# AODV in MATLAB

CpE 6420 Project Presentations

Stuart Miller

MISSOURI S&T

# Agenda

> **Background**

> Examples

> Traffic Statistics

> Future Work

> Conclusions

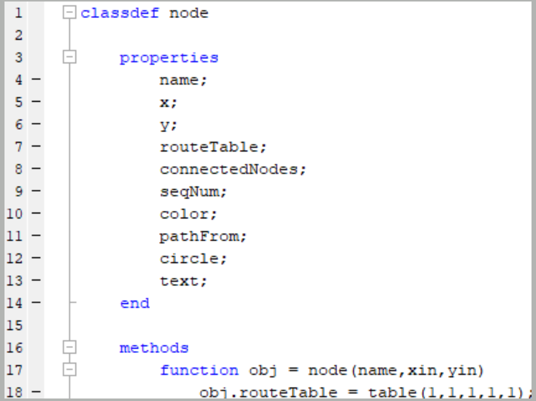
> Demo

# AODV Routing Algorithm

- Routing protocol for ad-hoc wireless networks
- Outlined in RFC 3561
  - C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003. (https://www.rfc-editor.org/rfc/rfc3561.txt)
- Utilized by Zigbee specification

# AODV Principles

- Reactive routing protocol
- Reduces network-wide broadcasts
- Lower overhead
- Discovers routes only as necessary
- Relies on flooding for route discovery
- Each node maintains its own route table



```
classdef node

    properties
        name;
        x;
        y;
        routeTable;
        connectedNodes;
        seqNum;
        color;
        pathFrom;
        circle;
        text;
    end

    methods
        function obj = node(name,xin,yin)
            obj.routeTable = table(1,1,1,1,1);
```

# AODV Route Messages

- Route Request (RREQ) →
  - Sent when a node doesn’t have a valid path to the destination, triggers flooding
- Route Reply (RREPL) →
  - Sent back to source when the destination is reached or an intermediate node has a route to the destination.
- Route Error (RERR) →
  - Send back up the path of propagation by a node when its link to the intended destination breaks
- Data →
  - Just normal packets containing actual information

# Approach

> Implement in MATLAB, make use of high-quality GUI environment

> Focus on resiliency throughout node movement (broken links, etc.)

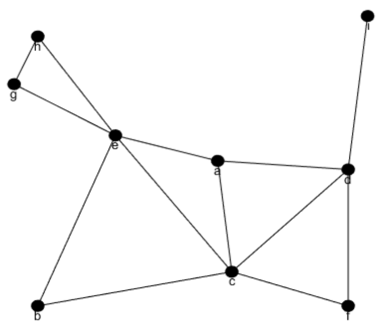
> Serve as more of a demonstration tool rather than an in-depth simulation

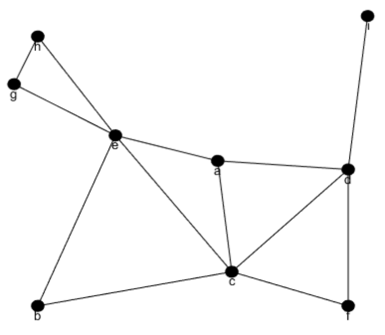
> Focus on showing step-by-step progress of algorithm

# Agenda

- > Background
- > **Examples**
- > Traffic Statistics
- > Future Work
- > Conclusions
- > Demo

# Example 1



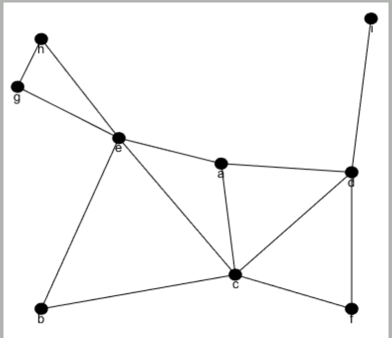


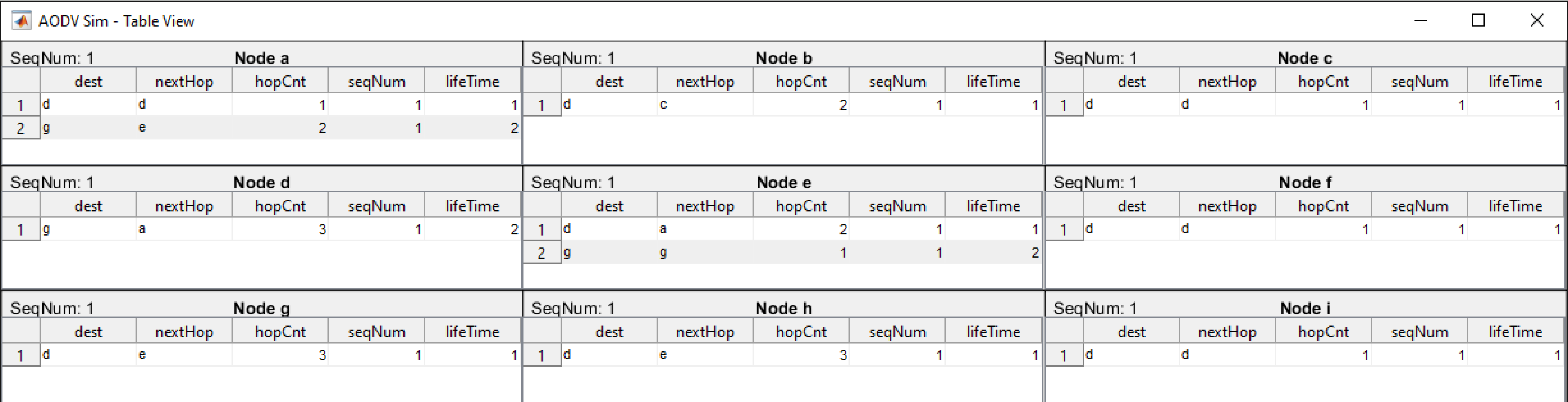
  - Sending node D -> G
  - All routing tables start empty

