Demo: Using nepi-ng for Mesh Networks Experiments

Yonathan Bleyfuesz, Thierry Parmentelat, Thierry Turletti, Farzaneh Pakzad, Mohamed Mahfoudi, Walid Dabbous

Demo: Using nepi-ng for Mesh Networks Experiments

Yonathan Bleyfuesz
Université Côte d'Azur, Inria
Yonathan.Bleyfuesz@inria.fr

Thierry Parmentelat
Université Côte d'Azur, Inria
Thierry.Parmentelat@inria.fr

Thierry Turletti
Université Côte d'Azur, Inria
Thierry.Turletti@inria.fr

Farzaneh Pakzad
Aptira, Australia
farzaneh@aptira.com

Mohamed Naoufal Mahfoudi
Université Côte d'Azur, Inria
Mohamed-Naoufal.Mahfoudi@inria.fr

Walid Dabbous
Université Côte d'Azur, Inria
Walid Dabbous@inria.fr

ABSTRACT

We describe a demonstration run on R2lab, an open wireless testbed located in an anechoic chamber at Inria Sophia Antipolis. The demonstration consists in easily deploying a Wi-Fi mesh network. The nodes provisioning, configuration and the scenario orchestration and control are automatically done using the `nepi-ng` experiment orchestration tool. A performance comparison of two wireless mesh routing protocols in presence of controlled interference is shown.

1. INTRODUCTION

A wireless mesh network allows dynamic and non-hierarchical communication between its nodes, without any infrastructure such as access points. In order to evaluate the performance of wireless mesh routing protocols such as OLSR [1] and BATMAN [2], some special considerations should be taken when deploying an experiment:

- **Number of nodes:** Need to handle a significant number of nodes.
- **Multi-hop context:** Need to control channel conditions in order to allow multiple-hop situations and make possible performance comparison of the two mesh routing protocols.
- **Radio conditions**: Need to ensure the same physical layer conditions for each iteration of the experiment.

Several studies have compared OLSR and BATMAN performance, see [3]. In this paper, we demonstrate `nepi-ng` as a tool to facilitate the deployment of wireless network experiments and performance comparison of routing protocols.

2. MESH NETWORK DEPLOYMENT USING NEPI-NG AND R2LAB

In this demonstration, we use `nepi-ng` and the FIT R2lab wireless testbed to meet the challenges described above. R2lab is a wireless testbed located in an anechoic chamber at Inria Sophia Antipolis, providing a controlled environment that ensures the same RF conditions for the different wireless experiments. We are deploying our experiments using `nepi-ng`, a tool for orchestrating network experiments in an automated and efficient way. See [4] for more details on `nepi-ng` and R2lab. Indeed, the `nepi-ng` parallel component allows to configure all of our nodes simultaneously. Thanks to `nepi-ng`'s nested schedulers [4], we are able to reuse portions of code from previous scripts. We have reused modules for configuring the wireless interfaces and for measuring the RSSI, which allowed us to quickly pinpoint the different parameters to characterize precisely our environment and produce a multi-hop scenario. Furthermore, we run the experiments directly from our local machine without having to manually access the gateway or the wireless nodes. This approach even-though practical and straightforward is more prone to connection losses to the target testbed. However, the results produced by the experiment remain available to the experimenter, as the scripts used by `nepi-ng` are pushed to the target nodes to run locally.

3. EXPERIMENT SCENARIO

The scenario consists of a wireless mesh network as depicted in Figure 1, using OLSR and BATMAN. We compare performance of these two routing protocols using 3 different metrics: end-to-end latency, packet delivery ratio, and number of hops. We use olsrd v0.6.6.2-1ubuntu1 and batmand v0.3.2-17 on Ubuntu 16.04 nodes with a 4.4.0-21-generic kernel. We chose the Atheros chipset configured on the 802.11g mode and operating on the 2.457GHz frequency, which provides a mesh interface. To maximize the number of hops, we use the minimal Tx power of 5 dBm and the maximal data rate of 54 Mbps that has the lowest range with a single antenna element. To create controlled interference, we generate additive Gaussian noise with a gain of -12 dB using the uhd_siggen tool from gnuradio 3.7.11.1 on an

USRP N210. We set the mesh_fwd option to 0 to prevent path forwarding.

