University of South Carolina

- Founded in 1801, University of South Carolina (USC) is the flagship institution of the University of South Carolina System
- More than 350 programs of study, leading to bachelor’s, master’s, and doctoral degrees
- Total enrollment of approximately 50,000 students, with over 34,000 on the main Columbia campus as of Fall 2017
University of South Carolina

- Founded in 1801, University of South Carolina (USC) is the flagship institution of the University of South Carolina System
- More than 350 programs of study, leading to bachelor’s, master’s, and doctoral degrees
- Total enrollment of approximately 50,000 students, with over 34,000 on the main Columbia campus as of Fall 2017
University of South Carolina

- The College of Engineering and Computing includes:
  - Integrated Information Technology (IIT)
  - Computer Science
  - Electrical Engineering
  - Mechanical Engineering
  - Aerospace Engineering
  - Biomedical Engineering
  - Chemical Engineering
  - Civil and Environmental
University of South Carolina

- Other facts
- Countless extra curricular activities
- ~2 hours to the most beautiful beaches in the U.S.
- One of the best athletics in the country
Introduction to Science DMZ

• Science and engineering applications are now generating data at an unprecedented rate
• From large facilities to portable devices, instruments can produce hundreds of terabytes in short periods of time
• Data must be typically transferred across high-latency WANs

The Energy Science Network (ESnet) is a backbone connecting U.S. national laboratories, Internet2, research centers.
Enterprise Network Limitations

- Security appliances (IPS, firewalls, etc.) are CPU-intensive
- Inability of small-buffer routers/switches to absorb traffic bursts
- Even a small packet loss rate reduces throughput
- At best, transfers of big data may last days or even weeks

---

Science DMZ

- The Science DMZ is a network designed for big science data\(^1\)
- Main elements
  - High throughput, friction free WAN paths (no inline security appliances, routers / switches w/ large buffer size)
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security = Access-control list + offline appliance/s (IDS)

---

Science DMZ

- The Science DMZ is a network designed for big science data

- Main elements
  - High throughput, friction free WAN paths (no inline security appliances, routers / switches w/ large buffer size)
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security = Access-control list + offline appliance/s (IDS)
Science DMZ

- The Science DMZ is a network designed for big science data
- Main elements
  - High throughput, friction free WAN paths (no inline security appliances, routers / switches w/ large buffer size)
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security = Access-control list + offline appliance/s (IDS)
Science DMZ deployments as of 2016
RATE-BASED (BBR) VS WINDOW-BASED LOSS-BASED CONGESTION CONTROL: IMPACT OF MSS AND PARALLEL STREAMS ON BIG FLOWS

With Zoltan Csibi
BBR Brief Overview

- TCP BBR has been recently proposed as a congestion control algorithm (2016/17)\(^1\)
- BBR represents a disruption from the window-based loss-based congestion control used during the last decades\(^2\)
- BBR uses ‘pacing’ to try to match the bottleneck rate

MSS and Parallel Streams

- Two of the main features impacting big flows
  - Maximum segment size (MSS)
  - The use of parallel streams
MSS

• Large MSS produces a faster recovery after a packet loss

\[
\text{TCP throughput} = \frac{c \cdot MSS}{RTT \cdot \sqrt{p}}
\]

MSS: maximum segment size
RTT: round-trip time
p: loss rate
c: constant

Note: the above equation does not apply to BBR

Parallel Streams

- Opening parallel connections essentially creates a large virtual MSS on the aggregate connection

CP: Control process
DTP: Data transfer process
Scenario

- Sender/receiver connected by a 10 Gbps path, 20 ms RTT, running CentOS 7
- Memory-to-memory tests using iPerf3
- Network Emulator (Netem) used to adjust loss rate
- At 20 ms RTT, throughput already collapses when subject to a small loss rate
Scenario

• Each experiment lasted 70 seconds (first 10 seconds were not taken into account)
• For each test condition, ten experiments were conducted and the average throughput was computed
Results

1. Single Stream - Corruption Rate: 0.001%
   - Throughput (Gbps) vs. MSS (Bytes)
   - Green line: BBR
   - Blue line: HTCP
   - Black line: Cubic
   - Red line: Reno

2. Single Stream - Corruption Rate: 0.001%
   - Normalized Throughput (%)
   - Blue line: HTCP
   - Black line: Cubic
   - Red line: Reno