- D floods with **RREQ**
- G replies with **RREPL**
- **Data** sent once route established
- Subsequent transmissions require no new overhead

# Example 1

- > Reverse routes set up during flooding
- > Forward route set up during reply





SeqNum: 1 **Node a**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |
| 2 | g | e | 2 | 1 | 2 |

SeqNum: 1 **Node b**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | c | 2 | 1 | 1 |

SeqNum: 1 **Node c**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |

SeqNum: 1 **Node d**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | g | a | 3 | 1 | 2 |

SeqNum: 1 **Node e**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | a | 2 | 1 | 1 |
| 2 | g | g | 1 | 1 | 2 |

SeqNum: 1 **Node f**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |

SeqNum: 1 **Node g**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | e | 3 | 1 | 1 |

SeqNum: 1 **Node h**

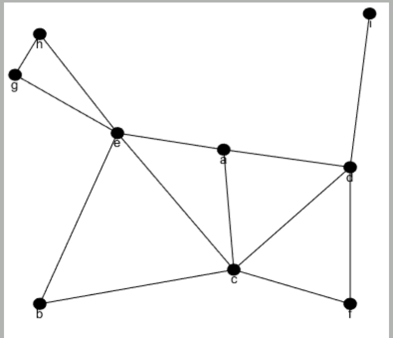
| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | e | 3 | 1 | 1 |

SeqNum: 1 **Node i**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |

# Example 2

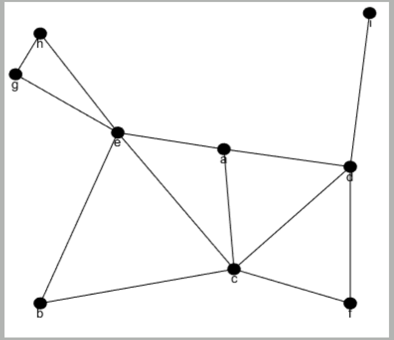
- – Sending node C -> G
- – Intermediates nodes have route info from Ex. 1

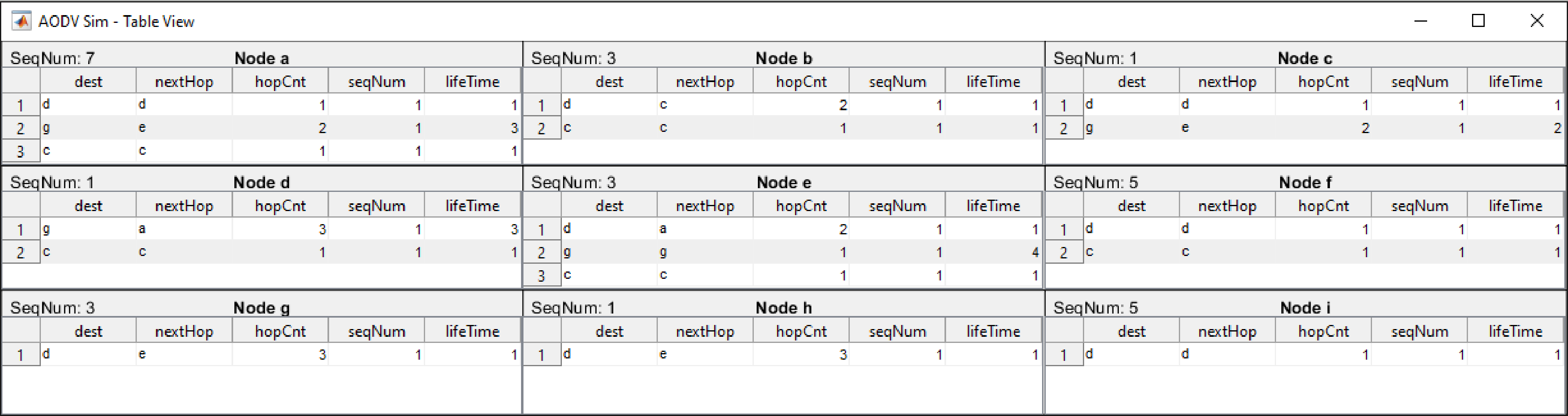


- > C floods with **RREQ**
- > D,A,&G all reply with **RREPL**
- > C picks reply with smallest hop count to destination
- > **Data** sent once route established

# Example 2

> More routes added




AODV Sim - Table View

SeqNum: 7 **Node a**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |
| 2 | g | e | 2 | 1 | 3 |
| 3 | c | c | 1 | 1 | 1 |

SeqNum: 3 **Node b**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | c | 2 | 1 | 1 |
| 2 | c | c | 1 | 1 | 1 |

SeqNum: 1 **Node c**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |
| 2 | g | e | 2 | 1 | 2 |

SeqNum: 1 **Node d**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | g | a | 3 | 1 | 3 |
| 2 | c | c | 1 | 1 | 1 |

SeqNum: 3 **Node e**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | a | 2 | 1 | 1 |
| 2 | g | g | 1 | 1 | 4 |
| 3 | c | c | 1 | 1 | 1 |

SeqNum: 5 **Node f**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |
| 2 | c | c | 1 | 1 | 1 |

SeqNum: 3 **Node g**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | e | 3 | 1 | 1 |

