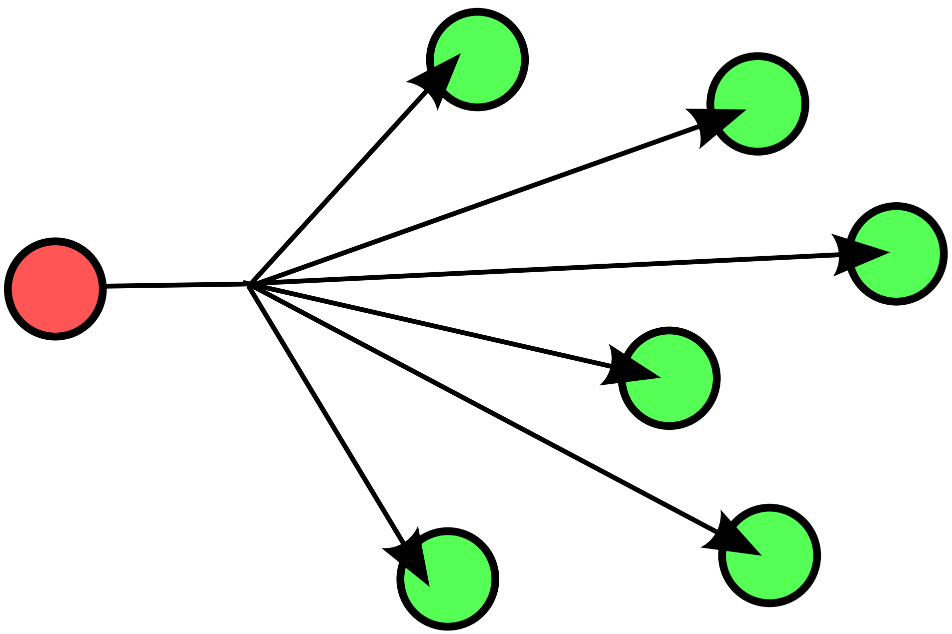
Energy-Efficient Broadcast Tree Protocol

Final Presentation



Source: https://en.wikipedia.org/wiki/Broadcasting_(networking)

23.10.2020 | Kevin Küchler | kkuechler@seemoo.tu-darmstadt.de

Introduction

- Energy-Efficient Broadcast Tree Construction
- Previous work
- Goals
- Energy-Efficient Broadcast Tree Protocol (EEBTP)
  - Changes to base design
  - Frame types
  - Cycle detection/prevention
  - Problems and solutions
- Evaluation
- Discussion
- Future work

Design with Potential Game

- Broadcast message to all nodes in a multi-hop network
- Build broadcast tree to minimize transmit power and then transmit data
- Potential game
  - Each node represents a rational player
  - Player reacts to actions of other players
  - Each action comes with a cost
  - Use weakly dominant strategy
    - Change between actions with the same cost
- Decentralized algorithm
- Use broadcast nature of wireless networks

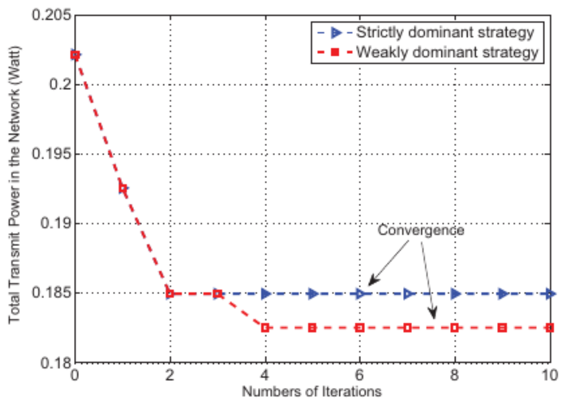


Fig. 3: Benefit of exploiting weakly dominant strategy for the proposed algorithm. There are 10 nodes in the network.

Source: [1]

“I want to play a game”

- Initiator or source node starts a game
- Players react to action of other players
- Player selects its parent
  - Cost for new parent is lower than or equal for the old parent
  - Connection must not create a cycle
  - New parent is connected to initiator
- Player finishes its game
  - After N rounds without changes
- Algorithm terminates after all players have finished

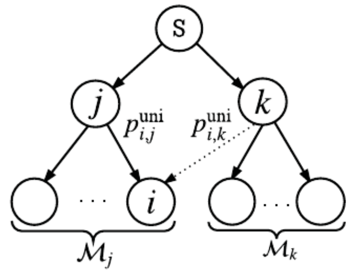


Fig. 1: A sample network. Solid and dashed arrows show current and possible connections a in broadcast tree, respectively. In this network node j is a parent node for node i and a child node for S.

Source: [1]

Previous Work

- Bachelor thesis of Sergío Domínguez
- Protocol design
  - Definition of frame types
  - Actions on frame types
  - Cycle prevention
    - Send own path to source with each discovery frame
    - Proposed in paper
- Energy model
  - Linear energy model
  - Energy curve for MAX2831 chip

Goals – Idealized Assumptions

- Addressing idealized assumptions
  - Node cannot know every action of every node
  - Nodes must communicate to exchange information
  - Packet loss
    - Noisy channel
    - Collisions of packets
  - Implementation behavior

Goals – Protocol Design

SECURE MOBILE NETWORKING

- Based on algorithm of Mahdi Mousavi
- Protocol header and frame types based on Sergío's previous work
- Addressing identified idealized assumptions

Goals – Implementation

- Implementation in ns-3
  - Based on Sergío's previous work
- Different mechanisms for cycle prevention/detection
  - *Ping-To-Source*
  - *Mutex*
  - *Path-To-Source*
- Custom energy model
- Simple broadcast for comparison

Goals – Evaluation

- How much application data must be sent until EEBTP cost less energy than other algorithms? (Break-even point)
- Which mechanisms variants are better under what conditions?
- Costs and benefits of the different mechanisms/designs
- How many packets get lost?
- Energy cost to construct a broadcast tree
- Time to construct a broadcast tree

EEBT-Protocol - Changes

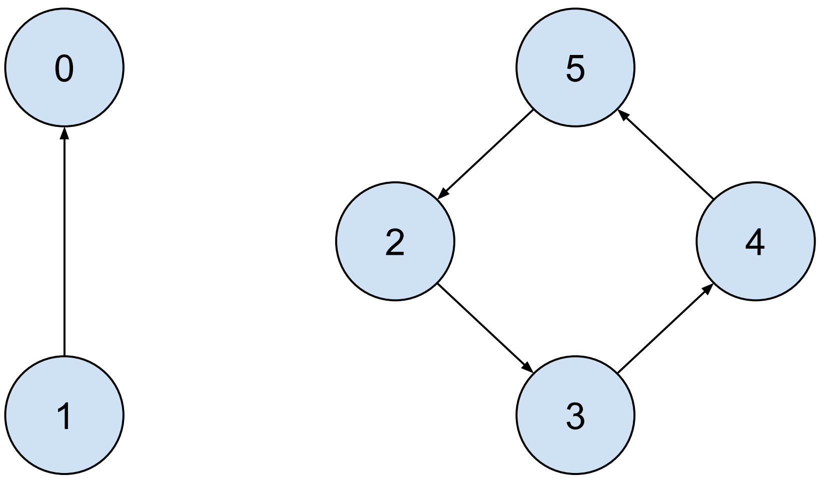
- Use Sergío’s previous design as base
- Header design changed
- Custom energy model
  - Models energy consumption of two chips (MAX2828, MAX2831)
    - Empirical data only for positive or negative dBm values (not both)
    - Approximation with only one chip too inaccurate
  - Easily expandable
- Multiple games at the same time
- Mechanisms for cycle detection/prevention

EEBT-Protocol – Frame Types

- Cycle detection/prevention
- Neighbor discovery
- Child request
- Child confirmation
- Child rejection
- Parent revocation
- End of game
- Application data

