IPv6 Routing Protocol for Low Power and Lossy Networks Implementation in ns-3

John Patrick Agustin
Joshua Jacinto

UP EEEI Computer Networks Laboratory
Jhoanna Rhodette I. Pedrasa, PhD
Wilbert Jethro R. Limjoco

PRESENTATION OUTLINE

I. Project Background
II. Problem Statement and Objectives
III. Methodology
IV. Results
V. Comparison to COOJA Simulation
VI. Expected Deliverables
VII. Conclusion
VIII. Recommendations

Wireless Sensor Networks

Dense wireless networks of small, inexpensive, low-power sensors which accumulate environmental data to facilitate monitoring and controlling of physical environments from remote locations

M. Abdullah, A. Ehsan.(2014) Routing Protocols for Wireless Sensor Networks: Classifications and Challenges

“ By 2021, annual shipments of WSN chipsets will approach 2.5B up from 680M last year”

M. Hatler, D. Gurganious, J. Kreegar. (2016) Wireless Sensor Network Markets: A Market Dynamics report

Image from http://microcontrollerslab.com/wireless-sensor-networks-wsn-applications/

RPL

IPv6 Routing Protocol for Low Power and Lossy Networks

Developed by the Internet Engineering Task Force (IETF), RPL is made for wireless networks that are deployed in lossy environment where it is likely to have lost messages and in networks that consists of battery powered nodes.

Janicijevic, N. Lukic, M. Mezei, I. (2011) Routing Protocol for Low-Power and Lossy Wireless Sensor Networks

RPL KEY FEATURES

- Support of dynamic routing metrics
- Storing and Non-storing mode
- Loop Avoidance and Loop Detection
- Global and Local Repair
- Timer Management

Winter, T. Thubert, P. Request for Comments: 6550. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks

PROBLEM STATEMENT

1

Existing RPL implementations are hardware-constrained and/or limited

2

Limitations of the Existing RPL Implementations

	Cooja	TOSSIM	Omnet++ Castalia	NS-3
Limitations	Emulates real hardware devices	Designed specifically for TinyOS applications to be run on MICA Motes.	Limited Implementation. Uses default hardware models	There are missing features that are yet to be implemented

PROBLEM STATEMENT

1 Existing RPL implementations are hardware-constrained and/or limited

No existing implementation for multiple RPL instances 2

Main Objectives

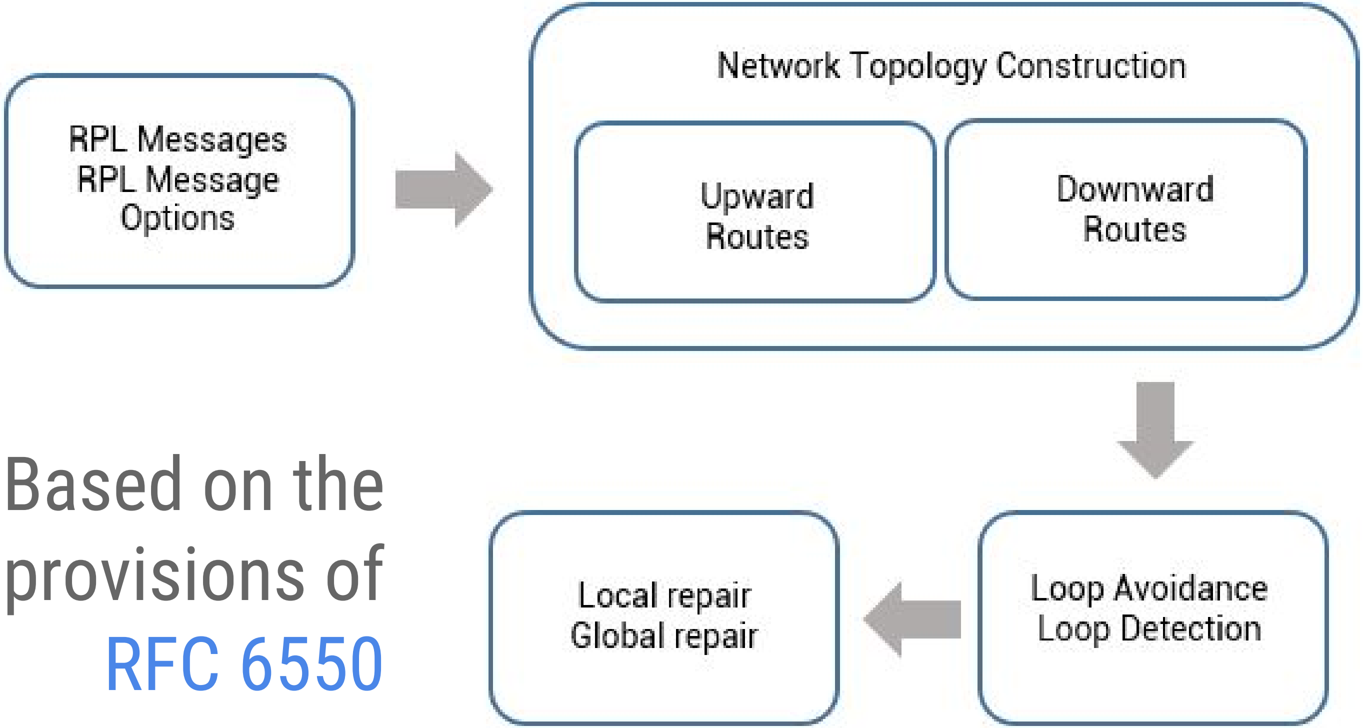
I. To fully implement RPL in ns-3 without hardware constraints or limitations

