User-Space Network Emulation for the SPLAY Platform

Valerio Schiavoni, Pascal Felber, Étienne Rivière

University of Neuchâtel, Switzerland

Network Emulation

Context and Problem

- Testbeds to evaluate distributed protocols
- Require privileged access
- Tedious setup, (expensive) dedicated HW
- Many under-used machines available

Challenges

- Emulation of internal nodes and links:
  - Congestion, BW, delays
  - No dedicate traffic shaping node
  - React to fast detection of traffic changes

Objectives

- Multiple network topologies at the same time
- Cross-platform execution
- No kernel modifications
- Reach performances of dedicated hardware

The SPLAY Platform

Application-Level Network Emulation for SPLAY

Decentralized Network Congestion Monitoring

- Lightweight diffusion protocol to notify nodes about current data flows
- Nodes adjust BW rates according to the competing flows on bottleneck links
- At T2, the congestion on bottleneck link R1-R2 triggers BW throttling

Classical Solutions

- ModelNet
- Emulab

Implementation

- Pure Lua
- Congestion monitoring using UDP/Multicast (for LANs)
- BW shaping using Hierarchical Token Bucket
- Support for ModelNet’s and Emulab’s topology DSL

Evaluation

- Congestion detection and BW throttling
- Delay accuracy

Ongoing Evaluation

- Comparison with Emulab and ModelNet
- Test with real distributed protocols:
  - BitTorrent, Chord
  - PlanetLab Emulation

www.splay-project.org

Network Topology to Emulate

Topology Representation in DSL

Dedicated Cluster

User access through web services

Ideal, congestion-free

with topology

Router

Controller

Emulator

Nodes about current data flows

Nodes adjust BW rates according to the competing flows on bottleneck links

At T2, the congestion on bottleneck link R1-R2 triggers BW throttling

Submitted job

Congestion on R1-R2

Congestion monitoring using

Queues

Queues

Deployed on commodity HW

Command implementation of algorithms in Lua

Standby daemon

Active daemon

Local testing & debugging (no code change)