Routing protocol evaluation for the IoT

Requirement analysis and experiment design for large-scale test beds.

Lotte Steenbrink
lotte.steenbrink@haw-hamburg.de

ABSTRACT

In order to gather realistic experience with routing in the IoT, routing protocol evaluation research needs to shift from simulation towards the use of large-scale testbeds. In this paper, a testbed-based evaluation approach for routing protocols is presented, with a strong focus on IoT applications. The approach presented was designed with respect to modularity and extensibility, so as to allow adaption to the high variation in network characteristics in different IoT use cases. Using this approach as a base, routing protocols can be evaluated concerning their suitability for different IoT scenarios, and possibilities for improvements can be uncovered.

Keywords

IoT, routing, MANET, LLN, RPL, AODV, RIOT, test beds

1. INTRODUCTION

The Internet of Things (IoT) envisions autonomous communication between small computers installed in everyday objects or distributed across industrial facilities in order to advance human interaction as well as productivity and security. In 2014, the IoT reached the peak of Gartner's hype cycle for emerging technologies[1]. It is both a growing market and a thriving research field. One central aspect of IoT communications is routing: finding the best paths between nodes and towards sink nodes and gateways is crucial to ensure energy-efficient and smooth network operations. However, practical experience with IoT routing is sparse, and scientific evaluation of such environments is rare. Most routing protocol evaluations are simulation-based, and even fewer of these evaluations have been designed with the IoT in mind. This paper presents a testbed-based evaluation approach tailored to the IoT. The goal is to enable the evaluation of routing protocols which have been created for Low Power and Lossy Networks (LLNs) or Mobile Ad-hoc Networks (MANETs) with regard to their suitability for the IoT.

[1] `https://www.gartner.com/doc/2809728`

The remainder of this paper is organized as follows: First, the need for experimental work is highlighted. Then, the different domains and use cases that form the IoT are assessed. Based on these findings, a network configuration is presented on which all experiments shall be run. Following up, experiment goals, design, realization and evaluation details are assessed. Finally, a conclusion of all findings as well as an outlook into future steps is provided.

1.1 Related work

To date, testbed experiments for the IoT featuring real hardware are rare. However, research on the foundations needed for testbed experiments has been done for about two decades, and is increasingly focused on the IoT.
[1] provides a summary of issues which should be considered when evaluating a routing protocol. Routing requirements for IoT-like scenarios of home and building automation, as well as urban LLNs are described in [2], [3] and [4]. [5] discusses influences on transmission range in food monitoring use cases, in particular monitoring bananas during transport. results were achieved both through mathematical analysis as well as a simple testbed consisting of four nodes. [6] presents the features and failings of different Wireless Sensor Network Testbeds, along with a requirement analysis for IoT-ready testbeds.

2. EXPERIMENTATION VS. SIMULATION

To date, most IoT, LLN, and MANET routing research has been conducted with the help of simulations. While this has the benefits of being cost-effective and widely available, simulations can only say so much about reality: Without "real life" data to check against, the accuracy of a simulation model cannot be determined. As [7] has shown, the assumptions made during network simulation often don't hold in the real world, eschewing simulation results. Matters are complicated further that especially in wireless networks, the quirks of an environment are many. Moving objects, reflection or outside noise can impact the performance of a network severely and are very hard to model in a simulation. The absence of these side effects can be of great benefit when studying specific traits of a protocol, but to determine a protocol's compatibility with the real world, its performance under such disturbances must be assessed, too. Consequently, it is necessary to obtain more realistic testing experience with the help of a growing fleet of test beds, made possible by technological progress and dedicated effort.

3. IOT DOMAINS AND USE CASES

By its very nature, the IoT encompasses a broad spectrum of environments and use cases. Surveys such as [8] and [9] divide the IoT into domains such as Transportation, Healthcare, Smart Environment, and Personal & Social ([8]) or Personal & Home, Enterprise, Utilities and Mobile([9]). The network characteristics vary widely between the different domains, and even within each domain, the variety of characteristics is high. For example,[8] considers both smart homes and industrial plants to be a part of the Smart Environment domain. Still, the networks established in a smart home may differ vastly from the network of an industrial farming facility. The floor space of a single-family house is much smaller than that of an industrial plant, and IoT home applications may be focused on interaction, while industrial IoT is focused on sensing and reporting. Among other things, this implies different network sizes and traffic patterns.
Therefore, while grouping by application domain is useful to map the impact of the IoT and its possibilities for interoperation and interdisciplinary collaboration, this approach to categorizing the IoT is not feasible when it comes to network modeling. Instead, IoT networks can be categorized along a number of metrics on a relatively fine-grained scale. An IoT network can be characterized in terms of:

Network Topology: The network may be organized in a star or mesh pattern.