Main Objectives

I. To fully implement RPL in ns-3 without hardware constraints or limitations

II. To be able to support multiple RPL instances

Design and Implementation



Based on the provisions of RFC 6550

Design and Implementation

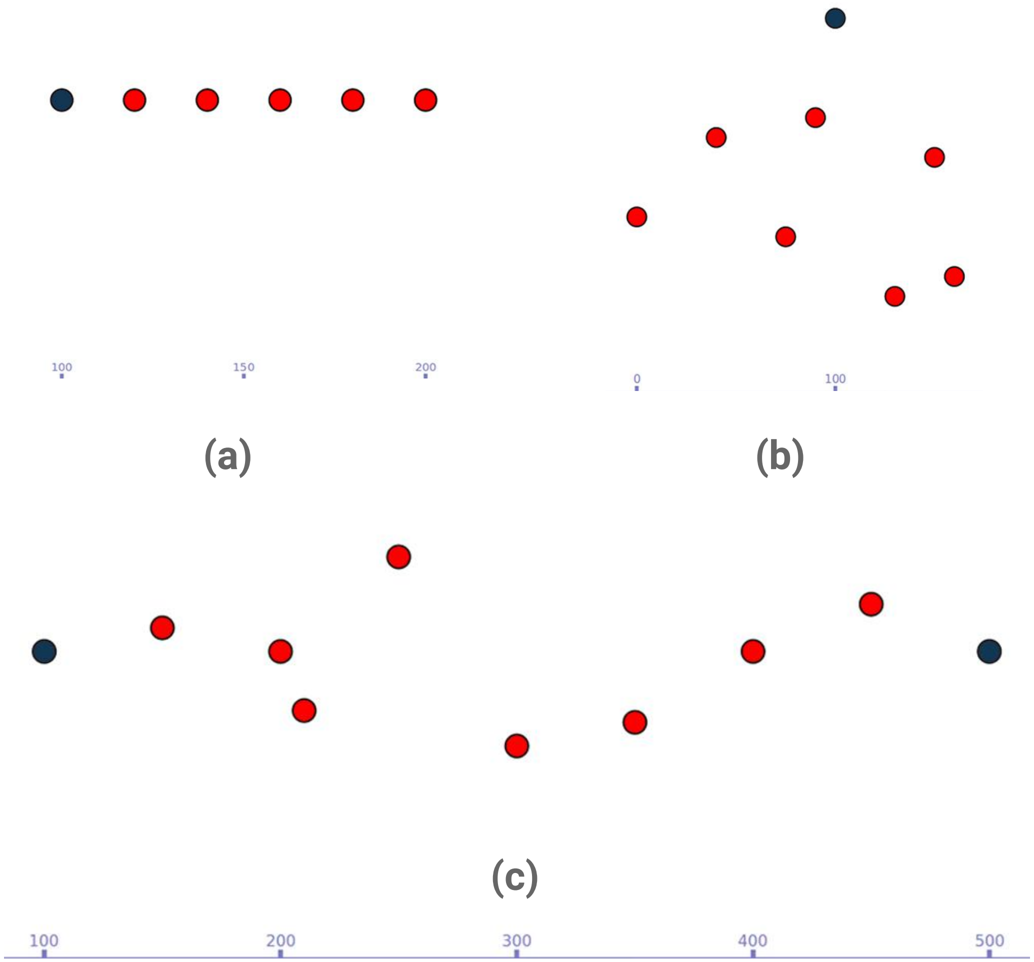
RFC 6550 does not cover the behavior of RPL multiple instances so the design will be up to us

Three possible ways:

- **Nodes in an existing instance can switch to another instance**
- Have at least two separate coexisting instances
- Two coexisting instances communicating with each other

