Measurement Results from Wireless Battle Mesh Version 7

Type: Measurement Analysis (work in progress)
Creation date: May 19, 2014

Event:
Sublab. Leipzig, Germany
12th to 18th of May 2014
http://battlemesh.org/BattleMeshV7

Contents

1 Introduction

WBM...

2 Data and System Repositories

http://wibed.confine-project.eu

https://github.com/battlemesh/wibed (buildroot)
https://github.com/battlemesh/wibed-battlemesh-experiment (package)
http://wiki.confine-project.eu/wibed:start
https://github.com/axn/wbm2pdf (this stuff, branch wbmv7 in future)

3 Testbed Descripiton

3.1 Nodes and Locations

NodeID	Location			
164a7a	deathroom			
3b3a90	workshopRoom			
3b3d70	????			
3e9db0	deathroom??	e16:9db0->1ab0!		e19:Mob:9db0->4174
51aac8	halleAnfang			e19:Mob:aac8->4174
8a417e	deathroom	e16:417e->4174!	e17:417e->1ab0!	
c24174	DST MOBILE HalleEnde		e17:4174->1936!	
c2427a	deathroom??			e19:Mob:427a->4174
ce3360	EloiTable			
e4b63a	mustiTable			
e60a62	halleMitte			
e60aac	deathroom			
e60ad6	deathroom			
e61936	SRC axelsTable	e16:1936->4174!	e17:1936->4174!	e19:Mob:1936->4174
f41ab0	kloschiOffice??	e16:1ab0->4174!	e17:1ab0->417e!	e18:Mob:1ab0->4174

3.2 Topology

4 Ping Measurements (hops, rtt, loss)

4.1 Stationary Scenarios

4.2 Ping Results Table

The folloing verbatim table lists statistics per experiment (EXP) and group (GRP) as calculated by the lua-based evaluation script based on the raw ping-measurements data and outputted to the file ping.stat. Event based results are given for each received icmp response in ping.data.

4.3 Stationary Nodes Measurements

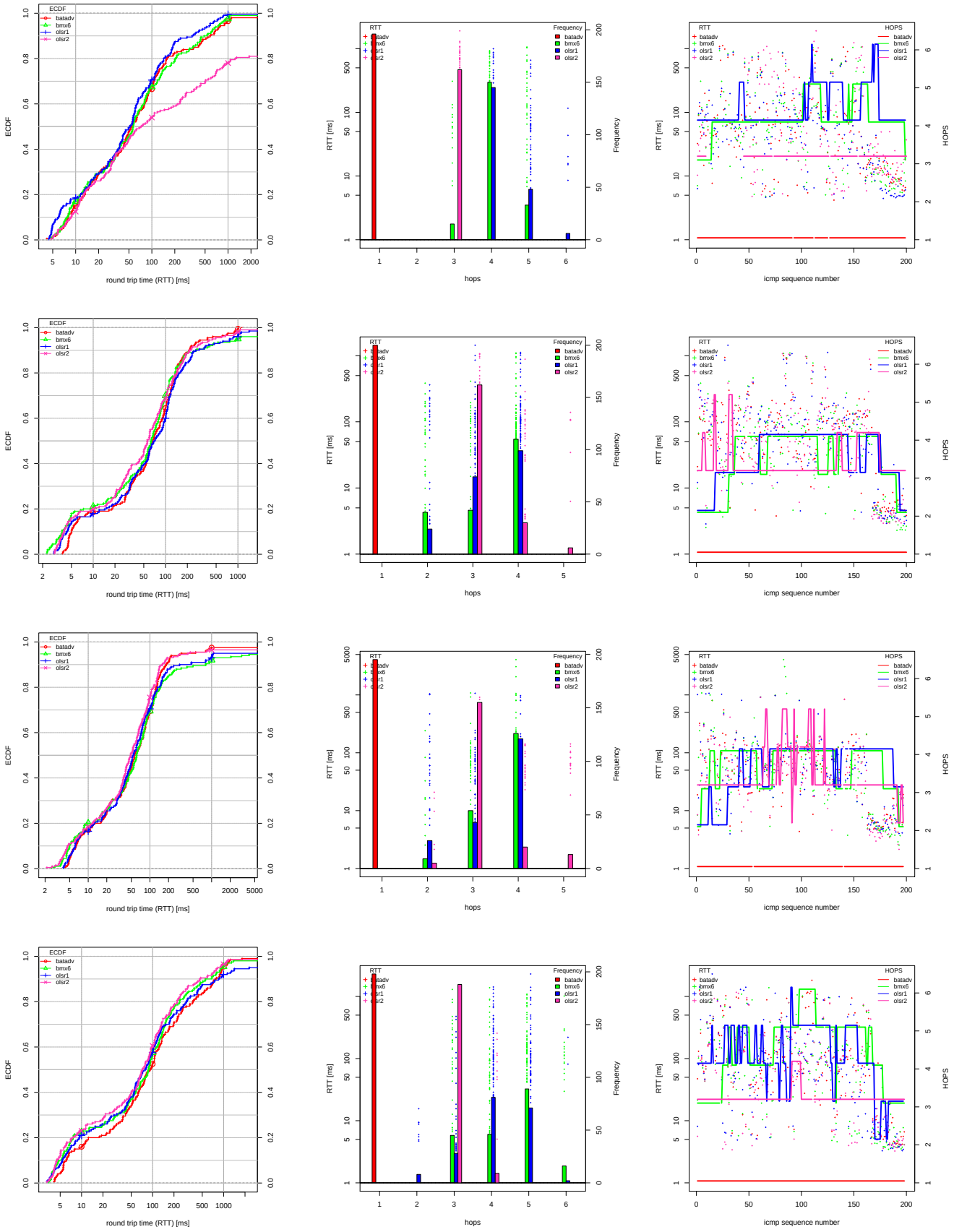

Table 1: End-to-end ping6 performance between two stationary nodes: 9db0-1ab0, 417e-4174, 1936-4174, 1ab0-4174