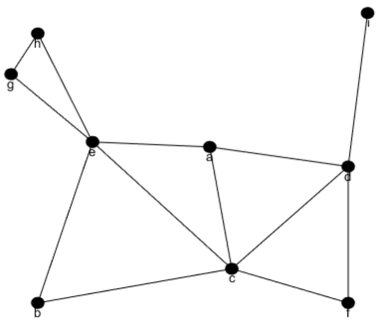
SeqNum: 1 **Node h**

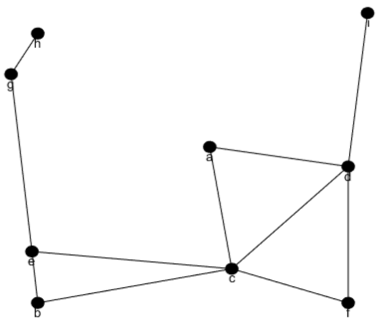
| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | e | 3 | 1 | 1 |

SeqNum: 5 **Node i**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |

# Example 3





- Sending node D -> G again
- Node E has been moved, breaking links

> D tries sending normally

> A can’t reach E anymore, so replies with **RERR**

> A now knows no routes, so must flood

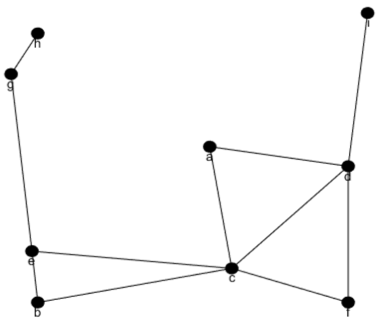
> C knows a route to G so **replies**

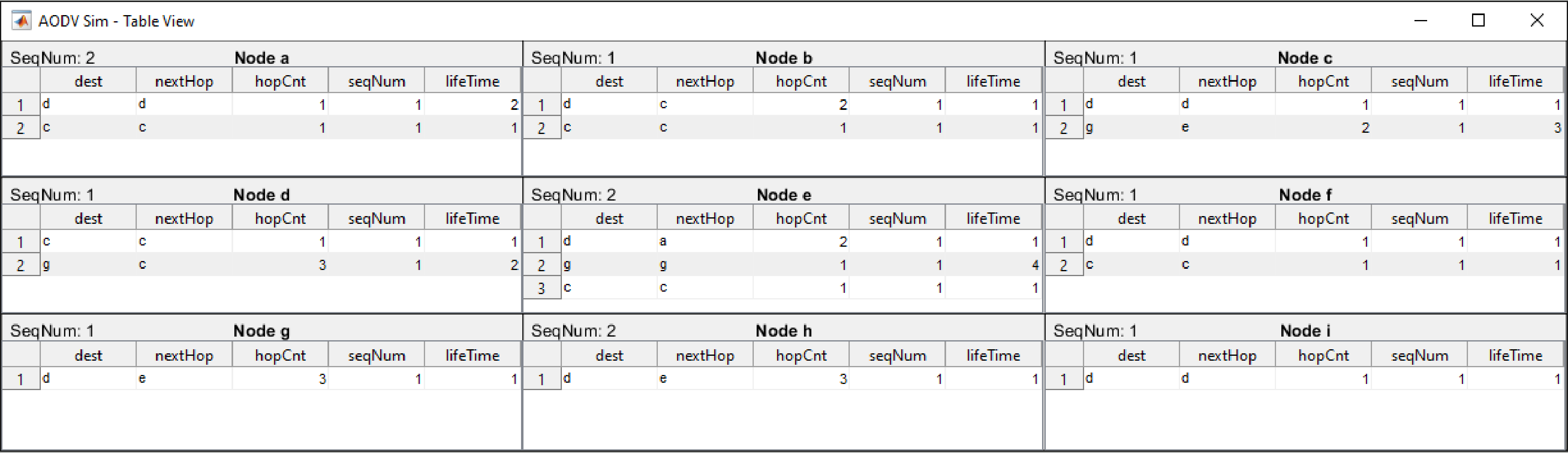
> D sends to G

# Example 3

> The **RERR** canceled out A->G

> The sequence numbers changed





SeqNum: 2 **Node a**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 2 |
| 2 | c | c | 1 | 1 | 1 |

SeqNum: 1 **Node b**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | c | 2 | 1 | 1 |
| 2 | c | c | 1 | 1 | 1 |

SeqNum: 1 **Node c**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |
| 2 | g | e | 2 | 1 | 3 |

SeqNum: 1 **Node d**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | c | c | 1 | 1 | 1 |
| 2 | g | c | 3 | 1 | 2 |

SeqNum: 2 **Node e**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | a | 2 | 1 | 1 |
| 2 | g | g | 1 | 1 | 4 |
| 3 | c | c | 1 | 1 | 1 |

SeqNum: 1 **Node f**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |
| 2 | c | c | 1 | 1 | 1 |

SeqNum: 1 **Node g**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | e | 3 | 1 | 1 |

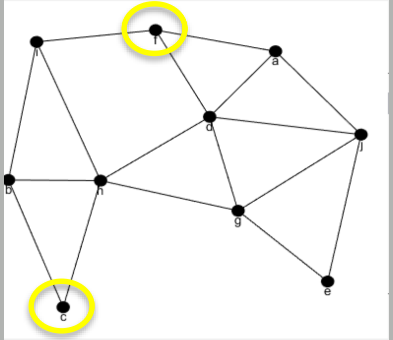
SeqNum: 2 **Node h**

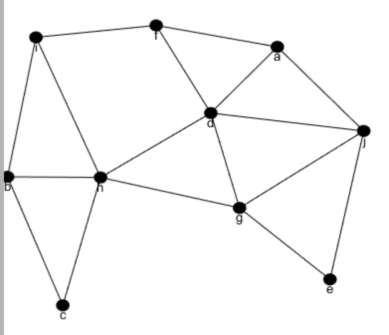
| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | e | 3 | 1 | 1 |

