PHOLD Performance of Conservative Synchronization Methods for Distributed Simulation in ns-3

School of Electrical and Computer Engineering

Georgia Institute of Technology, Atlanta, GA

Jared Ivey, Dr. Brian Swenson, Dr. George Riley

Overview

- Discrete Event Simulation to Parallel Discrete Event Simulation
- Synchronization Methods (Optimistic and Conservative)
- Experimental Setup
- Results (Performance Trends and Connectivity Thresholds)
- Conclusions and Future Work

Overview

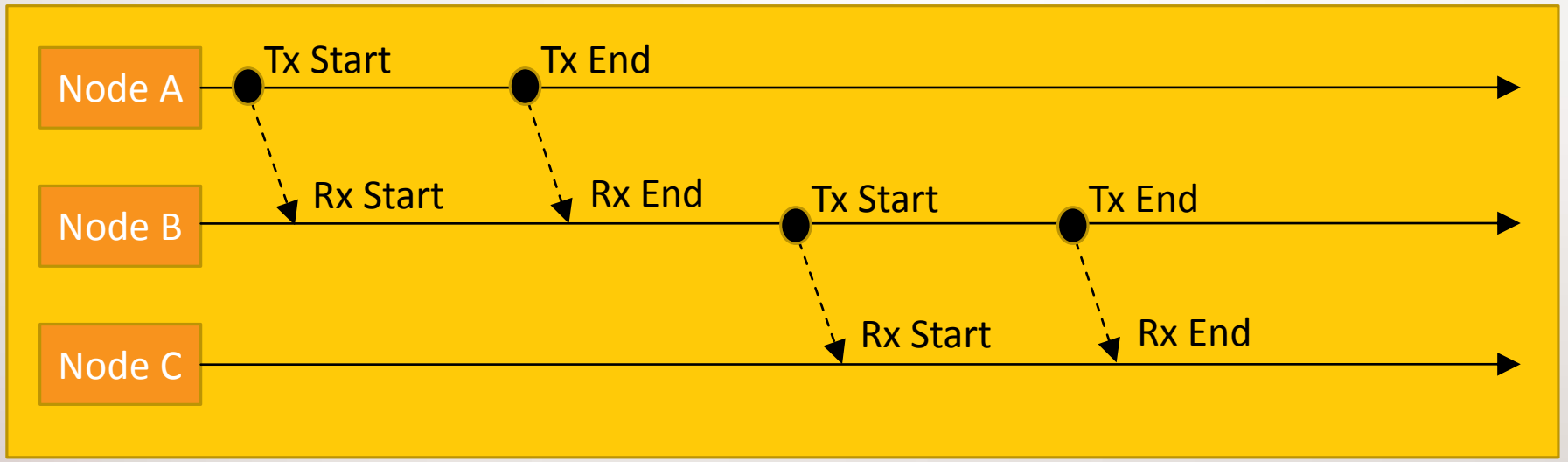
- Discrete Event Simulation to Parallel Discrete Event Simulation
- Synchronization Methods (Optimistic and Conservative)
- Experimental Setup
- Results (Performance Trends and Connectivity Thresholds)
- Conclusions and Future Work

Discrete Event Simulation

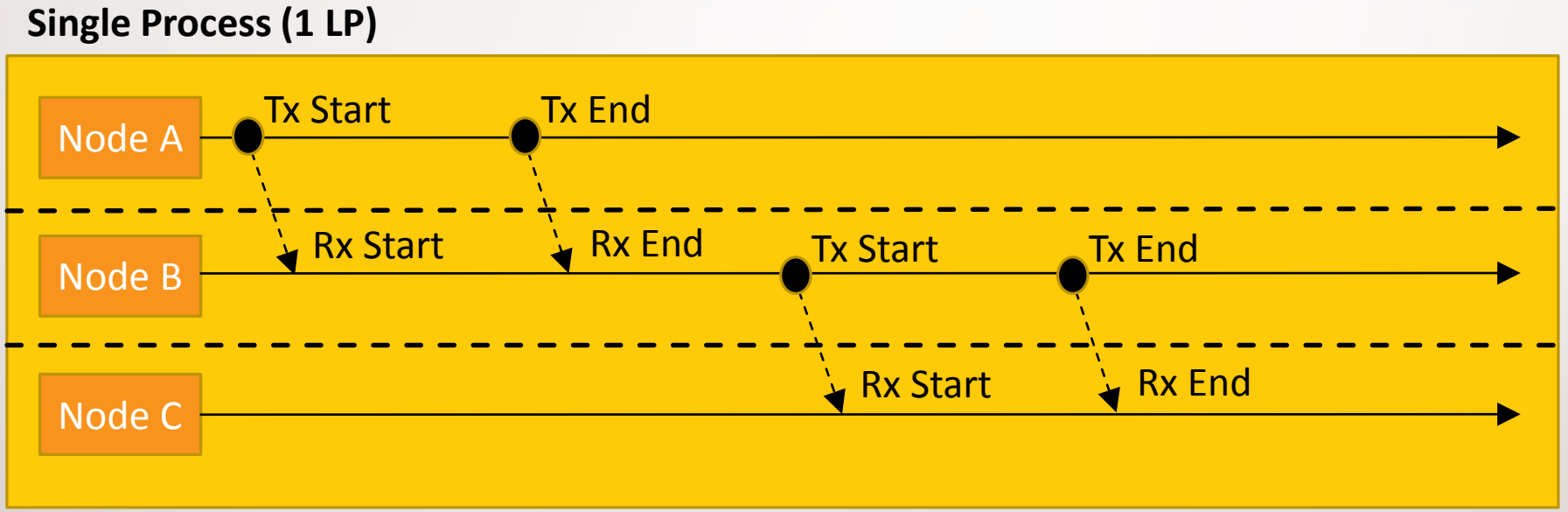
- Series of time-ordered events
- Advance simulation time based on next event
- Exit at specific time or when no events left

