Mobile Matrix: A Multihop Address Allocation and Any-To-Any Routing in Mobile 6LoWPAN

DCC DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO UFMG

Bruno P. Santos, Olga Goussevskaia, Luiz F. M. Vieira, Marcos A. M. Vieira, and Antonio A.F. Loureiro

{bruno.ps,olga,lfvieira,mmvieira,loureiro}@dcc.ufmg.br

Introduction and Motivation

Mobile Matrix is a routing protocol for 6LoWPAN that uses hierarchical address allocation to perform any-to-any routing and mobility management without changing a node's IPv6 address.

Mobility is a major factor present in everyday life. It makes life easier and turns applications more flexible. The usage of many devices for IoT can benefit from it, thus IoT becomes even more ubiquitous.

- μMatrix has low routing memory footprint and adjustable control message overhead.
- It manages routing and mobility without ever changing nodes IPv6.

Design Overview

- μMatrix control plane encompasses the *Mobility Engine* capable to handle nodes mobility.

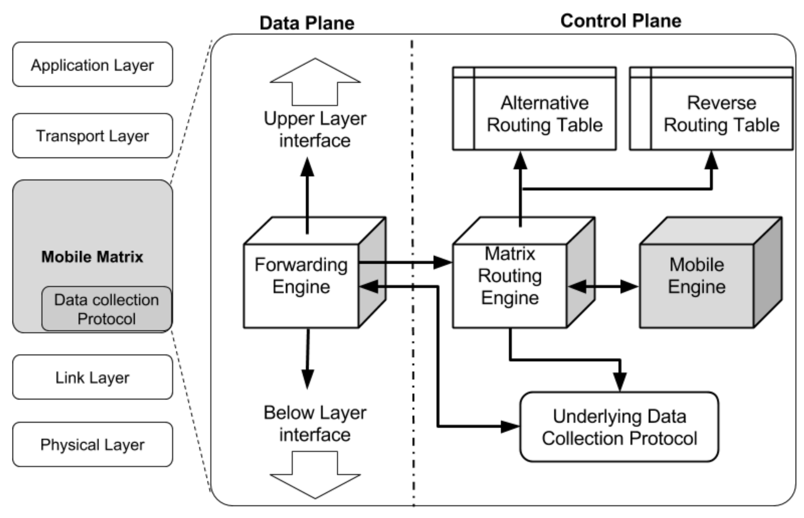


Figure 1: μMatrix architecture.

- μMatrix provide inexpensive routing memory requirements.
- Node mobility management requires only few routing table updates.

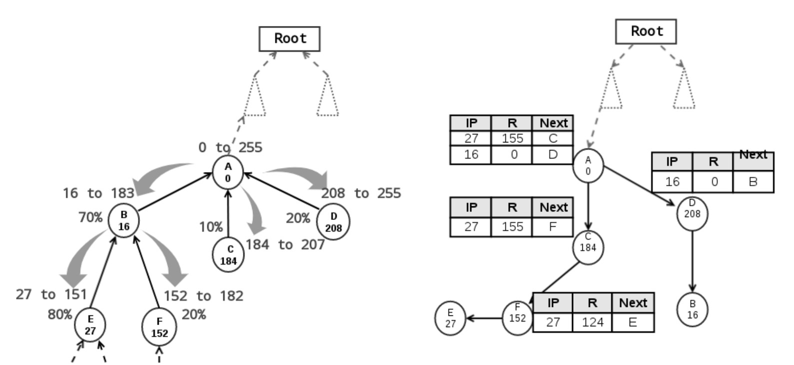


Figure 2: μMatrix hierarchical address assignment and mobility management.

Handling Mobility

- μMatrix's mobility engine uses a finite state machine to keep tracking nodes state.