SeqNum: 1 **Node i**

| | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 1 |

# Example 4





- Much shuffling, routes have almost all changed

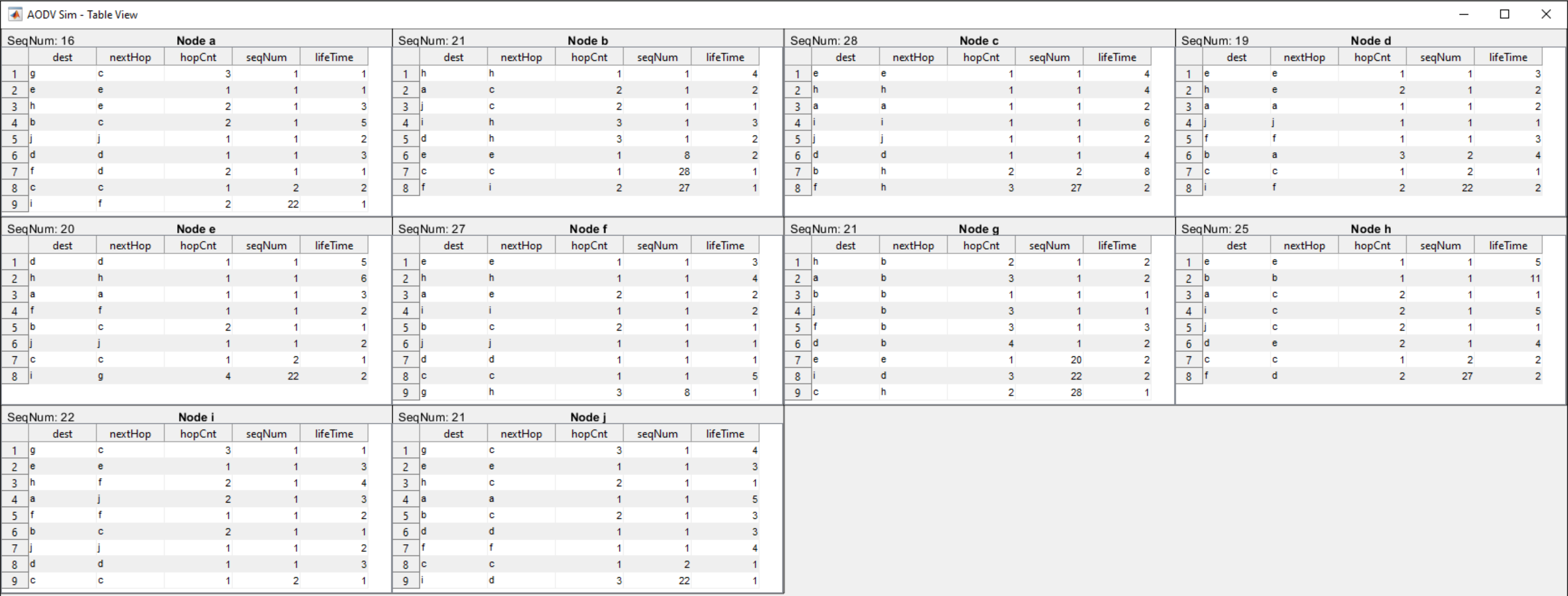
> Multiple route **cancellations**

> Once C receives a reply, it sends out data, to the node with the lowest hop count to destination, expecting it to make it

> Floods multiple times

> D’s route to F is still valid though

# Example 4



SeqNum: 16 **Node a**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | g | c | 3 | 1 | 1 |
| 2 | e | e | 1 | 1 | 1 |
| 3 | h | e | 2 | 1 | 3 |
| 4 | b | c | 2 | 1 | 5 |
| 5 | j | j | 1 | 1 | 2 |
| 6 | d | d | 1 | 1 | 3 |
| 7 | f | d | 2 | 1 | 1 |
| 8 | c | c | 1 | 2 | 2 |
| 9 | i | f | 2 | 22 | 1 |

SeqNum: 21 **Node b**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | h | h | 1 | 1 | 4 |
| 2 | a | c | 2 | 1 | 2 |
| 3 | j | c | 2 | 1 | 1 |
| 4 | i | h | 3 | 1 | 3 |
| 5 | d | h | 3 | 1 | 2 |
| 6 | e | e | 1 | 8 | 2 |
| 7 | c | c | 1 | 28 | 1 |
| 8 | f | i | 2 | 27 | 1 |

SeqNum: 28 **Node c**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | e | e | 1 | 1 | 4 |
| 2 | h | h | 1 | 1 | 4 |
| 3 | a | a | 1 | 1 | 2 |
| 4 | i | i | 1 | 1 | 6 |
| 5 | j | j | 1 | 1 | 2 |
| 6 | d | d | 1 | 1 | 4 |
| 7 | b | h | 2 | 2 | 8 |
| 8 | f | h | 3 | 27 | 2 |

SeqNum: 19 **Node d**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | e | e | 1 | 1 | 3 |
| 2 | h | e | 2 | 1 | 2 |
| 3 | a | a | 1 | 1 | 2 |
| 4 | j | j | 1 | 1 | 1 |
| 5 | f | f | 1 | 1 | 3 |
| 6 | b | a | 3 | 2 | 4 |
| 7 | c | c | 1 | 2 | 1 |
| 8 | i | f | 2 | 22 | 2 |

SeqNum: 20 **Node e**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | d | d | 1 | 1 | 5 |
| 2 | h | h | 1 | 1 | 6 |
| 3 | a | a | 1 | 1 | 3 |
| 4 | f | f | 1 | 1 | 2 |
| 5 | b | c | 2 | 1 | 1 |
| 6 | j | j | 1 | 1 | 2 |
| 7 | c | c | 1 | 2 | 1 |
| 8 | i | g | 4 | 22 | 2 |