Graph Cycles

- Cycle prevention/detection is important
  - A cycle causes isolation of nodes from the network
  - Isolated nodes will not get the application data (theoretically)

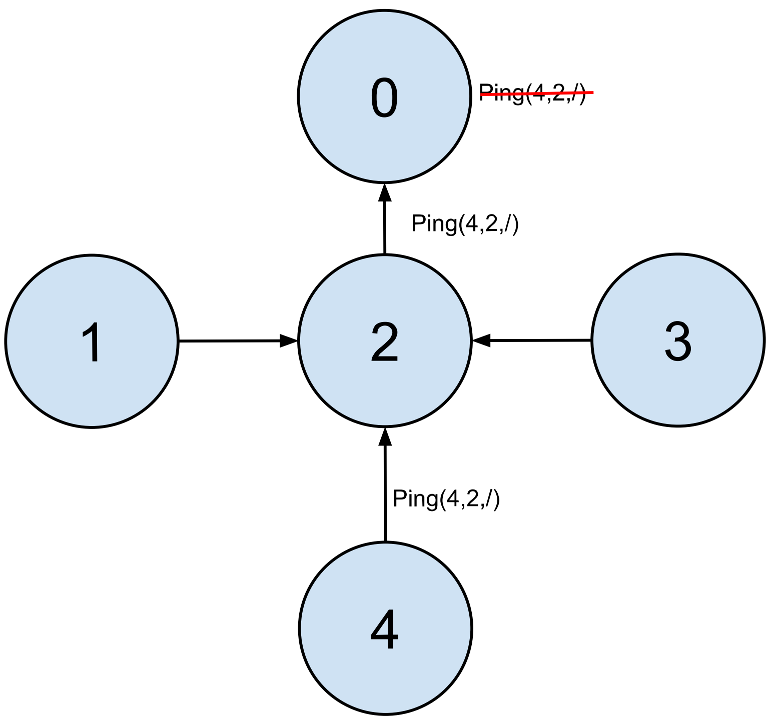


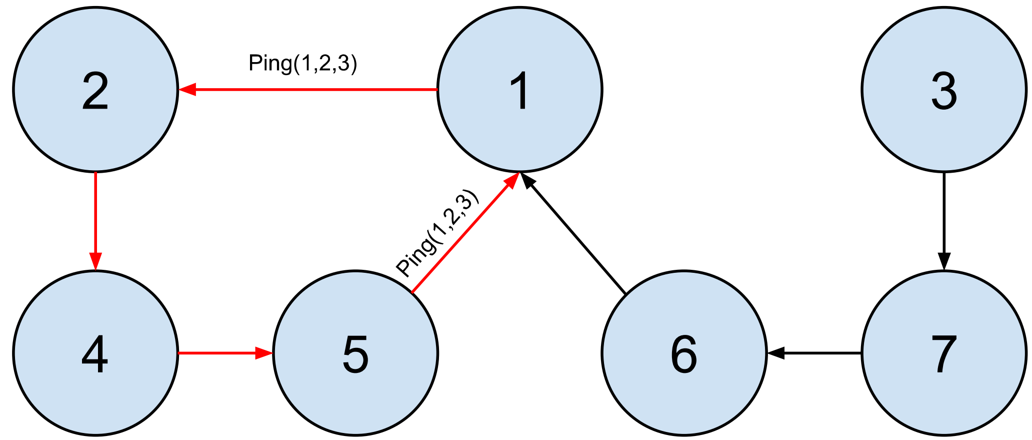
Cycle Detection - Ping to Source

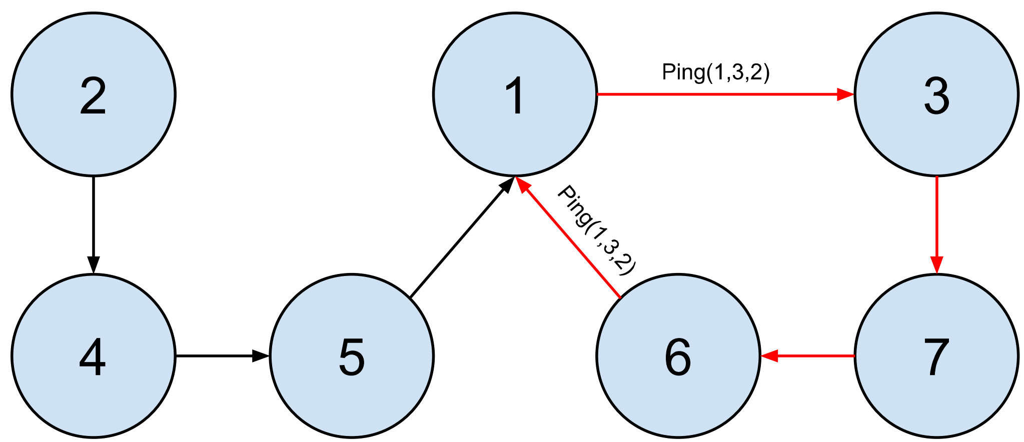
- Send ping (message) to source via new parent node
- Message contains information about
  - The sender
  - Old parent node
  - New parent node
- If message arrives at the sender
  - Cycle is detected
  - Connect to old parent
- Maintain blacklist and old parent stack
  - Prevents “Ping-Pong” effect
- Low resource cost

Cycle Detection - Ping to Source

Def: Ping(originator, newParent, oldParent)