Discrete Event Simulation

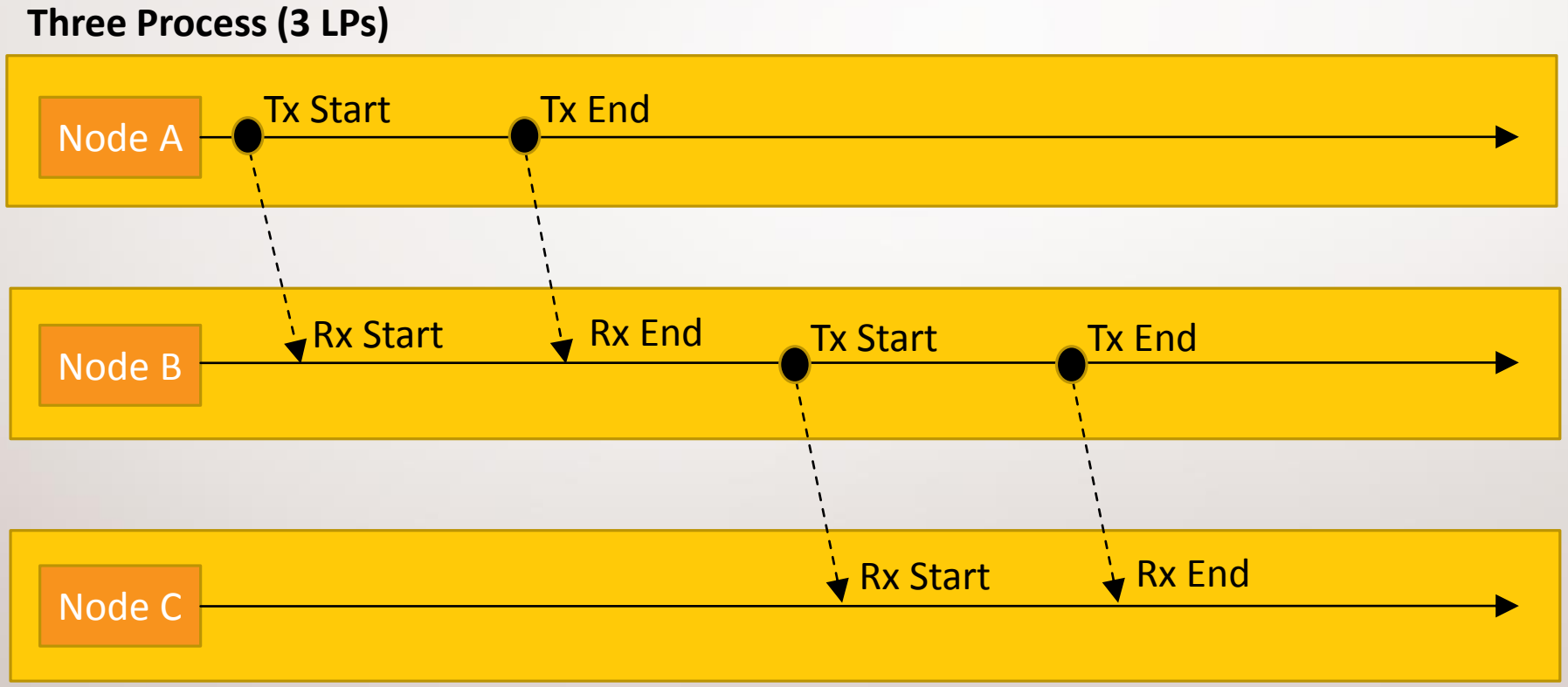
Single Process (1 LP)



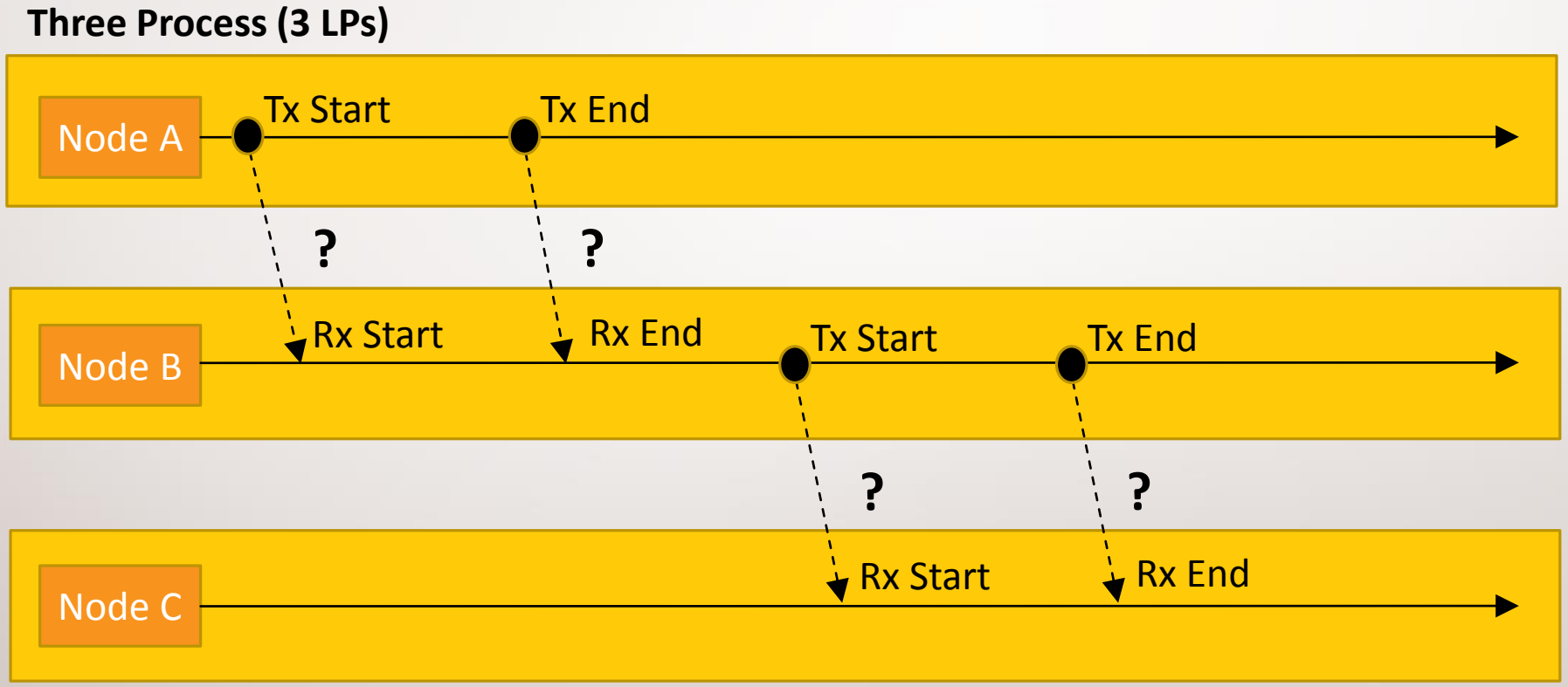
Discrete Event Simulation



Parallel Discrete Event Simulation



Parallel Discrete Event Simulation



Parallel Discrete Event Simulation

- Allocates processing requirements across multiple logical processes (LPs)
- More LPs = more processing power and (typically) decreased execution time
- Sequential (1 LP) vs. Parallel
  - Must produce identical simulated results
- Causality constraint
  - Multiple LPs require synchronization to ensure events are not out of order

