Measurement: Techniques, Strategies, and Pitfalls

David Andersen
CMU 15-744

Many (most) slides in this lecture from Nick Feamster's measurement lecture

Internet Measurement

• Process of collecting data that measure certain phenomena about the network
  – *Should be a science*
  – *Today: closer to an art form*

• **Key goal:** Reproducibility

• “Bread and butter” of networking research
  – *Deceptively complex*
  – *Probably one of the most difficult things to do correctly*
Types of Data

Active

- traceroute
- ping
- UDP probes
- TCP probes
- Application-level “probes”
  - Web downloads
  - DNS queries

Passive

- Packet traces
  - Complete
  - Headers only
  - Specific protocols
- Flow records
- Specific data
  - Syslogs …
  - HTTP server traces
  - DHCP logs
  - Wireless association logs
  - DNSBL lookups
  - …
- Routing data
  - BGP updates / tables, ISIS, etc.

Outline: Tools and Pitfalls

- Aspects of Data Collection
  - **Precision**: At what granularity are measurements taken?
  - **Accuracy**: Does the data capture phenomenon of interest?
  - **Context**: How was the data collected?

- Tools
  - **Active**
    - Ping, traceroute, etc.
    - **Accuracy pitfall example**: traceroute
  - **Passive**
    - Packet captures (e.g., tcpdump, DAG)
    - Flow records (e.g., netflow)
    - Routing data (e.g., BGP, IS-IS, etc.)
    - **Context pitfall example**: eBGP multihop data collection
Outline (continued)

• Strategies
  – Cross validate
    • consistency checks
    • multiple “overlapping” measurements
  – Examine Zeroth-Order

• Database as secret weapon

• Other considerations
  – Anonymization and privacy
  – Maintaining longitudinal data

Active Measurement

• Common tools:
  – Ping
  – traceroute
  – scriptroute (see homework)
Sample Question: Topology

- What is the topology of the network?
  - At the IP router layer
  - Without “inside” knowledge or official network maps
  - Without SNMP or other privileged access

- Why do we care?
  - Often need topologies for simulation and evaluation
  - Intrinsic interest in how the Internet behaves
    - “But we built it! We should understand it”
    - Emergent behavior; organic growth

How Traceroute Works

- Send packets with increasing TTL values

- Nodes along IP layer path decrement TTL
- When TTL=0, nodes return “time exceeded” message
Problems with Traceroute

- Can’t unambiguously identify one-way outages
  - Failure to reach host: failure of reverse path?

- ICMP messages may be filtered or rate-limited

- IP address of “time exceeded” packet may be the outgoing interface of the return packet

![Traceroute Diagram]

Famous Traceroute Pitfall

- **Question:** What ASes does traffic traverse?
- **Strawman approach**
  - Run traceroute to destination
  - Collect IP addresses
  - Use “whois” to map IP addresses to AS numbers

- **Thought Questions**
  - What IP address is used to send “time exceeded” messages from routers?
  - How are interfaces numbered?
  - How accurate is whois data?
More Caveats: Topology Measurement

- Routers have multiple interfaces
- Measured topology is a function of vantage points

**Example:** Node degree
- Must “alias” all interfaces to a single node (PS 2)
- Is topology a function of vantage point?
  - Each vantage point forms a tree
  - See Lakhina *et al.*
- *(preview of homework! :)*

Less Famous Traceroute Pitfall

- Host sends out a sequence of packets
  - Each has a different destination port
  - Load balancers send probes along different paths
  - Equal cost multi-path
  - Per flow load balancing

*Possible traceroute outcome:*

**Question:** Why won’t just setting same port number work?

Soule *et al.*, “Avoiding Traceroute Anomalies with Paris Traceroute”, *IMC 2006*
Designing for Measurement

- What mechanisms should routers incorporate to make traceroutes more useful?
  - Source IP address to “loopback” interface
  - AS number in time-exceeded message
  - ??

- More general question: How should the network support measurement (and management)?
Two Main Approaches

• Packet-level Monitoring
  – *Keep packet-level statistics*
  – *Examine (and potentially, log) variety of packet-level statistics. Essentially, anything in the packet.*
  – *Timing*

• Flow-level Monitoring
  – *Monitor packet-by-packet (though sometimes sampled)*
  – *Keep aggregate statistics on a flow*

Packet Capture: tcpdump/bpf

• Put interface in promiscuous mode
• Use bpf to extract packets of interest

Accuracy Issues

• Packets may be dropped by filter
  – Failure of tcpdump to keep up with filter
  – Failure of filter to keep up with dump speeds

*Question:* How to recover lost information from packet drops?
Traffic Flow Statistics

- **SNMP (Simple Network Management Protocol)**
  - Get # of packets across interface per 5min
  - or other similar very coarse stats

- **Flow monitoring (e.g., Cisco Netflow)**
  - Statistics about groups of related packets (e.g., same IP/TCP headers and close in time)
  - Records header information, counts, and time
  - May be sampled

What is a flow?