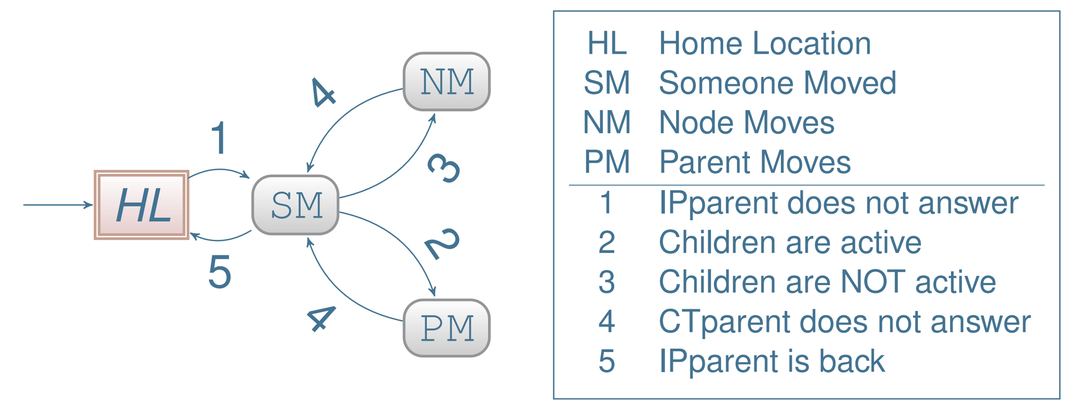


Figure 3: Mobile Engine state machine.

- μMatrix is capable to detect mobility by active or passive fashion. Active approach the node uses additional hardware (ex: GPS) to infer mobility, passive aproach the node infer mobility by itself.

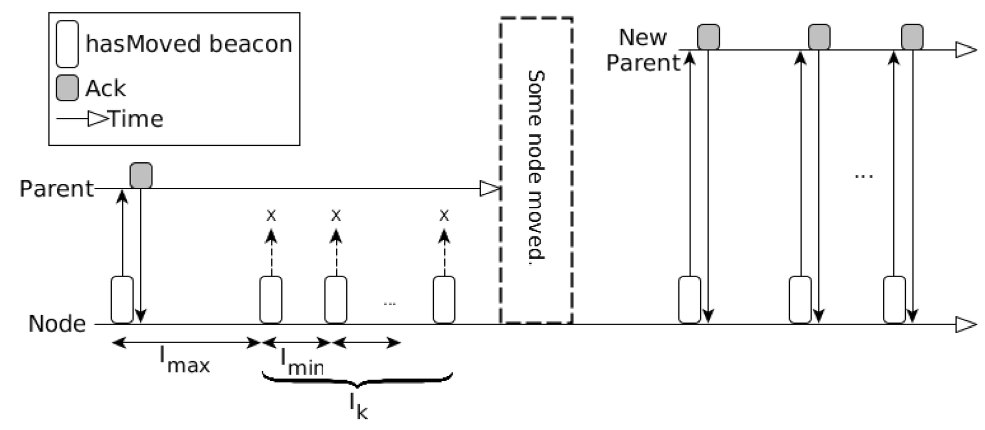
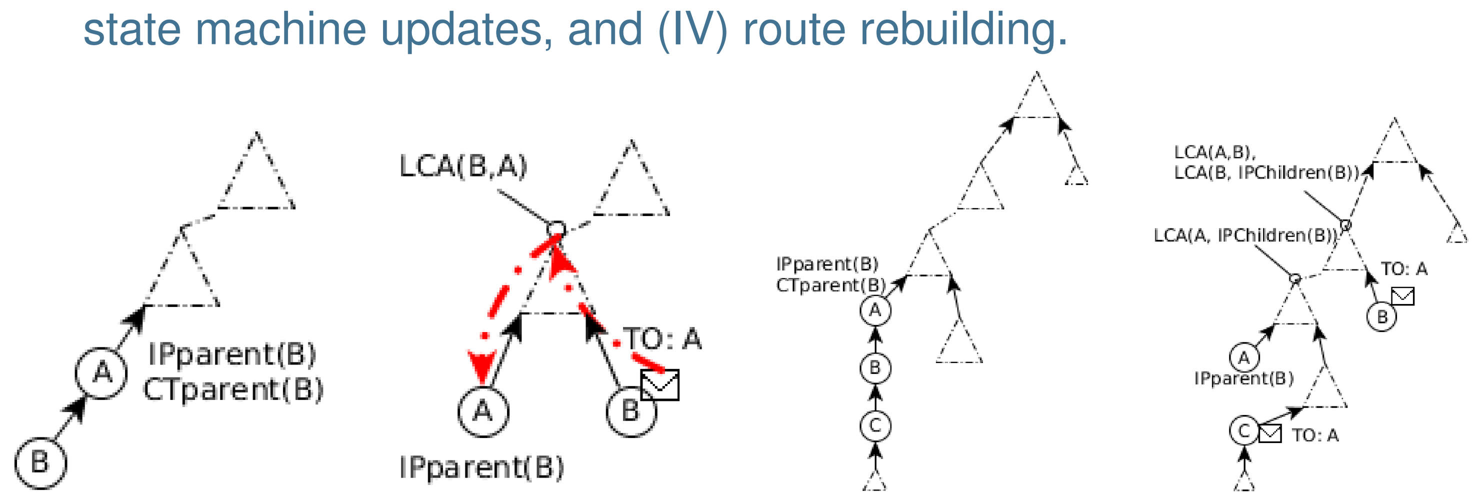