Overview

- Discrete Event Simulation to Parallel Discrete Event Simulation
- Synchronization Methods (Optimistic and Conservative)
- Experimental Setup
- Results (Performance Trends and Connectivity Thresholds)
- Conclusions and Future Work

Synchronization Methods

- Two types
  - Optimistic
  - Conservative

Optimistic Synchronization

- LPs execute freely and synchronize when errors are detected
- Rollback to state prior to error
- Send anti-message for each event message after error
- Global Virtual Time to save memory (fossil collection)

Conservative Synchronization

- Avoid processing events out of order
- Two styles:
  - Synchronous: Granted Time Window
  - Asynchronous: Chandy-Misra-Bryant “Null Message”

Granted Time Window

- Based on Distributed Snapshot algorithm proposed by Mattern
- Integrated into ns-3.8
- Global determination of the lowest bound timestamp (LBTS)
- Transient message check (event messages that have been sent by an LP but not yet handled by the recipient)
- Remote point-to-point channels → minimum channel delay = lookahead
- Granted Time = Lookahead + LBTS

Granted Time Window

```
void DistributedSimulatorImpl::Run (void)
{
    m_lookAhead = CalculateLookAhead ();
    m_stop = false;
    while (!m_globalFinished)
    {
        GrantedTimeWindowMpiInterface::ReceiveMessages ();
        Time nextTime = Next ();
        if (nextTime > m_grantedTime || IsLocalFinished () )
        {
            TestSendComplete ();
            LbtsMessage lMsg (GetRxCount (), GetTxCount (), m_myId, IsLocalFinished (), nextTime);
            m_pLBTS[m_myId] = lMsg;
            MPI_Allgather (&lMsg, sizeof (LbtsMessage), MPI_BYTE, m_pLBTS, sizeof (LbtsMessage), MPI_BYTE, MPI_COMM_WORLD);
            m_globalFinished = GlobalFinishCheck ();
            if ( TransientMessageCheck() )
            {
                m_grantedTime = CalculateGrantedTime ();
            }
            if ( (nextTime <= m_grantedTime) && (!IsLocalFinished ()) )
            {
                ProcessOneEvent ();
            }
        }
        NS_ASSERT (!m_events->IsEmpty () || m_unscheduledEvents == 0);
    }
}
```

Null Message

- Based on the Chandy-Misra-Bryant (CMB) algorithm
- Integrated into ns-3.19
- Asynchronous (no global communication)
- Null message
  - Not an actual simulation event
  - Earliest time an LP may expect to receive an actual event from the sender (Guarantee time)
  - Guarantee time = current simulation time + lookahead

Null Message

```
void NullMessageSimulatorImpl::Run (void)
{
     CalculateLookAhead ();
     RemoteChannelBundleManager::InitializeNullMessageEvents ();
     m_stop = false;
     while (!IsFinished ())
     {
          Time nextTime = Next ();
          if ( nextTime <= GetSafeTime () )
          {
               ProcessOneEvent ();
               NullMessageMpiInterface::ReceiveMessagesNonBlocking ();
          }
          else
          {
               NullMessageMpiInterface::ReceiveMessagesBlocking ();
          }
          CalculateSafeTime ();
          NullMessageMpiInterface::TestSendComplete ();
     }
}
```

Null Message

- Sending null message
  1. Schedule next null message
- Sending event message
  1. Cancel pending null message
  2. Reschedule next null message

Overview

- Discrete Event Simulation to Parallel Discrete Event Simulation
- Synchronization Methods (Optimistic and Conservative)
- Experimental Setup
- Results (Performance Trends and Connectivity Thresholds)
- Conclusions and Future Work