- **Source IP address**
- **Destination IP address**
- **Source port**
- **Destination port**
- **Layer 3 protocol type**
- **TOS byte (DSCP)**
- **Input logical interface (ifIndex)**
Flow Record Contents

Basic information about the flow...
- Source and Destination, IP address and port
- Packet and byte counts
- Start and end times
- ToS, TCP flags

...plus, information related to routing
- Next-hop IP address
- Source and destination AS
- Source and destination prefix

Aggregating Packets into Flows

- **Criteria 1:** Set of packets that “belong together”
  - Source/destination IP addresses and port numbers
  - Same protocol, ToS bits, …
  - Same input/output interfaces at a router (if known)

- **Criteria 2:** Packets that are “close” together in time
  - Maximum inter-packet spacing (e.g., 15 sec, 30 sec)
  - *Example:* flows 2 and 4 are different flows due to time
Packet Sampling

- Packet sampling before flow creation (Sampled Netflow)
  - 1-out-of-m sampling of individual packets (e.g., m=100)
  - Create flow records over the sampled packets
- Reducing overhead
  - Avoid per-packet overhead on (m-1)/m packets
  - Avoid creating records for a large number of small flows
- Increasing overhead (in some cases)
  - May split some long transfers into multiple flow records
  - … due to larger time gaps between successive packets

Problems with Packet Sampling

- Determining size of original flows is tricky
- Flow records can be lost
- Small flows may be eradicated entirely
- Flow sampling can provide better accuracy
  - But requires measuring every packet still
- Lots of research looking at sampling techniques, etc.
Routing Data

- IGP
- BGP
  - Collection methods
    - eBGP (typically “multihop”)
    - iBGP
  - Table dumps: Periodic, complete routing table state (direct dump from router)
  - Routing updates: Continuous, incremental, best route only

Why Trust Your Data?

- Measurement requires a degree of suspicion
  - Why should I trust your data? Why should you?
- Resolving that...
  - Use current best practices
    - e.g., paris-traceroute, CAIDA topologies, etc.
  - Don’t trust the data until forced to
    - Sanity checks and cross-validation
    - Spot checks (when applicable)
Context Pitfall: AS-Level Topologies

- **Question:** What is the Internet’s AS-level topology?
- **Strawman approach**
  - Routeviews routing table dumps
  - Adjacency for each pair of ASes in the AS path

- **Problems with the approach?**
  - **Completeness:** Many edges could be missing. Why?
    - Single-path routing
    - Policy: ranking and filtering
    - Limited vantage points
  - **Accuracy**
  - **Coarseness**

Context Pitfall: Routing Instability

- **Question:** Does worm propagation cause routing instability?
- **Strawman approach:**
  - Observe routing data collected at RIPE RIRs
  - Correlate routing update traffic in logs with time of worm spread
  - Finding: Lots of routing updates at the time of the worm spreading!
  - *(Bogus) conclusion:* Worm spreading causes route instability

**Missing/Ignored Context:** Instability + eBGP multihop …

Figure 5: A zoom-in on the BGP message storm of 18–22 September.

Cowie et al., “Global Routing Instabilities Triggered by Code Red II and Nimda Worm Attacks”
Strategy: Examine the Zeroth-Order

- Paxson calls this “looking at spikes and outliers”
- **More general:** Look at the data, not just aggregate statistics
  - *Tempting/dangerous to blindly compute aggregates*
  - *Timeseries plots are telling (gaps, spikes, etc.)*
  - *Basics*
    - Are the raw trace files empty?
      - Need not be 0-byte files (*e.g.*, BGP update logs have state messages but no updates)
    - Metadata/context: Did weird things happen during collection (machine crash, disk full, etc.)

---

Strategy: Cross-Validation

- Paxson breaks cross validation into two aspects
  - *Self-consistency checks (and sanity checks)*
  - *Independent observations*
    - Looking at same phenomenon in multiple ways
- What are some other examples of each of these?
Example Sanity Checks

- Is time moving backwards?
  - Paxson’s probing example
  - Typical cause: Clock synchronization issues

- Has the speed of light increased?
  - E.g., 10ms cross-country latencies

- Do values make sense?
  - IP addresses that look like 0.0.1.2 indicate bug

Cross-Validation Example

- Traceroutes captured in parallel with BGP routing updates

- Puzzle
  - Route monitor sees route withdrawal for prefix
  - Routing table has no route to the prefix
  - IP addresses within prefix still reachable from within the IP address space (i.e., traceroute goes through)

- Why?
  - Collection bugs … or
  - Broken mental model of routing setup: A default route!
Databases: Secret Weapon

• Easy way to get lots of summary statistics
  – Regular first-order stats (cf. Paxson’s recommendation)
    • Latest timestamp, number of updates, etc.
  – Cross-validation becomes easier (quick and dirty SQL)
  – Joint analysis of diverse datasets is a common need

• Caveats!
  – Insertion must be done properly
    • Always, always save raw data

Horror Story #1: Buggy Postprocessing

• Logs maintained at each host
• Files collected and merged to compute one-way delays
• If corresponding ends of logfile missing: set receive time to zero.

“Does the extra effort matter?” (Paxson)

• What if the log files don’t match up in time properly?
• What about missing log files?
Longitudinal measurement hard

- Accurate distributed measurement is tricky!
- Lots of things change:
  - Host names, IPs, software
- Lots of things break
  - hosts (temporary, permanently)
  - clocks
  - links
  - collection scripts
- Paxson's “master script” can help a bit

Anonymization

- Similar questions arise here as with accuracy
- Researchers always want full packet captures with payloads
  - …but many questions can be answered without complete information
- Common methods:
  - Nulling out low-order IP bytes
  - hashing IP addresses
- Privacy / de-anonymization issues
PlanetLab for Network Measurement

- Nodes are largely at academic sites
  - Other alternatives: RON testbed (disadvantage: smaller, less software support)

- Repeatability of network experiments is tricky
  - Proportional sharing
    - Minimum guarantees provided by limiting the number of outstanding shares
  - Work-conserving CPU scheduler means experiment could get more resources if there is less contention