

Emulab Delay Agent Presentation

Ryan Jackson and Jonathon Duerig

Purpose of Delay Agent

- Control Link Properties
 - Bandwidth
 - Delay
 - Packet Loss

Goals of New Implementation

- Support link up/down, bandwidth limits, delay
- Support Linux for end node shaping
- Support Linux as a delay node
- Clean API between OS-independent front-end and OS-specific back-ends

New Delay Agent Capabilities

- Bring links up and down using OS loss implementation
- Modify bandwidth limits
- Change packet delay

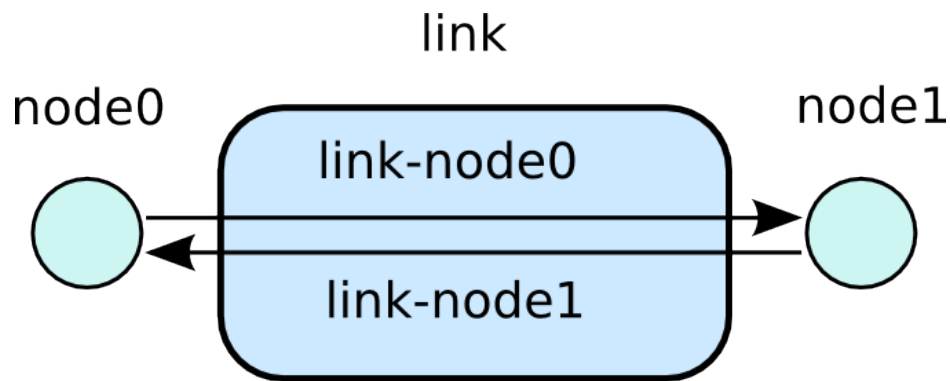
OS Support

- FreeBSD
 - End node shaping
 - Delay Nodes
- Linux
 - End node shaping

Design (Front-End)

- From Events
 - Receives events from Emulab event system
 - Events modify link parameters or bring them up/down
- To Back-ends
 - Demuxes events to appropriate pipe(s)
 - Parses parameters to make a Parameter object
 - Save parameter object and sends it to back-end.
 - Link up/link down are a special case.

Multiple Pipes



- Each link has two pipes
- Events to 'link' apply to both
- Events to 'link-end' apply to one
- An agent may handle multiple links

Design (Back-End)

- Receives parameters from front-end
- Parameters handled based on type
 - BANDWIDTH, DELAY, LINK_UP
- Call OS-dependent routines to modify desired property

Experimental Results

	Old Agent	FreeBSD	Linux
Bandwidth (kbps)			
Initial	968	968	996
Incremental	489	490	488
Link Down	0	0	0
Link Up	5170	5170	5169
Delay (ms)			
Initial	41	41	42
Incremental	43	43	43
Link Down	100% Loss	100% Loss	100% Loss

Experimental Analysis

- FreeBSD performed almost identically
 - No underlying OS change
- Linux differences due to different OS environment

Conclusion

- Within project requirements, the new agent is a drop-in replacement for the old one
 - Remaining requirements (packet loss, RED, etc.) will be added later
- All project goals were met, except supporting Linux as a delay node
 - This requires modification of Emulab infrastructure outside of the delay agent, but the agent code supports this configuration