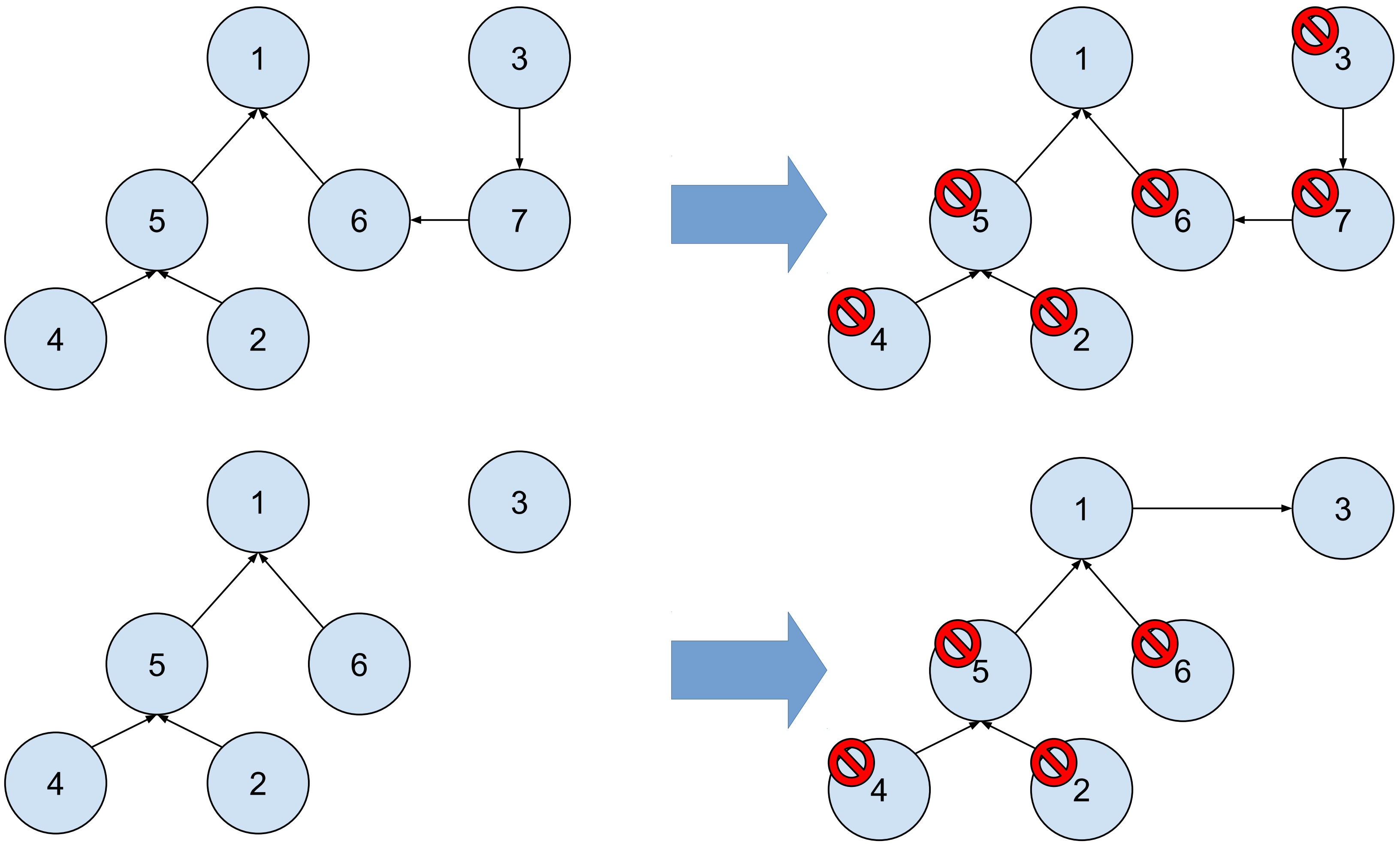




Cycle Prevention - Mutex

- New connection
  - Lock all child nodes in subtree
  - Connect to new parent node
  - Release lock
- New parent is locked
  - Possible cycle
  - New parent rejects child request
- Nodes are not allowed to do anything when finally locked
  - Node is finally locked when all child nodes responded
- High resource cost but effective
  - Costs a lot of time

Cycle Prevention - Mutex



Cycle Prevention - Path to Source

- Adopted from base design of Sergío Domínguez
- Each node maintains its path to the source node
- Every neighbor discovery frame contains this path
  - Receiver checks if it’s on the path of that node
  - If not, it can connect to the sender
- High resource cost and can fail
  - Costs less time due to async behavior
  - Information can get lost along the path
  - Cycles can occur

Ping-Pong Effect

- Node switches between different parent nodes
- Can have different causes
  - Parent has second node with same TX power
  - Noisy channel which leads to disconnects
  - Isolated group of nodes
- Solutions
  - Connection counter
  - Last parent stack
  - Rejection counter
  - Blacklist
- Problem remains sometimes

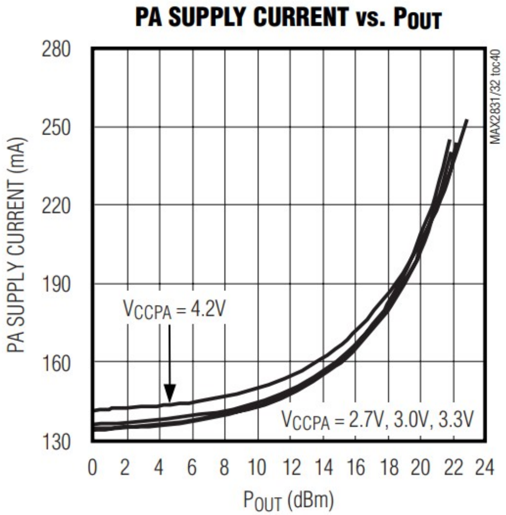
Problems with ns-3

- Missing event trigger in version 3.30.1
  - Solution: Fixed with version 3.31
- TX power per packet
  - Changed wifi-phy.cc (PHY-Layer)
  - Add packet tag with tx power information
- Error in wifi-phy-state-helper / energy model
  - Energy model switched to CCA_Busy during packet transmission when another packet arrives
  - Causes energy model to calculate energy for CCA_Busy mode and not TX mode
  - Results in very low energy consumption
  - Solution: Added exception for TX mode in wifi-phy-state-helper.cc

Custom Energy Model

SECURE MOBILE NETWORKING

- Standard ns-3 energy model is linear
  - Energy curve of real wifi chips is rather an exponential function
- Energy curve works only for positive values
  - Sergío fitted a polynomial
  - For larger negative values the result gets larger
  - Unrealistic energy consumption
- Lack of useful data
  - Many manufactures only specify max power
- Fitted exponential function from two chips
  - Data for positive and negative dBm values
  - Measured data are more realistic than a linear function



Source: [3]

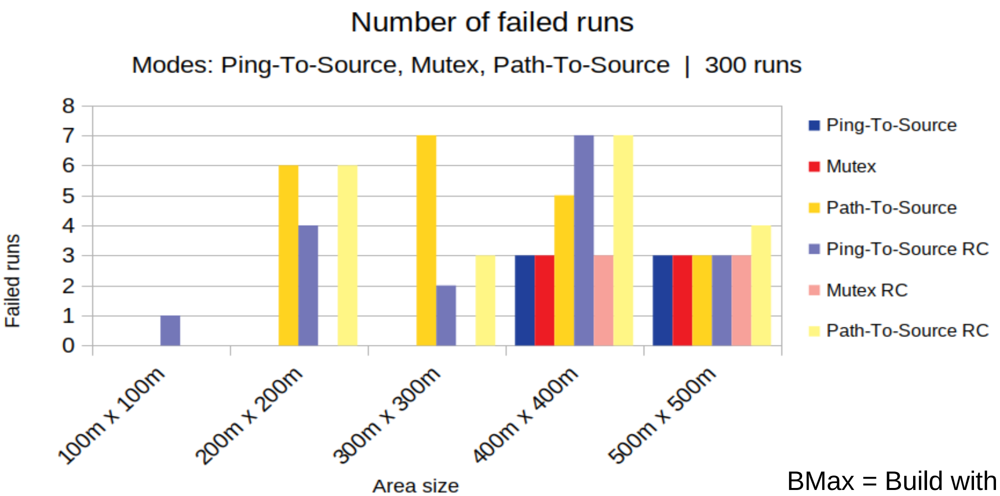
Evaluation

- Failed runs
- Ping-To-Source vs Mutex vs Path-To-Source
  - Energy to build a broadcast tree
  - Accumulated TX power
  - Time to build a broadcast tree
  - Tree depth and packet loss
  - Energy to transfer 1MB of data
  - Unconnected nodes
- Simple broadcast
- Request-To-Send – Clear-To-Send (RTS/CTS;RC)
- Construct tree with max TX power

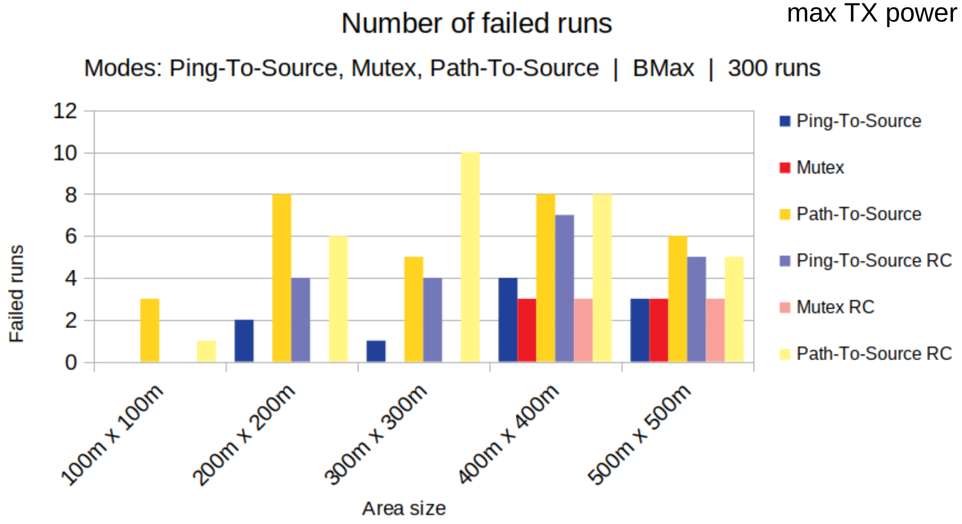
Evaluation – Failed Runs

SEMO SECURE MOBILE NETWORKING

- Build time measured with finish time of source node
- Build phase took too long (more than 10 seconds)
- Build phase never completed
  - Ping-Pong effect
  - Area too large
  - Isolation



