Strategies for Sound Internet Measurement

Vern Paxson

ICSI Center for Internet Research (ICIR)
International Computer Science Institute
Berkeley, CA

vern@icir.org

October 26, 2004

Disclaimers:

- There are no new research results in this talk.
- Many of the problems discussed are mundane. Experienced measurement practitioners: feel free to work on your laptops.
- A number of the points also apply to Internet simulation, large-scale systems work in general.
- Unfortunately, just about all of the strategies involve extra work ("discipline").
- There's no easy answer to the question "how much extra work is merited?"

Strategic Areas:

- Errors & imperfections.
- Dealing with large volumes of data.
- Ensuring reproducible analysis.
- Community datasets.

Precision:

Precision: limit of a measurement device's resolution.

Consider a `tcpdump` timestamp:

```
1092704424.276251 IP 192.168.0.122.22 > 192.168.0
```

How precise is it?

Answer: at <u>most</u> to 1 μsec. But perhaps much less.

Precision, con’t:

Notion applies to discrete measurements, too.

How precise are the packets captured by `tcpdump`?

Depends:

- “Snapshot” length limits total data.
- *Filtering* does too.

Precision, con't:

If you look in a `tcpdump` trace file, you can determine:

- snapshot length (savefile header)

You can be told:

- timestamp precision (savefile header)

  . . . but it's wrong

You can't determine filtering.

Strategy #1: Maintain Meta-Data

- Identify auxiliary information necessary for soundness.
- Determine how to measure <u>it</u>.
- Devise a mechanism to keep it associated with measurements (e.g., database).
- Note: unfortunately, existing tools tend to be weak here.

* Of much broader relevance than just precision.

* Can have a lifetime way beyond initial measurement.

Accuracy: Measurement's Degree of Fidelity

Much broader problem than precision.

E.g., clocks can:

- be arbitrarily off from true time; jump forward or backward; fail to move; run arbitrarily fast or slow

E.g., packet filters can:

- fail to record packets ("drops"); fail to report drops; report drops that didn't occur; reorder packets; duplicate packets record the wrong packets

The problem of *misconception*:

Misconception: not measuring what you think you're measuring.

E.g., measuring packet loss by counting retransmissions.

E.g., measuring Web fetches that hit hidden caches.

E.g., `ttcp` with large socket buffers, small data volume.

E.g., computing TCP connection size based on SYN/FIN sequence difference.

E.g., Mark Allman's 10 msec to establish a TCP connection with a host 100 msec away, transfer data to it, close it down . . . but *the remote machine was powered off*!

Strategy #2: run your intended methodology by colleagues.

Calibration:

Goal: detect problems of loss of precision / limited accuracy / data reduction bugs / misconception.

Possible additional goal: adjust for these effects *post facto*.

Or: simply identify & remove tainted measurements (careful to consider bias).

Calibration, con't:

Strategy #3a: examine outliers and spikes

- *e.g., what's the biggest and smallest RTT, and why?*
- problems often manifest here
- easy to find

We can often detect measurement errors *if we have enough additional information*.

Calibration, con't:

Strategy #3b: employ self-consistency checks

E.g., protocol information:

– if a TCP receiver acknowledges data never sent, the packet filter must* have dropped the sent data.

($* =$ Or: the packet took another route. Or: the data was sent before you started measuring. Or: the TCP receiver is broken.)

Calibration, con't:

