Network Congestion-Control Emulator

Open Source with SLU

Anh Nguyen, Andrew Berry, Halima Malik (project lead)
Department of Computer Science, Saint Louis, MO 63110

> motivation.c

<why> Congestion control is a mechanism employed in a network to control the rate in which packets are sent to avoid congestion and collapse of the network. Different congestion control algorithms have been developed in response to observed congestion and collapse while fully utilizing the bandwidth of the network. These algorithms use different packet queue management mechanisms, scheduling algorithms and network feedback signals to avoid congestion. Therefore, it is hard to test and compare different protocols across the board when subjected to the same network conditions. </why>

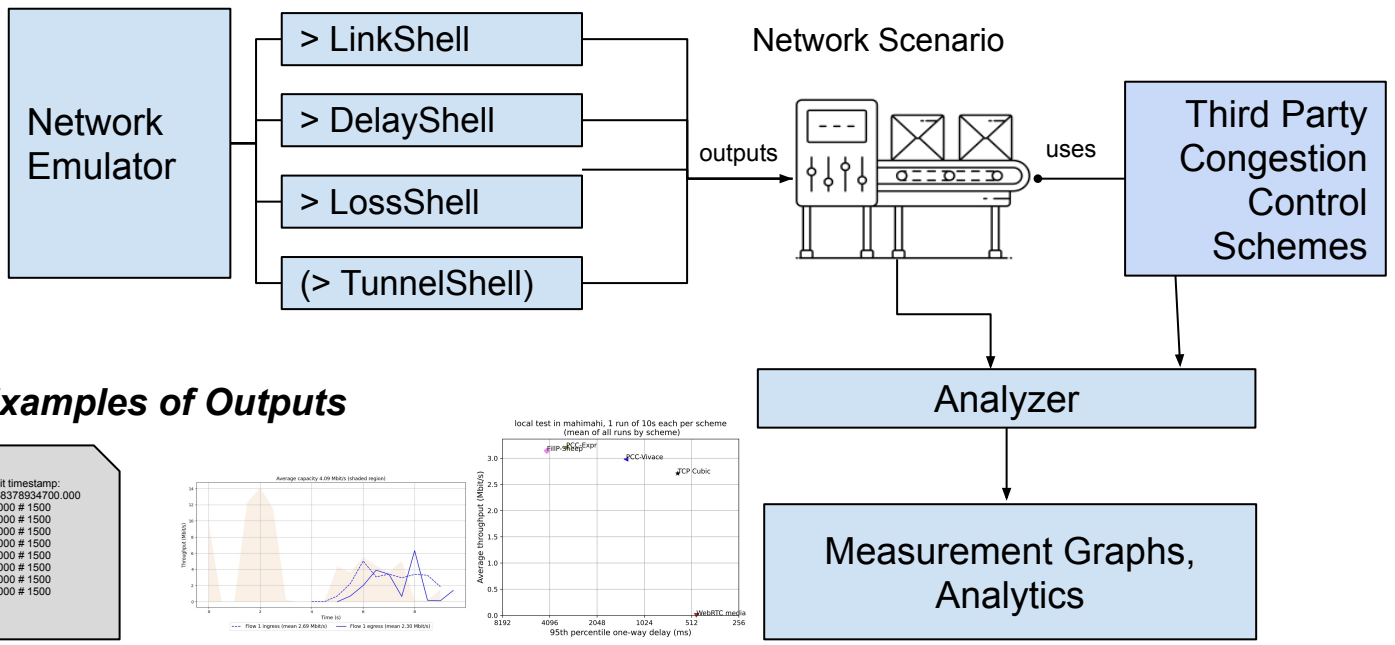
<what> This purpose of this project is to create a software capable of emulating different network scenarios while evaluating these different congestion control protocols. The software generates evaluation reports in terms of measurement metrics such as throughput, delays and jitters. </what>

<who> The client requesting the software is Dr. Flavio Esposito from our department. Our work would benefit network engineers and the general networking research community. </who>