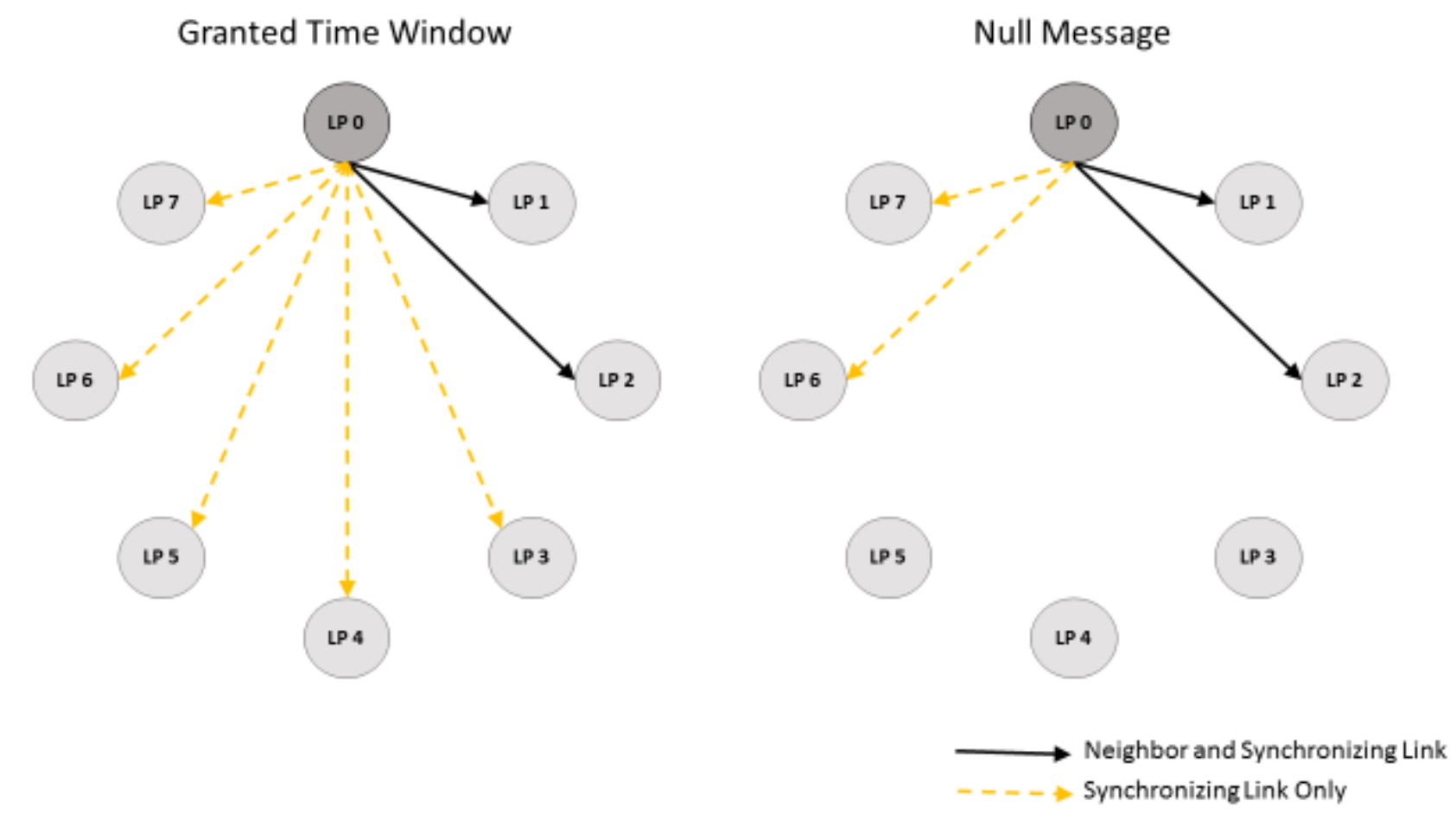
Experimental Setup

- PHOLD Performance Model
  - User-defined application in ns-3
  - Provides a synthetic workload by artificially creating both local and remote network traffic for the system.
  - Process:
    1. Each LP sends a message to either itself or a neighboring LP at a random time in the future.
    2. When a message is received by an LP, it schedules another local or remote event at a random time in the future.
    3. Repeat until either a predetermined simulation time or number of transmitted messages is achieved.

Experimental Setup

- Modified versions of both synchronization options supported by ns-3 (Granted Time Window and Null Message)
- Allow all LPs to transmit packet messages to each other as individual nodes
  - No simulated network routing
  - No simulated IP overhead
- Send MPI messages directly between applications
  - Only using the event scheduler and synchronization algorithms of ns-3

Experimental Setup



Experimental Setup

- Initial packet seed: 128 local messages
- Transmit time: Exponential distribution with μ = 0.9 seconds
- LP selection: Uniform distribution
- System size: 1024 LPs
- Total number of messages transmitted: 16,384 messages per node (Total: 16,777,216 messages)
- Variables:
  - Lookahead (1 - 28 ms)
  - Number of selectable neighbors (1 - 512)
  - Level of remote traffic (0%, 10%, and 50%)

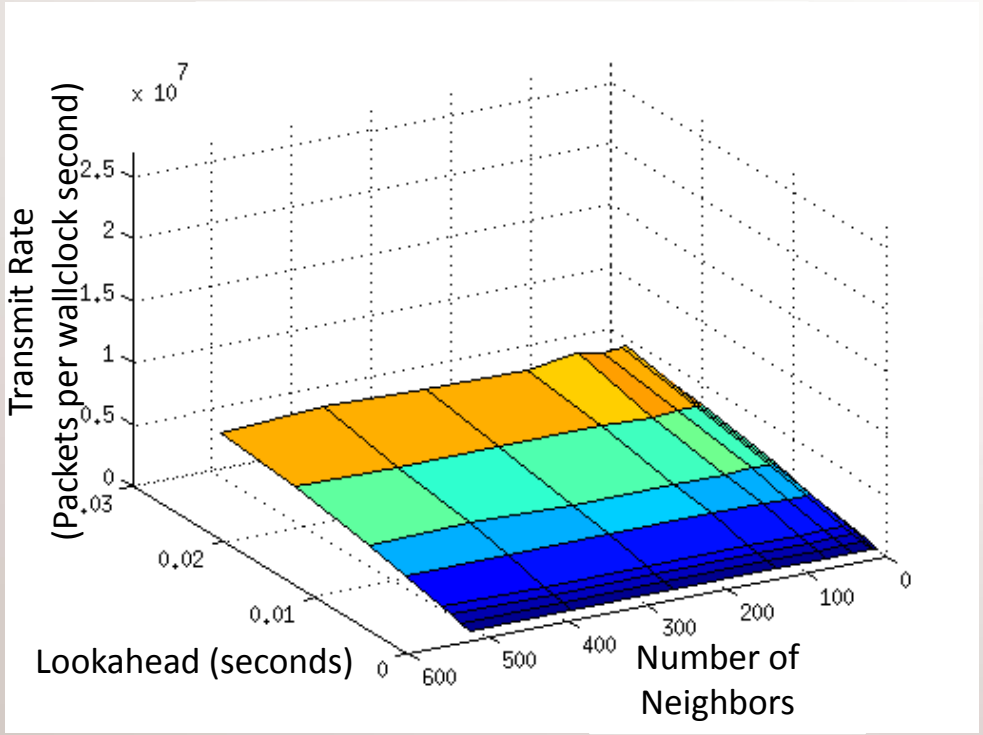
Experimental Setup

- Hardware:
  - Cab computing cluster at Lawrence Livermore National Laboratory
  - 2.6GHz Intel Xeon 8-core E5-2670 processors
  - 32GB of memory per node
  - 1024 cores used for the PHOLD experiments, with each core acting as a single node in the distributed simulation.

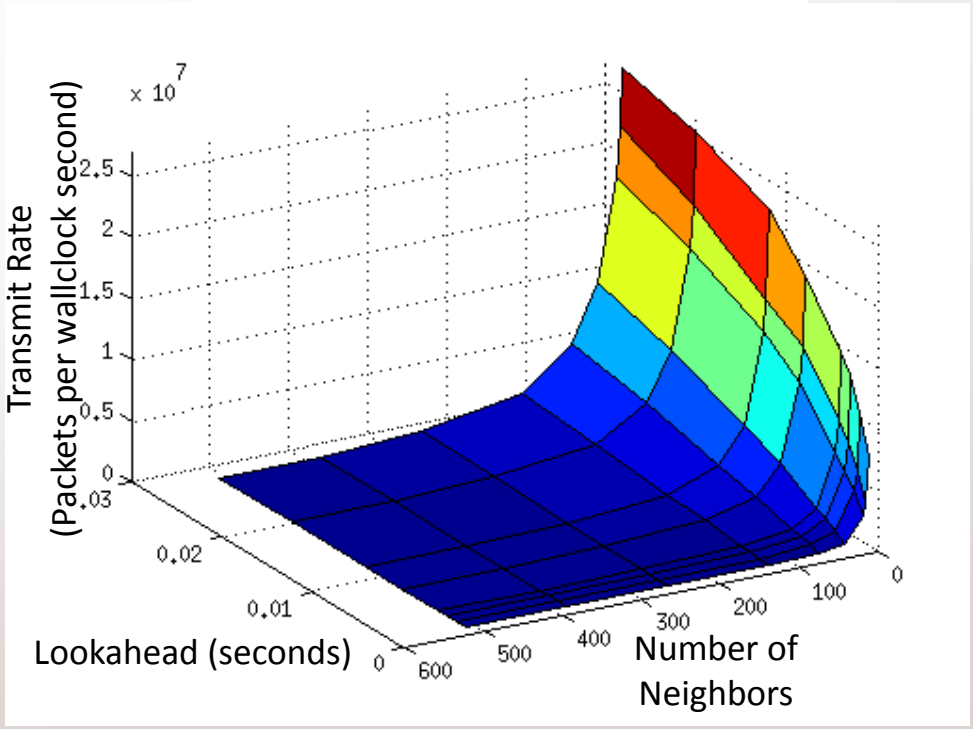
Overview

- Discrete Event Simulation to Parallel Discrete Event Simulation
- Synchronization Methods (Optimistic and Conservative)
- Experimental Setup
- Results (Performance Trends and Connectivity Thresholds)
- Conclusions and Future Work