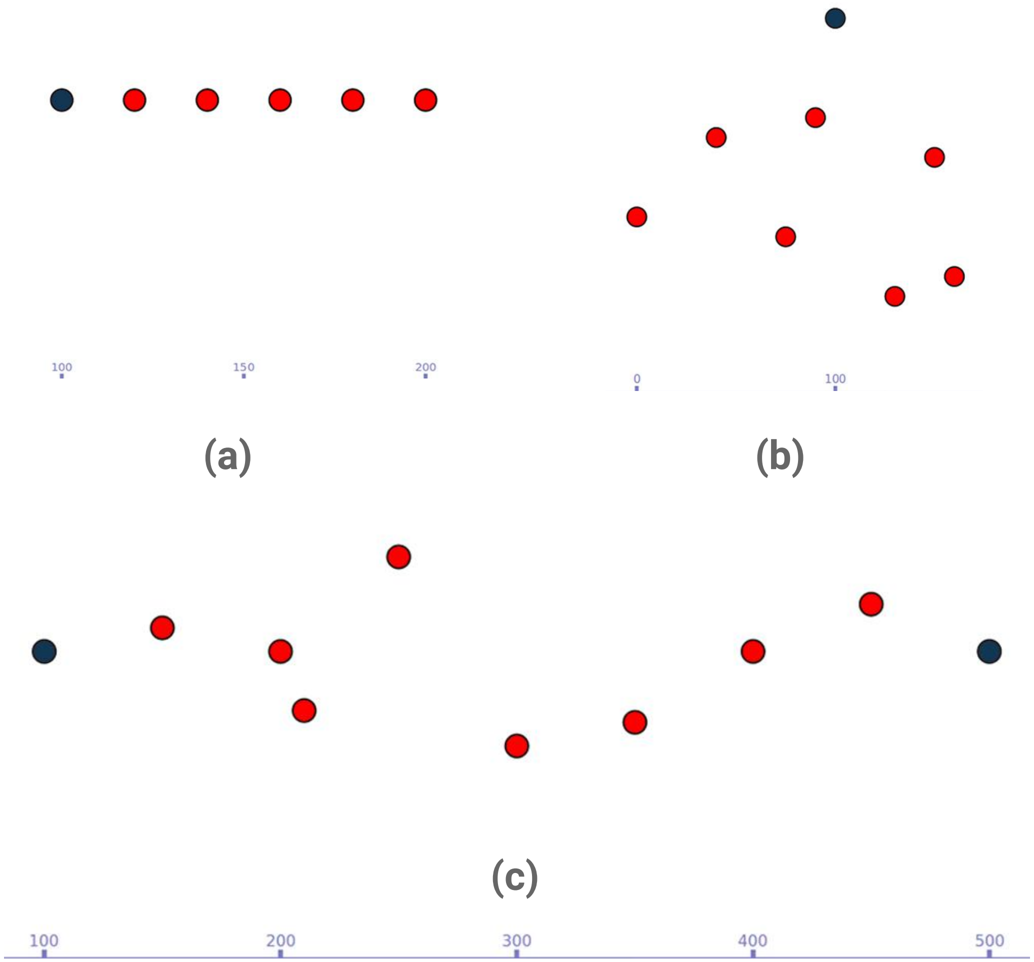
Testing Scenarios

(a) Linear topology
(b) Branching tree with single root
(c) Branching tree with multiple roots
(d) Test network with broken link

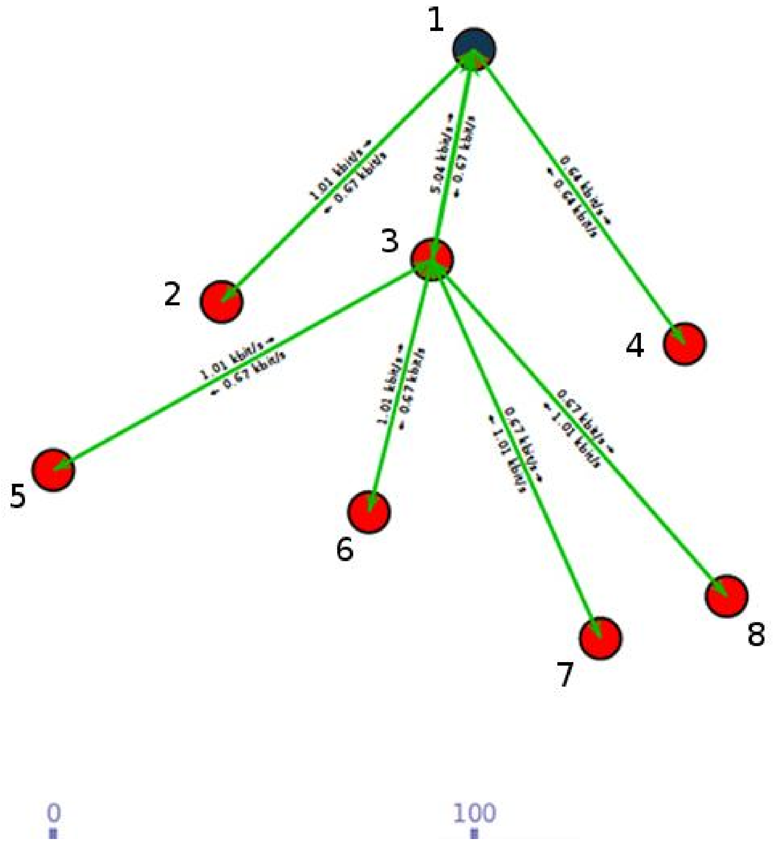


Testing Environment

WiFi module instead of 6LowPAN (IPv6 over Low power Wireless Personal Area Networks)