SeqNum: 27 **Node f**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | e | e | 1 | 1 | 3 |
| 2 | h | h | 1 | 1 | 4 |
| 3 | a | e | 2 | 1 | 2 |
| 4 | i | i | 1 | 1 | 2 |
| 5 | b | c | 2 | 1 | 1 |
| 6 | j | j | 1 | 1 | 1 |
| 7 | d | d | 1 | 1 | 1 |
| 8 | c | c | 1 | 1 | 5 |
| 9 | g | h | 3 | 8 | 1 |

SeqNum: 21 **Node g**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | h | b | 2 | 1 | 2 |
| 2 | a | b | 3 | 1 | 2 |
| 3 | b | b | 1 | 1 | 1 |
| 4 | j | b | 3 | 1 | 1 |
| 5 | f | b | 3 | 1 | 3 |
| 6 | d | b | 4 | 1 | 2 |
| 7 | e | e | 1 | 20 | 2 |
| 8 | i | d | 3 | 22 | 2 |
| 9 | c | h | 2 | 28 | 1 |

SeqNum: 25 **Node h**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | e | e | 1 | 1 | 5 |
| 2 | b | b | 1 | 1 | 11 |
| 3 | a | c | 2 | 1 | 1 |
| 4 | i | c | 2 | 1 | 5 |
| 5 | j | c | 2 | 1 | 1 |
| 6 | d | e | 2 | 1 | 4 |
| 7 | c | c | 1 | 2 | 2 |
| 8 | f | d | 2 | 27 | 2 |

SeqNum: 22 **Node i**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | g | c | 3 | 1 | 1 |
| 2 | e | e | 1 | 1 | 3 |
| 3 | h | f | 2 | 1 | 4 |
| 4 | a | j | 2 | 1 | 3 |
| 5 | f | f | 1 | 1 | 2 |
| 6 | b | c | 2 | 1 | 1 |
| 7 | j | j | 1 | 1 | 2 |
| 8 | d | d | 1 | 1 | 3 |
| 9 | c | c | 1 | 2 | 1 |

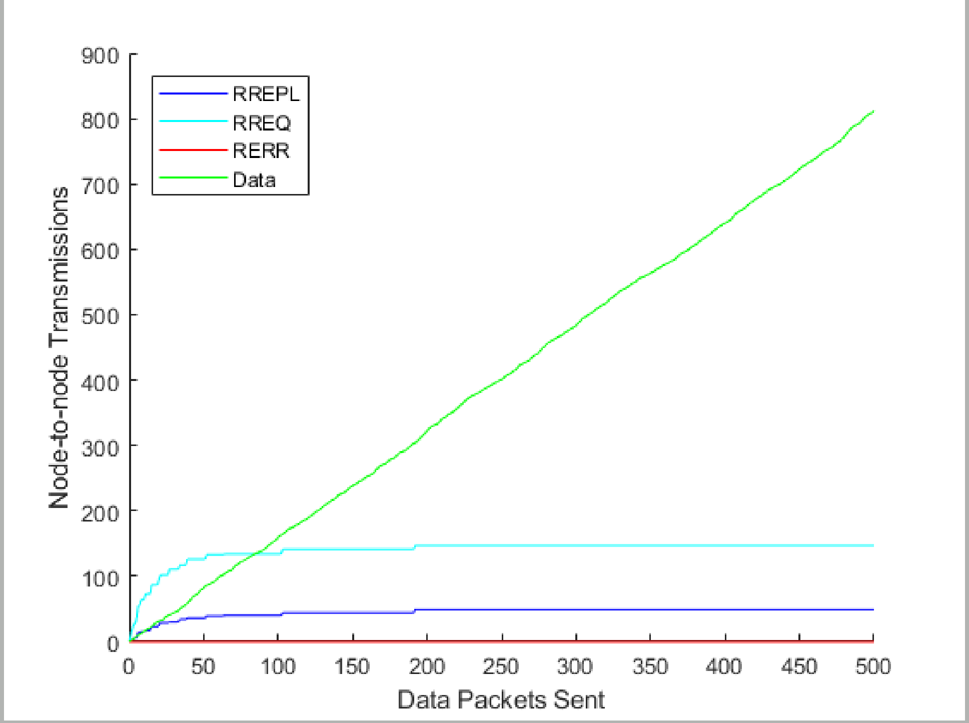
SeqNum: 21 **Node j**

|  | dest | nextHop | hopCnt | seqNum | lifeTime |
|---|---|---|---|---|---|
| 1 | g | c | 3 | 1 | 4 |
| 2 | e | e | 1 | 1 | 3 |
| 3 | h | c | 2 | 1 | 1 |
| 4 | a | a | 1 | 1 | 5 |
| 5 | b | c | 2 | 1 | 3 |
| 6 | d | d | 1 | 1 | 3 |
| 7 | f | f | 1 | 1 | 4 |
| 8 | c | c | 1 | 2 | 1 |
| 9 | i | d | 3 | 22 | 1 |

