Personal Keystroke Dynamics Leakage in On-The-Fly Web Applications

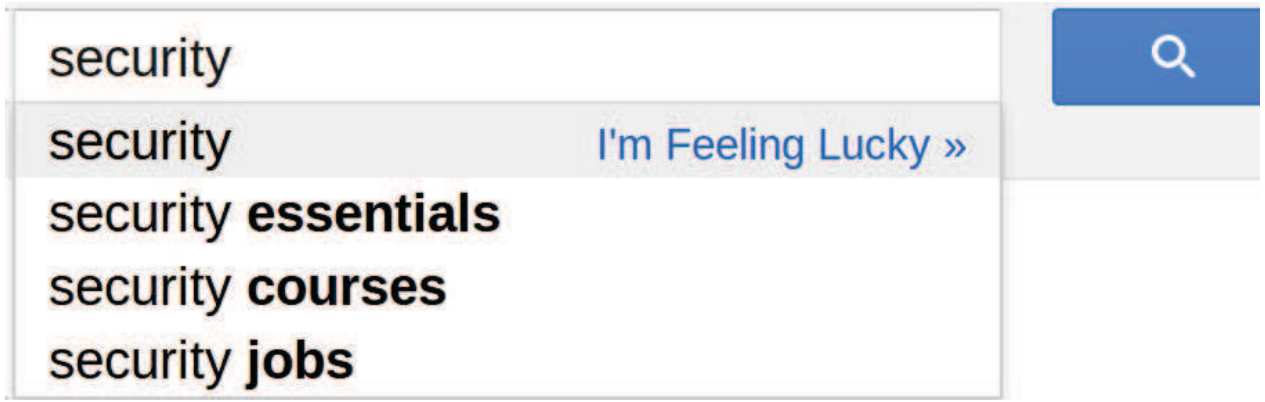
Debin Gao

Assistant Professor

School of Information Systems

Singapore Management University

Google Instant



- Allows user to view the results/suggestions on–the–fly while typing search queries.
- Frequent communication between frontend Javascript and the backend server.

Example of traffic

```
GET /s?...&q=s&...
GET /s?...&q=se&...
GET /s?...&q=sec&...
GET /s?...&q=secu&...
GET /s?...&q=secur&...
GET /s?...&q=secure&...
```

Overview of project

- Study communication between web applications and servers.
- Analyze delay between keystroke and packet timings.
- Identify cause of variable delay (noise).
- Mitigate effect of noise.
- Recover inter-keystroke timings from inter-packet timing

Trends in Javascript applications

- Richer and more complex
- Sophisticated GUI updates
- Fast client- server communications in the background using AJAX.
- Approaching capabilities of traditional desktop applications.

AJAX and XMLHttpRequest

- AJAX
  - Asynchronous JavaScript and XML.
  - Allows dynamic retrieval of data in web applications
- XMLHttpRequest
  - API available to web applications
  - Used to send HTTP(S) requests directly to a web server.
  - Web server responds with data (in various format: JSON, XML, etc)
  - Web application can use data to update document dynamically.

Does high interactivity result in information leakage?

- Prior work: Chen 2010
  - packet sizes of encrypted packets leak content
- Open question:
  - timing side channel of Javascript applications.
  - harvesting of typing pattern from Javascript applications

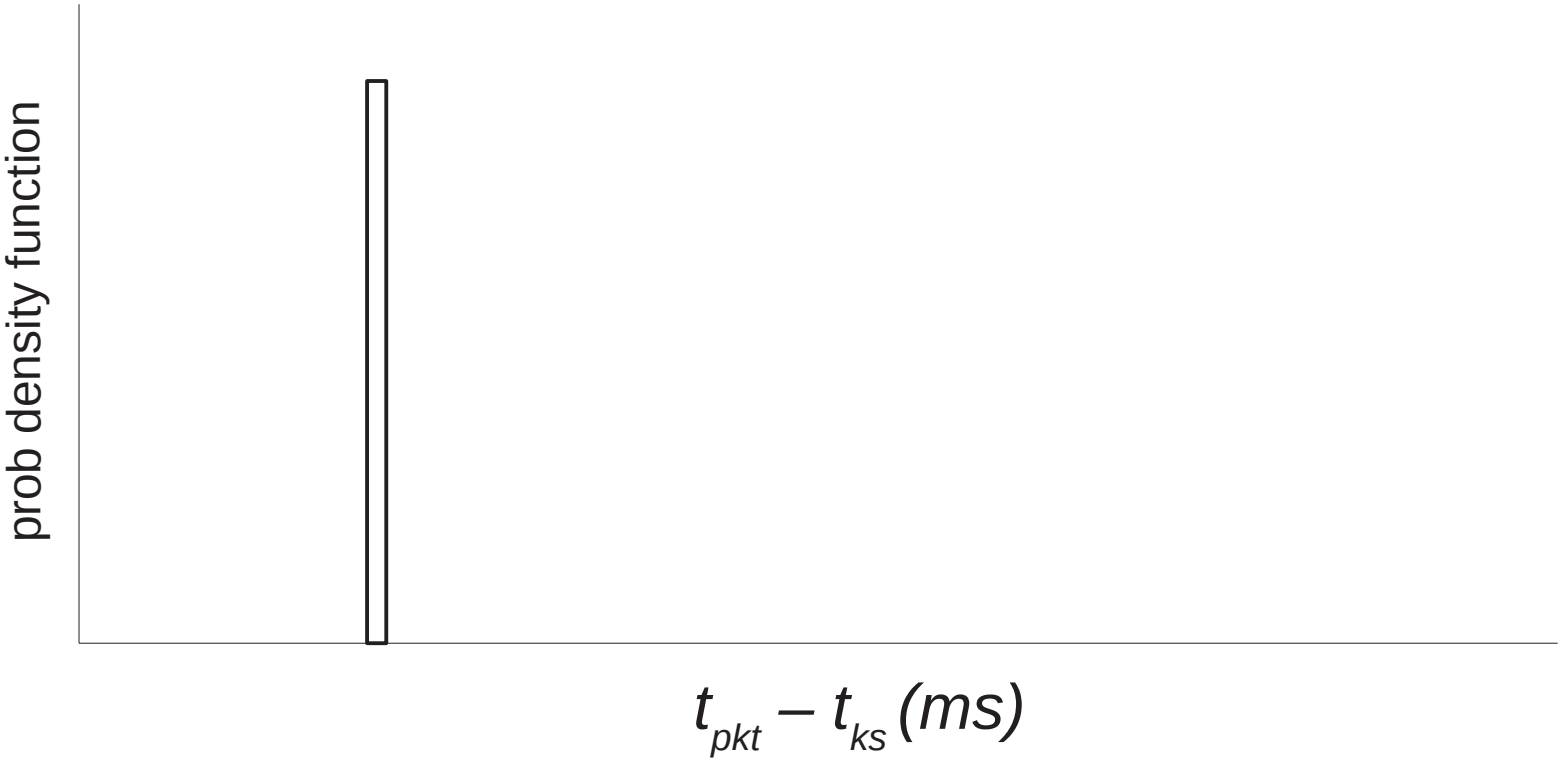
Why typing pattern is important?