Results – Performance Trends (0%)

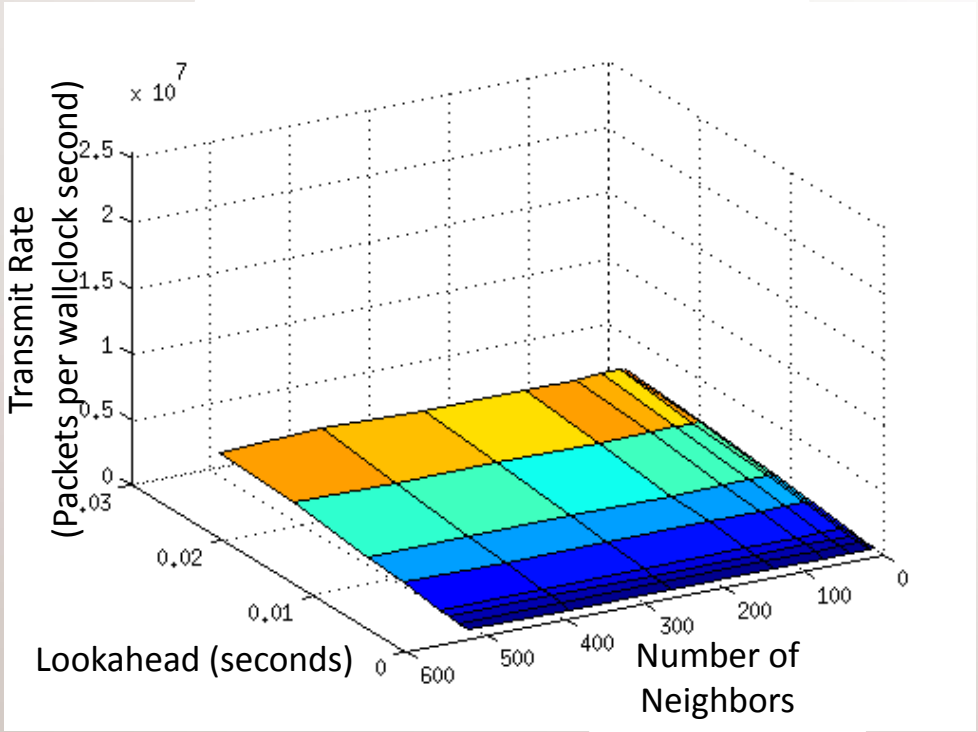


Granted Time Window

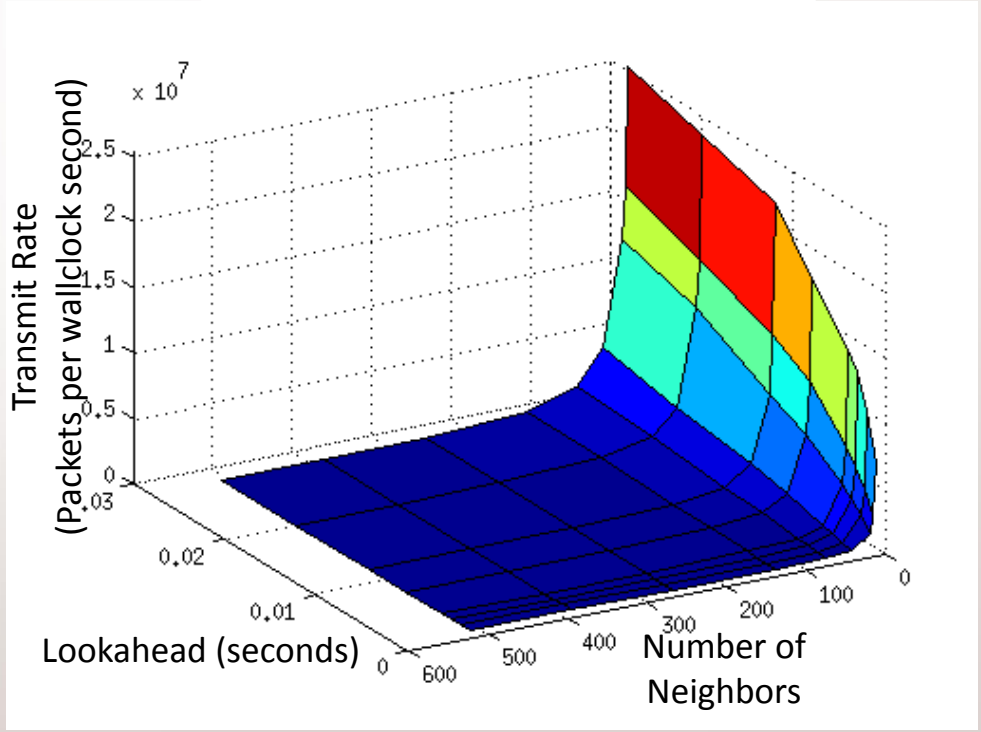


Null Message

Results – Performance Trends (10%)