Figure 4: Reverse Trickle a passive mobility detection mechanism.

- μMatrix step-by-step: (I) host configuration, (II) mobility detection, (III) state machine updates, and (IV) route rebuilding.



Static situation (nodes at HL).

B moves, then it transits to NM, and rebuilds the route.

B a non leaf node before move.

B moves, then C transits to PM and rebuilds the route.

Figure 5: μMatrix operation after mobility events.

Complexity Analysis

- μMatrix memory footprint to each node $\mu \in$ Ctree is

$$\mathcal{M}(\mu) = O(depth(Ctree)).$$

- The control message complexity of μMatrix is

$$\mathcal{M}sg(\mu Matrix(Ctree)) = O\left(\frac{m \times I_k}{I_{min}} + \frac{n}{I_{max}}\right) + O\left(\frac{m \times \Delta}{\delta} depth(Ctree)\right).$$

Experiments

- Cyclical Random Waypoint Mobility Model (CRWP) is a mobility model based on the Random Waypoint. CRWP models scenarios where entities move to different destinations, and eventually, they return to their initial positions.
- Simulation: Cooja/ContikiOS; Mobility: BoonMotion

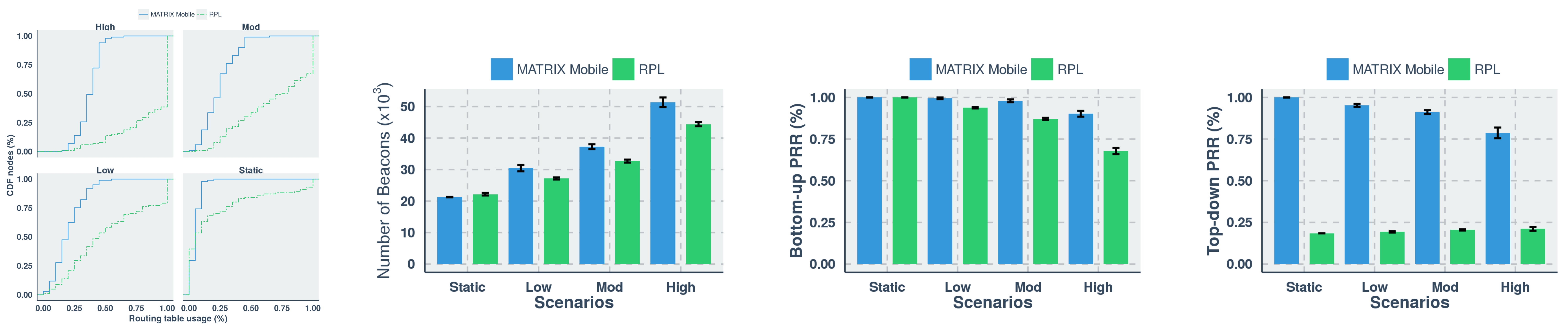
Table 1: Simulation parameters

Simulation parameter	Values		
# Nodes	1 center root, 100 nodes in grid		
Application data packets	20 pkt/node, Rate = 1 pkt/min		
Radio environment	50 m UDGM constant loss		
Area of deployment	400 m $\times$400 m		
Reverse Trickle	$I_{max} = 60$ s, $I_{min} = 1$ s, $I_k = 3$		
RPL Trickle	$I_{max} = 60$ s		
keepRoute beaconing period	$\delta = 60$ s		
Mtable	$TTL_{max} = 90$ s, Size = 20 entries		
RPL downwards table	Size = 20 entries		
# mobility traces	10 traces/scenario		
Number of experiments	10 runs/trace		
Node Speed	constant 4 m/s		
T_{pause}	constant 300 s		
# node stops	Uniform Dist. in [1, 3] stops		
	Low	**Moderate**	**High**
PerMobNode	**5%**	**10%**	**15%**

Table 2: Mobility metrics

Mobility Metrics	Low Mob. sce.	Mod. Mob. sce.	High Mob. sce.
Avg. Link Breaks	1621	3057	4838
Avg. Link duration	761.90	457.4	345