- Privacy concern:
  - User identification through typing pattern (Araujo 2005, Killourhy 2012, Monrose 2000, Peacock 2004)
- Security concern:
  - Prerequisite for keystroke biometrics imitation attack (Tey 2013)
  - Facilitates attacks on other timing side channels (Song 2001)

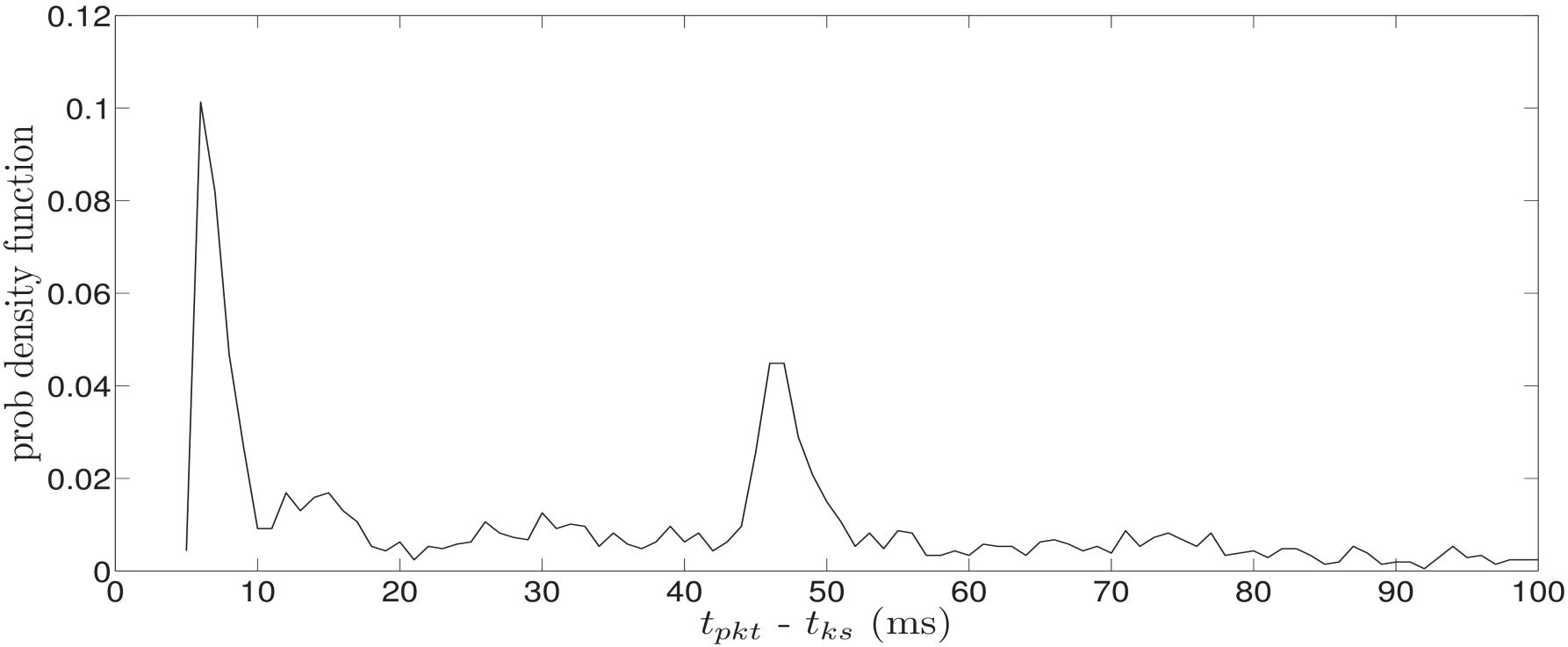
Challenges - weak correlation between keystroke and packet timings

- Javascript applications far slower than native binary applications.
  - Interaction with DOM more complex.
  - UI updates take longer compared to the computation intensive tasks such as SSH (Song 2001).
  - Single threaded co-operative multitasking execution model.

