

Demo: Disruption Free Live Reconfiguration in a Wireless Testbed

Mustafa Al-Bado, Anja Feldmann, Thorsten Fischer, Thomas Hühn,
Ruben Merz, Harald Schiöberg, Julius Schulz-Zander, Cigdem Sengul, Benjamin Vahl

Deutsche Telekom Laboratories
An-Institut der Technischen Universität Berlin
<firstname>@net.t-labs.tu-berlin.de

*This demo has already been presented at Sigcomm 2009.
It requires at least 4Mbps of Internet bandwidth.*

1 Introduction

The Bowl¹ Testbed is a generic wireless testbed that focuses on evaluation of new research proposals in a realistic environment. The testbed is deployed on the roofs of the TU Berlin campus, and can be used by all students of the university to access the Internet. We incorporate all aspects of the system, not only the data plane, but also control plane aspects such as mobility management and fail-over strategies. Each component is designed for clear separation of functionalities and modularity for easy deployment of new experiments.

Despite our goal to test experimental software under realistic conditions, it is essential to guarantee stable service to the users in the presence of reconfiguration or failures. The network is presented as flat layer-2 transparent infrastructure Wifi network to the user throughout all modes of operation. Internally the nodes of the network have three different states of operation to guarantee reliability of the end-user service: In *live* mode the users are actually served via the experimental protocols. Constant monitoring of the network health enables us to immediately revert to a more stable state even if severe faults occur. In *rescue* mode, the node acts as an access point and serves all traffic using the wired infrastructure, using the Linux software bridge. In *transition* mode the user traffic is duplicated to the experimental Click based forwarding, yet only the data from the bridge is actually delivered to the user.

1.1 Setup

For this demo, the testbed is configured as a wireless mesh network. Two nodes are set up at the venue and can be attached to arbitrary locations of the main mesh, and traffic from the venue is routed through the main mesh to the Internet. All effects of the demo scenarios can be seen live on on-site clients as well as a (Google-Maps) overlay visualisation of the traffic in the main testbed.

1.2 Scenarios

The demo starts with all mesh nodes in *rescue* mode. All user traffic is served by the wired infrastructure, Click is disabled, and only Linux bridging is used. In the next stage, *transition* mode,

the click forwarding process is started, along with either the commodity *olsrd* routing daemon or the click internal *dsr* routing. Traffic is duplicated while the routing tries to explore a path to the Internet gateway. As soon as a path to the Internet gateway is established, the nodes automatically switch to *live mode* and disable the Linux bridge. Further instabilities can then be introduced, and the system will react by temporarily falling back to a lower mode of operation, with only very brief disruptions of end-user connectivity.

2 Architecture

The architecture consists of the Click Router as the forwarding engine and a control framework based on distributed Ruby. Within the Click Router, IP-IP encapsulation is used to manage end-system mobility: packets arriving at the edge of the network are first processed by an *IPRouteTable* Element, called the *Mesh Location Table*. It maps the packet's destination IP address to the mesh node where the actual client is attached. It is maintained by the location manager part of the Ruby control framework. Packets are then encapsulated in an additional IP Header, using the target mesh node as the destination address. Then they are passed to the *Mesh Routing Table*, which is another *IPRouteTable* in the case of *olsr*, or a *DSRRouteTable* Element. This element does the actual forwarding within the mesh, until the Packet reaches the exit node, where it is decapsulated and delivered. Some ARP hacks on the external interfaces are used to complement the transparent view of the network. Click is also used to add an additional layer of IP-IP encapsulation to attach the remote demo nodes transparently to the main mesh.

3 Future Work

We plan to open the testbed by the end of 2009 to the research community to test new approaches and protocols under realistic conditions. In addition, by formalizing our implementation into a framework with clearly defined components, we hope to foster the development of mutually compatible components for different network layers. Such a framework will help clarify the interactions between the various layers, as well as the true comparability of different approaches. We furthermore plan to include more aspects than just forwarding, such as the transport layer, application layer services and obviously the MAC layer in our framework.

¹<http://bowl.net.t-labs.tu-berlin.de>