Traffic Patterns: Some networks experience bursty traffic based on outside events, for example a home automation system when the homeowners return. Others have a regular, scheduled stream of sensor data. Yet others employ a request/response-cycle based on outside events or internal calculations. Packets may travel towards a central *sink node* in a multipoint-to-point fashion, or flooding the network from a central node as point-to-multipoint traffic, or simple point-to-point. The data rate is typically small.

Mobility: Some or all of the nodes of a network may move at different speeds. Movements can be shortlived and caused by displacement, such as the rearrangement of furniture, or constant, such as moving vehicles. Depending on a node's "host", different movement patterns may emerge: Cars drive on a fixed grid of streets at nonlinear speeds, while humans moving though an open field are less restricted in terms of directions, but also less prone to extreme acceleration.

Energy efficiency requirements: Some IoT devices may be built-in to a host with a constant energy supply and therefore not constrained by battery. Other nodes can and will be charged regularly, while yet others must run without maintenance for years.

Scale: According to [2], [3] and [4], network sizes can range from 250 to more than 10.000 nodes, depending on the use case. Some networks, like building automation deployments, are broken down into several subnets containing up to 250 nodes each.

Depending on these characteristics, networks differ in their requirements for a routing protocol. For example, a building automation installation in a factory might feature 1000 nodes arranged in a star topology with scheduled multipoint to point traffic, no mobility, and high energy efficiency requirements, as nodes are expected to operate on one battery for 5 years[3]. A routing protocol suitable for this environment thus has lower requirements for latency and code size, but its high energy-efficiency requirements call for a routing protocol with high route stability and reliability. On the other hand, a solution monitoring the insides of a food truck features a mesh topology made necessary by the high density of the truck's contents which result in low radio ranges and bursty traffic and node mobility whenever the goods are unloaded or rearranged [5]. However, these goods are stored and monitored in boxes, which could be recharged upon arrival, lowering energy efficiency requirements. Thus, an optimal routing protocol for this environment differs vastly from the protocol suitable for the building automation installation described above: While energy-efficiency is less important here, code and storage size is a relevant factor, since the nodes installed should be as cheap and lightweight as possible. Since boxes can be rearranged during unloading/reloading, timely failure recovery is necessary.

In general, Routing Protocol performance metrics for the IoT can be summarized as:

Latency: The latency with which routes are found or packets are sent is crucial to some applications. Networks with high mobility, for example in transport or emergency scenarios, may require quick route establishment and usage before the connection is disrupted, while static farming arrangements are not affected if sensor data arrives at the sink node with higher latency.

Failure recovery: Especially in highly mobile networks, route disruptions should be recognized and– if possible– fixed in a timely manner.

Route stability: Networks with frequently changing routes can be expensive both in terms of latency as well as battery usage: unless constant routing information is maintained[2], route rediscoveries require increased activity of the transceiver, which is the most battery-hungry component of IoT nodes.

Code & storage size: With the exception of border routers and sink nodes, IoT devices typically have constrained memory storage resources. Devices which are used in bulk as "throwaway hardware" for monitoring even more so than devices in household appliances etc. Therefor, protocol complexity and thus code size can be relevant criteria. Another factor is memory usage on operation: extensive routing tables, such as those maintained by proactive protocols, can become a problem especially in large-scale networks, since they increase linearly with the network size. Many embedded OSes use fixed buffer sizes which can be regulated accordingly if the memory usage of a protocol is well-known.

[2] which is the case with most proactive protocols like OLSR[10]

Energy-efficiency: Sending and receiving data is very battery-consuming, so it is advised to keep control traffic as low as possible. Additionally, low handling complexity will help keep retain a high energy efficiency.

Reliability: Routes which experience a high amount of packet loss are prone to triggering packet retransmissions (effectively draining batteries) or losing valuable data. Therefor, it should be ensured that the most reliable route is chosen. A significant part in this is played by *route metrics*: these are the traits by which a protocol decides which link or route to use. Popular metrics include Hop Count (albeit considered suboptimal in many cases) and Expected Transmission Count (ETX).

Based on these characteristics and metrics, an experiment design will be presented in the following.

4. NETWORK MODEL

It can be seen in section 3 that the IoT is a very heterogeneous field in terms of network characteristics, and that a one-scenario-fits-all approach to studying IoT routing is unlikely to be feasible. Therefore, a specific scenario will be studied, modeled and modified in detail over the course of this paper, with the hope that some of the building blocks may be reused as research expands. To achieve this, the characteristics listed in table 1 have been chosen as the base scenario to be modeled, as they can be found in a wide range of applications, and are among the most challenging for routing protocols. In order to study which protocol excels in which scenario, variations have to be created. Therefor, **Case 1** and **Case 2** describe different characteristics which are switched up per experiment round.