3. Ten Streams - Corruption Rate: 0.001%
   - Throughput (Gbps) vs. MSS (Bytes)
   - Green line: BBR
   - Blue line: HTCP
   - Black line: Cubic
   - Red line: Reno

4. Ten Streams - Corruption Rate: 0.001%
   - Normalized Throughput (%)
   - Blue line: HTCP
   - Black line: Cubic
   - Red line: Reno
Results

1. Single Stream - Corruption Rate: 0.01%
   - Throughput vs. MSS (Bytes)
   - BBR, HTCP, Cubic, Reno

2. Single Stream - Corruption Rate: 0.01%
   - Normalized Throughput (%)
   - HTCP, Cubic, Reno

3. Ten Streams - Corruption Rate: 0.01%
   - Throughput vs. MSS (Bytes)
   - BBR, HTCP, Cubic, Reno

4. Ten Streams - Corruption Rate: 0.01%
   - Normalized Throughput (%)
   - HTCP, Cubic, Reno
Results

- When not limited by network bandwidth, parallel streams improved BBR’s throughput by more than a factor of 3.
- The improvement factor for loss-based CC is lower.
- When parallel streams are used, the performance of HTCP, Cubic, and Reno are similar.
TRAFFIC CHARACTERIZATION USING NETFLOW
Motivation

• Offline scalable security appliances are required in Science DMZs
• Flow statistics can be available
• Flow-based Intrusion Detection System (IDS) is more scalable than payload-based IDS\(^1\)
• Goal: characterize normal traffic behavior by using flow information only (e.g., IPs, ports, transport protocol)

---
Motivation

- One approach for flow characterization is to measure the *randomness* or *uncertainty* of elements of a flow.
Motivation

- One approach for flow characterization is to measure the *randomness* or *uncertainty* of elements of a flow.

External port: low uncertainty; most external ports expected to be 80 (http)
### Motivation

- Entropy provides a measure of randomness or uncertainty.
- For a variable $X$, entropy of $X = \sum_{x \in X} p_x \log_2 \left( \frac{1}{p_x} \right)$.
- For the previous port example, let $X$ be the variable indicating the external port.

\[
X = \begin{cases} 
80 \text{ with probability } p_1 = \frac{5}{6} \\
22 \text{ with probability } p_2 = \frac{1}{6}
\end{cases}
\]
Motivation

- Entropy provides a measure of randomness or uncertainty.
- For a variable $X$, entropy of $X = \sum_{x \in X} p_x \log_2 \left( \frac{1}{p_x} \right)$.
- For the previous port example, let $X$ be the variable indicating the external port.

$$X = \left\{ \begin{array}{l}
80 \text{ with probability } p_1 = \frac{5}{6} \\
22 \text{ with probability } p_2 = \frac{1}{6}
\end{array} \right.$$ 

Entropy External Port $= \sum_{i=1}^{2} p_i \log_2 \left( \frac{1}{p_i} \right) = \frac{5}{6} \log_2 \left( \frac{1}{\frac{5}{6}} \right) + \frac{1}{6} \log_2 \left( \frac{1}{\frac{1}{6}} \right) = 0.65$
Motivation

- Entropy provides a measure of randomness or uncertainty
- For a variable $X$, entropy of $X = \sum_{x \in X} p_x \log_2 \left( \frac{1}{p_x} \right)$
- For the previous port example, let $X$ be the variable indicating the external port

$$X = \begin{cases} 
80 \text{ with probability } p_1 = \frac{5}{6} \\
22 \text{ with probability } p_2 = \frac{1}{6}
\end{cases}$$

Entropy External Port

$$= \sum_{i=1}^{2} p_i \log_2 \left( \frac{1}{p_i} \right) = \frac{5}{6} \log_2 \left( \frac{1}{\frac{5}{6}} \right) + \frac{1}{6} \log_2 \left( \frac{1}{\frac{1}{6}} \right) = 0.65$$

- 0 entropy -> no uncertainty (e.g., all external ports are 80)
- 1 entropy -> random -> high uncertainty
Scenario

- Small campus network ~15 buildings
- Inbound traffic is used as a reference (external IP address is in the Internet, campus IP address is in campus)
- The collector organizes flow data in five-minute time slots