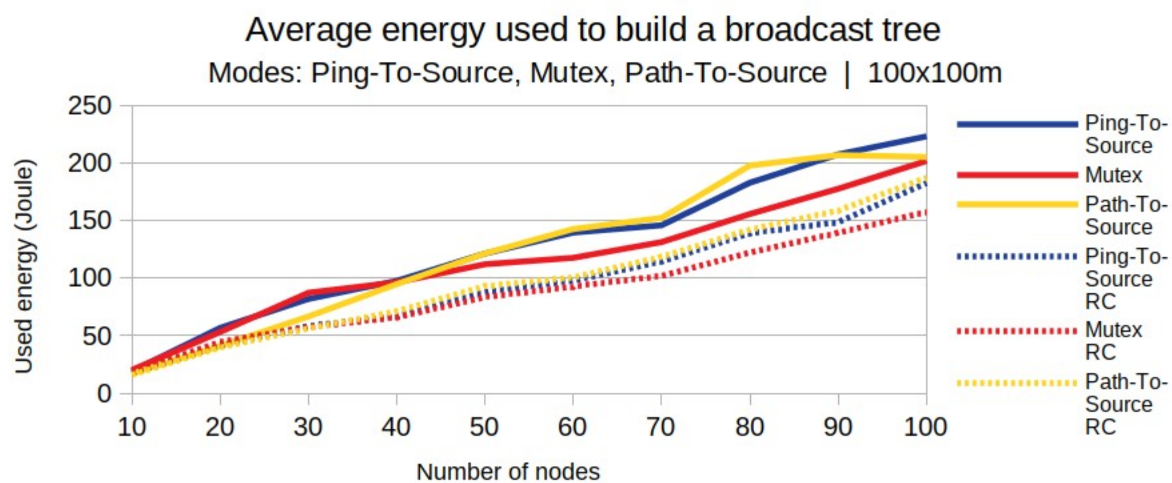
BMax = Build with max TX power

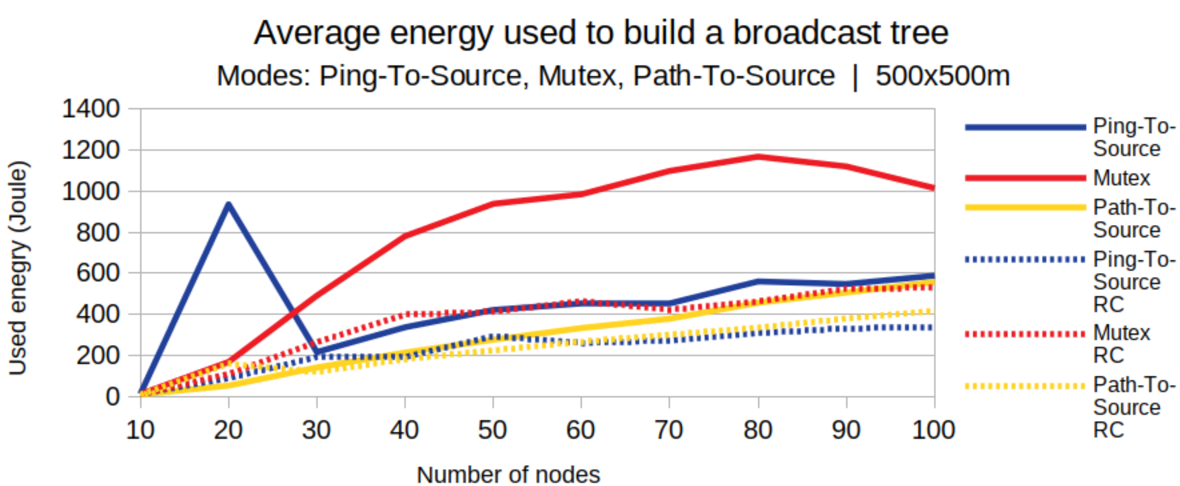


Evaluation – Build Energy

SECURE MOBILE NETWORKING

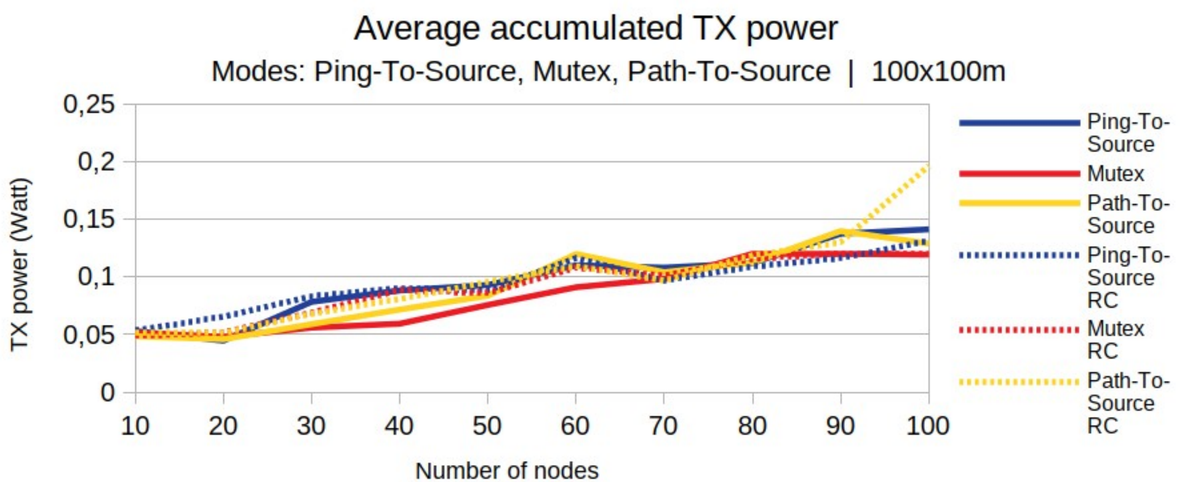
- *Mutex* needs more energy since it needs to send lock requests and responses. Child request get rejected when node is locked.
- RTS/CTS helps to reduce the needed build energy.

