Implementation of Stateless Transport Protocols in ns-3

D.Ju. Chalyy

e-mail: chaly@uniyar.ac.ru

P.G. Demidov Yaroslavl State University,
Yaroslavl, Russia

May 14, 2015
TCP Architecture

- Originally from 1980s: in-order delivery, loss recovery, congestion control;
- Many major improvements: new congestion control algorithms etc.;
- Tons of minor improvements: bug fixes, specific corrections etc.
Stateless architecture

- **Store no application state on your servers**
  
  *AWS Tips I Wish I’d Known Before I Started*
  
  ([https://wblinks.com/notes/aws-tips-i-wish-id-known-before-i-started/](https://wblinks.com/notes/aws-tips-i-wish-id-known-before-i-started/))

![Diagram showing stateless and stateful architectures](chart)

Must retain in-order delivery, loss recovery, congestion control.
Trickles protocol

- Developed in Cornell in 2000s:

- Sample implementation exists for Linux kernel (last commit in 2006)
Trickle congestion control

- Trickle management: split, continue, terminate;
- Follows Reno congestion control algorithm: slow start, congestion avoidance, fast retransmit.
Overview of Trickles model in ns-3

Source code is available at https://github.com/dchaly/stateless
Stages of modeling in ns-3

- Build a model of your own protocol or use existing ones.
- Describe an experiment using ns-3 classes.
- Compile, run and get a trace file:
Research pipeline

Author

Raw data → Analytic data → Computational results

Cleaning data → Analytic code → Presentation code

Figures → Tables → Article

Numerical summaries → Text

Reader
Research pipeline with ns-3

Author

Raw data → Analytic data → Computational results

Analytic code → Presentation code

Cleaning data

Figures

Tables

Numerical summaries

Article

Text

Reader

P.G. DEMIDOV
YAROSLAVL STATE UNIVERSITY

A tool for filling the pipeline

Key insight: use data science methods. Use R!
- Free software for statistical computing
- Tons of packages for different needs (CRAN)
- R-studio is a free IDE for developing
- http://www.r-project.org
Research pipeline

Author

Raw data

Use R!
Cleaning data

R data frame

Analytic data

Use R!
Analytic code

Computation results

Use R!
Presentation code

Figures

Tables

Numerical summaries

Article

Text

Reader


P.G. DEMIDOV
YAROSLAVL STATE UNIVERSITY
Literate programming

- Introduced by D. Knuth in 1984
- Essentially a programming language and a documentation system
- ns-3 uses C++ and Doxygen, thus it follows the literate programming ideas
- How to introduce the concept of literate programming to experiments?
- R is a programming language
- Markdown is a documentation language
- knitr is an R package that ties R and Markdown and produces a report
- R executes ns-3 using shell commands and parses stdout and stderr
Simple experiment

- $r = \{1.0\, \text{Mb/s}, 2.0\, \text{Mb/s}, 3.0\, \text{Mb/s}\}$;
- $b = \{10\, \text{ms}, 55\, \text{ms}, 100\, \text{ms}\}$;
- queue length varies from 70 to 100 with the step of 5 packets.

R Markdown source code is available at

https://github.com/dchaly/stateless/ns-3.20/ns3-stateless-report.Rmd
Experimental evaluation

- TCP Reno
- Trickles

Queue size, packets vs. Performance, MB

Queue size, packets: 70, 80, 90, 100

Performance, MB: 0, 1, 2, 3, 4, 5, 6, 7, 8

Queue size, packets (1 Mb): 70, 80, 90, 100

Queue size, packets (2 Mb): 70, 80, 90, 100

Queue size, packets (3 Mb): 70, 80, 90, 100
Future work

- More model evaluations, testing and debugging
- Develop new stateless protocols
- Performance analysis of stateless protocols in different settings: wired, wireless, SDN etc.