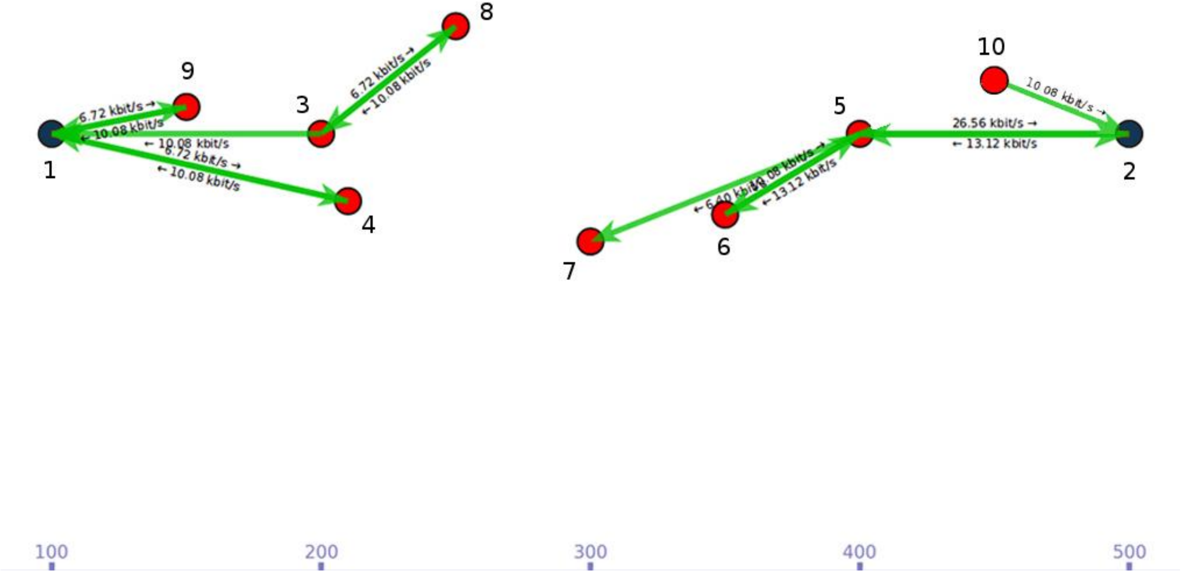
RESULTS - DEMO



Branching tree with single root

UPWARD ROUTES

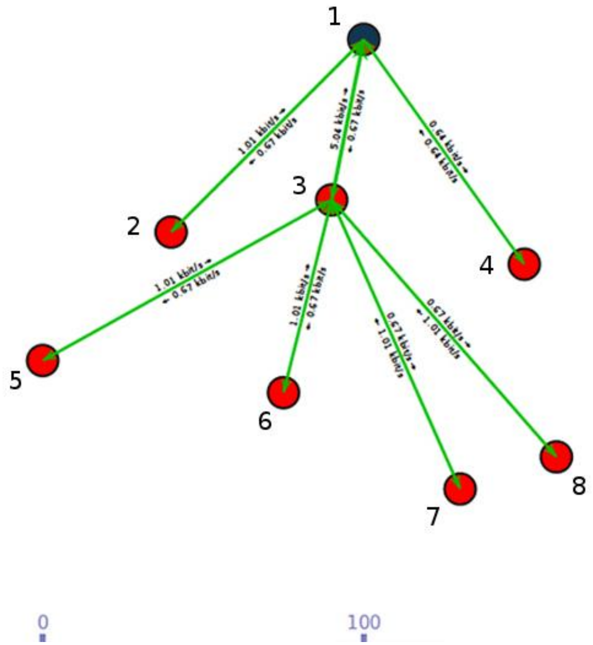
NODE	RANK (OF0)	PARENT
1 (Root)	1	-
2	769	1
3	769	1
4	769	1
5	1537	3
6	1537	3
7	1537	3
8	1537	3



Branching tree with multiple roots

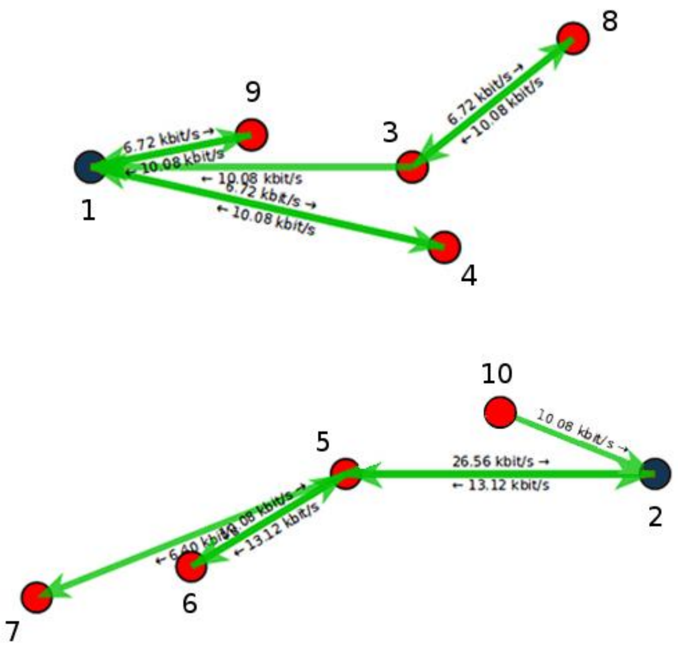
NODE	RANK (OF0)	PARENT
1 (Root)	1	-
3	769	1
4	769	1
8	1537	3
9	769	1
2 (Root)	1	-
5	769	2
6	1537	5
7	1537	5
10	769	2