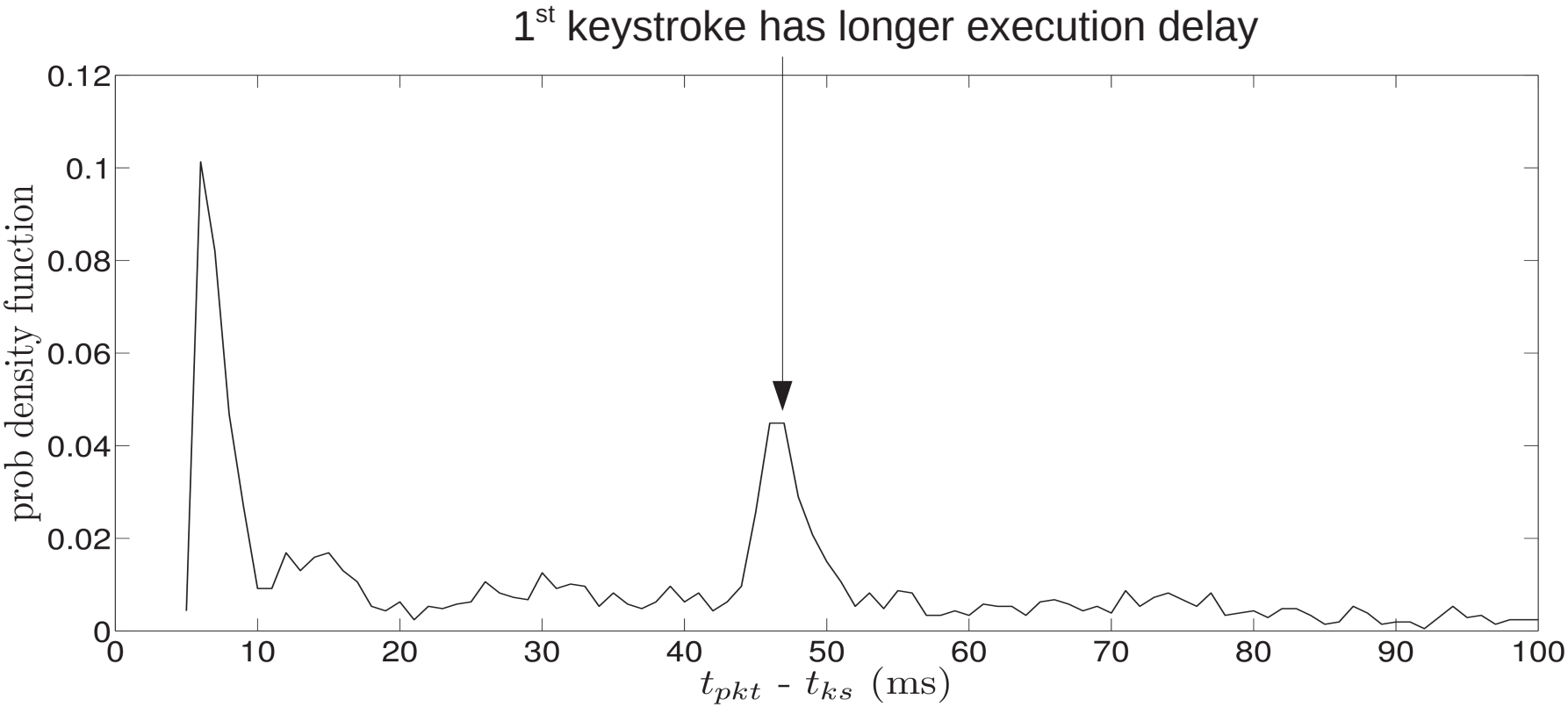
Ideal delay between keystroke and packet timing



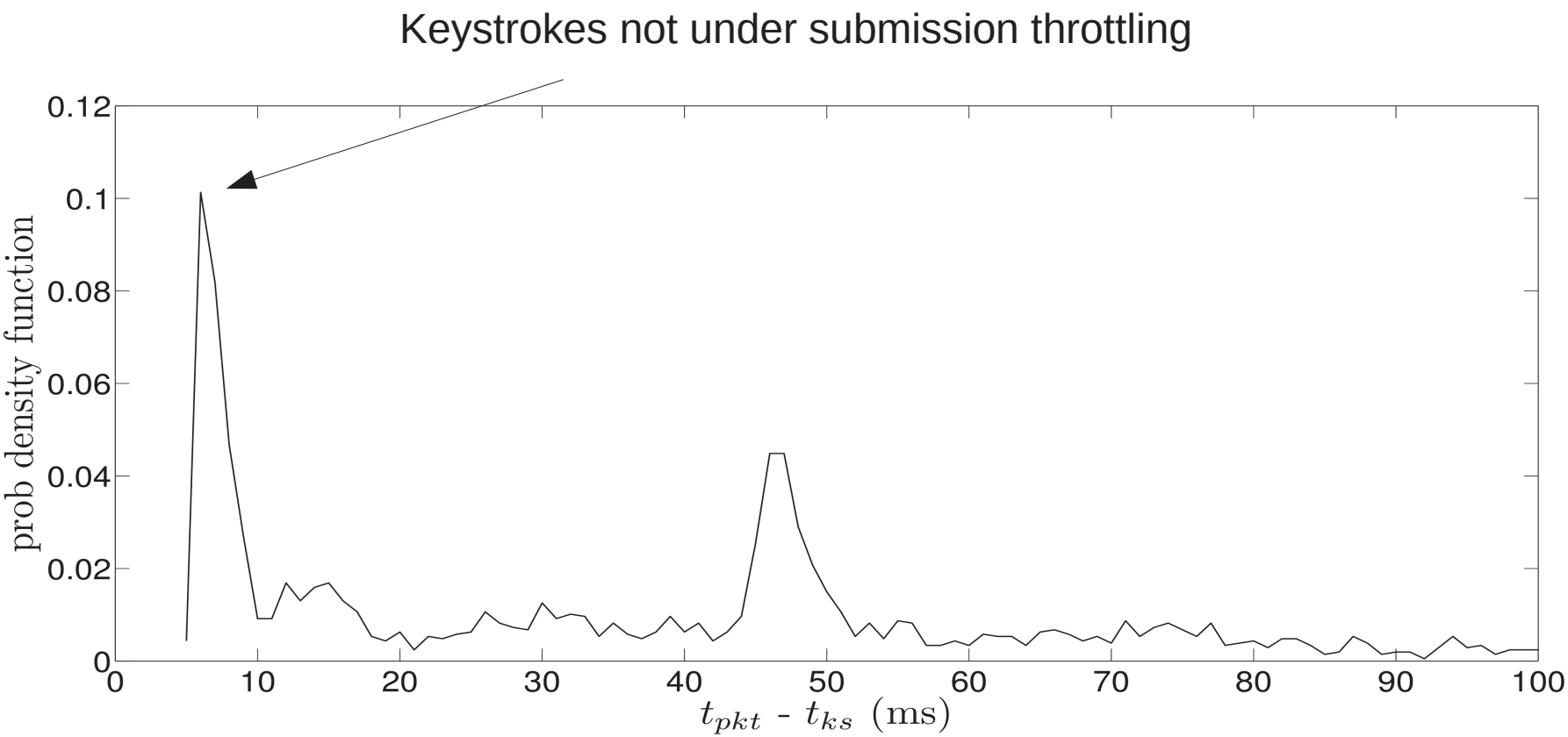
Reality - Variable delay between keystroke and packet timing