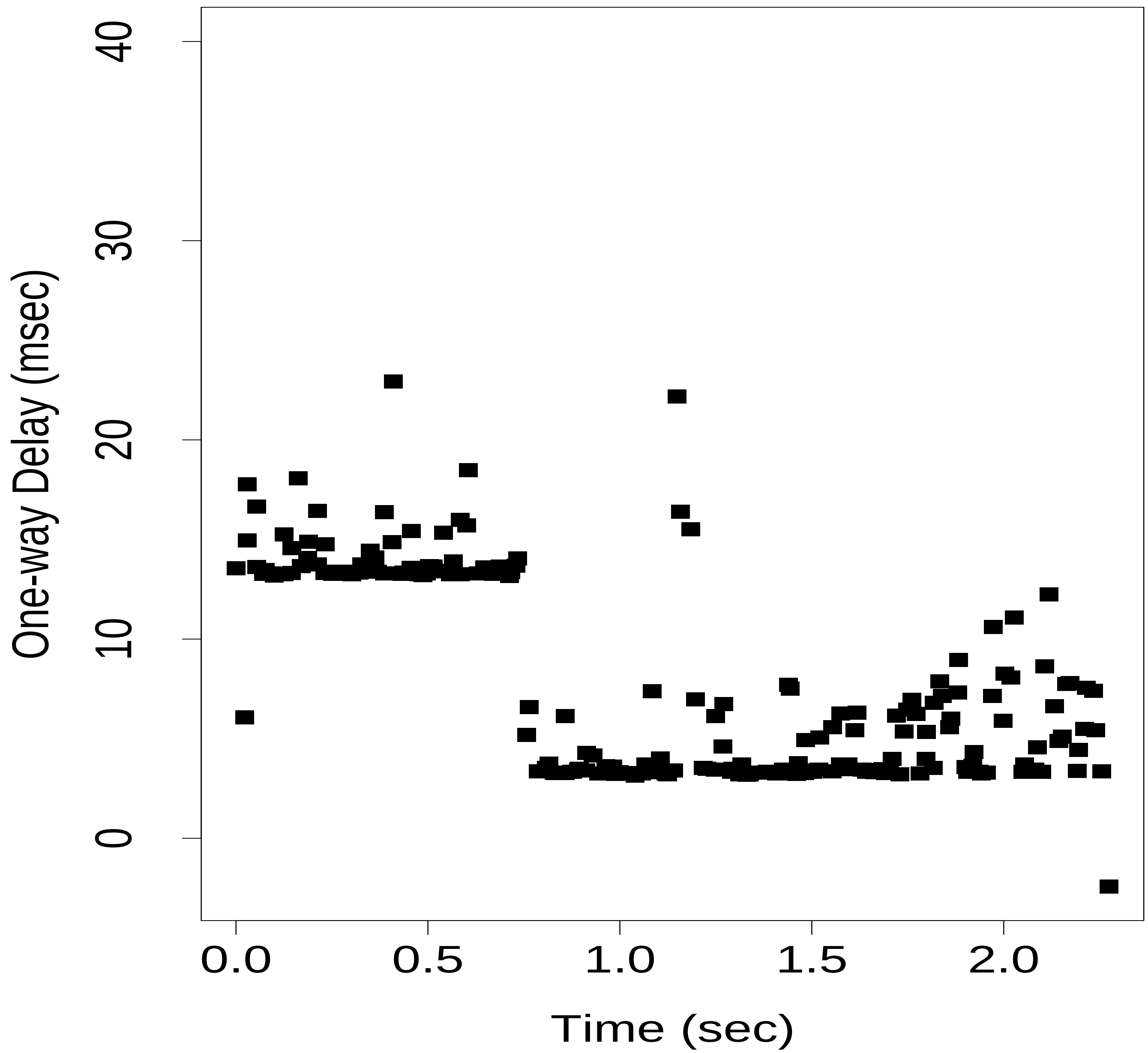
Strategy #3c: compare multiple measurements/computations.

E.g.: when tracing, compare monitor packet count vs. receiver's.

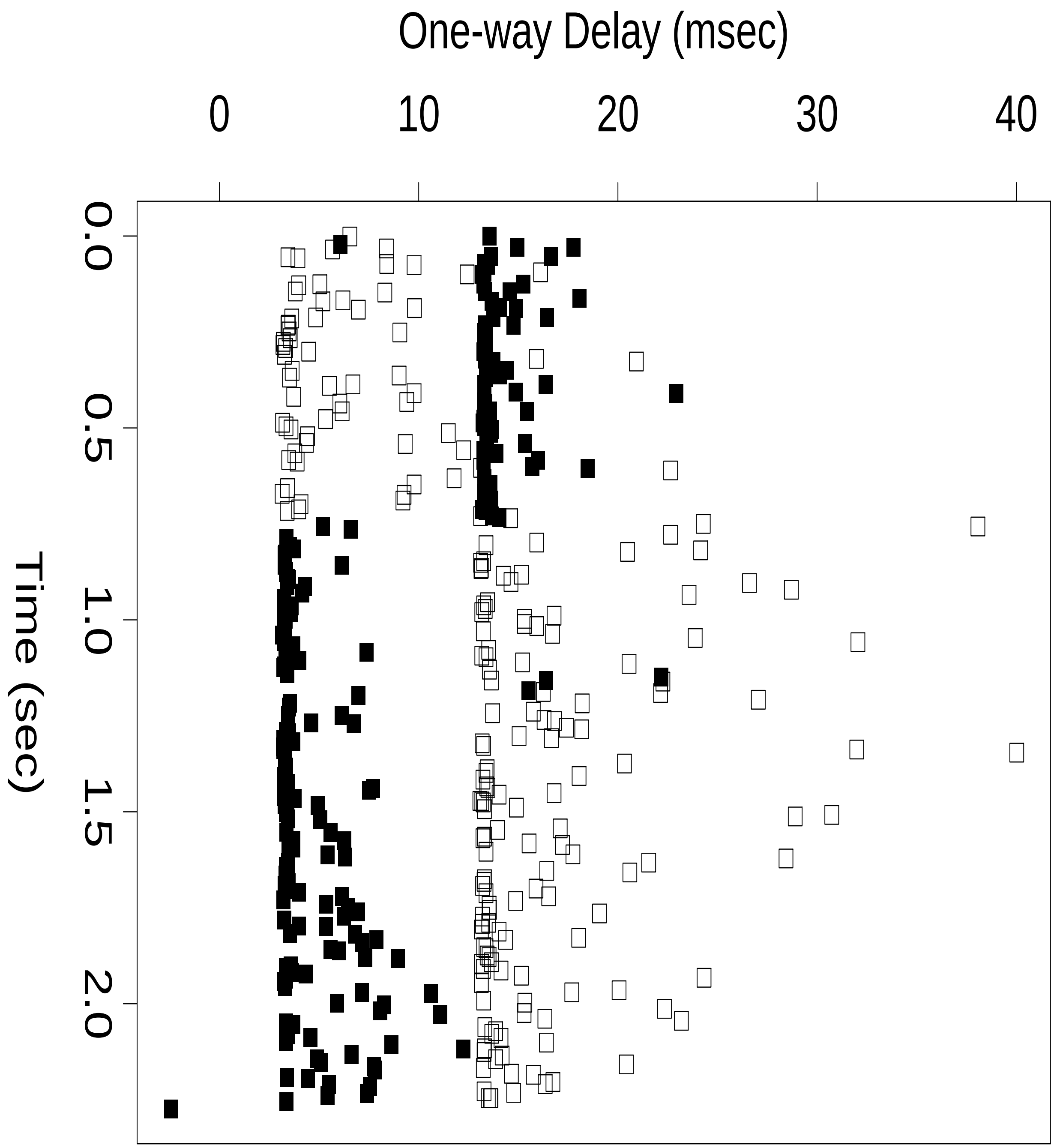
E.g.: compare bytes reassembled vs. SYN/FIN seq. #'s.