DOWNWARD ROUTES



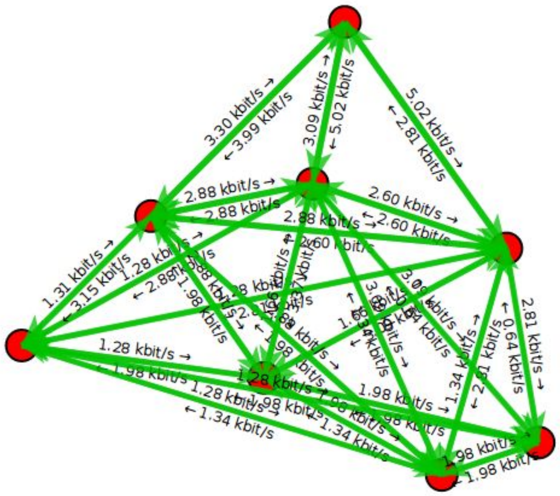
NODE	NON STORING	STORING	PARENT
1 (Root)	2, 3, 4, 5, 6, 7, 8	2, 3, 4, 5, 6, 7, 8	-
2	1	1	1
3	1	1, 5, 6, 7, 8	1
4	1	1	1
5	2, 3, 4	2, 3, 4	3
6	2, 3, 4	2, 3, 4	3
7	2, 3, 4	2, 3, 4	3
8	3, 4	3, 4	3

DOWNWARD ROUTES

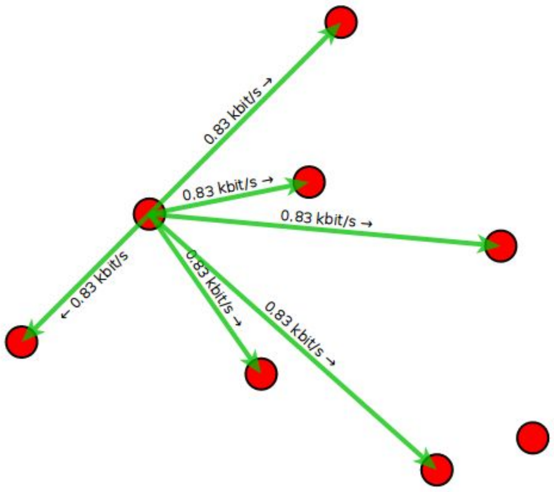


NODE	NON STORING	STORING	PARENT
1 (Root)	3, 4, 8, 9	3, 4, 8, 9	-
3	1	1, 8	1
4	1	1	1
8	3, 4, 9	3, 4, 9	3
9	1	1	1
2 (Root)	5, 6, 7, 10	5, 6, 7, 10	-
5	2	2, 6, 7	2
6	5, 10	5, 10	2
7	5	5	5
10	2	2	2