Reality - Variable delay between keystroke and packet timing



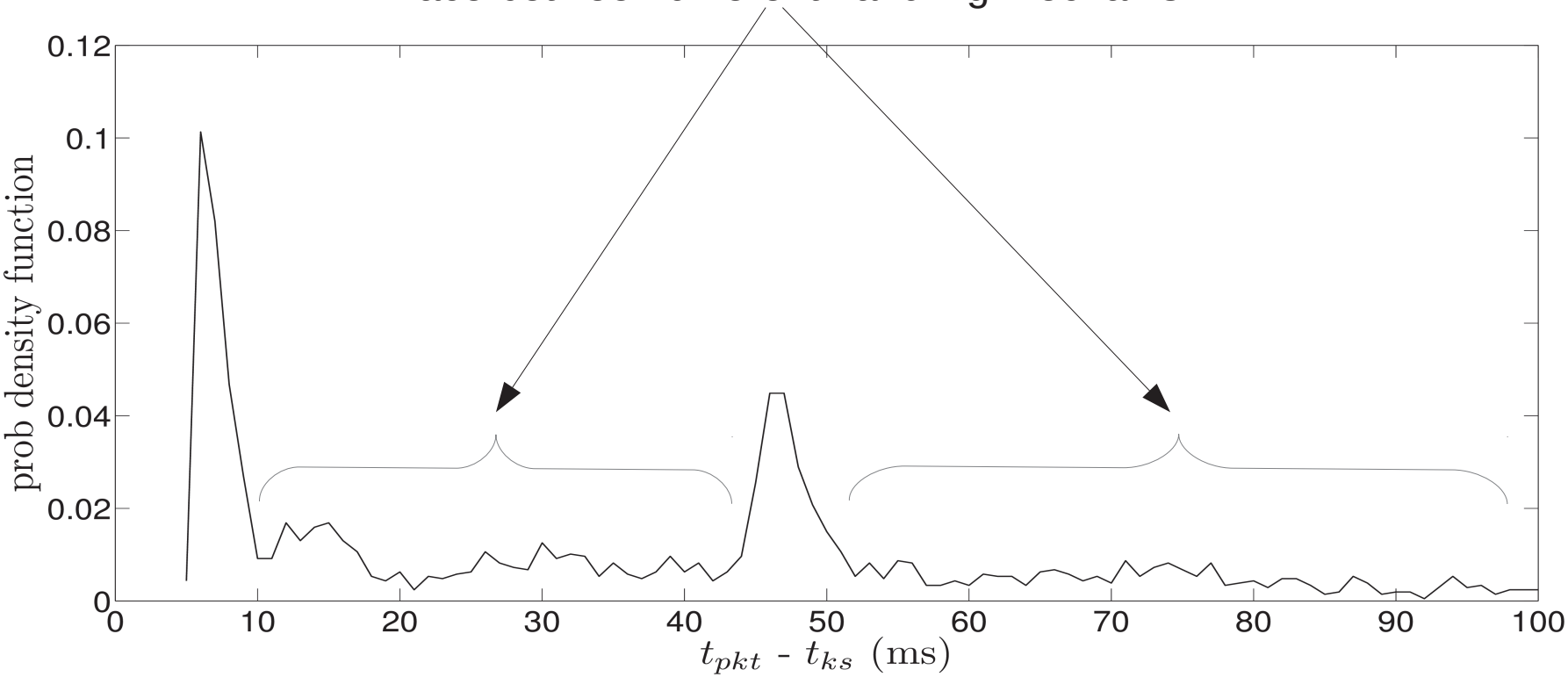
Reality - Variable delay between keystroke and packet timing