# Agenda

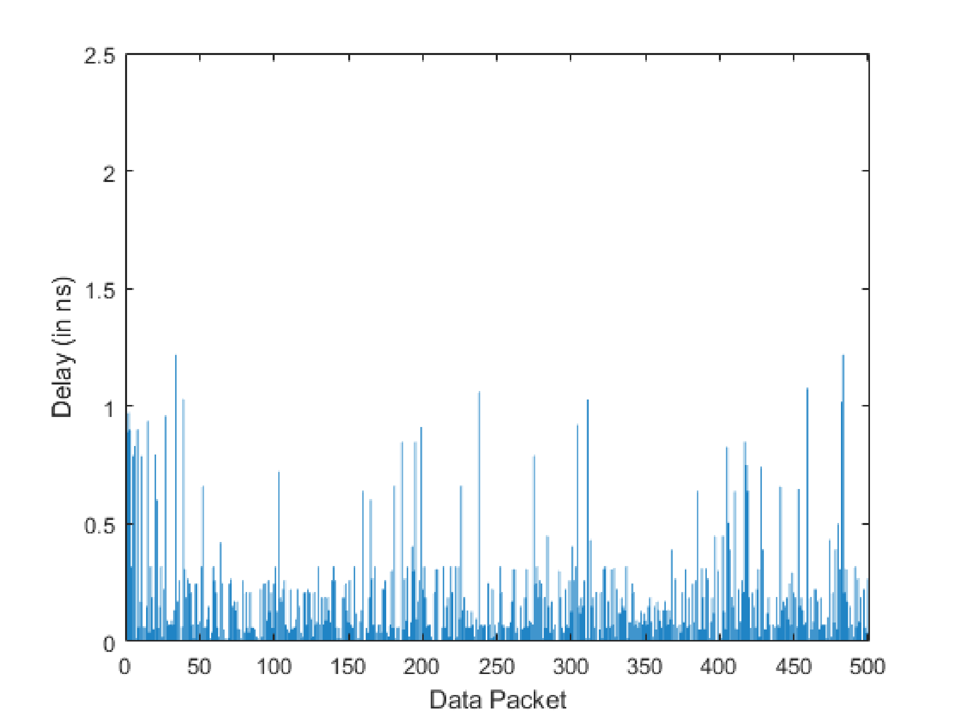
> Background
> Examples
> **Traffic Statistics**
> Future Work
> Conclusions
> Demo

# Static Network



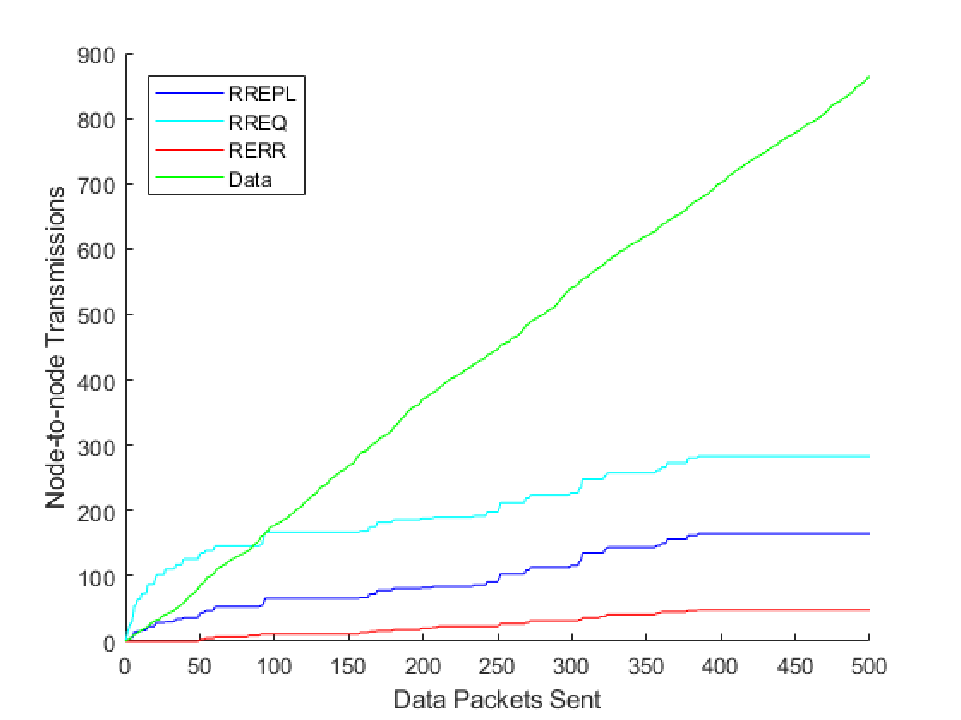
- 500 random packets sent
- No movement, nodes remain in the same place
- No **RERR**s