TRICKLE TIMERS

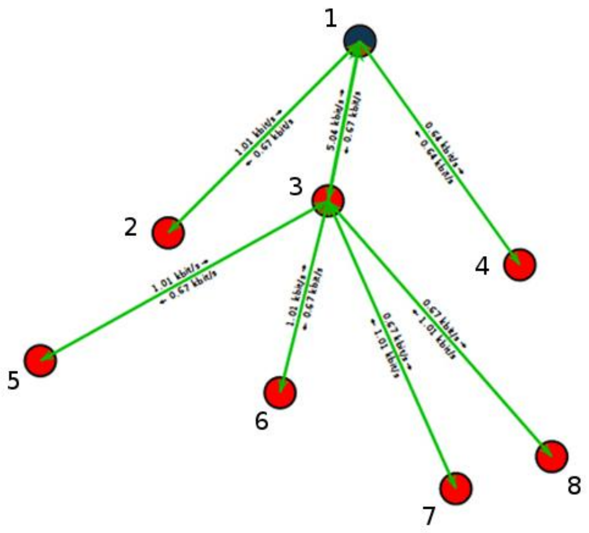


(a) At bootstrap

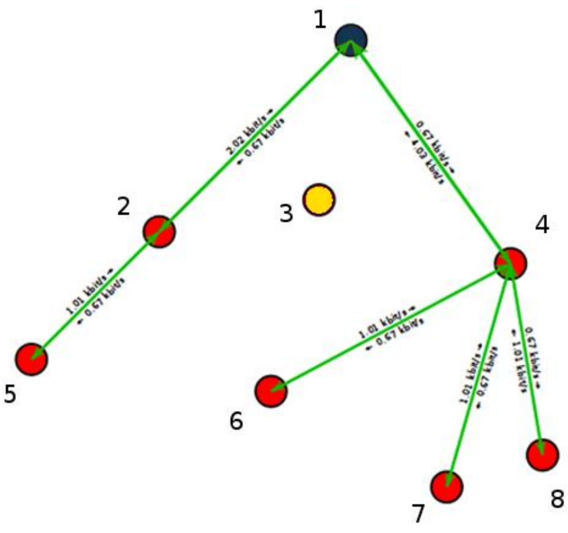


(b) At time = 100 s

LOCAL REPAIR



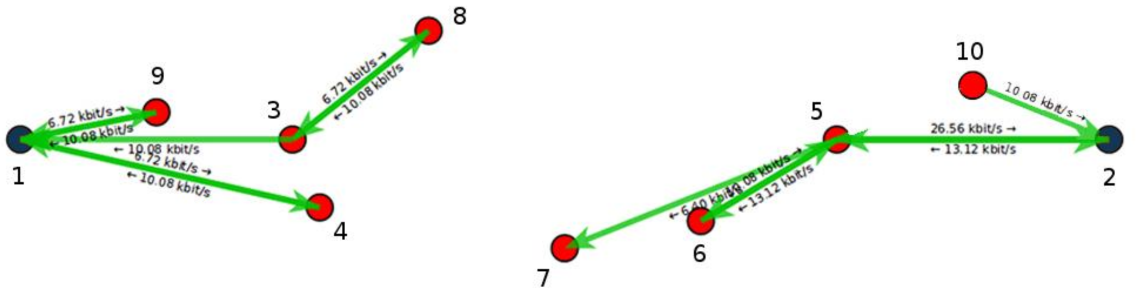
(a) Before repair



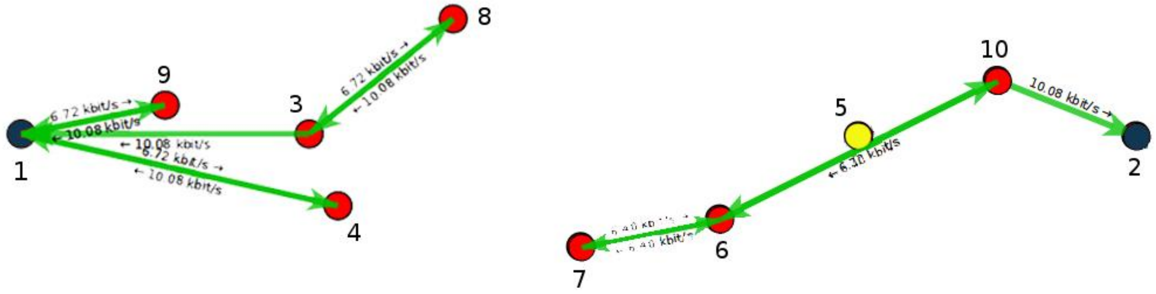
(b) After repair

0 100 0 100

LOCAL REPAIR



(a) Before repair



(b) After repair

GLOBAL REPAIR

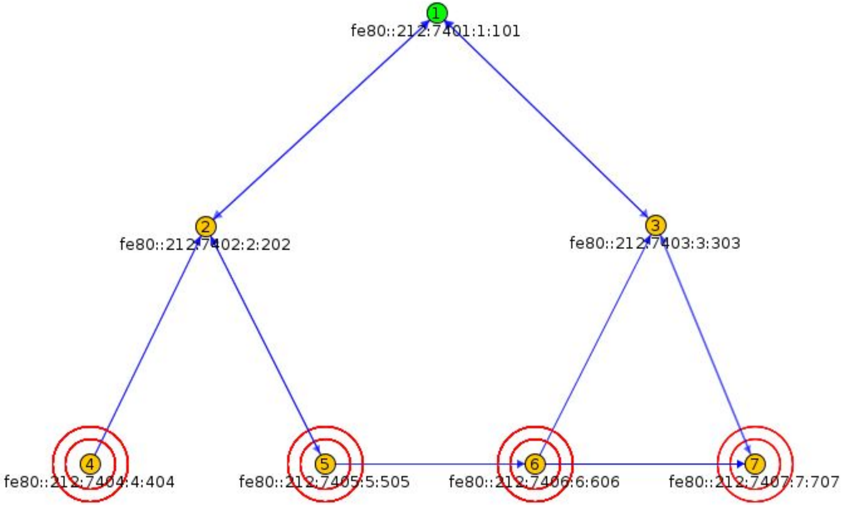
NODE	RANK BEFORE REPAIR	RANK AFTER REPAIR	PARENT
1 (Root)	1	0	-
2	769	1	1
3	769	1	1
4	769	1	1
5	1537	2	3
6	1537	2	3
7	1537	2	3
8	1537	2	3

