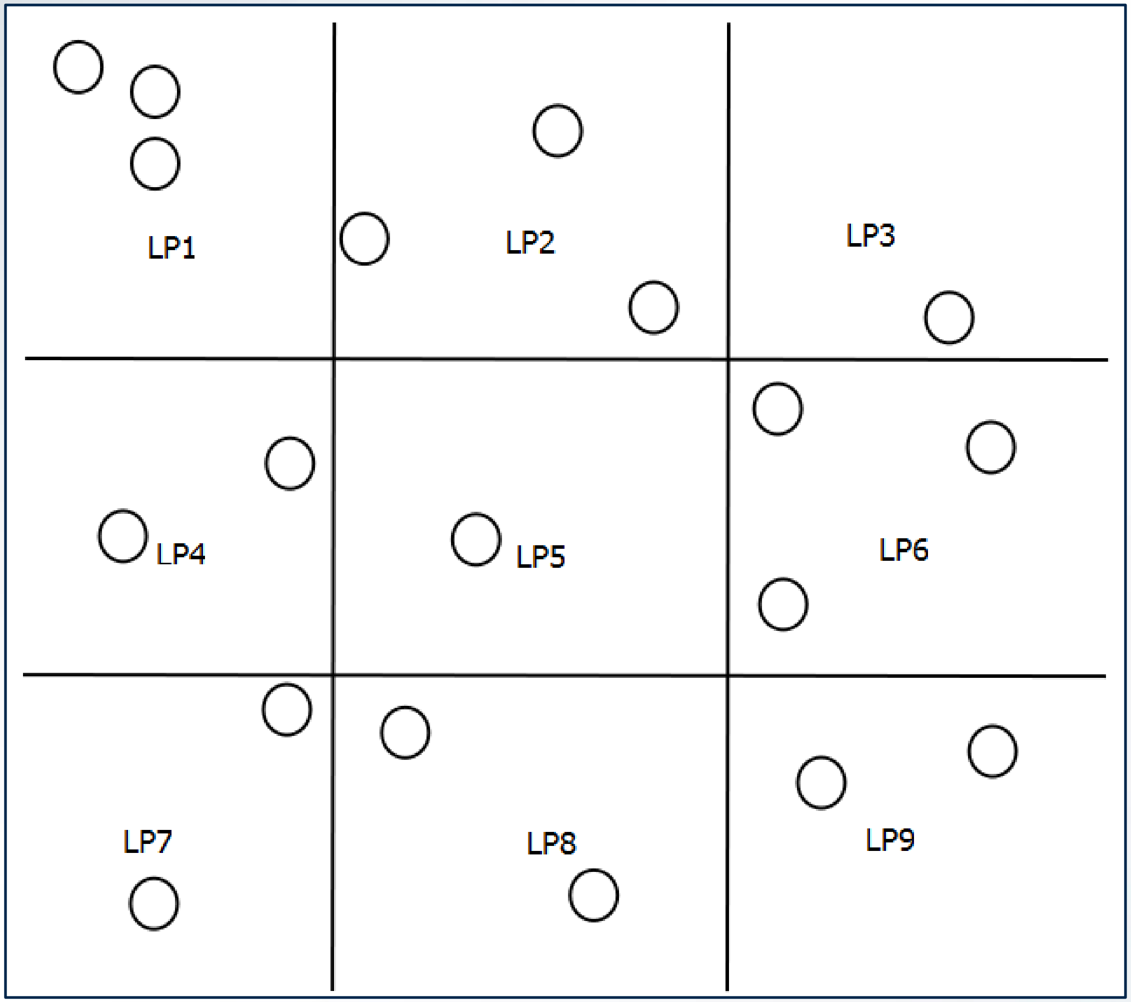


Figure 12. Wireless network topology, partitioned

# Node-based Partitioning

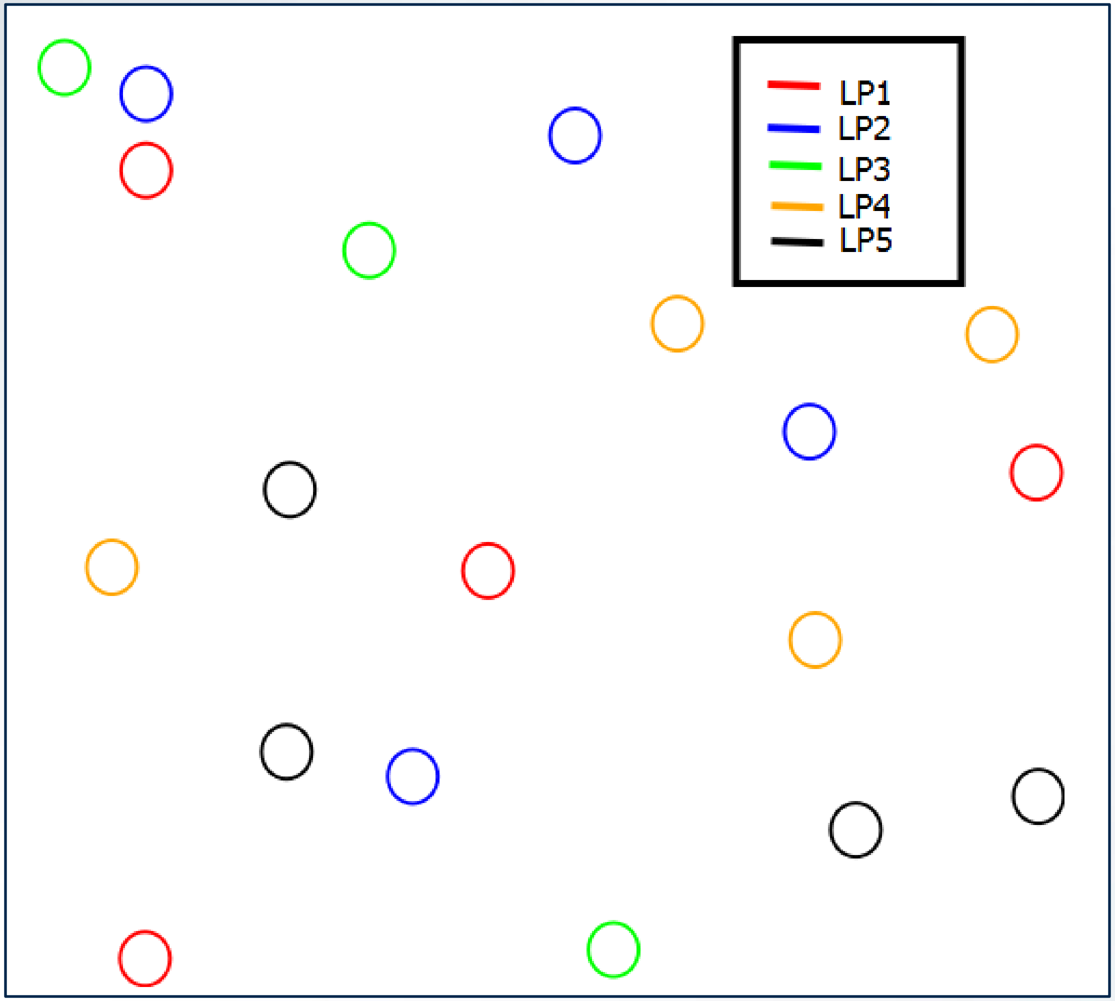


Figure 13. Wireless network topology, partitioned

Georgia Tech

# Lookahead

- Typical wireless scenarios present small lookahead due to node distances and the speed of light
- Small lookahead is detrimental to distributed simulation performance
- Possible optimizations
  - Protocol lookahead [2]
  - Event lookahead [3]

# Wireless Simulation Events

- Wireless simulations require a large number of events
- Increased inter-LP communication (bad)
- Event Reduction [4]
  - Decreases overhead
  - However, must ensure simulation fidelity

Georgia Tech

# Event Reduction Techniques

- Set a propagation limit
  - Carrier Sensing Threshold (too inaccurate?)
  - Popular distance limit [5]
- Lazy Updates
  - Leverage protocol mechanics and simulator knowledge
  - Ex: Lazy MAC state update [6]
- Event Bundling
  - Send fewer events but deliver the same information
  - Ex: LP-Rx event [3]

Georgia Tech

# Initial Development Plans

- Geographic and node-based partitioning
- Simple lookahead
  - Assume minimal lookahead
- Event Reduction
  - Use carrier sensing threshold for propagation limit
  - Use event bundling

Georgia Tech

# Distributed Wireless Summary

- People want distributed wireless
- Implementing distributed wireless simulation should be easy
- Optimizing distributed wireless simulation is hard
- The good news is a great amount of research and previous implementations give us direction for optimization

# References


[1] R.M. Fujimoto. *Parallel and Distributed Simulation Systems.* Wiley Interscience, 2000.

[2] J. Liu and D. M. Nicol. Lookahead revisited in wireless network simulations. In *Proceedings of the 16th Workshop on Parallel and Distributed Simulation*, 2002.

[3] Peschlow, P., Voss, A., and Martini, P. 2009. Good news for parallel wireless network simulations. In *Proceedings of the 12th ACM international Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2009

[4] Z. Ji, J. Zhou, M. Takai, and R. Bagrodia. Optimizing parallel execution of detailed wireless network simulation. In *Proceedings of the 18th Workshop on Parallel and Distributed Simulation*, 2004.

[5] Z. Ji, J. Zhou, M. Takai, and R. Bagrodia. Scalable simulation of large-scale wireless networks with bounded inaccuracies. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2004.

[6] Y. J. Lee and G. F. Riley. Efficient simulation of wireless networks using lazy MAC state update. In *Proceedings of Advanced and Distributed Simulation*, 2005.