# Static Network



- No movement
- Propagation delays

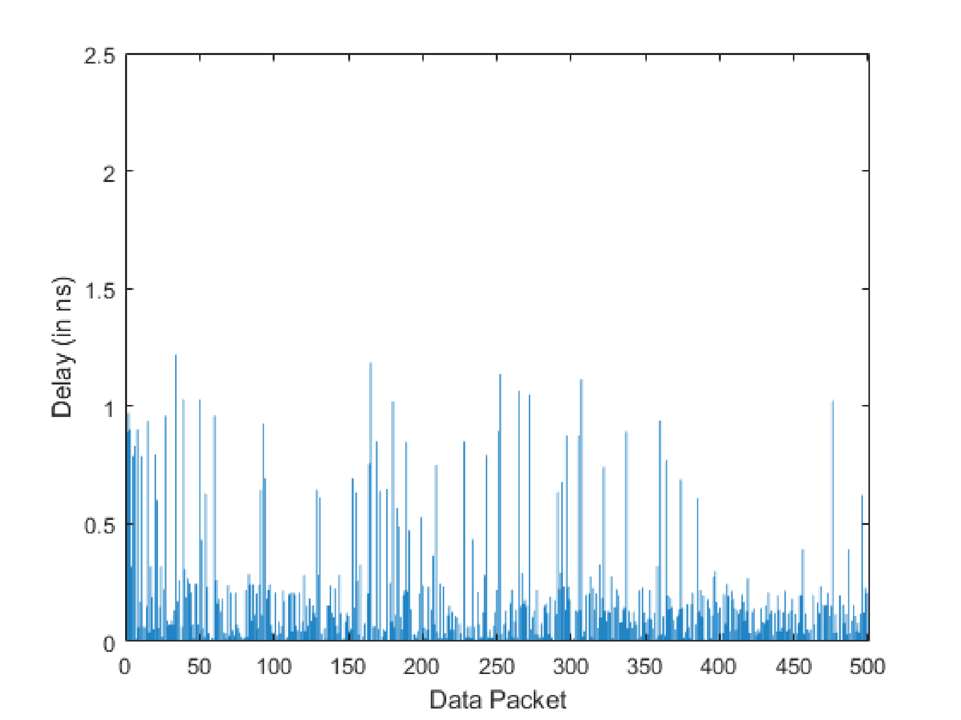
# Mobile Network



- > 500 random packets sent
- > Movement every 50 packets

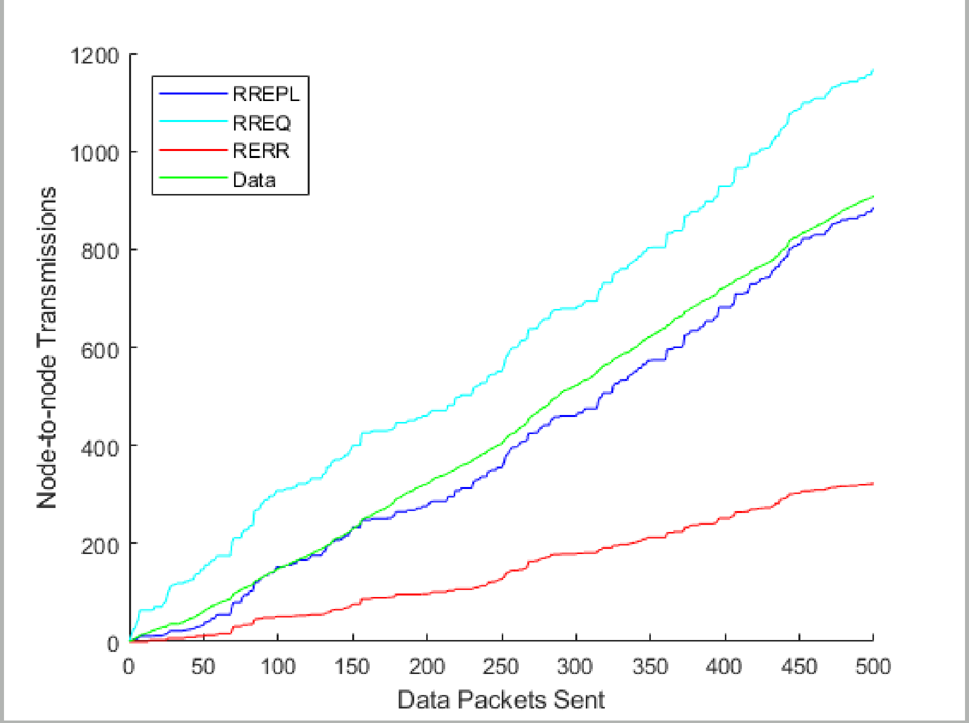
MISSOURI S&T

# Static Network



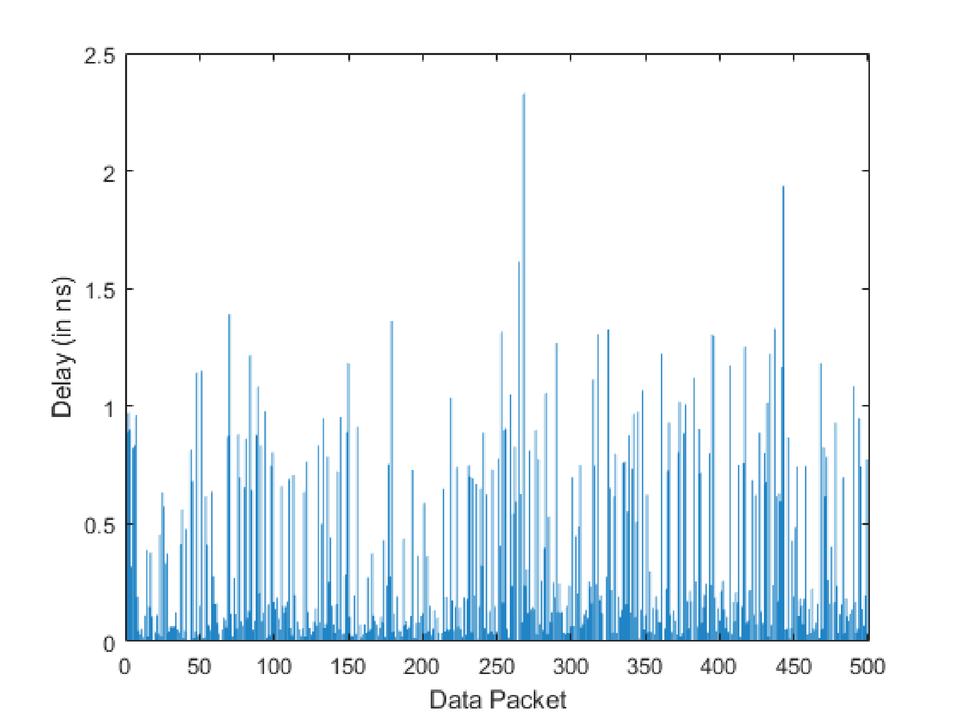
- Movement every 50 packets
- Propagation delays