Submission throttling: keystrokes occuring within timeout window are delayed

Reality - Variable delay between keystroke and packet timing

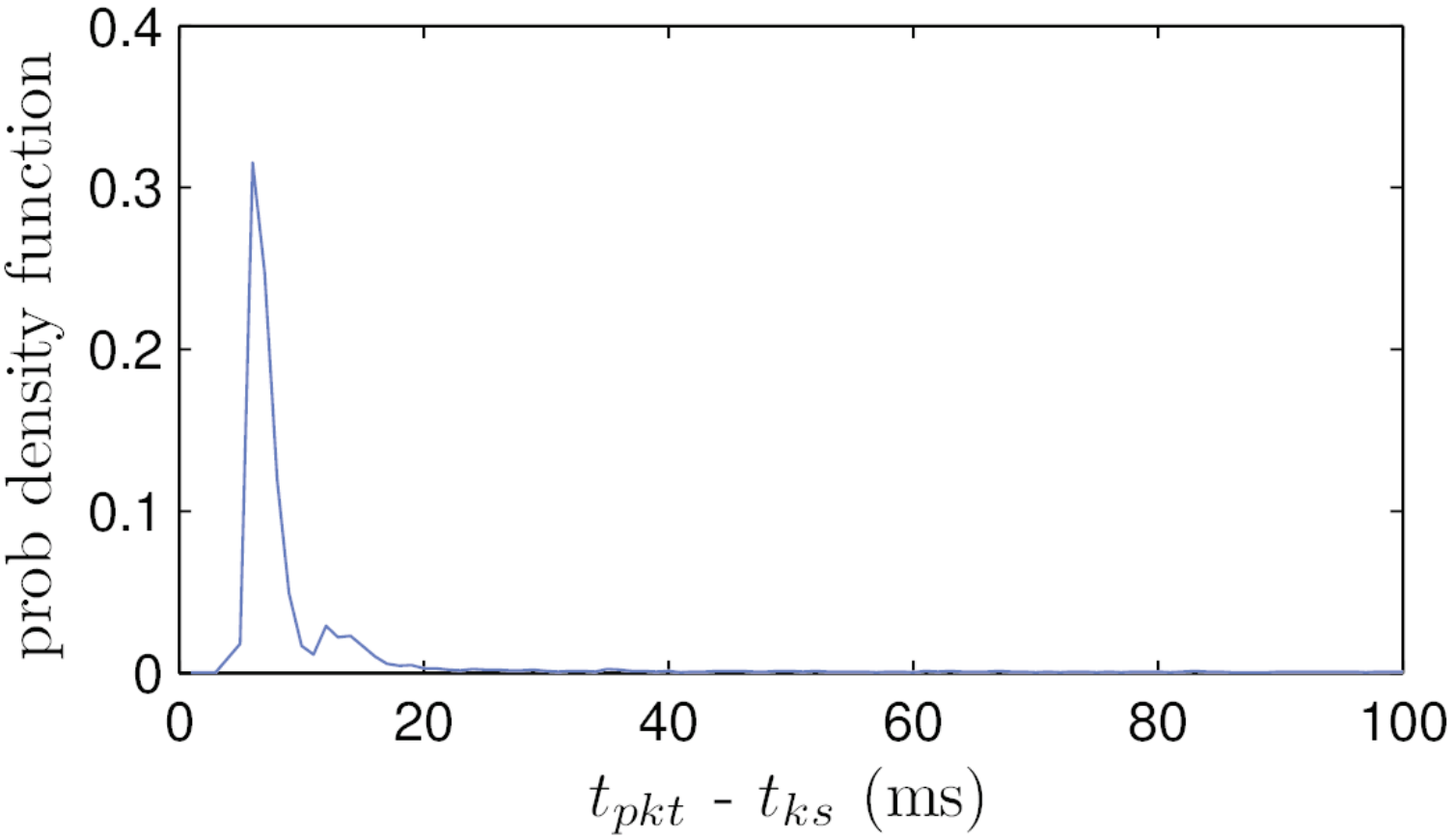
- Submission throttling
- Single threading concurrency delay
- race between different handling mechanism



Mitigating influence of noise

- Execution delay
  - Unable to mitigate effect of race and concurrency
  - Mitigate additional execution path by ignoring 1st packets
- Submission throttling
  - Identify and ignore packets affected by submission throttling.