NSF # 1541352, “Northern's Network Expansion for Large Science and Engineering Data Flows.”
External IP
- In general, high entropy, ‘many’ external IP addresses
- External IPs dispersed in the Internet
- Abnormal low entropy points
- Entropy near zero (no uncertainty of the external IP address), or ‘very low’ level (few external IP addresses dominate the distribution)

External port
- Higher entropy during the night, weekends
- Low entropy during the day, noon
- Large volume of http flows when students are on campus (less uncertainty/entropy on external port)
- Abnormal high entropy points
Results

Campus IP
- In general, low entropy, ‘few’ IP addresses on campus
- Higher entropy on weekends and at night
- Lower entropy when students are on campus
- A handful of public IP addresses used for regular Internet connectivity (network address translation)

Campus Port
- Lower entropy at night
- High entropy (close to uniform distribution) at noon
- Dynamic ports used by browsers when students connect to the Internet
- Abnormal low entropy points
Results

- Anomalies are detected by a single feature or by correlating multiple features
- E.g., event I: low campus port’s entropy, high external port’s entropy, low external IP’s entropy
## Results

- Correlation of entropy time-series

<table>
<thead>
<tr>
<th></th>
<th>Campus IP</th>
<th>Campus port</th>
<th>External IP</th>
<th>External port</th>
<th>Total traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Weekday</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-tuple</td>
<td>0.23</td>
<td>0.1</td>
<td>0.6</td>
<td>-0.02</td>
<td>-0.05</td>
</tr>
<tr>
<td>Campus IP</td>
<td>-0.85</td>
<td>0.6</td>
<td>0.89</td>
<td></td>
<td>-0.8</td>
</tr>
<tr>
<td>Campus port</td>
<td></td>
<td>-0.37</td>
<td>-0.98</td>
<td></td>
<td>0.78</td>
</tr>
<tr>
<td>External IP</td>
<td></td>
<td></td>
<td>0.45</td>
<td>-0.36</td>
<td></td>
</tr>
<tr>
<td>External port</td>
<td></td>
<td></td>
<td></td>
<td>-0.81</td>
<td></td>
</tr>
<tr>
<td><strong>Weekend</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-tuple</td>
<td>-0.23</td>
<td>-0.12</td>
<td>0.56</td>
<td>0.06</td>
<td>-0.03</td>
</tr>
<tr>
<td>Campus IP</td>
<td>0.15</td>
<td>-0.38</td>
<td>0.06</td>
<td>-0.38</td>
<td></td>
</tr>
<tr>
<td>Campus port</td>
<td></td>
<td>-0.48</td>
<td>-0.93</td>
<td>0.31</td>
<td></td>
</tr>
<tr>
<td>External IP</td>
<td></td>
<td></td>
<td>0.48</td>
<td>-0.05</td>
<td></td>
</tr>
<tr>
<td>External port</td>
<td></td>
<td></td>
<td></td>
<td>-0.39</td>
<td></td>
</tr>
</tbody>
</table>
FUTURE RESEARCH
Rate-based CC with P4 Switches

- BBR results indicate that rate-based congestion control (CC) can improve throughput
- BBR is still an end-to-end CC algorithm and uses implicit information (RTT)
- What if intermediate devices provide explicit feedback?
  - Queue’s length
  - Latency
  - Bandwidth usage
Rate-based CC with P4 Switches

- P4 is a programming language for switches, currently under standardization process
- Software-defined Networking (SDN) allows devices to program the control plane
- P4 switches permit to program the forwarding (data) plane
  - Add proprietary features: invent, differentiate, own
  - Telemetry and measurement
  - Reduce complexity

Barefoot’s Tofino (Dec. 2016)
Rate-based CC with P4 Switches

- My switch.p4
- Compiler
- Switch OS
- Driver
- Programmable Switch
Rate-based CC with P4 Switches

• What if rate at a sender node is adjusted based on feedback provided by a P4 switch?
• Engineers now have the capability of defining their own protocols, processed by a programmable P4 switch
• Feedback may include queue’s length, packet latency, and others
Rate-based CC with P4 Switches

• Many more opportunities…
  • New approaches to congestion control
  • New encapsulations and tunnels
  • New ways to tag packets for special treatment
  • New approaches to routing: e.g. source routing
  • New ways to process packets