# Highly Mobile Network



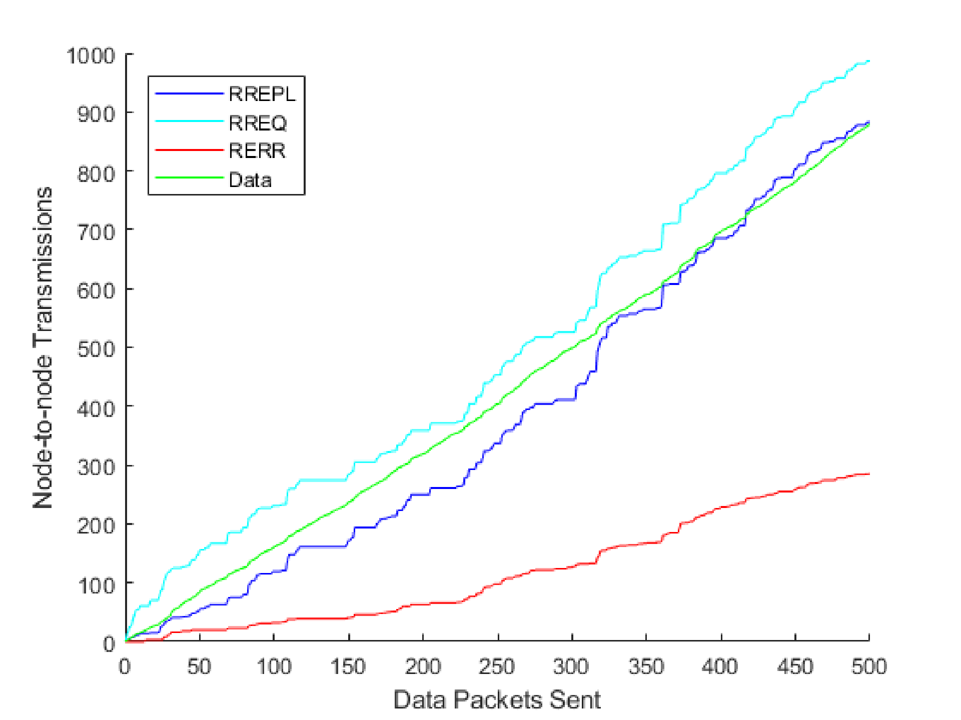
- 500 random packets sent
- Movement every 5 packets sent

# Static Network



- Movement every 50 packets sent
- Propagation delays

# One-Dimensional Network



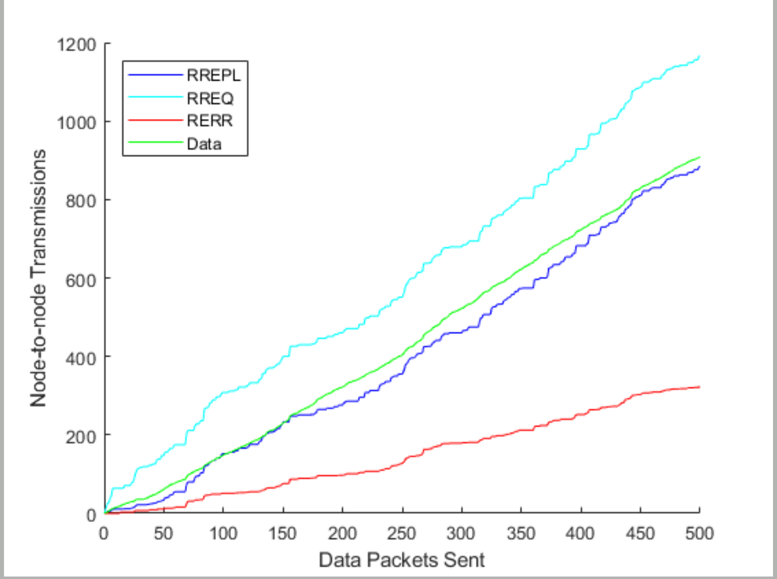
> 500 random packets sent

> Movement every 5 packets sent

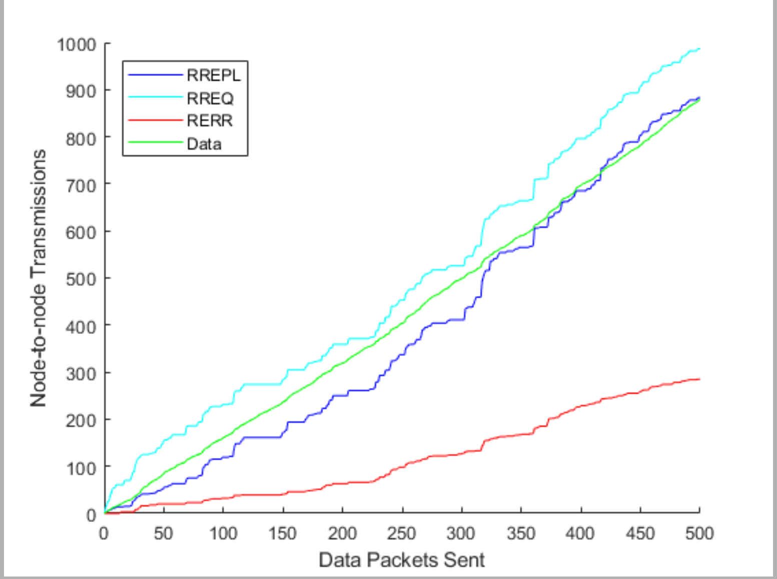
> Nodes only move in the X direction

# One-Dimensional Network

## Two-Dimensional Network



## One-Dimensional Network



# Agenda

- > Background
- > Examples
- > Traffic Statistics
- > **Future Work**
- > Conclusions
- > Demo

# Future Work

- Compare to other protocol like DSDV or DSR
- Implement other sources of delay such as queuing

# FONTS

- &gt; Background
- &gt; Examples
- &gt; Traffic Statistics
- &gt; Future Work
- &gt; **Conclusions**
- &gt; Demo

# Conclusions

- Highly congested networks are a burden for any protocol
- AODV handles link breakage with minimal overhead in simplistic cases
- Works best when there aren’t multiple routes to choose or cancel out

# Agenda

> Background

> Examples

> Traffic Statistics

> Future Work

> Conclusions

> **Demo**

Missouri S&T

# Demo

# Questions?