Simple Network Performance Tomography

Nick Duffield
AT&T Labs-Research, Florham Park, NJ
Network Performance Tomography

- Basic Idea
  - Infer performance of interior links from end-to-end measurements

- Setup
  - Mesh of intersecting network paths
  - Measure end-to-end performance along each path (loss, delay)
  - Correlate measurements to infer performance on common subpaths
Correlations and Network Probing

- Packet-level correlation by design
- Perform end-to-end measurements using probe traffic
- Couple probing on different network paths at the packet level
- Multicast-based tomography
- Send probes down multicast tree
- One-to-one map from link-performance to end-to-end performance
- Packet performance identical on common path
- Invert map infer link performance
- Example: Infer link loss rates from measure end-to-end loss rates

Ratnasamy and McCanne, MINC (AT&T, ICIR, UMass)
Weaker Correlations and Network Probing

- Unicast-based tomography
  - Emulate multicast probes: groups of closely spaced unicast probes
    - Striped across multiple destinations
  - Performance within group on common links is correlated
    - But not identical
    - Need additional model parameters to describe correlations
  - Many-to-one map from link-performance to end-to-end performance
  - Cannot simply invert to perform inference: too many link parameters

- Two approaches to unicast inference
  1. If correlations strong within group: use multicast inference methods
     - Tailor selection of probes to enhance correlation
     - Verify correlation strength through direct measurement
     - UINC Project (as for MINC)
  2. Couple unknown parameters through queueing model
     - Incite Project (Rice)
Tomography Without Strong Packet Correlations?

- Previous focus
  - Inferring packet loss, delay, per link

- Simpler question
  - Which are the badly performing links?

- Padmanabhan et. al. Infocom 2003
  - Simple end-to-end loss measurements down source tree
    - Determined from measured TCP retransmissions from web server
  - No packet level correlation designed into measurements
  - Apply several inference methods to end-to-end loss measurements

- Can identify badly performing links quite well
  - Certain performance models
  - Better when bad links are rarer
Motivation and Summary

- How can inference of worst links work w/o packet correlations?
- First component: link performance model
  - Separable performance: disjoin between “good” and “bad” links
  - Investigate conformance with separable performance model
- Second component: assume bad links are rare
  - Easier to reliably identify with low false positive rate
Separable Link Performance Model

- Associate performance metric \( \varphi_i \) with each link, path \( i \)
  - e.g. parameter or quantile of distribution of loss or delay
  - Measurement interval may be part of definition

- Separate possible values of \( \varphi_i \) into two disjoint subsets:
  - Call these subsets “good” and “bad”

- Separable performance model assumption
  - A path is bad if and only if contains at least one bad link
  - Cannot make a bad path out of “partially” bad links
Examples of Separable Performance

- **Example 1 (trivial)**
  - Performance metric is packet loss
  - Links either lose no packets (= good), or lose all packets (= bad)

- **Example 2 (Padmanabhan et. al., Infocom 2003)**
  - Source tree: links of two types:
    - Good: loss uniformly distributed in between 0 and 1%
    - Bad: loss uniformly distributed between 5% and 10%
  - Separable if depth of (logical) tree ≤ 5

- **Example 3 (Delay spike measurements: Zhang et. al., IMC 2001)**
  - End-to-end measurements exhibit delay spikes
    - Periods of greatly elevated delay
    - "Bad": if max delay encountered by stream of packets ≥ (high) threshold

  **Requirements for (approximately) separable performance**
  - Assume spikes due to delays on one or more links
  - High chance for probe stream to encounters delay spike on bad link
  - Small chance for packet to encounter delay spike on multiple links

  **Satisfied reasonably closely in data for realistic probe set**
  - ~a few minutes audio transfer
Separability and Correlations

- Separability equiv. to exact correlation of multicast loss model
- Compare:
  - Separable: any path through a bad link is bad
  - Multicast: packet lost on any end-to-end path if lost on a common link
- Consequence
  - Model: link $i$ is independently bad with probability $\alpha_i$
  - Measure frequencies $\gamma_i$ of end-to-end badness over multiple intervals
  - Infer $\{\alpha_i\}$ from $\{\gamma_i\}$ by usual multicast inference algorithm (e.g. MINC)
- Limitation
  - Still need to coordinate measurements temporally
    - Multiple aligned measurement intervals for all paths.
  - Desirable to infer from measurements over single interval
Identification of Rare Bad Links

- Suppose bad links are rare:
  - intersecting bad paths most likely have common bad link

- Example:
  - Links independently bad with probability $p \approx 0$

- Suppose paths (12) and (13) both bad
  - Conditional probability that link 1 is bad
    
    \[
    \frac{p}{p + (1 - p)q^2} \approx 1 - p
    \]

  - Overwhelming likely for link to be bad

- Inference: identify link 1 as bad

- Strength:
  - low false positive rate

- Weakness: coverage
  - Suppose link 1 good, links 2,3, bad: don't detect this case.

duffield@research.att.com
Smallest Consistent Failure Set (SCFS) Algorithm

- Data: single snapshot of end-to-end performance on source tree
  - each path is good or bad
- Determine smallest set of links consistent with observed badness
  - Those nearest to root

Measured data
Good paths have all good links

After using SCFS
Attribute common loss to common cause
SCFS Performance

- Model: link $i$ independently bad with probability $p_i$
- Model amenable to direct analysis
- Performance Measures
  - False positive rate
  - Coverage (= proportion of bad links identified)
SCFS False Positive Rate

- Calculate uniform bounds of false positive rate
  - As function of 1 - maximum probability of link badness: \( \alpha = 1 - \max_i p_i \)
  - Depends on topology only through (minimum) branching ratio \( r \)

- Decreases rapidly as branching ratio increases
- Can be applied to subtrees with larger \( r, \alpha \)
SCFS False Positive Rate (2)

- Computations on selection of 1000 node trees,
  - Random branching ratios between 2 and 10

<table>
<thead>
<tr>
<th>p = Prob(link bad)</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>0.02%-0.08%</td>
</tr>
<tr>
<td>10%</td>
<td>0.09%-0.24%</td>
</tr>
<tr>
<td>20%</td>
<td>0.4%-0.9%</td>
</tr>
</tbody>
</table>
**SCFS Coverage: Dependence on Branching Ratio**

- Coverage relatively insensitive to branching ratio
SCFS Coverage: Dependence on Tree Depth

- Coverage decreases with depth
  - Coverage $\approx p / \text{Tree Depth}$: when $p = \text{Prob(Link Bad)} \approx 0$
SCFS Coverage: Constant Path Failure Rate

- Trees will not be arbitrarily deep for constant link badness
  - Otherwise all paths become unreliable
- Opposite: scale $p$ to keep path failure rate constant
  - Coverage insensitive to depth in this scaling
Comparisons

- SCFS Computationally very simple
- Comparison
  - with three reference methods in Padmanabhan et. al.
  - compared with published results, not reimplementation
- False positive rate:
  - seems at least as good in all cases
- Coverage:
  - nearly as good if badness rare, (less than 10%)
  - noticeably worse than simple reference methods if less rare
Summary

- Tomography and Correlations
  - Many existing methods depend on intricate probing methods
  - To arrange for packet level correlations in probes

- Separable Performance Model
  - Enables exploitation of correlations and probe streams; less intricate
  - Applicable to model in literature, observed network performance

- Inference from one time measurements
  - Smallest consistent failure set algorithm
  - Very low false positive rate to identify bad links, especially if rare

- Models very amenable to analysis

- Further work: need simulation/experimental evaluation