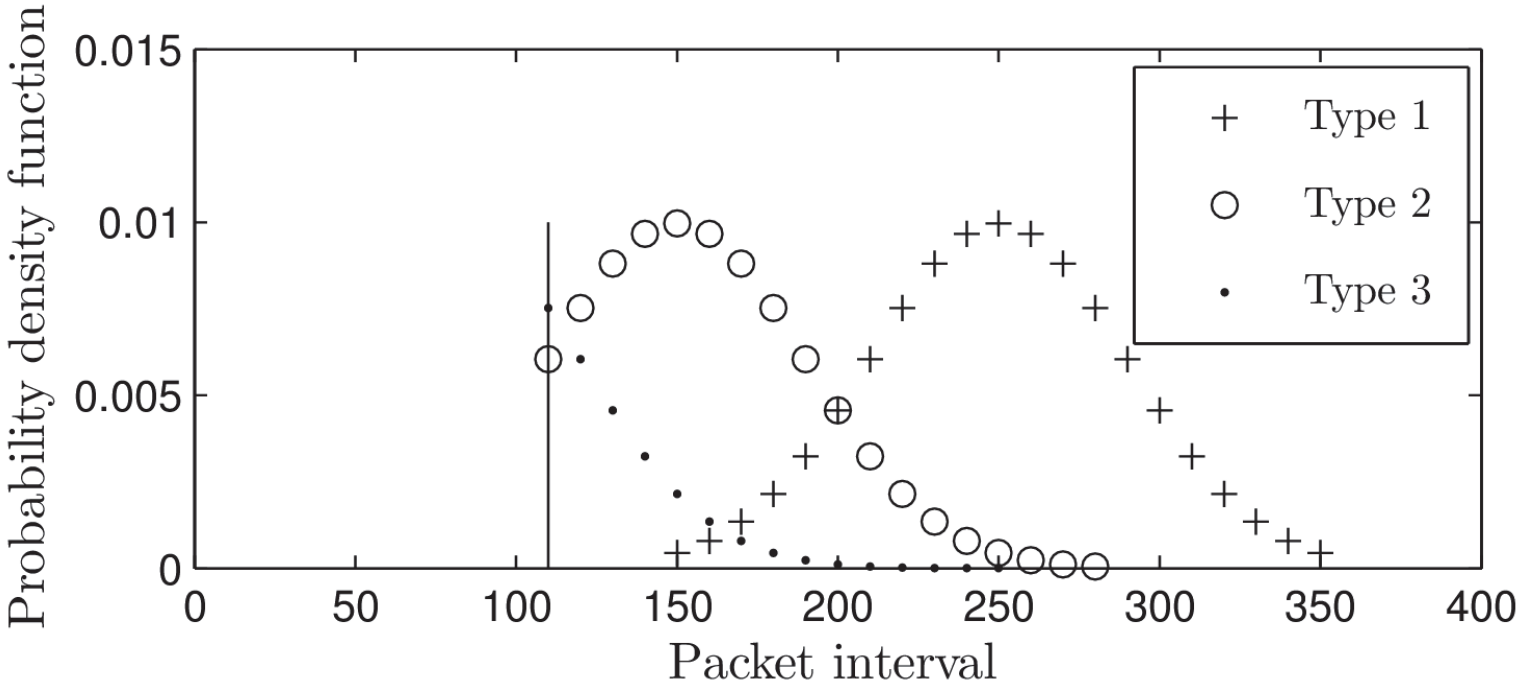
Improved noise model



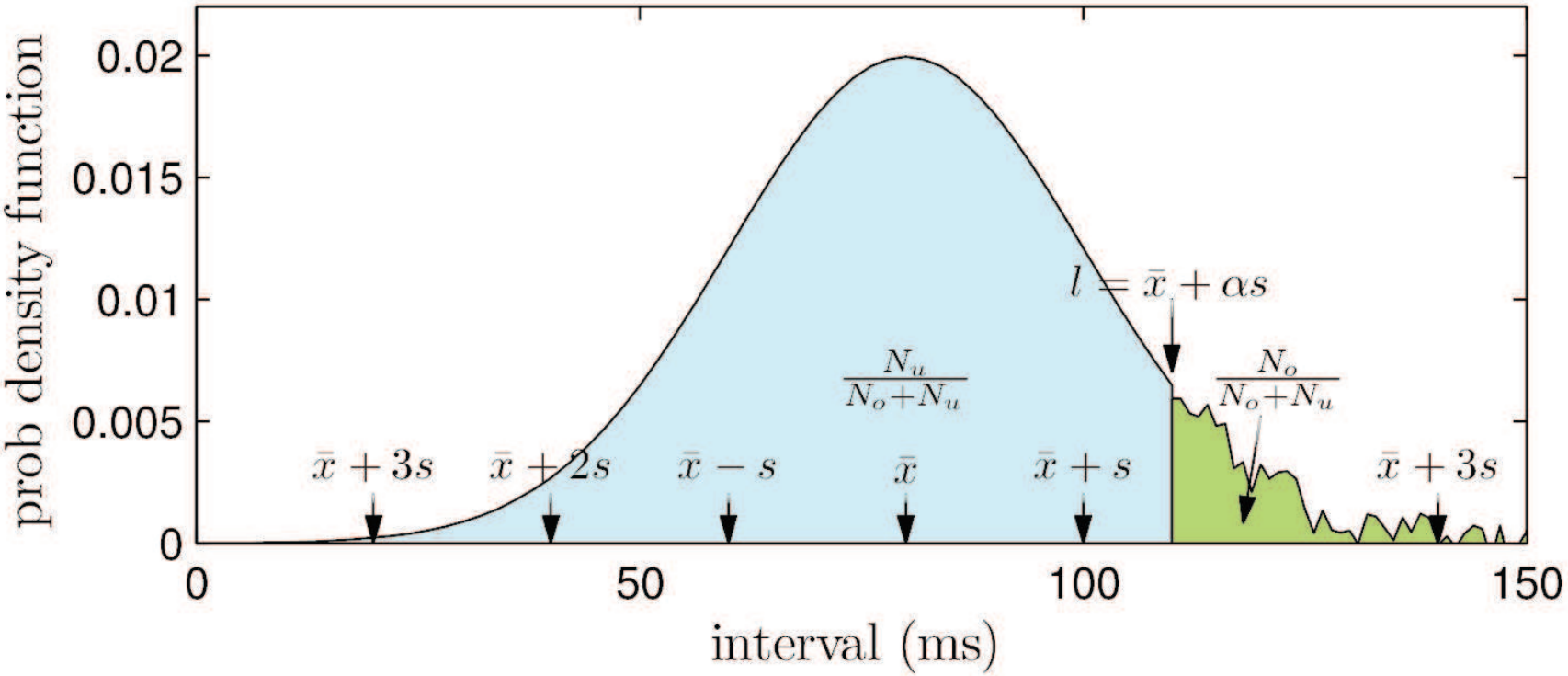
Constructing the typing pattern

- Type 1: all timings > throttling threshold
  - compute mean and variance directly
- Type 2: most timings > throttling threshold
  - reflect points about median
- Type 3: most timings < throttling threshold
  - curvefit the tail
- Type 4: all timings < throttling threshold
  - subtract digraph timing from trigraph timing