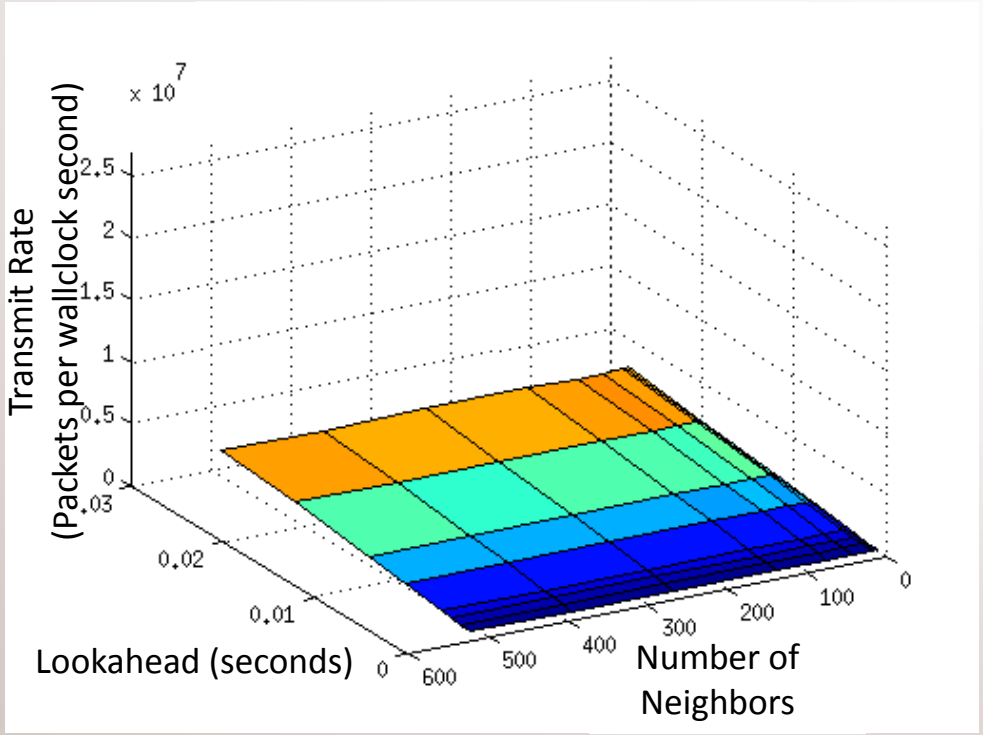


Granted Time Window

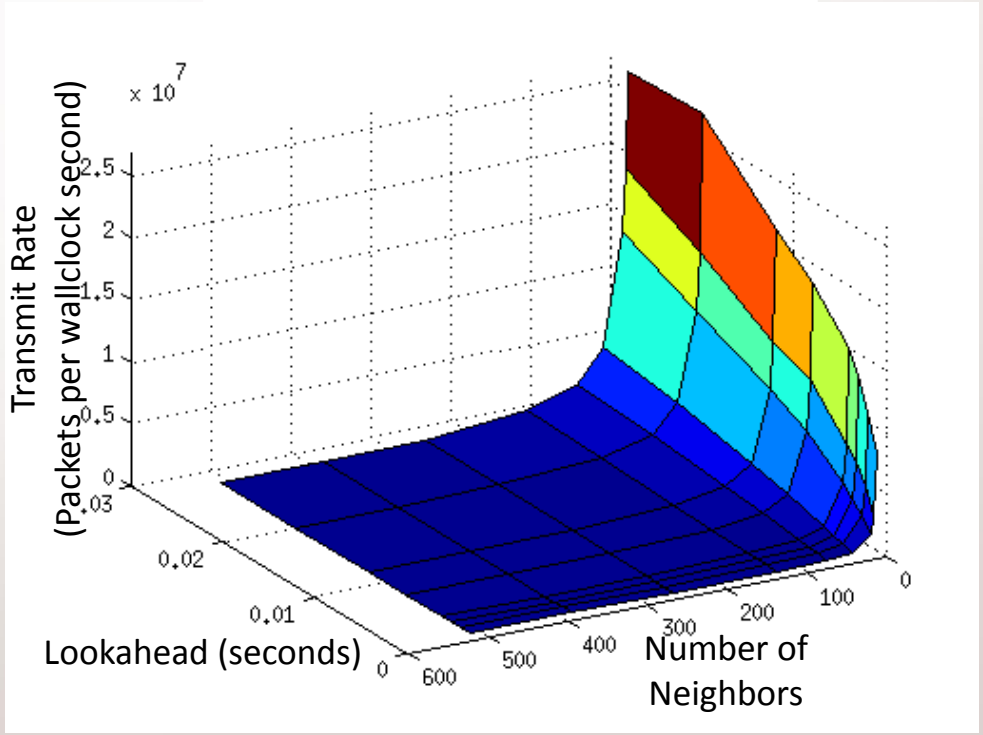


Null Message

Results – Performance Trends (50%)