E.g.: compare GET/POST/HEAD instances in logs vs. running "`strings`" on packet trace of traffic sent to server (and *understand the discrepancy*).

E.g.: errors in a single clock are often undetectable, but apparent when comparing clocks.



One-way Delay (msec)
0
10
20
30
40
0.0
0.5
1.0
1.5
2.0
Time (sec)


One-way Delay (msec)
0
10
20
30
40
Time (sec)
0.0
0.5
1.0
1.5
2.0

Cautions re Calibration:

- Devising a consistency check can be a lot of work . . .
- . . . but *real* work is then investigating the inconsistencies.
- Often, you find nothing. Occasionally, you find scandal.

$\Rightarrow$ *Big* payoff if you can automate consistency-checking.

An All-Too-Familiar Scenario:

You work on your measurement study at a crazy feverish pace due to Deadline Crunch.

Months later, you receive feedback.
The reviewers ask that you redo an element of the analysis with a modest tweak.

Do you (1) introduce the tweak, recrunch the numbers, update the tables, and Call It Done . . . ?

An All-Too-Familiar Scenario, con’t:

. . . or (2) first run *without* the tweak to ensure you understand the process you used to get the numbers in the first place?

Clearly, (2) is more sound . . .

An All-Too-Familiar Scenario, con't:

. . . But: for a good-sized measurement study, unless you Strategy #4: structure for reproducible analysis, you very likely will *not* be able to reproduce the exact earlier numbers!

$\Rightarrow$ You've lost the previous mental context of fudge factors, glitch removals, script inconsistencies.

Does it matter?

For a paper of mine: 2X performance difference!

An example of structuring for reproducible analysis:

- Enforce discipline of using a single (master) script that builds all analysis results from the raw data.
- Maintain all intermediary/reduced forms of the data as explicitly ephemeral (caches).
- Maintain a *notebook* of what was done and to what effect.
- Use version control for scripts & notebook.

$\Rightarrow$ But also really needs: ways to visualize what's changed in analysis results after a re-run.

Provides "paper trail" *and* systematizes data exploration.

Community Datasets:

Two issues arise when datasets are captured by one party for use by another:

- data <u>soundness</u> concerns
- data <u>sensitivity</u> concerns

For data soundness, experience has shown the utility of

Strategy #5: periodically analyze ongoing measurements

- let's you discover when data acquisition *broken*
- ensures you're collecting (some) *meta-data*

Community Datasets, con't:

For data sensitivity, anonymization is getting very challenging as analysis increasingly needs packet *contents*.

Alternate approach: consider using Strategy #6: package analysis for "data reduction requests".

- send data analysis software to dataset holder
- they run it, inspect results, & return them

Benefit: packaging up analysis for others forces well-specified analysis steps, great aid for reproducibility.

Drawback: access to data ephemeral; data-gatherers may find it too much hassle.

Summary of Strategies:

Strategy #1:	*maintain meta-data*
Strategy #2:	*run your intended methodology by colleagues*
Strategy #3a:	*examine outliers and spikes*
Strategy #3b:	*employ self-consistency checks*
Strategy #3c:	*compare multiple measurements/computations*
Strategy #4:	*structure for reproducible analysis*
Strategy #5:	*periodically analyze ongoing measurements*
Strategy #6:	*package analysis for "data reduction requests"*
Strategy #7:	*subsample large datasets, assess variability*

What’s Needed:

- Data management: databases, version control
- Scriptable analysis environments
- Visualization & test suites to investigate *differences*
- Electronic “scientist’s laboratory notebook”
- Publication of measurement management tools/environments
- Funders supporting the development of such tools

Is it really worth the extra effort?

Measurement is hard enough already.

But:

- These strategies really can make the difference in soundness and *confidence*.
- Care in measurement engenders more thought about the *meaning* underlying analysis.
- Offers opportunities for *serendipity*.