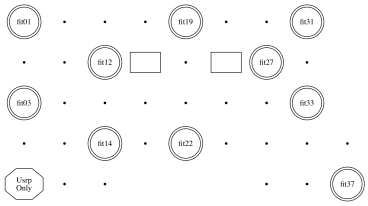

Figure 1: Physical location of used nodes used in R2lab

3.1 Experiment flow

The experiment script consists in the six following steps :

- Check if we have reserved the R2lab testbed, turn on and load images on the nodes on demand.
- Configure wireless cards and routing protocol on nodes.
- Let the wireless network settle for 60 *s*.
- Snapshot the routing tables of all nodes.
- Start the pings between the source and destinations.
- Stop the routing processes; retrieve the generated data.

Moreover, the experimenter has the following options:

- Enable a tcpdump capture on every node and create a mini textual heatmap of the R2lab testbed.
- Enable an co-channel interference with a white noise generator in the anechoic chamber.
- Launch the service to snapshot the routing tables on each node every 500 ms.

Table 1 represents the results of 20 runs. We can infer that BATMAN is more reliable on 1-hop routes. However when more than 2 hops are required to reach the destination, both protocols choose highly variable routes. We note that OLSR uses more hops in general and achieves a higher PDR for nodes 19, 27, 31 and 33. Regarding latency, performance are similar except for node 19 with BATMAN, which suffers many retransmissions due to bad channel conditions.

4. CONCLUSION

This demonstration shows how wireless mesh networks can be easily and rapidly deployed using nepi-ng on R2lab. More complete performance results on Batman vs OLSR are also available on our github [5] in the form of a Jupyter notebook.

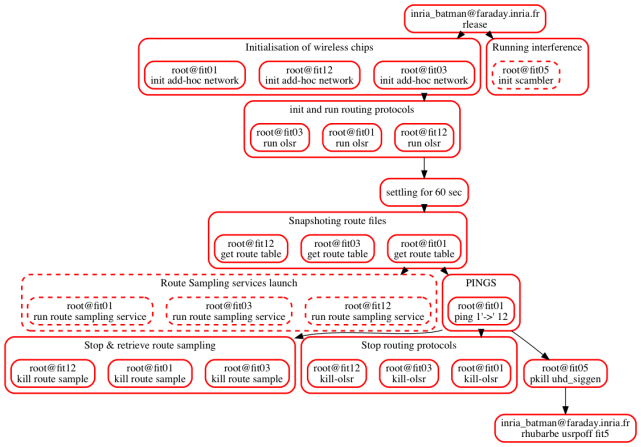

Figure 2: Example of an OLSR run with some modules.

Table 1: Hop count, PDR and RTT for 20 experiments

Destination	3	12	14	19	22	27	31	33	37
Avg hops	2	1	1	1.32	2	2.63	2.95	3.05	2.84
	2	1	1	3.79	2	3	3.95	3.16	0
Avg PDR	87.1	100	99.85	26.95	97.75	84.85	54.15	82.25	0
	96.65	95	97.8	41.15	95	94.75	73.7	88.15	0
Stdev PDR	20.91	0	0.65	31.31	9.58	27.93	42.43	31.72	0
	3.35	21.79	9.59	38.98	21.79	21.75	40.16	26.07	0
Avg RTT (ms)	9.11	11	1.94	112.28	4.06	5.80	11.01	7.85	0
	7.35	11.12	1.87	46.88	4.08	5.94	10.65	8.11	0

Note: Node 1 is the source. Each cell includes 2 results: the top one is for BATMAN, the other for OLSR.

Acknowledgments

This work and the R2lab testbed are funded by the French ANR through the "Investments for the Future" Program under grants ANR-11-LABX-0031-01 (LABEX UCN@Sophia) and ANR-10-EQPX-0031-01 (EQUIPEX FIT).

5. REFERENCES

[1] T. Clausen et al. OLSR. RFC 3626, October 2003.

[2] A. Neumann et al. Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.). Internet-Draft, IETF, 2008.

[3] Farzaneh Pakzad. *Towards Software Defined Wireless Mesh Networks*. PhD thesis, The University of Queensland, 2017.

[4] Thierry Parmentelat et al. nepi-ng: an efficient experiment control tool in R2lab. In *ACM WiNTECH 2018 Workshop*, New Delhi, India, November 2018.

[5] `https://github.com/fit-r2lab/r2lab-demos`. Batman vs OLSR.