5. EXPERIMENT GOALS

It is assumed that all routing protocols involved are fully functional, but behave differently under different circumstances. The main goal is not to test them for functionality, but to examine which protocol performs best[3] under which circumstance, and which factors impact routing protocol operations negatively. These factors may be unforeseen quirks which did not occur during previous simulations, or specific network configurations, or something completely different.

6. EXPERIMENT DESIGN

After all prerequisites have been discussed, a specific experiment can be designed.

6.1 Choosing the testbed

In order to run the experiments in a lifelike, but still controlled environment, a testbed is needed. Ideally, a testbed suitable for the IoT should be able to provide their users with at least several hundreds, but ideally several thousands of nodes, a diverse range of hardware, and a number of mobile nodes. [6] compares several testbeds with regard to suitability for the IoT, and concludes that the FIT-IoTLab[4] is one of the most suitable facilities. Located all over France, the FIT-IoTLab offers 2,728 nodes in total, featuring three different hardware platforms of different capabilities:

The WSN430 Node featuring a MSP430 MCU with 48kB Flash, 10kB RAM, an IEEE 802.15.4. radio interface, as well as sensors for ambient sensor light and temperature.

The M3 Node featuring an ARM Cortex M3 MCU with 64kB RAM, an IEEE 802.15.4. radio interface, as well as sensors for ambient sensor light, atmospheric pressure and temperature, a gyrometer, and an accelerometer.

The A8 Node featuring an ARM Cortex-A8 microprocessor with 256 MV RAM, an ethernet interface, a gyrometer, and an accelerometer.

Additionally, the IoT-Lab offers node mobility through a fleet of toy trains. This allows for the use of realistic, but controllable mobility patterns. The two more constrained platforms of the IoT-Lab, the Wsn430 and M3 nodes, offer support for RIOT[11][5]. This combination is unique among all available testbeds, and provides every feature needed to conduct the described experiments. Therefore, it is advised to run the experiments described this paper on the FIT-IoTLab testbed.

6.1.1 Involved routing protocols

RIOT currently features implementations of two routing protocols: RPL[12] and AODVv2[13]. The former is a proactive, point-to-multipoint-protocol designed for LLNs, while the latter is a reactive point-to-point protocol designed for MANET. Additionally, implementations of the proactive MANET protocol OLSR[10] and the Ant Routing Algorithm (ARA)[14] are in progress.[6] All protocols vary vastly in their characteristics and application scenarios, so it would be advisable to involve as many as possible in the experimentation. In addition to RPL and AODVv2, any of the other protocols which is available by the time of the experiment should be used.

6.2 Experiment setup and execution

In preparation of the experiment, the following is created:

1. a randomized list containing the IDs of all participating nodes. It should contain some duplicates. This is needed to create point-to-point traffic.
2. a randomized list containing tuples with randomized pairings of the IDs of all *except for one* participating nodes. It should contain some duplicates. This is needed to create multipoint-to-point traffic.
3. a sample packet with a payload of 20 bytes, resulting in a 61-byte packet including IEEE 802.15.4. and IPv6 headers with applied 6LoWPAN header compression.

These lists must never be changed throughout the experiment, and should be stored along with the experiment data

[3] with regards to the metrics listed in section 3
[4] `https://www.iot-lab.info`
[5] `https://www.iot-lab.info/operating-systems/`, accessed 19.05.2015
[6] `https://github.com/RIOT-OS/RIOT/pull/2294` `https://github.com/mfrey/RIOT/tree/ara`

Characteristic	Case 1	Case 2
Network Topology	Mesh	
Traffic Pattern	Multipoint-to-point, with most traffic traversing several hops. Scheduled data transmissions.	point-to-point across the network. Scheduled data transmissions.
Mobility	Sparse and bursty: Nodes move seldomly and not constantly	
Scale	100	500

Table 1: Characteristics of the modeled network(s)

for future reference.

The complete experiment consists of several runs, which are grouped per scenario. Each experiment will be run on four scenario combinations, as noted in table 1. The scenario combinations will be run through one by one, and each routing protocol is tested on the scenario before moving on to the next one. Still, the scenario is torn down and built from scratch for each new protocol, so as to prevent any unwanted side effects. The resulting running order is as follows:

Scenario #	Traffic Pattern	Scale	Protocol
1	Case 1	Case 1	AODVv2
1	Case 1	Case 1	RPL
1	Case 1	Case 1	(other)
2	Case 1	Case 2	AODVv2
2	Case 1	Case 2	RPL
2	Case 1	Case 2	(other)
3	Case 2	Case 1	AODVv2
3	Case 2	Case 1	RPL
3	Case 2	Case 1	(other)
4	Case 2	Case 2	AODVv2
4	Case 2	Case 2	RPL
4	Case 2	Case 2	(other)

