On the Complexity of Traffic Traces and Implications

CHEN GRINER

CHEN AVIN, MANYA GHOBADI, CHEN GRINER, STEFAN SCHMID

“It is not the strongest of the species that survives, nor the most intelligent that survives. But the one that is most adaptable to change.”

(Leon C. Megginson)
Mapping a Landscape of Network Traffic

- Define
- Categorize
- Quantify
- *Map different types of structure*
- Better design of networks?
What is Structure?
Visible Structure?

• Can we see patterns in real network traffic?
  • Yes!

Rack-to-rack, Frontend cluster
FB @ SIGCOMM 2015
Structure: Two Types

- Temporal
- Non temporal(spatial)
ML Example: Non temporal structure

- A traffic matrix of a distributed ML application
- Source destination denoted by color
- Height denotes amount of traffic
- How would a matrix without structure look like?
Temporal structure

- Not all structure is related to frequency

- *Temporal* structure, represents the dependency of future events on recent events
  
  - Example: bursts of traffic
  
  - Both traces have the same traffic matrix but are different in time
How to Measure Structure
An information theoretic perspective

• In Information theory entropy (entropy rate) is a measure “randomness”

• Lower entropy ⇒ More predictable traffic

• Traffic with more “structure” is less “complex”

• Compression offers a way to estimate entropy

• But how to represent traffic?
### Traffic as a Network Trace

- A Simplified **time ordered** list of source-destination pairs
- How to use it?

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>(192.168.1.3)=s_0</td>
<td>(192.168.1.1)=d_0</td>
</tr>
<tr>
<td>s_1</td>
<td>d_1</td>
</tr>
<tr>
<td>s_2</td>
<td>d_2</td>
</tr>
<tr>
<td>s_3</td>
<td>d_3</td>
</tr>
<tr>
<td>s_4</td>
<td>d_4</td>
</tr>
<tr>
<td>s_5</td>
<td>d_5</td>
</tr>
<tr>
<td>s_6</td>
<td>d_6</td>
</tr>
</tbody>
</table>
Methodology

• Measuring complexity with two steps:

1. Sequentially randomize a trace, remove a specific types of structure

2. Compress the trace, compare their size.
Systematic randomization

A trace

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>$d_0$</td>
</tr>
<tr>
<td>$s_1$</td>
<td>$d_1$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$d_2$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>$d_3$</td>
</tr>
<tr>
<td>$s_4$</td>
<td>$d_4$</td>
</tr>
<tr>
<td>$s_5$</td>
<td>$d_5$</td>
</tr>
<tr>
<td>$s_6$</td>
<td>$d_6$</td>
</tr>
<tr>
<td>$s_7$</td>
<td>$d_7$</td>
</tr>
<tr>
<td>$s_8$</td>
<td>$d_8$</td>
</tr>
<tr>
<td>$s_9$</td>
<td>$d_9$</td>
</tr>
<tr>
<td>$s_{10}$</td>
<td>$d_{10}$</td>
</tr>
</tbody>
</table>
Systematic randomization

Increasing complexity

A trace

Original trace $\sigma$
- ✓ Temporal
- ✓ Non Temporal

Row Randomized $\Gamma(\sigma)$
- X Temporal
- ✓ Non Temporal

Uniform Trace $u(\sigma)$
- X Temporal
- X Non Temporal
Compression
Formal Definitions of Trace Complexity

Let $c(\sigma)$ is the size of a compressed trace $\sigma$:

We define:

Total complexity: $\psi(\sigma) = \frac{c(\sigma)}{c(u(\sigma))}$

Temporal complexity: $T(\sigma) = \frac{c(\sigma)}{c(\Gamma(\sigma))}$

Non-temporal complexity: $NT(\sigma) = \frac{c(\Gamma(\sigma))}{c(u(\sigma))}$

Complexity is in the range of $[0 ... 1]$
Mapping Trace Complexity
The Complexity Map

- **X axis**: temporal complexity
- **Y axis**: non-temporal complexity
- **Rule of thumb**: Closer to the axis’s origin, means lower complexity
The Complexity Map

- **Uniform Traffic**
  - Lacking any structure, maximal entropy

- **Bursty Traffic**
  - Has temporal correlations

- **Skewed Traffic**
  - From a skewed distribution

- **Skewed & Bursty Traffic**
  - Has both temporal and non-temporal elements
Case Study

- ML
- Facebook
- Database, Web, Hadoop
- High Power Computing (HPC)
- pFabric
What can we expect to learn from trace complexity?

- Lower complexity means better optimization
- Identify and quantify different structures
- Compare different traces?
- Differentiate between different workloads?
Future work

• Test trace with more metadata
  • Interarrival times, ports etc

• What are the other dimensions of complexity?

• Practical implementation of complexity in online algorithms?

• Trace website:
  • https://self-adjusting.net

• Further details are found in the paper:
  On the Complexity of Traffic Traces and Implications.
  Chen Avin, Manya Ghobadi, Chen Griner, Stefan Schmid. Sigmetrics 2020