Exploring and Investigating TCP Traffic Flow Control and Congestion Control by Using Network Simulator (NS-3)

Jonathan Kua
4240189@student.swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Overview

- Introduction to Network Simulator 3 (NS-3)
- Dynamics of Transmission Control Protocol (TCP)
  - TCP Flow Control
  - TCP Congestion Control
- Experimental Work: Simulation in NS-3
- Results and Data Analysis
- Thoughts on Internship
Network Simulator 3 (NS-3)

- NS-3 is an open source, discrete event network simulator
- Software architecture is built on C++, object oriented with integration of various protocols and models
- Basic user paradigm:
  - Getting NS-3 source code – Two ways
  - Integration: Run build system with build.py, bake, ./waf.
- Detailed tutorials and API documentation at www.nsnam.org

NS-3 Key Features and Architecture

- Simulations performed on an abstract model and models are represented in C++ by classes
- Key abstractions in NS-3
  - Node, Application, Net Device, Channel, Topology Helpers
- Sophisticated simulation features
  - Tracing system, callbacks, logging, pcap output
  - Network Simulation Cradle (NSC): Uses real world protocol stacks
Network Architecture in NS-3

Dynamics of TCP

- **TCP Goal:** Reliable end-to-end delivery of byte stream
- **ACK self clocking** – Regulates TCP traffic flow
- **Flow Control:** Sender Window (swnd) = min (Sender Buffer, Receiver Advertised Window (awnd))
- **Congestion Control:** swnd = min (Congestion Window (cwnd), awnd)
Bandwidth Delay Product

- Throughput is bounded by the bottleneck link of a network path
  - Throughput = Window Size / RTT
  - Max throughput = Bottleneck bandwidth
  - Max Window = Bandwidth x RTT = BDP

Bandwidth Delay Product (BDP)

- Perfect synchronization, full utilization of network capacity
- Sender's window size >= BDP for full utilization

TCP Congestion Control

- Prevent Congestion Collapse (1986)
- Goal: Achieve high network utilization without congestion
- Source-based control - Sender prevents overwhelming the network
- Defined in RFC 5681, it has four main parts:
  - Slow Start: Exponential Increase
  - Congestion Avoidance: Linear Increase (AIMD)
  - Fast Retransmit: Retransmission after 3 dupACKs
  - Fast Recovery: Window inflation and deflation
TCP Congestion Control

- Prevent Congestion Collapse (1986)
- Goal: Achieve high network utilization without congestion
- Source-based control - Sender prevents overwhelming the network

- Defined in RFC 5681, it has four main parts:
  - **Slow Start**: Exponential Increase
  - **Congestion Avoidance**: Linear Increase (AIMD)
  - **Fast Retransmit**: Retransmission after 3 dupACKs
  - **Fast Recovery**: Window inflation and deflation

---

**Experimental Work: Simulation in NS-3**

- **Sender**
  - 1 Gbps Link, Delay 0ms
  - 10.1.1.0/30

- **Queue**
  - Bottleneck Dumbbell Topology

- **Router**
  - 10 Mbps Link, Variable Delay
  - 10.2.1.0/30
  - 10.3.1.0/30

- **Receiver**
  - 10.3.1.0/30

---

Bottleneck Dumbbell Topology
Results and Data Analysis – Scenario 1

![Graph showing Congestion Window and Queue Graph of Delay10ms and MaxPackets100](graph.png)

- **Congestion Window and Queue Graph**
- **Delay10ms and MaxPackets100**
- **Graph Details:**
  - Cwnd flow
  - Enqueue Flow
  - Packet Drop

---

Results and Data Analysis – Scenario 1

![Graph showing Congestion Window and Queue Graph (1.5–3s) of Delay10ms and MaxPackets100](graph2.png)

- **Congestion Window and Queue Graph**
- **Delay10ms and MaxPackets100**
- **Graph Details:**
  - Slow Start
  - Fast Retransmit/
  - Fast Recovery
  - Congestion Avoidance

---

8 August 2013
Results and Data Analysis – Scenario 2

Congestion Window and Queue Graph of Delay25ms and MaxPackets500

- Zoom In
  - Congestion Epoch

- cwnd grows negatively – Why?
  - Multiple packet losses
Issues Encountered

- NS-3 does not support window scaling
  - Hack TCP stack to “deceive” sender
- No TCP traffic when send size is greater than socket buffer
  - Does not behave like real world application
- Manual settings
  - Manual TCP tuning, set Slow Start Threshold (ssthresh), buffer sizes, window values, initial cwnd, queue sizes

Thoughts on Internship

- My winter internship at CAIA is an enriching and wholesome learning experience - Great exposure to the telecommunications and network engineering research field
- Developed new technical skills
  - UNIX Operating System - FreeBSD
  - Network Simulator 3 (NS-3)
  - R - Statistical Computing and Graphing Tool
  - GDB Debugger
  - Lyx, Latex and Apache Open Office
Thoughts on Internship

- Enhanced soft skills and qualities
  - Research skills
  - Greater confidence in exploring new things and solving problems
  - Stronger analytical thinking skills
  - Technical report writing and presentation skills
  - Patience and discipline
- Absolutely invaluable and rewarding experience

Conclusion

- Appreciation for the wonder of TCP Flow Control and Congestion Control mechanisms
  - Slow Start and Congestion Avoidance
  - Fast Retransmit and Fast Recovery
  - Analysis of cwnd behavior
- NS-3 as a versatile network simulation tool
  - Provides ideal network simulation environment
  - Simulations results are accurate and reliable
- Appreciation for research work
A Big Thank You

I would like to express my heartfelt appreciation for all CAIA staff who provide support throughout my internship.

Thank you Lawrence Stewart for your supervision, incredible patience and encouragement.

Thank you Prof. Grenville Armitage, Dr. Philip Branch and Dr. Jason But for giving me this opportunity to undertake this internship.

Thank you fellow intern Michal Scigocki and Djuro Mirkovic for your support and help.

Useful Resources (NS-3)

- NS-3 Source Code
  - http://code.nsnam.org/

- NS-3 Tutorial

- NS-3 Documentation
Question and Answer

Questions?

Thank You

Thank You!