GLOBAL REPAIR

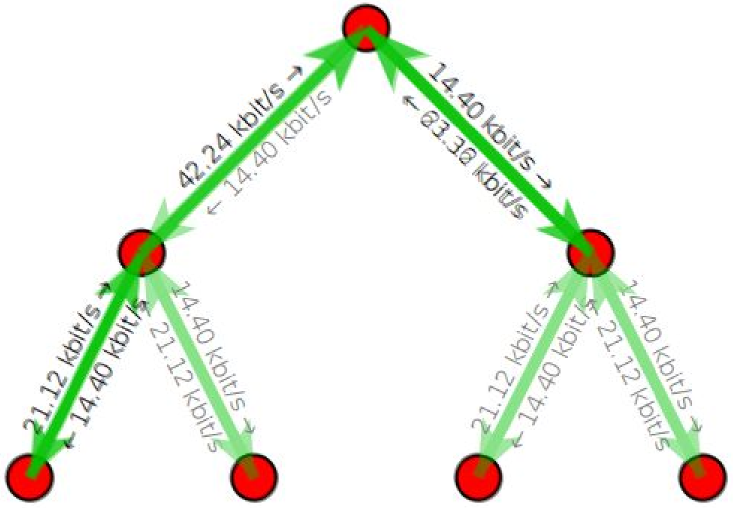
NODE	RANK BEFORE REPAIR	RANK AFTER REPAIR	PARENT
1 (Root)	1	0	-
3	769	1	1
4	769	1	1
8	1537	2	3
9	769	1	1
2 (Root)	1	0	-
5	769	1	2
6	1537	2	5
7	1537	2	5
10	769	1	2