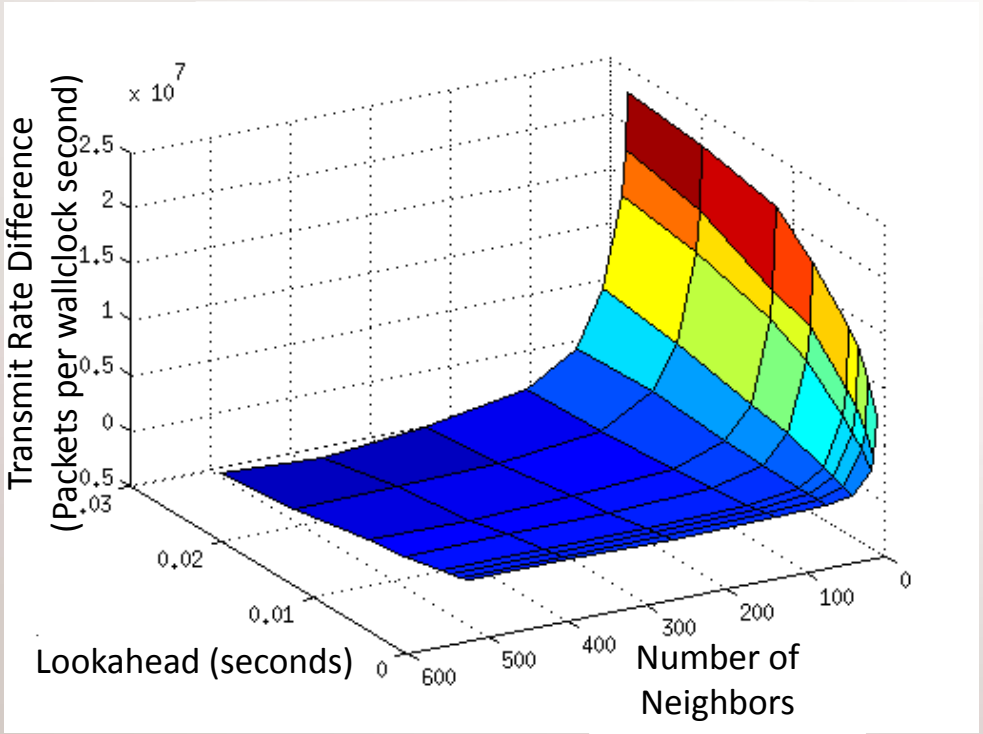


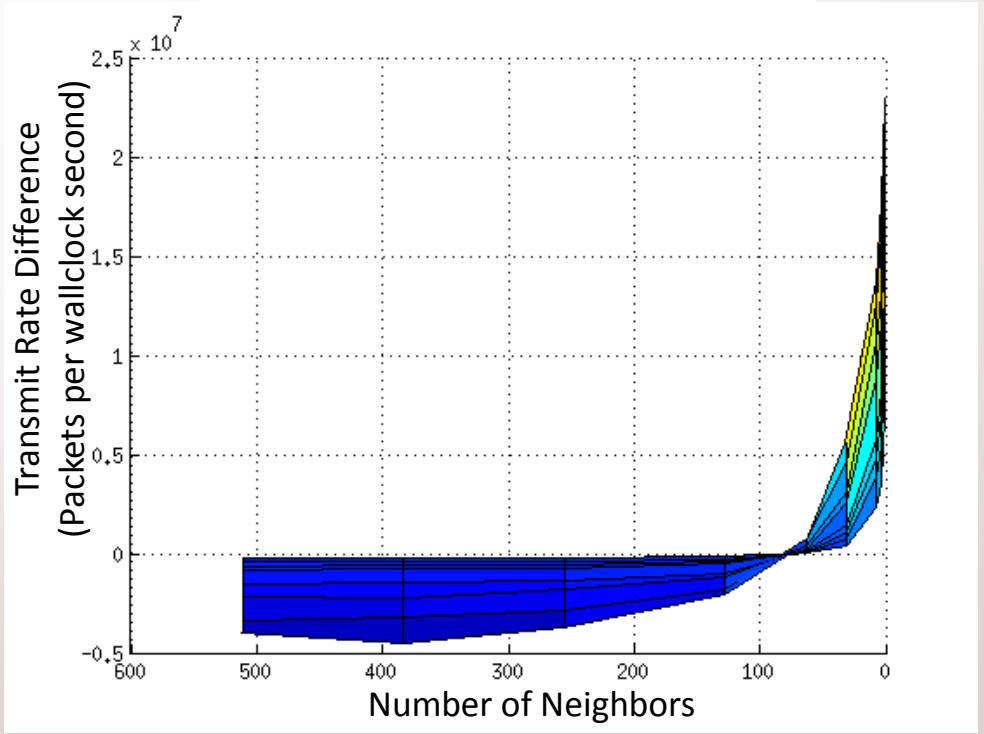
Granted Time Window



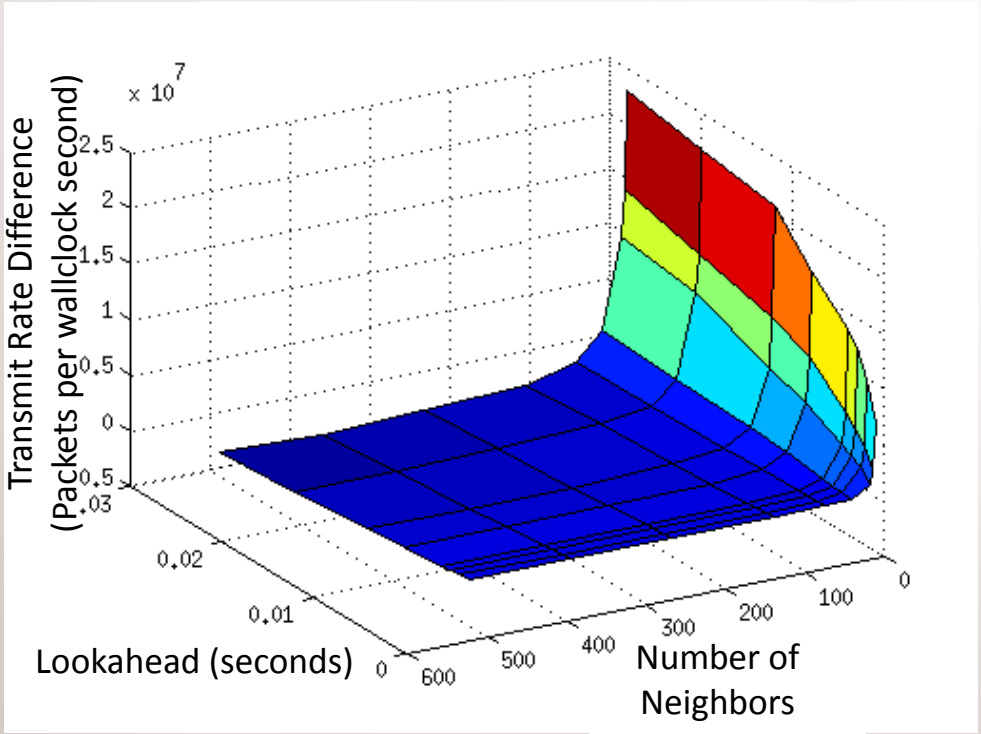
Null Message

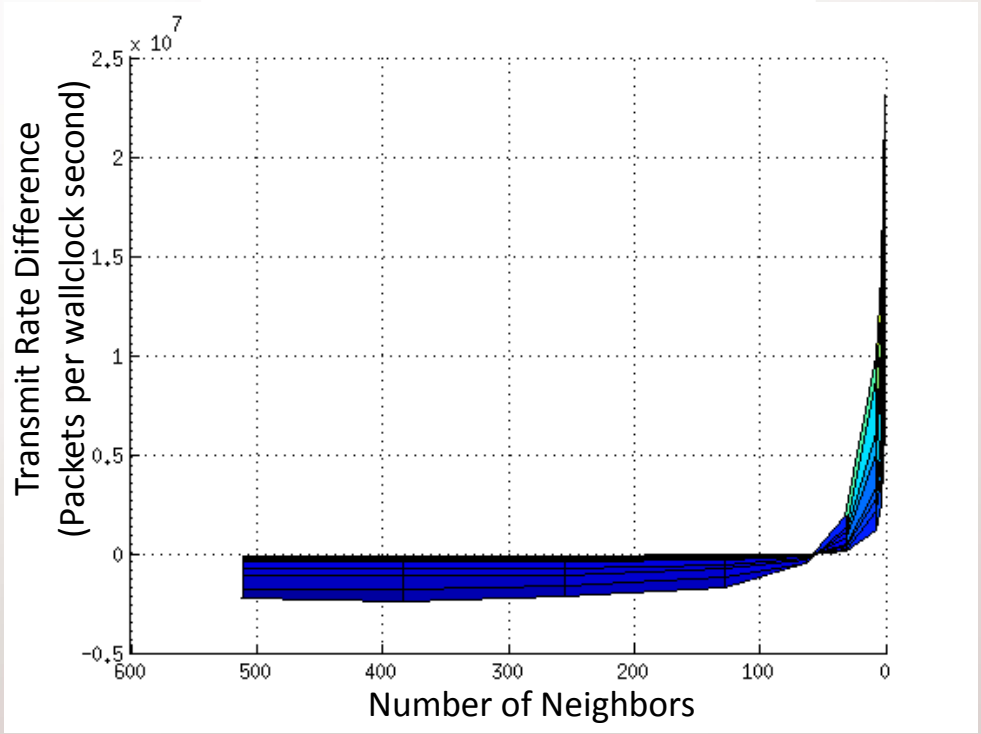
Results – Connectivity Thresholds (0%)