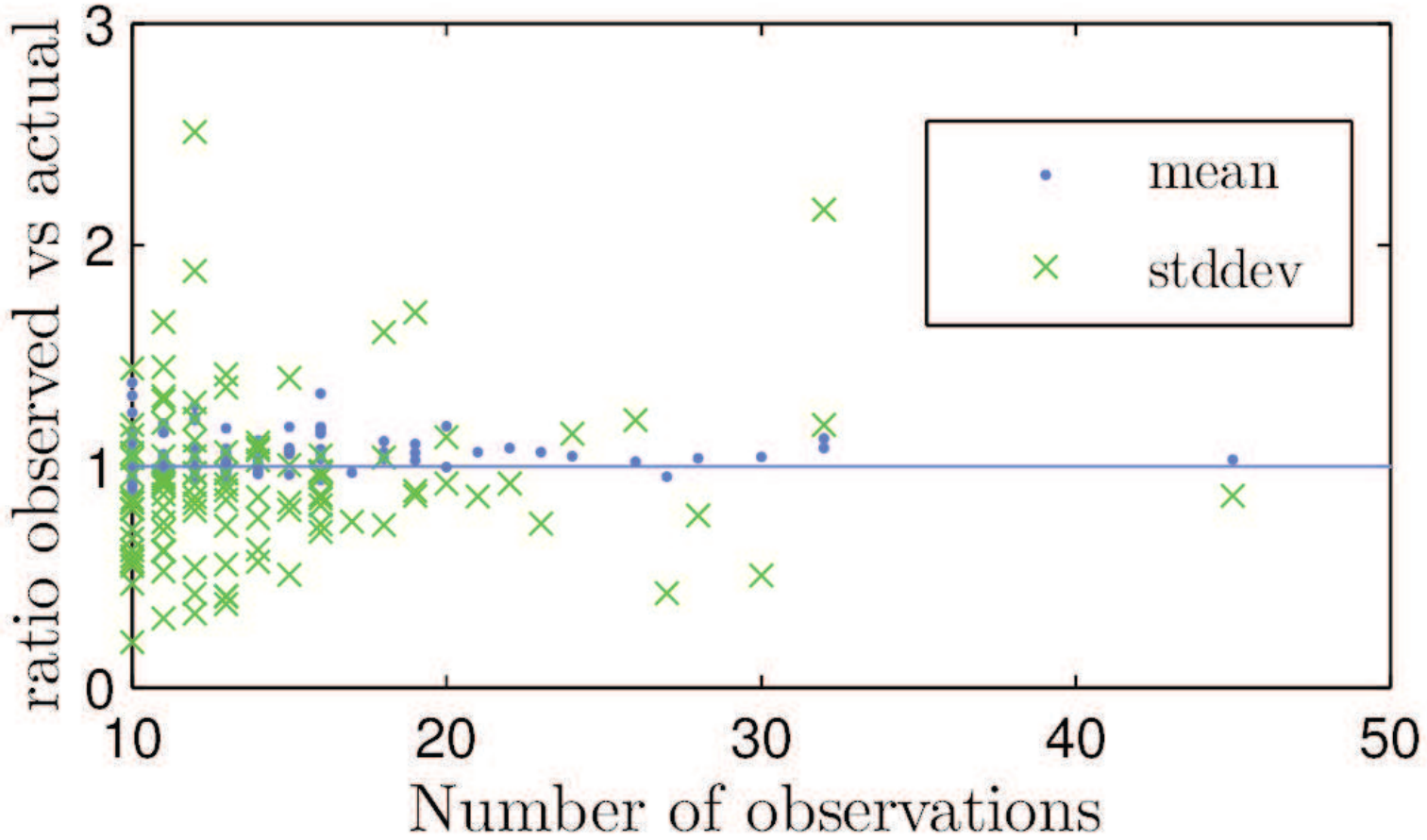
Characterisation (Type 4 not shown)



