Network Troubleshooting from End-Hosts

Renata Teixeira
Laboratoire LIP6
CNRS and UPMC Sorbonne Universités
Internet problems are hard to troubleshoot

Is my connection ok?

Is it google?

Is the problem in one of the networks in path?

DNS? NAT? DHCP? ???

Net1

Stanford network

Net2

Net3
Nobody has the full picture to diagnose problems

- End-users can’t know what happens in network
- Network operators can’t know user experience
End-to-end measurements to rescue

- Available data depends on who you are
  - Network operators: Data from their own network
  - End-users: Data from their machine

- End-to-end measurements compensate for missing data
  - Network operators: Deploy monitoring hosts
  - End-users: Monitor some paths and cooperate with other users

- Troubleshooting is mostly manual and ad-hoc
Goal

Automatically troubleshoot network faults and performance disruptions

- Fault identification: where is the problem?
  - Most common: traceroute
  - Assist in pinpointing the location of the problem
  - How to measure to obtain accurate location?

- Automatic detection: is there a problem?
  - Detect problems before users
  - Fast detection requires continuous monitoring
  - How to monitor with less overhead?
Overview

- Identification
  - Paris traceroute
  - Network tomography for fault diagnosis

- Detection
  - Minimizing active probing cost
  - Performance problems as perceived by users

- End-host measurement platform
  - Fathom: A browser-based platform
Paris traceroute

B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien (LIP6) D. Veitch (U. Melbourne)
Traceroute

**Actual path**

TTL exceeded from B.1

TTL exceeded from A.1

TTL = 1

Inferred path

TTL = 2
Traceroute under load balancing

Actual path

TTL = 2

Inferred path

TTL = 3

Missing nodes and links

False link
Errors happen even under per-flow load balancing

- **Traceroute uses the destination port as identifier**
  - Needs to match probe to response
  - Response only has the header of the issued probe
Paris traceroute: Removing false links

- Solves the problem with per-flow load balancing
  - Probes to a destination belong to same flow

- Changes the location of the probe identifier
  - Use the UDP checksum

![Diagram showing traceroute process with TTL and checksum values.]

**Node A**: TTL = 2, Port 1, Checksum 2

**Node B**: Port 1

**Node C**: TTL = 3, Port 1, Checksum 3

**Node D**

**Node E**: t
Paris traceroute: Tracing all the paths

- Adaptive probing strategy
  - Vary flow id of probes
  - Send enough probes to enumerate all interfaces with high confidence
Implications

- Paris traceroute used for topology mapping
  - More accurate routes

- Identification with Paris traceroute
  - Forwarding loops and some reachability problems
  - Traceroute identifies the effect, not the cause
Network tomography for fault diagnosis
Fault diagnosis in collaborative environments

- Faults are persistent reachability problems
- Correlate measurements from multiple end-hosts
Binary tomography

- **Given**
  - Complete network topology
  - Path reachability

- **Find the smallest set of links that explains bad paths**
  - If link is bad, all paths that cross the links are bad
  - Given bad links are uncommon

[Duffield, 2006]
Practical issues with tomography

- **Given**
  - Complete network topology
  - Path reachability
- **Find the smallest set of links that explains bad paths**
  - If link is bad, all paths that cross the links are bad
  - Given bad links are uncommon

- Often unknown
- Hard to measure correlated reachability
- Links can fail for some paths, but not for all
Measurement methods for tomography

- Methods to measure path reachability
  - Failure confirmation
  - Failure aggregation

- Methods to track topology

- NetDiagnoser: A binary tomography algorithm
  - Handles path-specific failures (e.g., misconfigurations)
  - Takes advantage of routing information
Predicting and tracking Internet path changes

with
I. Cunha (UPMC/Technicolor)
D. Veitch (U. Melbourne)
C. Diot (Technicolor)
Challenges of tracking topology with traceroute

- Cannot measure fast enough to detect all changes
  - Network and system limitations

- Accurate measurements require extra probes
  - Identify all paths under load balancing
Frequent vs. accurate measurements

- Paris traceroute: High frequency, high accuracy
- Traceroute: Low frequency, low accuracy
- Tracetree: High frequency, low accuracy
- Doubletree: Low frequency, high accuracy
Approach

- Observation: Internet paths are mostly stable
  - Current techniques waste probes
- Probe according to path stability
- Separate tasks
  - Change detection: lightweight probing for speed
  - Path remapping: accuracy with Paris traceroute
Contributions

- **NN4: Predicting Internet path changes**
  - Distinguish between stable and unstable paths

- **DTrack: Tracking Internet path changes**
  - Lightweight probing process to detect changes
  - Allocates more probes to unstable paths
Predicting path changes

- Prediction goals
  - Time until the next change
  - Number of changes in a time interval
  - Whether a path will change in a time interval

- Identify features that can help with prediction
  - Features must be computable from traceroutes
    - Characteristics of the current path
    - Characteristics of the last path change
    - Behavior of the path in the recent past
Feature selection

- RuleFit to identify feature importance
  - Fraction of time path active in the past (prevalence)
  - Number of changes in the past
  - Number of previous occurrences of the current path
  - Path age

- Four most important features carry all the predictive information
NN4 predictor

- RuleFit is hard to integrate in other systems
- NN4 is based on the nearest-neighbor scheme
  - Compute neighbors by partitioning the path feature “state-space”
  - Prediction computed as the average behavior of all neighbors
NN4: summary

- NN4 is lightweight, easy to integrate, and as accurate as RuleFit
- Prediction is not highly accurate
  - It is possible to distinguish unstable/stable paths
DTrack

- Goal: Given a probing budget, detect as many changes as possible
- Allocates probing rates *per path* using NN4’s predictions
- Targets probes along each path
  - Reduce redundant probes at shared links
  - Spread probes over time
Probe rate allocation

- Allocate rates that minimize number of missed changes
- Model changes in each path as a Poisson process
  - Estimate the rate of changes using NN4
- Compute missed changes as function of probing rate
Probe targeting overview
DTrack summary

- Detects more changes than optimized traceroute
  - 2.2 times more changes in trace-driven simulations
  - 5 times more changes in PlanetLab
Summary

Focus: Two steps of network troubleshooting
- Identification of fault location
- Detection of faults and performance disruptions

Contribution: Measurement methods with practical assumptions
- To obtain accurate measurements
- To measure efficiently
Fathom: A browser-based network measurement platform

with

M. Dhawan (Rutgers), J. Samuel (UCB), M. Allman, C. Kreibich (ICSI)
V. Paxson (UCB/ICSI)
End-host measurements are challenging

- Measurement from end-hosts are vital
  - Researchers to understand Internet
  - Practitioners to diagnose user problems

- Hard to deploy measurements
  - Developers: Portability, safety
  - Users: need to install new software
A browser-based measurement platform

- Why browser?
  - Flexibility, deployability
  - Ubiquity of browser

- Fathom: Firefox extension
  - Measurement API in JavaScript
  - Web page performance
  - System performance
  - Active measurements
Thank you!

- **HostView**: [http://cmon.lip6.fr/EMD](http://cmon.lip6.fr/EMD)
  - Measure end-host performance with user feedback
  - Mac OS and Linux

  - One-shot test
  - Home network configuration
  - WiFi performance
  - Internet access performance