4.4 Mobile Node Measurements

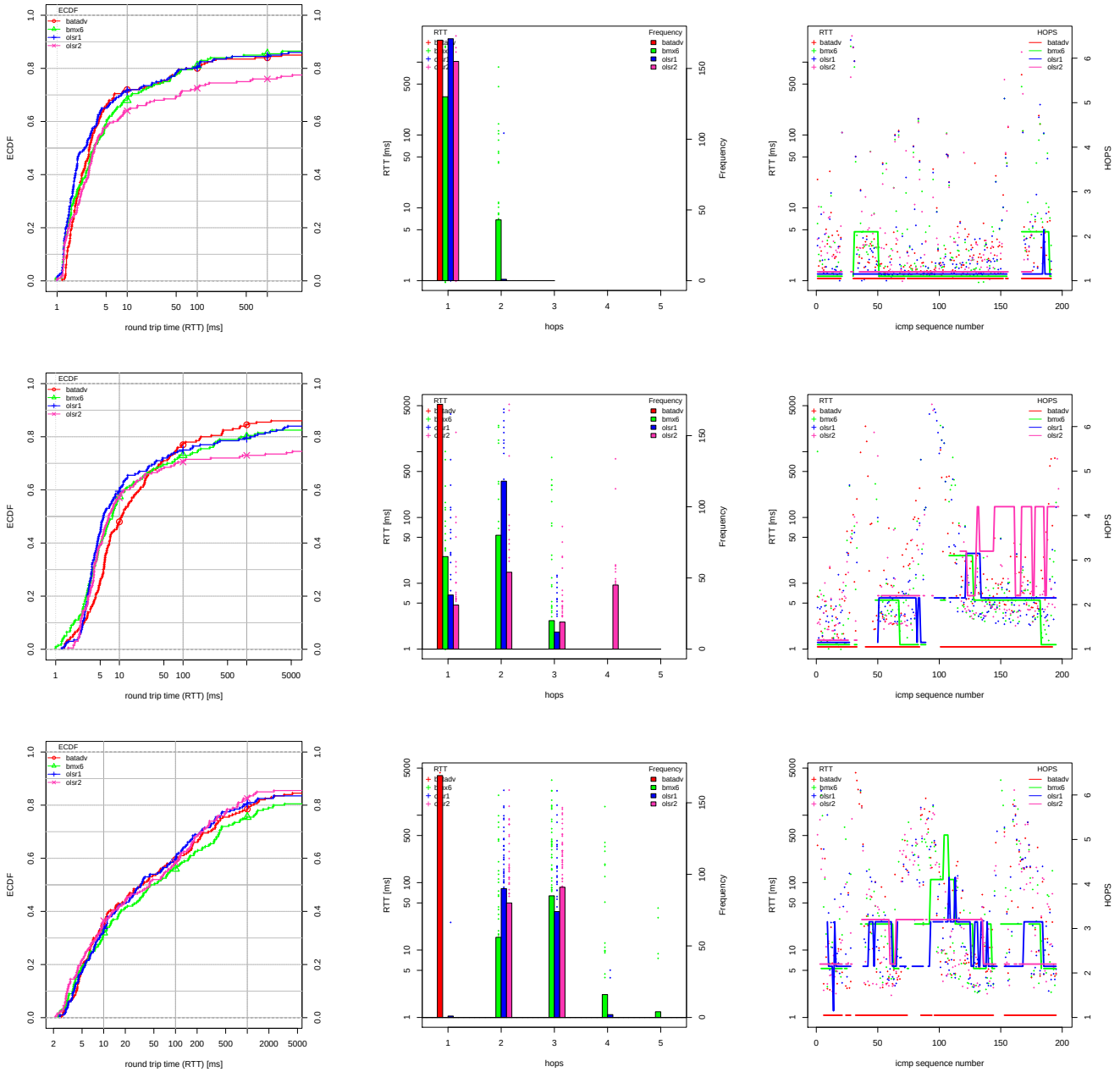

Table 2: End-to-end ping6 performance to mobile node 4174 from aac8, 1936, 1ab0

4.5 Mobile Scenarios

5 TCP Throughput Measurements

6 Recommendations for next battlemesh

- Traceroute and mrt often show high packet for intermediate nodes. This is due to a kind of denial-of-service mechanism enabled by default in Linux kernel. WIth this mechanism the kernel simply discards frequent icmp responses (eg due to exceeded TTL values). This behavior can be disabled by lowering the default net.ipv6.icmp.ratelimit=1000 setting, eg via: sysctl -w net.ipv6.icmp.ratelimit=10
- Most important: Measure protocol traffic overhead in parallel

7 Appendix