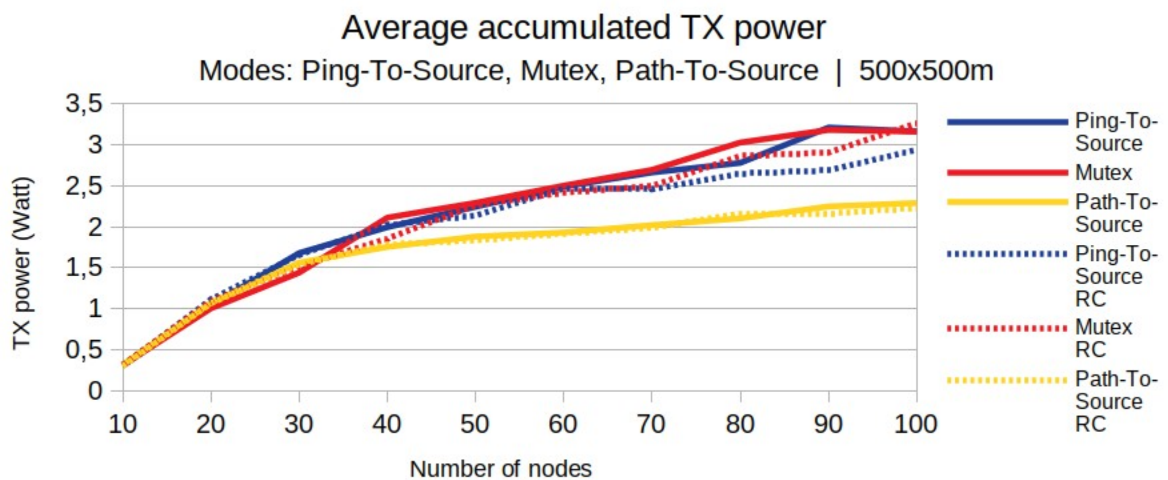




Evaluation – Accumulated TX Power

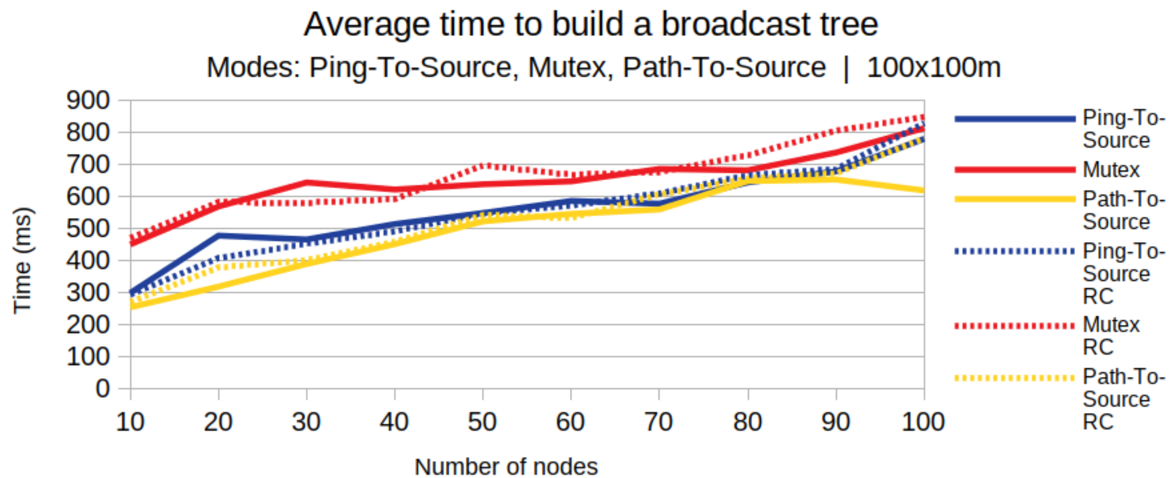
- Not much difference in dense areas
- RTS/CTS helps a little to reduce the TX power
- *Path-To-Source* scores better in large and low density areas but can create cycles

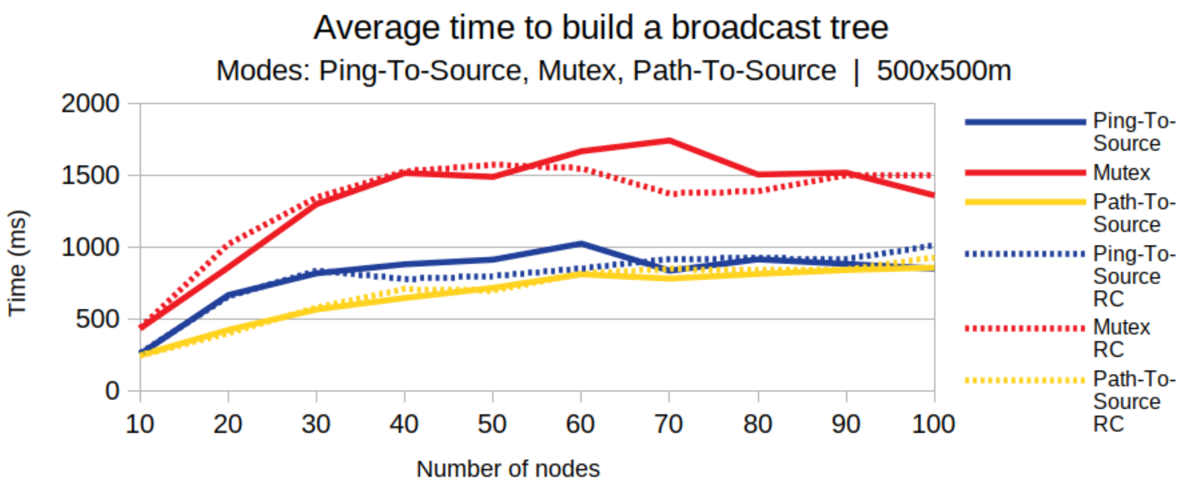




Evaluation – Build Time

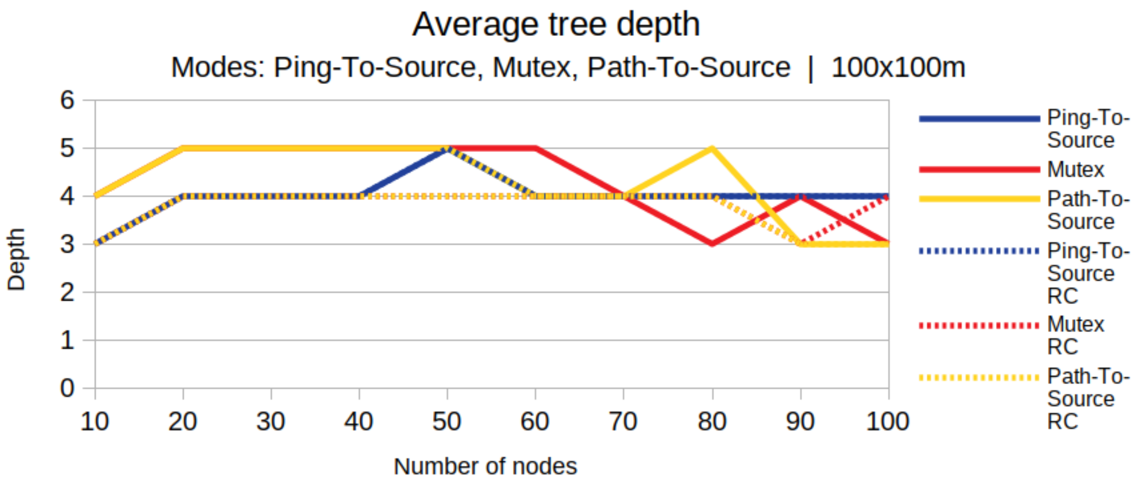
- Time increases with node count and area size
- *Mutex* takes longer due to the lock requests and responses
- RTS/CTS has a negligible effect