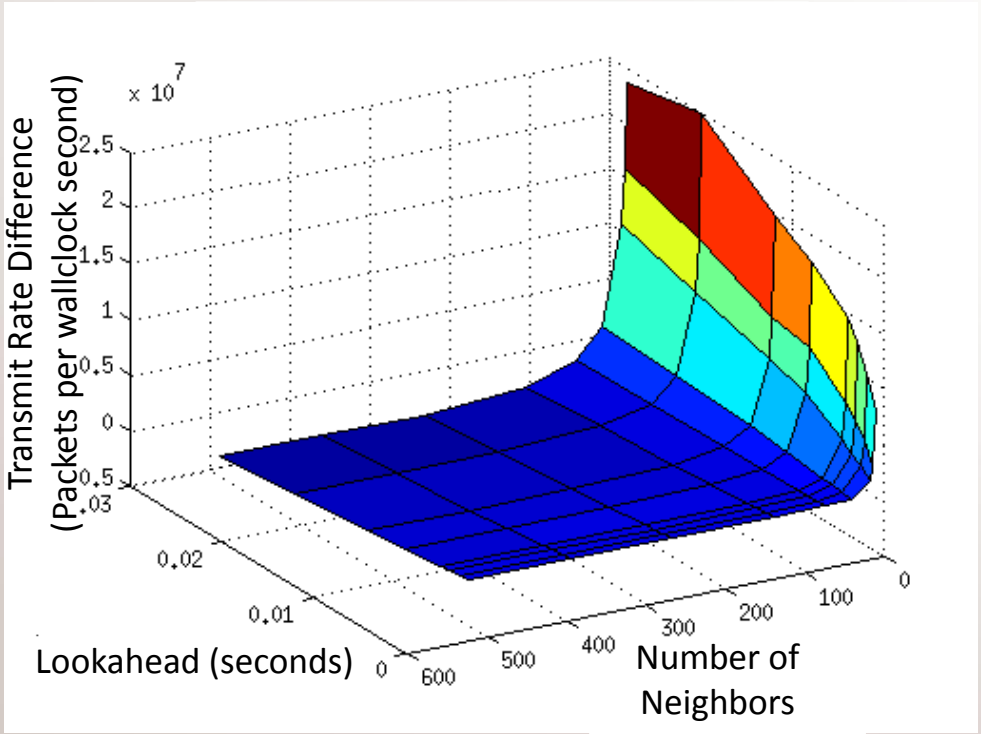


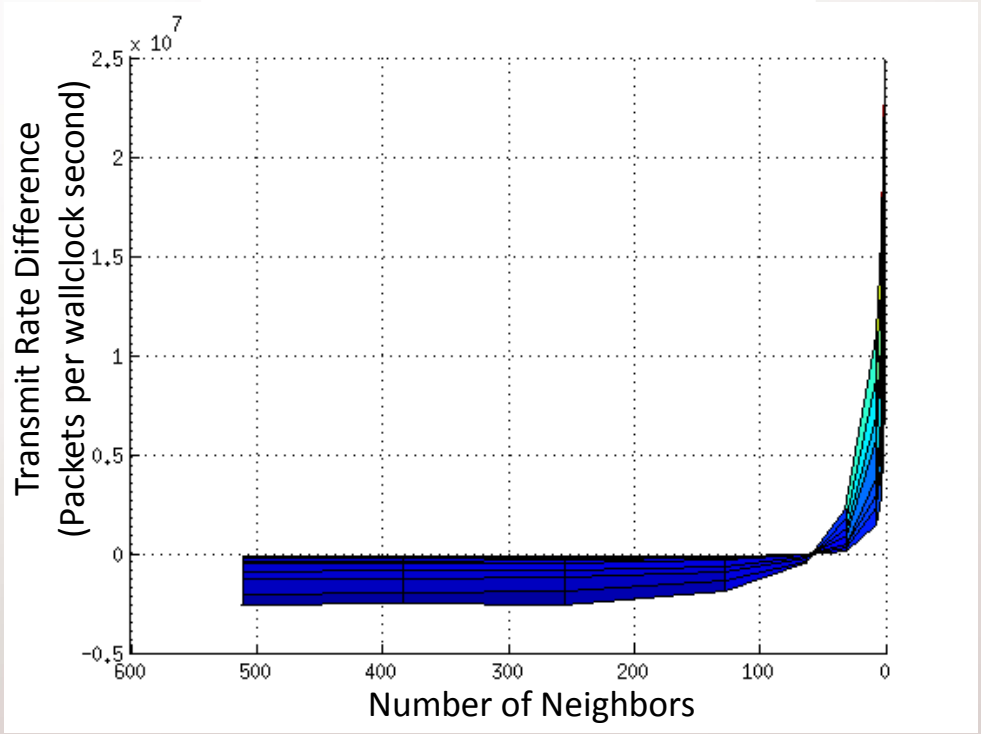
Results – Connectivity Thresholds (10%)





Results – Connectivity Thresholds (50%)





Conclusions

- Examined PHOLD performance of Granted Time Window and Null Message in ns-3
- Analyzed the transmit rate of local and remote messages
- Informal proof of concept for distributed simulations in ns-3 operating independent of the simulated networking overhead of ns-3
- Transmit rates increased with increasing lookahead
- Greater variability for Null Message (1000 to 27 million packets per second)
- Neighbor connectivity thresholds:
  - 0% remote traffic: **Neighbors <= 64** use **Null Message; Neighbors >= 128 neighbors** use **Granted Time Window**
  - 10% and 50% remote traffic: **Neighbors <= 32** use **Null Message; Neighbors >= 64 neighbors** use **Granted Time Window**

Future Work

- More concretely define the neighbor connectivity thresholds
- Future plans for ns-3: incorporation of optimistic synchronization methods
  - Will require performance comparisons as well

Questions?

- Thank you!