Automated Online Reconfigurations in an Outdoor Live Wireless Mesh Network

Mustafa Al-Bado Anja Feldmann Thorsten Fischer Thomas Huehn Ruben Merz
Harald Schioeberg Julius Schulz-Zander Cigdem Sengul Benjamin Vahl*
Deutsche Telekom Laboratories at TU-Berlin
{mustafa, anja, thorsten, thomas, harald, julius, bvahl}@net.t-labs.tu-berlin.de
{ruben.merz, cigdem.sengul}@telekom.de

1. INTRODUCTION

Wireless networking research lacks a comprehensive experimental environment that supports implementation, validation, deployment, performance evaluation and comparison of research ideas. While some open wireless testbeds [1, 2] have taken important steps to create such an environment, they typically provide researchers only access to a set of mesh nodes, an operating system distribution and scripts to run experiments. Hence, implementation of an idea still remains significantly difficult - for instance, even modification of MAC or transport protocols require invasive kernel changes. What researchers need is an easy-to-use environment that provides them (1) *a protocol toolkit*: to implement and modify only the necessary elements and not an entire system and (2) *reconfiguration ability*: to easily configure nodes as well as network protocols both in an offline and online manner. To the best of our knowledge, no open wireless testbed exists that satisfy both (1) and (2).

Another deficiency of current wireless research networks is that they do not typically serve user traffic. This prohibits evaluating solutions in a realistic manner. Having real users in a testbed creates a unique opportunity for testing protocols with real traffic and mobility patterns. However, it also creates the need to reconcile two different worlds: the worlds of users and researchers. Real users expect a reliable network access and demand privacy. On the contrary, a research network is an unstable environment with frequent outages. To support both *real users* and *ease-of-use*, all the components of the experimental environment need to be designed with these two factors in mind. Additionally, it is necessary to provide automated and online reconfiguration of the network. Furthermore, reconfigurations should be coupled with fail-safe mechanisms to ensure the reliability of the user traffic. These capabilities enable the network to run without downtime.

We are designing such an environment in Berlin. The Berlin Open Wireless Lab (BOWL)[1] is an open wireless mesh testbed designed to provide Internet access to real users (i.e. the TU-Berlin community) and an experimentation environment to researchers in wireless mesh networking. BOWL has two essential capabilities:

- Automated and online reconfigurations — It includes turning on/off functionality of a specific protocol and changing protocols at different network layers respecting inter-layer dependencies.
- Hosting real user traffic — This requires avoiding disruption to the user traffic during reconfigurations. Fail-safe mechanisms are triggered to automatically redirect user traffic during reconfigurations. The users are directed back once a reconfiguration is complete.

To the best of our knowledge, compared to other work (e.g., MeshMon) [3, 1, 2], BOWL is unique in providing (1) a protocol toolkit and (2) a reconfigurable experimental environment. Additionally, it has visualization support for illustrating these reconfigurations. In the rest of this paper, we will describe the possible configuration scenarios (Section 2) and how the Berlin Open Wireless Lab network (Section 3) makes them possible.

2. EXAMPLE CONFIGURATION SCENARIOS

To run their experiments in a mesh network, researchers need the capability to perform automated and online reconfigurations. This is essential for exploring the parameter space of a given protocol, or compare different protocols under similar network conditions. However, user traffic needs to be protected from disruptions during configurations. In the following, we describe various possible reconfigurations in BOWL in the order of increasing complexity, effect on ongoing user traffic, and value to the researchers.

2.1 Node and Network Reconfigurations

One of the possible configurations is to modify the network topology: to connect additional mesh nodes to or disconnect nodes from the network remotely. The level of network connectivity can be modified by changing the per-node transmission power levels.

Furthermore, these changes are made easily accessible to the researchers through visualization. As a visualization tool, a testbed map displays the current situation in the network, for instance, existing *real user* connections, current node configurations and protocol parameters. Using this map, important changes, such as new nodes or disruptions in the network, can be displayed accurately and efficiently. (Note that network disruptions may not be captured if fail-safe mechanisms are triggered.)

2.2 Switching to Different Protocol Parameters

In BOWL, protocol parameters can be easily modified, which is especially important for testing the effect of different parameters on the protocol behavior under similar

*Authors are listed in alphabetical order.
[1]http://bowl.net.t-labs.tu-berlin.de/

network conditions. For instance, we can change a MAC layer functionality: turn on and off RTS/CTS. Next, we can switch to different routing metrics, for instance, from hop count metric to ETX (Expected Transmission Count) in OLSR. At the end of such a reconfiguration, the testbed map reflects the changes in the cost of the links as well as potential route changes for some flows. During such reconfigurations, user traffic is expected to be disrupted. Therefore, fail-safe mechanisms are triggered to minimize the impact on user traffic.