Avg. Degree	4.12	4.36	4.44
Avg. Time to link break	227.6	216.1	204.5

Results



(a) CDF of routing table usage.

(b) Number of control packets

(c) Bottom-up routing success rate.

(d) Top-down routing success rate.

Figure 6: Simulation experiments

Related Work

Table 3: Routing protocol properties

Feature	μMatrix	RPL	Co-RPL	MMRPL	ME-RPL	mRPL	DMR	Hydro	XCTP
Bottom-p	✓	✓	✓	✓	✓	✓	✓	✓	✓
Top-down	✓	✓	✓	✓	✓	✓		✓	✓
Any-to-any	✓	✓	✓	✓	✓	✓		✓	
Address Allocation	✓								
IPv6 support	✓	✓	✓	✓	✓	✓	✓		
Memory efficiency	✓								
Fault Tollerance	✓								✓
Local Repair	✓								
Topological changes	Reverse Trickle	Trickle	Periodic fixed	Reverse Trickle-like	Trickle	Trickle	Trickle	Periodic fixed	Trickle
Constraints	Nodes should return to home location			Need static nodes	Need static nodes	Need static nodes	Need static nodes	Need static nodes	

Conclusions

We presented μMatrix: a memory efficient routing protocol for 6LoWPAN that performs any-to-any routing, hierarchical address allocation, and mobility management.

We also introduced the CRWP, a mobility model suited for scenarios with mobile nodes that have cyclical movement patterns.

As future work, we plan to run experiments with physical devices and extend experimental evaluation to more mobile models, such as faulty communications scenarios.

Selected References

[1] P. Levis, N. Patel, David Culler, and Scott Shenker. Trickle: A Self-regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. In *USENIX NSDI*, pages 2–2, 2004.

[2] Bruna S. Peres, Otavio A. de O. Souza, Bruno P. Santos, Edson R. Jr, Olga Goussevskaia, Marcos A. M. Vieira, Luiz F. M. Vieira, and Antonio A. F. Loureiro. Matrix: Multihop Address Allocation and Dynamic Any-to-Any Routing for 6LoWPAN. In *ACM MSWiM*, pages 302–309, 2016.

[3] Charles Perkins, David Johnson, and Jari Arkko. Mobility support in IPv6, 2011.

[4] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, 2012.

CAPES CNPq FAPEMIG