<how> For our project, we create a software that enables users to easily choose **TCP congestion-control schemes** and **emulated network conditions** for benchmarking analysis. Open-source software are available to emulate network conditions (Mahimahi) and analyze performances of different algorithms (Pantheon, end of support). Our development process involves:

 understanding the use of existent open-source software
 refactoring emulation features
 updating the legacy codebase to be compatible with the current network technology
 implementing a set of robust features to allow a wider range of developed algorithms
</how>

> architecture.c



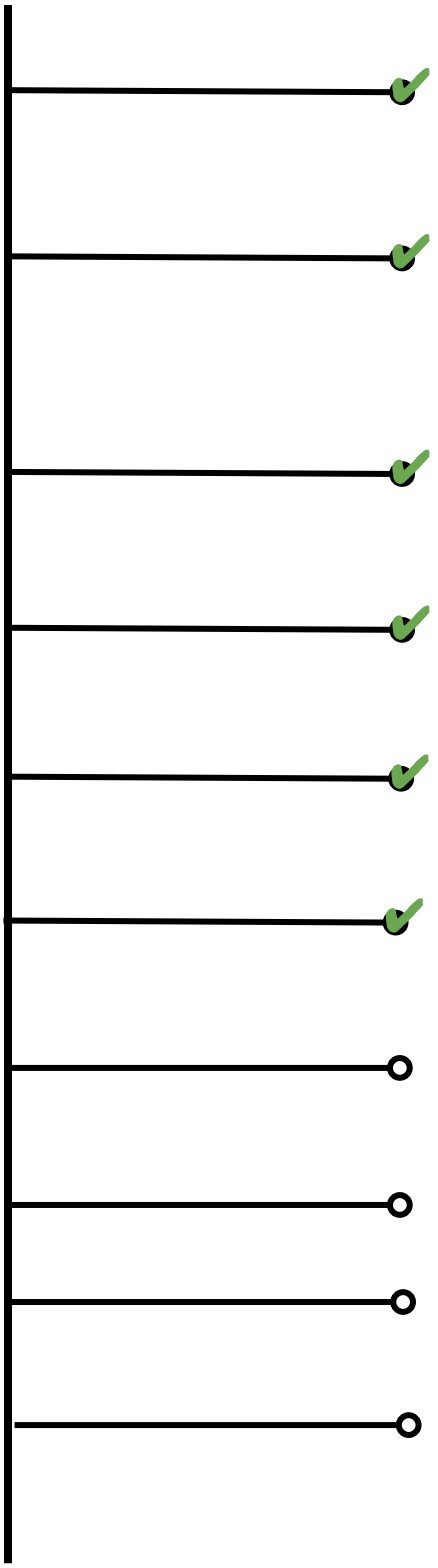
Examples of Outputs

Timestamp files per protocol

Throughput performance in 2 network scenarios

Benchmarking results across different protocols

> progress.c



- ✓ Understand the existing open-source software. Identify core components from the legacy code for our emulator
- ✓ Extract emulation features developed in Mahimahi to encapsulate network scenarios (fixed delay on a per-packet basis, packet loss, and delivery packet links based on input traffic traces).
- ✓ Update the legacy open-source codebase (language version, API changes, configuration parameters) to be compatible with current network application frameworks and libraries for future maintenance and development.
- ✓ Create a virtual machine for flexible testings and usability of the application.
- ✓ Combine both emulation and analysis features (latency, loss, throughput metrics computations) into a single network emulator application.
- ✓ Run experiments and test different congestion control protocols against different network congestion scenarios using our developed application.
- ○ Interface our software with eBPF (Extended Berkeley Packet Filter) and eXpress Data Path (XDP) to read signal information and enable developers to write low-level scripting.
- ○ Support a wider range of congestion control network protocols, including ML-based methods
- ○ Enable the process of learning congestion window size within our emulation system
- ○ Implement a set of more robust monitoring metrics throughout the whole transport process (from application to transport layer. Transport layer is where congestion occurs)

@ developers

a8nguyen

andrewb1243

HalimaMalik

github.com/oss-slu/Congestion-Control-Emulator