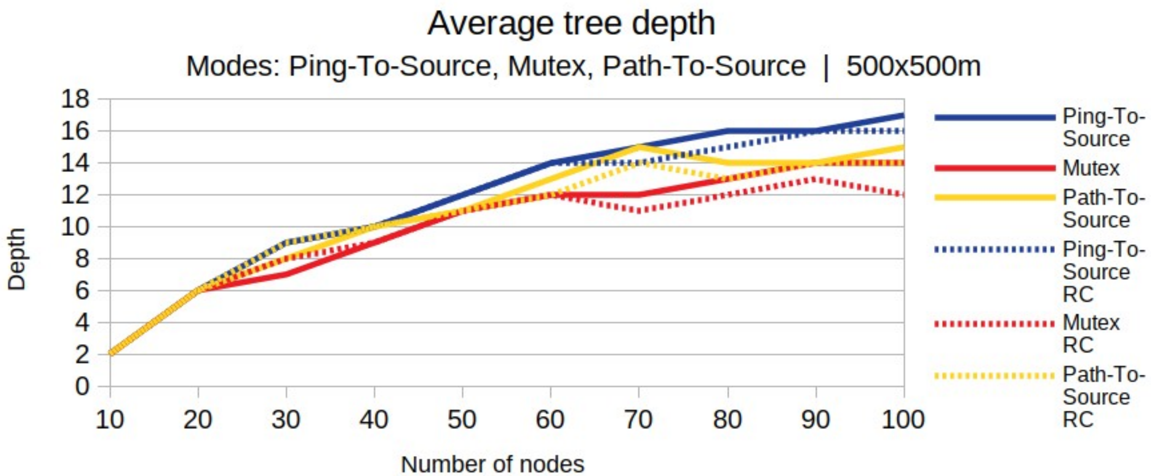




Evaluation – Tree Depth

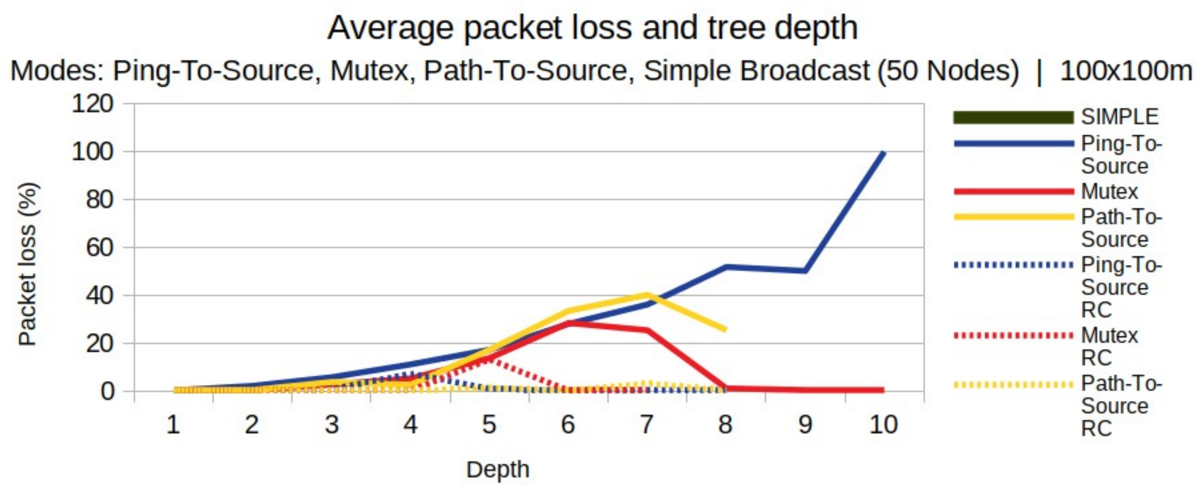
- Small, dense areas result in flat trees
- Large and sparse areas result in huge trees
- *Ping-To-Source* creates deeper trees than *Mutex*
- RTS/CTS has a negligible effect

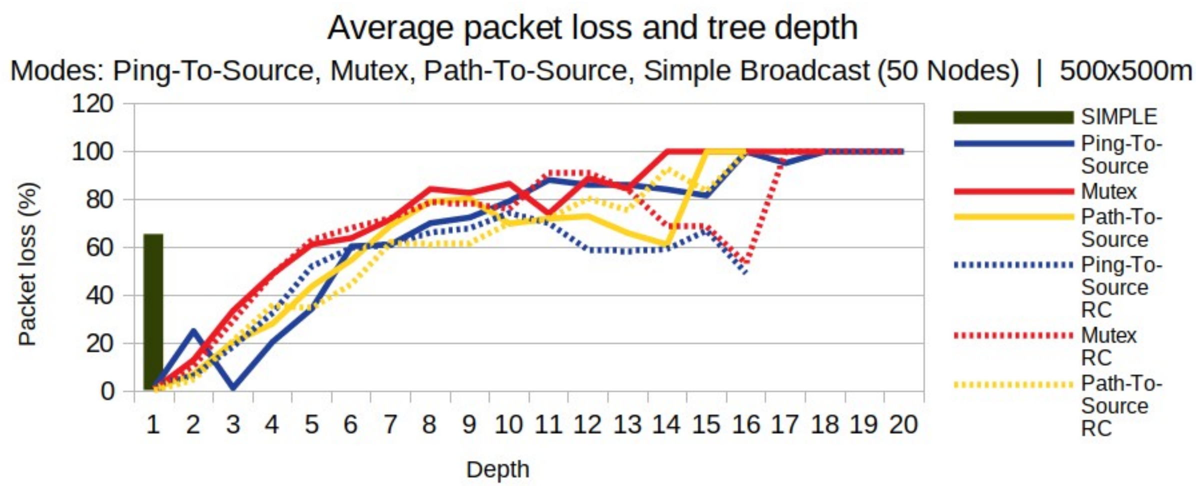




Evaluation – Tree Depth and Packet Loss

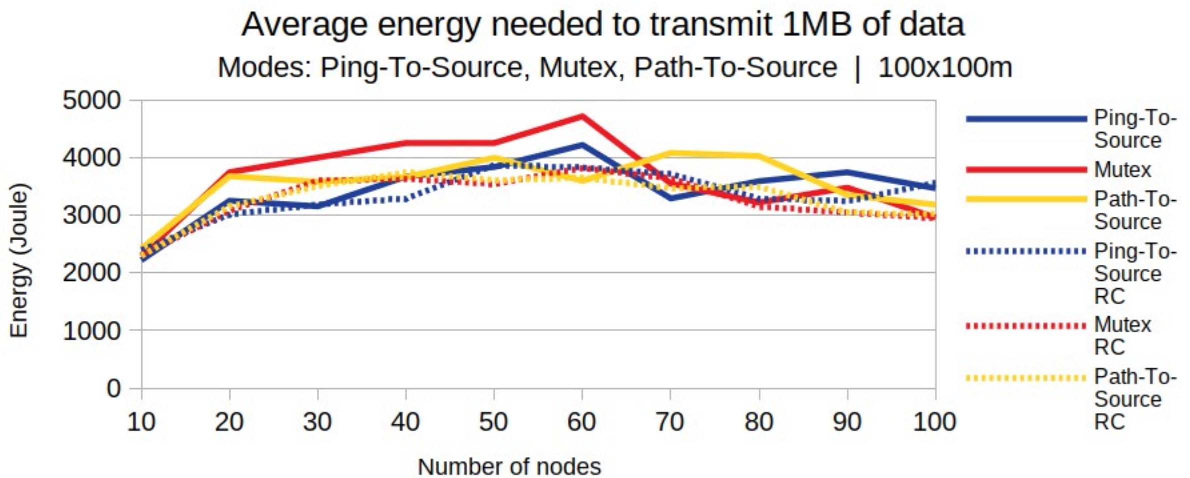
- Packet loss increases with tree depth
- Accepting packet from everyone and not only parent helps
- Large areas increase packet loss
- EEBTP can have a lower packet loss than Simple Broadcast