2.3 Switching to a New Routing Protocol

This is the most challenging scenario as switching to a new routing protocol is a more disruptive change. One of the challenges stems from different philosophies followed by routing protocols. A few examples are reactive vs. proactive protocols, source routing vs. distance vector routing. Consider a switch from a proactive routing protocol, OLSR to a reactive routing protocol, DSR (Dynamic Source Routing). In DSR, the only time a route is discovered is when a new flow is initiated. Therefore, additional steps need to be taken to first find routes for flows that exist in the network during reconfiguration. In the future, we plan to preserve the old routing state of OLSR and translate this information to a routing cache that can be used by DSR. Additionally, DSR does not require route lookups at the intermediate nodes as the route is carried in the packet. Hence, the following changes are expected during the reconfiguration. First, fail-safe mechanisms redirect active flows. After DSR is installed in all nodes, for each connection that is redirected, a route request is sent and route caches are populated. As routes for existing flows are discovered, the flows are redirected. At the end of the reconfiguration, the network map is expected to show different routes for the flows. Also, the packet traces will show source routes in packet headers.

3. BOWL TESTBED DESCRIPTION

Our testbed deployment for the Berlin Open Wireless Lab (BOWL) project is still an ongoing process as mesh nodes are continuously installed on the roofs of several buildings spanning the campus. The important components of our testbed that render automated and online reconfigurations possible are (1) the node and the network design, (2) configuration management and monitoring, and (3) the so-called protocol toolkit.

The node and the network design were shaped by the need to perform remote management and updates, in addition to fault-tolerance and recovery in the presence of wireless network failures. Since nodes are deployed on rooftops, physical access to the nodes is limited. Therefore, all nodes have a dedicated Ethernet connection to allow performing management tasks remotely. Furthermore, most nodes contain two independent units, the actual mesh router, and a simpler unit to manage the mesh router in case of a malfunction (e.g. due to loss of wireless connectivity or operating system failures). A watchdog daemon running on the mesh router automatically triggers a reboot of the mesh router in the presence of failures. In addition, a custom-built circuit allows us to remotely power on and off each board. The dual board design also allows for the easy upgrade of the hardware of the mesh router while the rest of the infrastructure remains in place. In our network, it is possible to provide reliability for the user traffic by redirecting the traffic, for instance, to the wired interface. This feature also is actively used by the configuration management and monitoring. (Note that the traffic can also be redirected to secondary wireless interfaces, if the network remains connected using these interfaces.)

The configuration management and monitoring component remotely and automatically manages, configures and checks the status of the nodes in the testbed. During reconfigurations, fail-safe mechanisms are triggered to protect user traffic. For instance, these mechanisms take the active flows off the wireless mesh network on to the wired and back. In the current state, it is triggered on-demand whenever we expect a major disruption of the network traffic and connectivity. All nodes participate in the process to ensure that both in-flight and future traffic is eventually safely delivered. The expected delay should be in terms of a few seconds. In the future, we will extend with a second operating mode, which will automatically invoke these mechanisms as a result of active probing and failure detection.

The protocol toolkit provides a comprehensive system environment where the researcher can focus on individual components without losing the whole system point of view. Hence, it becomes possible to add, switch between or modify protocols easily, which are essential to the configuration scenarios. The protocol toolkit is based on the Click Modular Router [6]. Its flexibility is enabled via well-defined elements, which interface to each other seamlessly. Elements can be dynamically added to and removed from the current running configuration using the hot swapping capability of Click.

4. CONCLUSION

This work describes the first steps towards an experimental research environment with real user traffic. For this purpose, we demonstrate how to run automated and online reconfigurations in a mesh network. We span different demands: (1) change the network topology, (2) change a protocol functionality and (3) change a protocol in a given layer. Such reconfigurations allow the BOWL testbed be an attractive tool for outside researchers, especially for demos. Essentially, the researchers can find the opportunity to demonstrate their solutions under live traffic.

5. REFERENCES

[1] School of Computing, University of Utah, "Emulab - total network testbed," http://www.emulab.net/.

[2] "Orbit," http://www.orbit-lab.org/.

[3] S. Nanda, "Mesh-mon: A monitoring and management system for wireless mesh networks," Ph.D. dissertation, Dartmouth College, 2008.

[4] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," http://www.ietf.org/rfc/rfc3626.txt, Oct. 2003.

[5] D. B. Johnson, D. A. Maltz, and J. Broch, *Ad Hoc Networking*, 2001, ch. DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks.

[6] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, 2000.