Each experiment run is conducted as follows:

For scenarios 1 and 2, List number 1 is used to model multipoint-to-point traffic. The node not contained in the list is appointed as the *sink node* towards which all traffic is directed. Then, the list is traversed sequentially to initiate sending:

```
for node in node_ids:
    node.send_packet(sink_node)
    wait(fixed_time)
```

For scenarios 3 and 4, List number 2 is used to model point-to-point traffic. Each tuple **(node_1, node_2)** represents a transmission from **node_1** to **node_2**. This list too is traversed sequentially to initiate sending:

```
for (node_1, node_2) in node_ids:
    node_1.send_packet(node_2)
    wait(fixed_time)
```

This ensures the exact same transmission sequence for each experiment run, eliminating possible side effects.

For each scenario combination, each experiment is run a fixed number of multiple times, so as to not eschew data by isolated incidents. All numbers used should be tweaked in case they are found to be unrealistic.

6.2.1 *Experiment implementation*

In order to provide a reusable and tweakable setup, modularity is to be kept in mind when implementing this experiment. Components should be parametrized wherever possible to allow further experiment variation. Additionally, the resulting experiment setup should be easy to use to enable the reproduction of experiments.

6.3 Experiment evaluation

The success or failure of each routing protocol is determined with the metrics listed in section 3. A protocol's performance regarding a certain metric is evaluated as follows:

Latency: The median difference between packet dispatching and arrival time is used to calculate the latency with which packets are sent. For proactive protocols, the median time in which routes appear in the Forwarding Information Base (FIB) is taken for route finding latency. For reactive protocols, each yet unknown tuple is examined: the median time it takes between sending attempt and appearance of the route in the FIB determines route creation latency.

Failure recovery: TODO

Route stability: TODO

Code & storage size: Memory usage is monitored during the experiment. The median as well as the maximum memory usage are used to determine memory efficiency, as well as the code size at compile time.

Energy-efficiency: Energy usage is monitored during the experiment as well and can be compared between protocols per experiment batch.

Reliability: Packets are recorded when they are sent at one end and received at the other. This way, the median number of lost packets can be determined.

7. CONCLUSION AND OUTLOOK

Over the course of this paper, the necessity of IoT routing protocol experimentation aided by testbeds has been discussed. Challenges in the creation of a setup have been discussed, along with possible solutions and a concrete experimentation setup. It has been stressed that this setup is to be extended to fully honor the diversity of IoT scenarios, and that it can be merely a starting point. These extensions may

not only be limited to further switching of parameters and increasing the network size. Instead of waiting for a fixed time interval between transmissions, the waiting time could be randomized, or physically close nodes could send simultaneously, simulating a local triggering event. Detailed mobility schemes could be developed, or more complex and/or hybrid traffic patterns. A wider range of protocols should be tested: protocols with similar characteristics could be compared against each other, or the same protocol could be tested with different route metrics.

Based on the findings of all of these experiments, a map of protocol characteristics suitable for different IoT scenarios can be created. Future work could also include the development of optimization extensions for any of the routing protocols involved.
Before all of this can be done, however, the provided experimentation scenarios will have to be implemented, put to the test, and tweaked.

8. REFERENCES

[1] M. S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, IETF, January 1999.

[2] A. Brandt, J. Buron, and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks," RFC 5826, IETF, April 2010.

[3] J. Martocci, P. D. Mil, N. Riou, and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks," RFC 5867, IETF, June 2010.

[4] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks," RFC 5548, IETF, May 2009.

[5] R. Jedermann, T. Pötsch, and C. Lloyd, "Communication techniques and challenges for wireless food quality monitoring," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2017, 2014.

[6] A.-S. Tonneau, N. Mitton, and J. Vandaele, "A survey on (mobile) wireless sensor network experimentation testbeds," in *Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on*, pp. 263–268, May 2014.

[7] D. Kotz, C. Newport, and C. Elliott, "The mistaken axioms of wireless-network research," tech. rep., Dartmouth Computer Science, July 2003.

[8] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, pp. 2787–2805, Oct. 2010.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[10] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, IETF, October 2003.

[11] O. Hahm, E. Baccelli, H. Petersen, M. Wählisch, and T. C. Schmidt, "Demonstration Abstract: Simply RIOT – Teaching and Experimental Research in the Internet of Things," in *Proc. of 13th ACM/IEEE Conference on Information Processing in Sensor Networks Demo Session (IPSN)*, (Piscataway, NJ, USA), IEEE Press, April 2014.

[12] 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, IETF, March 2012.

[13] C. Perkins, S. Ratliff, J. Dowdell, L. Steenbrink, and V. Mercieca, "Dynamic MANET On-demand (AODVv2) Routing," Internet-Draft – work in progress 09, IETF, May 2015.