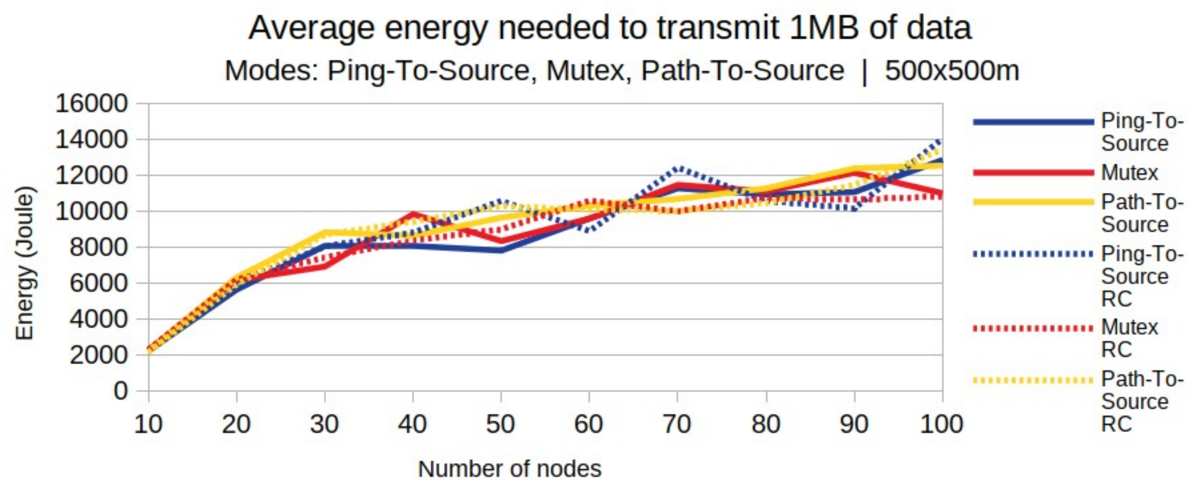




Evaluation – TX Energy

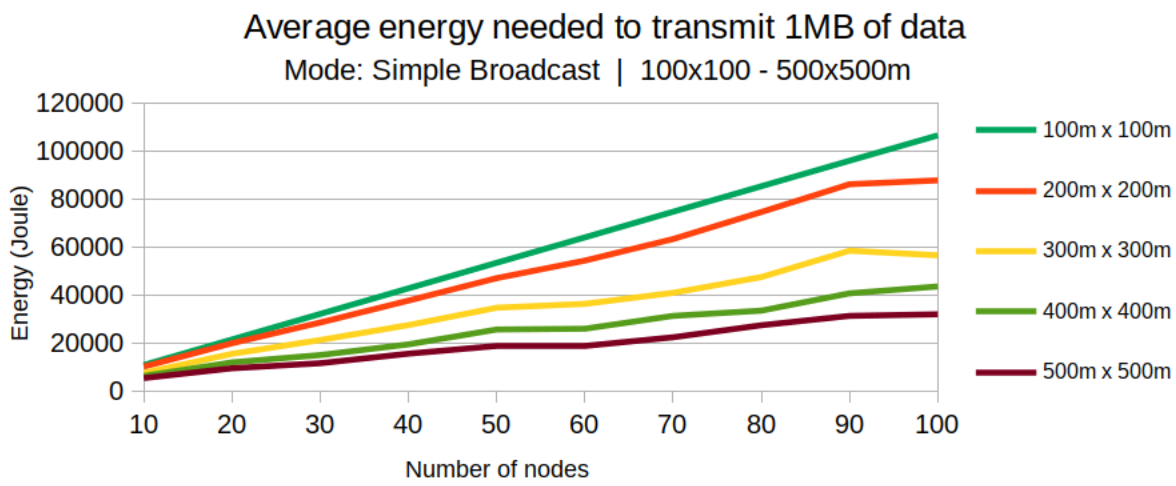
- RTS/CTS helps slightly
- Differences between mechanisms in large areas is small
- Large areas need more energy





Evaluation – TX Energy

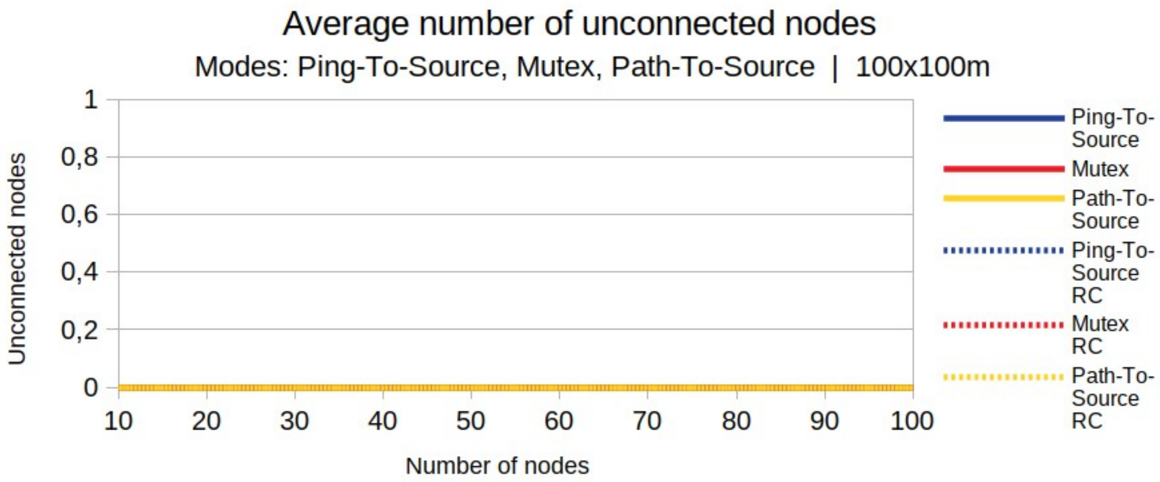
- Simple Broadcast uses much more energy
- In larger areas, packet loss gets higher and energy lower

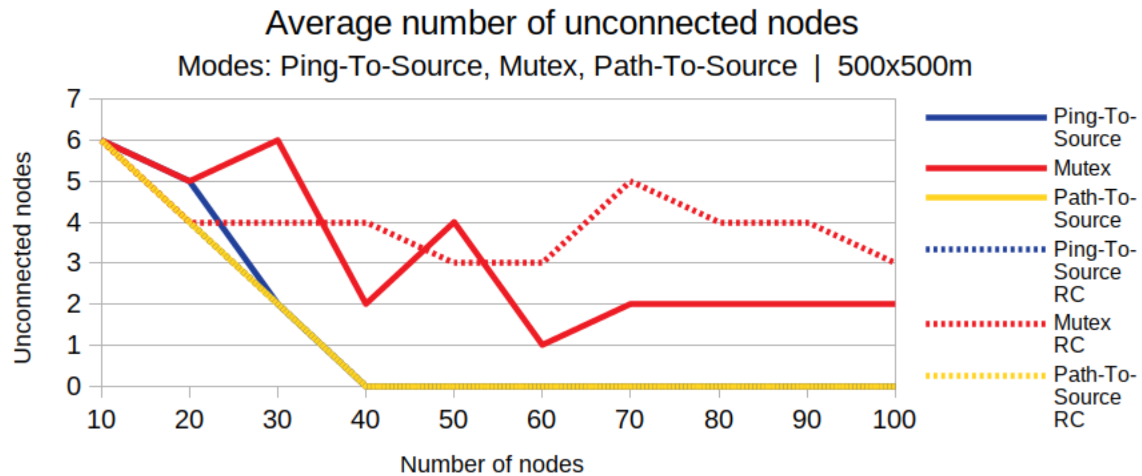


Evaluation – Unconnected Nodes

SECURE MOBILE NETWORKING

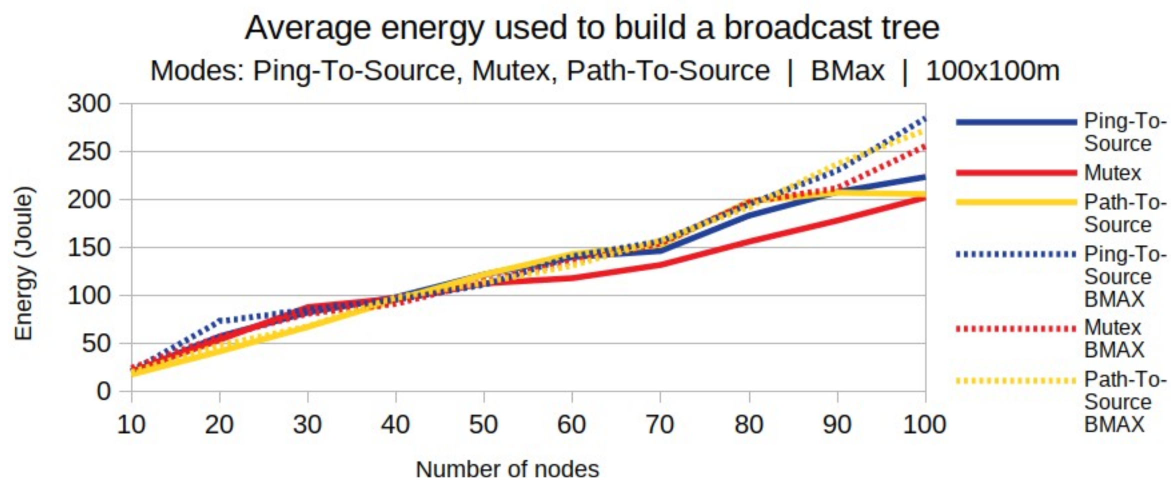
- In large areas, nodes are too far away to reach each other
- Mutex has more unconnected nodes due to high block times