COMPARISON TO COOJA SIMULATION

PARENT SELECTION AND RANK COMPUTATION



(a) Branching Tree in Cooja



(b) Branching Tree in ns-3

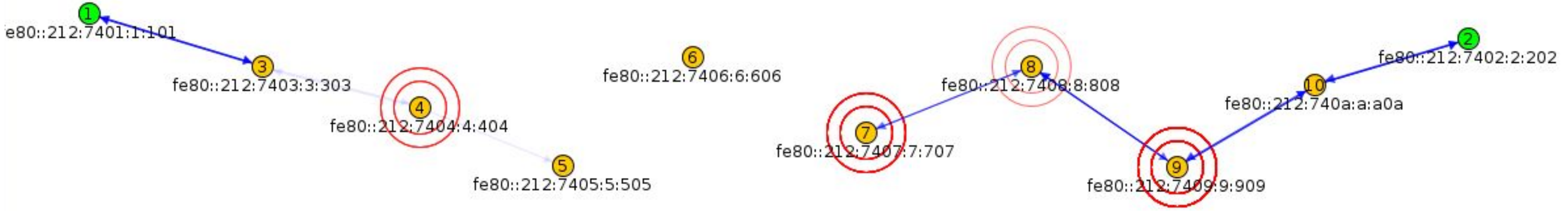
PARENT SELECTION AND RANK COMPUTATION

NODE	RANK (OF0)	PARENT
1 (Root)	1	-
2	512	1
3	512	1
4	768	2
5	768	2
6	768	3
7	768	3

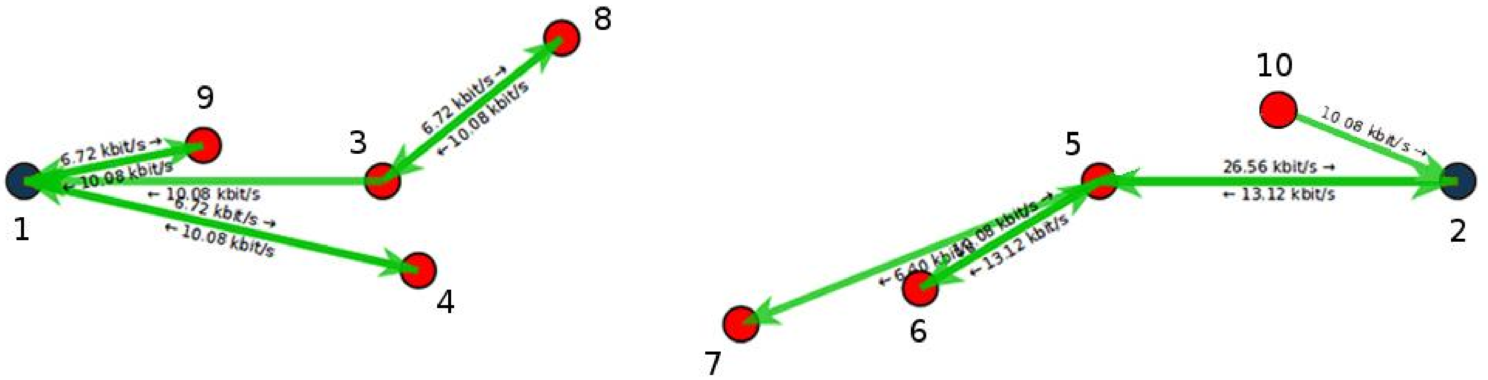
(a) Branching Tree in Cooja

NODE	RANK (OF0)	PARENT
1 (Root)	1	-
2	769	1
3	769	1
4	1537	2
5	1537	2
6	1537	3
7	1537	3

(b) Branching Tree in ns-3

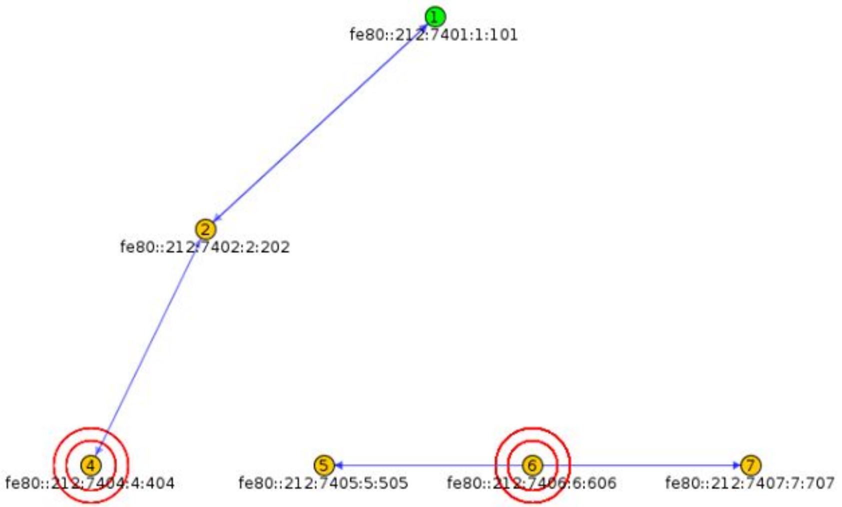


(a) Multiple roots in Cooja

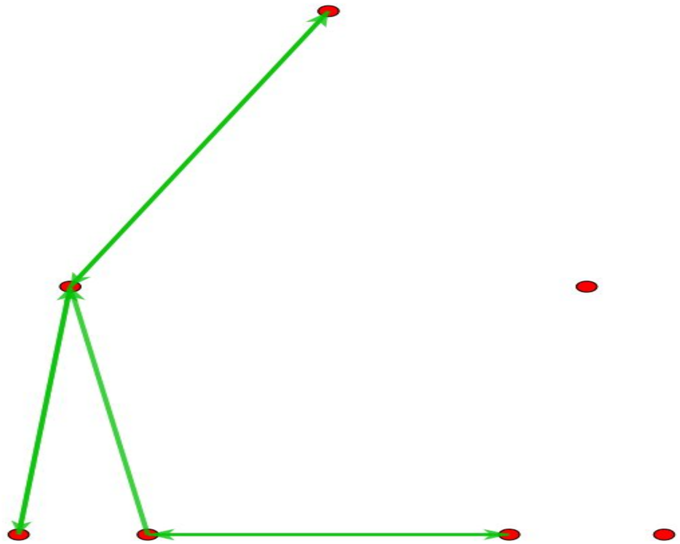


(b) Multiple roots in ns-3

LOCAL REPAIR



(a) Local repair in Cooja



(b) Local repair in ns-3

Difference between ns-3 and Cooja Simulation

NS - 3	COOJA
- Need to manually position nodes using allocators - Functionality not limited by the node’s capacity and hardware specifications	- Simulator is easier to use because of its GUI - Code and functionality that can be run is limited by mote memory

Halfway-point Deliverables

Run a single instance of the RPL module:

- [ ] Should be able to send RPL control messages
- [ ] OF0 implementation - Rank computation and Parent Selection
- [ ] Should be able to detect and avoid loops

Final Deliverables

Single RPL Instance & Support of multiple instances

- ☐ Downward routes - Storing and Non storing mode
- ☐ 2nd OF implementation - MRHOF w/ Hop Count Metric
- ☐ Implementation of multiple instances

CONCLUSION

RECOMMENDATIONS

Development of multiple RPL instances can be further explored

Addition of other Objective Functions

Utilization of control message flags and security features

THANK YOU!

REFERENCES

M. Abdullah, A. Ehsan. (2014) Routing Protocols for Wireless Sensor Networks: Classifications and Challenges. Journal of Electronics and Communication Engineering Research, Vol 2 Issue 2, pp: 05-15

Al-Karaki, J., & Kamal, A. (2004). Routing techniques in wireless sensor networks: A survey. IEEE Wireless Communications, 11(6), 6-28.

Janicijevic, N., Lukic, M., & Mezei, I. (2011). Routing protocol for low-power and Lossy wireless sensor networks. 2011 19th Telecommunications Forum (TELFOR) Proceedings of Papers

Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., & Alexander, R. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550 (Proposed Standard)

Hui, J. (2012). The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams. RFC 6553 (Proposed Standard)

Bartolozzi, L., Pecorella, T., & Fantacci, R. (2012). Ns-3 RPL module: IPv6 Routing Protocol for Low power and Lossy Networks. Proceedings of the Fifth International Conference on Simulation Tools and Techniques.

Saad, L.. Chauvenet, C. Tourancheau, B. (2011). Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: Two Case Studies. SENSORCOMM 2011 : The Fifth International Conference on Sensor