Wide Area Network Emulation

Bruno Quoitin (bruno.quoitin@uclouvain.be)
UCLouvain, Louvain-la-Neuve, Belgium

I. Introduction

A Wide Area Network (WAN) emulator is useful to test and validate the behaviour of networking applications and protocols in a testbed. A WAN emulator can be used to introduce latency, jitter or bandwidth constraints on traffic exchanged between multiple endpoints. How the prototype under test reacts can then be verified. A WAN emulator is also able to provide multiple forwarding paths with different characteristics over a complex network topology.

Most existing network emulation tools focus on the emulation of links characteristics only. Dummynet [5], for example, is limited to link emulation. It relies on the software Ethernet bridging capabilities of FreeBSD to enforce queue and bandwidth limitations, delays, packet losses, and multipath effects. Another link emulation tool targeted at the Linux platform is NetEm [2]. NetEm enhances the traffic control facilities of the Linux kernel.

The closest work to ours is NetPath [1], a network emulation tool mostly targeted at layer-2 switching. NetPath runs on top of Click but it does not offer any IP routing facilities and routing tables need to be populated by hand.

This abstract presents an application of the Click modular router [3] to WAN emulation. Our emulator simulates some forwarding characteristics of paths through a complex network topology. The emulator runs on a single computer. Moreover, each element of the paths through the emulated topology is visible to tools such as traceroute.

II. Methodology

We rely on Fig. 1 to describe our methodology. A simple testbed with two real hosts and the WAN emulator in the middle is shown. Both hosts are connected to different Ethernet ports on the emulator. In Fig. 1 host 1 sends an IP datagram to host 2 through the WAN emulator. The datagram has 192.168.2.1 as source address and 192.168.5.1 as destination. When the datagram is received by the WAN emulator, it must traverse the emulated topology.

The emulator receives the datagram through its 'en0' interface which is associated with the left link of virtual router 0.1.0.0. A lookup in the forwarding information base (FIB) of the virtual router is performed to find the nexthop for the datagram. In our example, the nexthop is virtual router 0.2.0.0. A link propagation delay of 20ms has been specified between 0.1.0.0 and 0.2.0.0, therefore the datagram is queued for 20ms before it is handed to router 0.2.0.0. Then, a lookup is performed in the FIB of 0.2.0.0 to find where the datagram must be sent. In this case, it must be sent through the right

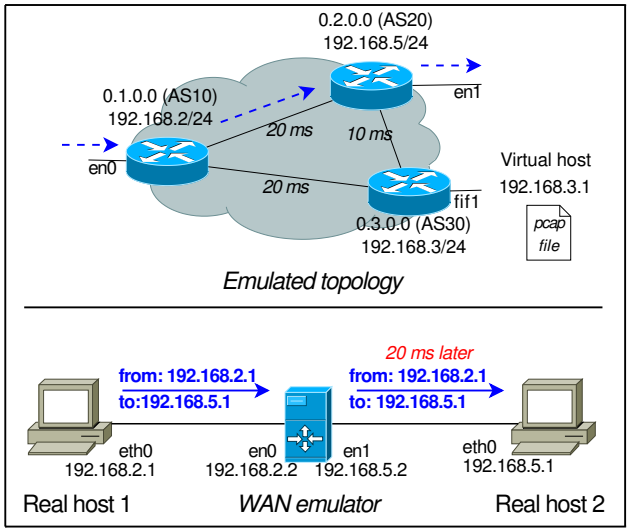

Fig. 1. Network emulation methodology

link of 0.2.0.0 which is associated with the 'en1' interface. The datagram is sent outside and reaches real host 2.

Our WAN emulator prototype has no "control plane", i.e. no routing protocol is running in the emulator. We use the C-BGP routing solver [4] off-line to compute how routes between the virtual routers are selected. We then filter the topology to keep only the nodes and links that are on the used forwarding paths. For each node in the obtained topology, we build a simplified IP router using the Click modular router. Each router has a StaticIPLookup element populated with the nexthops computed by C-BGP. Pure virtual links are modeled using Queue and DelayUnqueue elements configured with the appropriate link propagation delay. Links between the virtual nodes and the physical interfaces are based on To/FromDevice elements.

A prototype of our WAN emulator is publicly available on http://cbgp.info.ucl.ac.be/projects/cbgp2click.php

References

[1] S. Agarwal, J. Sommers, and P. Barford. Scalable Network Path Emulation. In *13th IEEE MASCOTS*, September 2005.
[2] S. Hemminger. Network Emulation with NetEm. In *Linux Conf Australia (LCA)*, April 2005.
[3] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. Kaashoek. The Click Modular Router. *ACM TOCS*, 18:263–297, August 2000.
[4] B. Quoitin and S. Uhlig. Modeling the Routing of an Autonomous System with C-BGP. *IEEE Network*, 19:12–19, November 2005.
[5] L. Rizzo. Dummynet. http://info.iet.unipi.it/~luigi/ip_dummynet/.