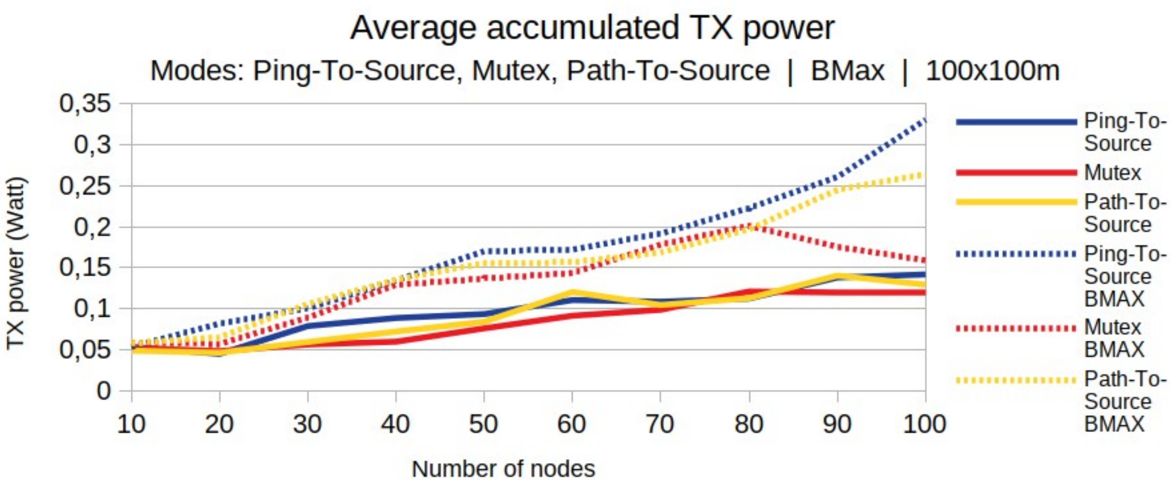




Evaluation – Build with Max TX

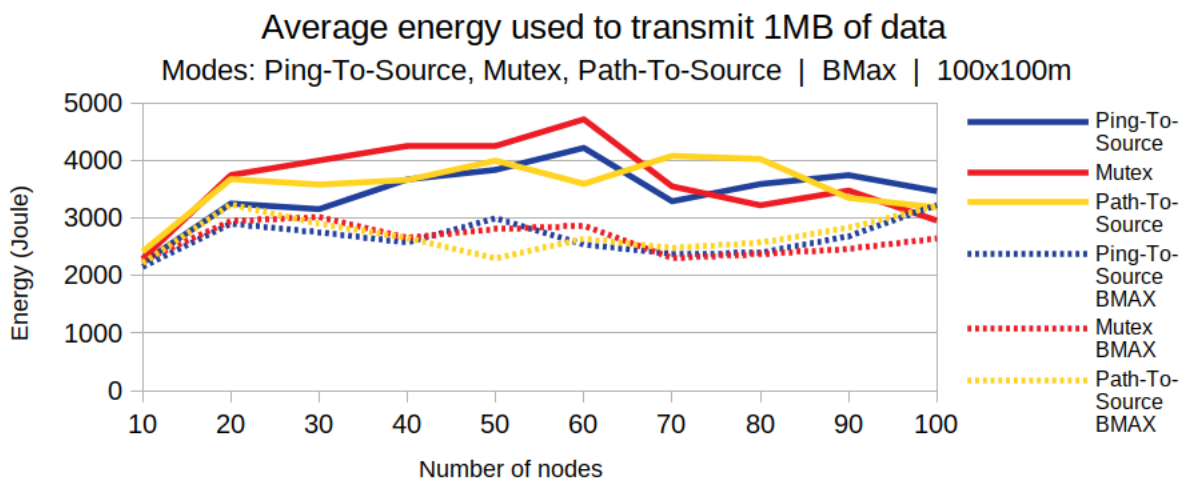
- Needs slightly more energy to build a broadcast tree
- Much higher accumulated TX power
- Build time is nearly the same

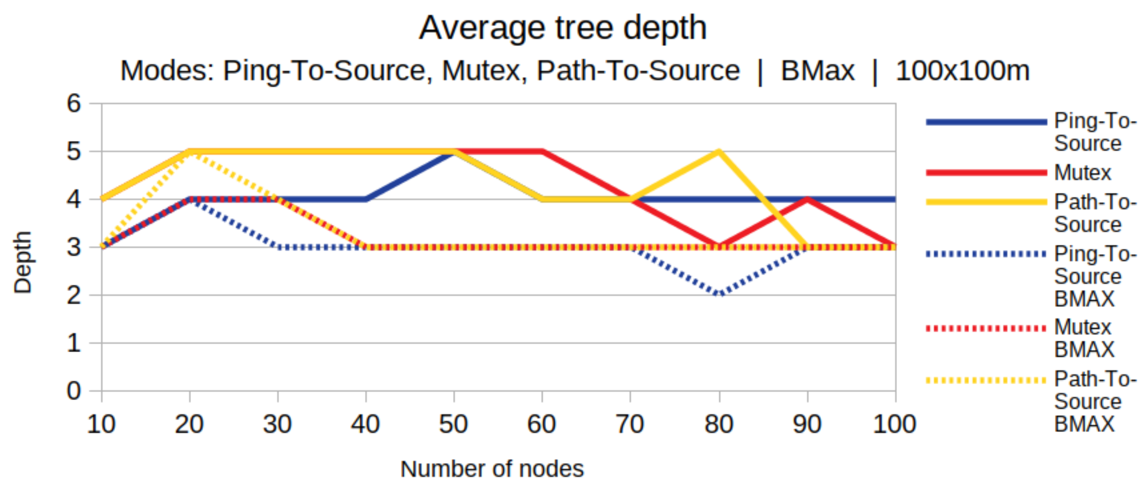




Evaluation – Build with Max TX

- Broadcast tree needd less energy to transfer 1MB of data
- Corresponds to smaller tree size
- Packet loss does not differ much
- Effect reduces in larger areas





Discussion

- EEBTP is capable of reducing the needed energy for data transmission
- EEBTP suffers from high noise levels
- Ping-To-Source is currently the optimal cycle detection mechanism
- Mutex never creates cycles but needs more time and build energy
- Path-To-Source is not suitable since cycles can occur due to lost information
- Nodes in a cycle can still receive data when they also accept data from nodes other than their parents

Discussion

- Large trees not suitable for VoIP or video streaming due to high packet loss
- RTS/CTS helps to improve results
- Using max TX power during build phase decreases the needed energy for data transmission
- Protocol may still fail to build a broadcast tree
  - Area too large
  - Endless build phase
    - Ping-Pong effect

Future Work

- Expand protocol to a mobile environment
- Compare to other decentralized algorithms in a more realistic environment
- Look further into a real test environment
- Better mechanisms to handle the effects of noise
- Prevent endless build phase due to “Ping-Pong” effect

Questions and Feedback

- [1] Mahdi Mousavi et al., “Energy-Efficient Data Dissemination in Ad Hoc Networks: Mechanism Design with Potential Game”, ISWCS, 2015
- [2] Sergio Domínguez Moreno, “Practical Broadcast Tree Construction with Potential Game for Energy-Efficient Data Dissemination in Ad-Hoc Network”, Bachelor Thesis, 2018
- [3] Datasheet of the MAX2831/MAX2832 RF Transceivers, https://datasheets.maximintegrated.com/en/ds/MAX2831-MAX2832.pdf