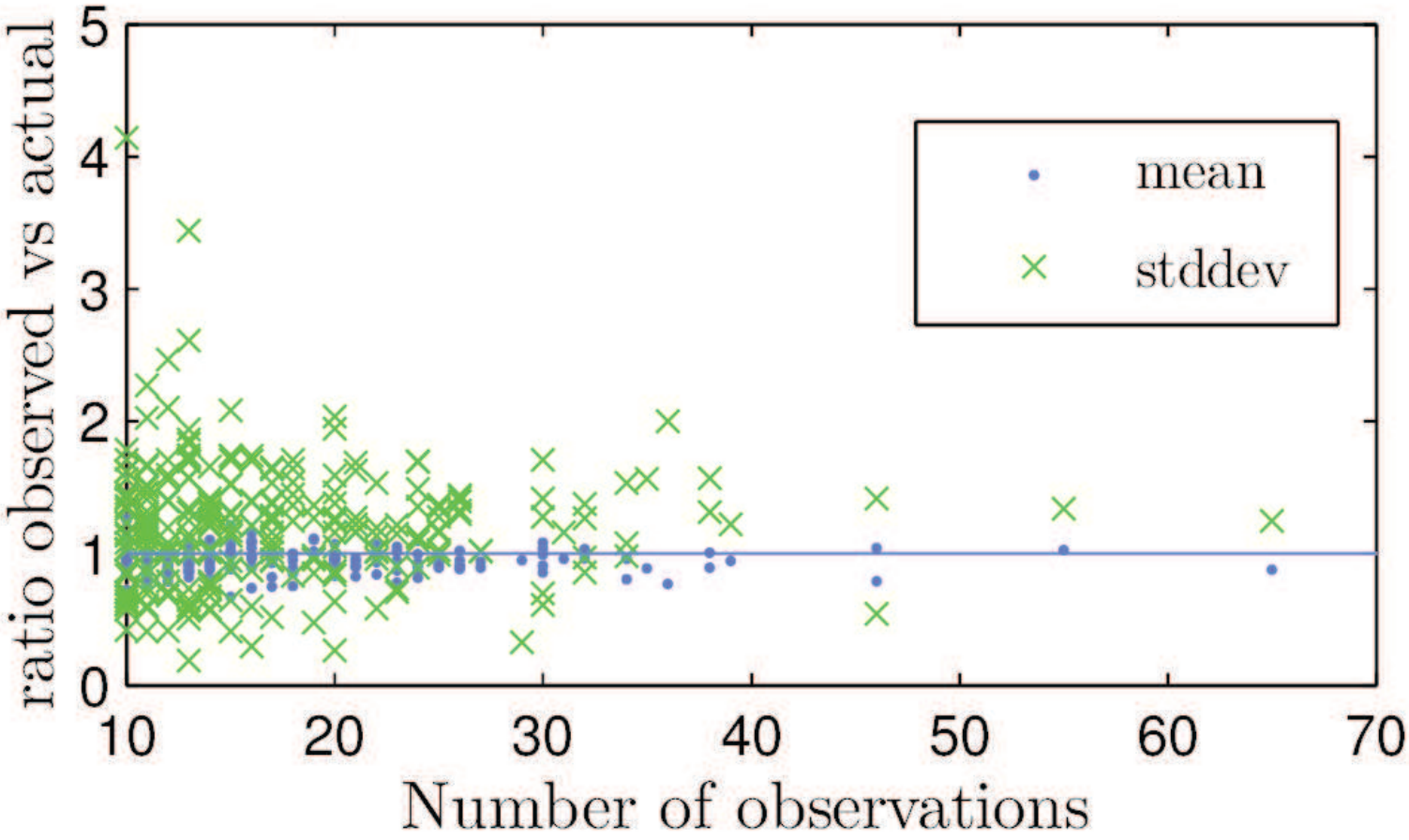
Type 3 curve fitting



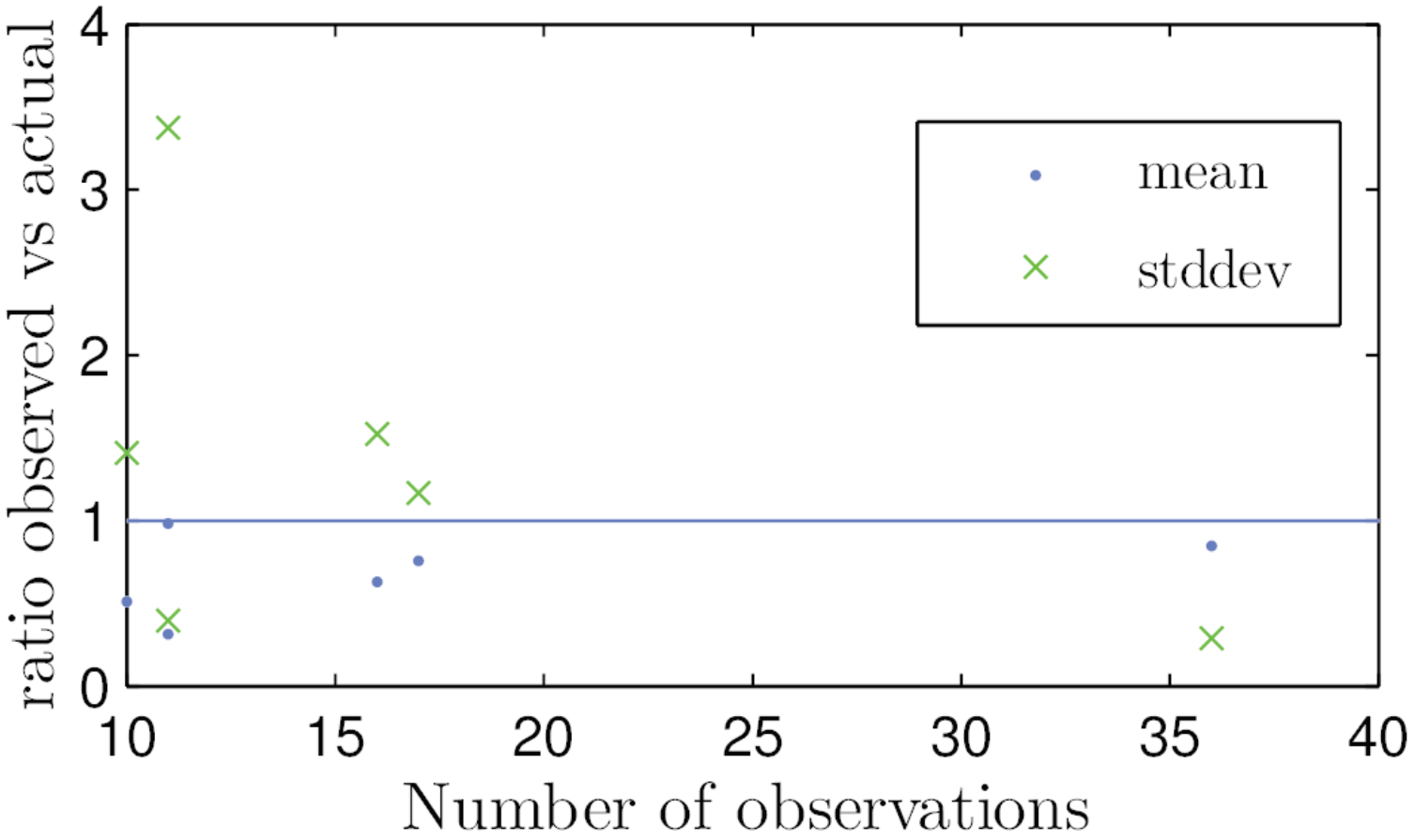
Results - Type 1 (direct)



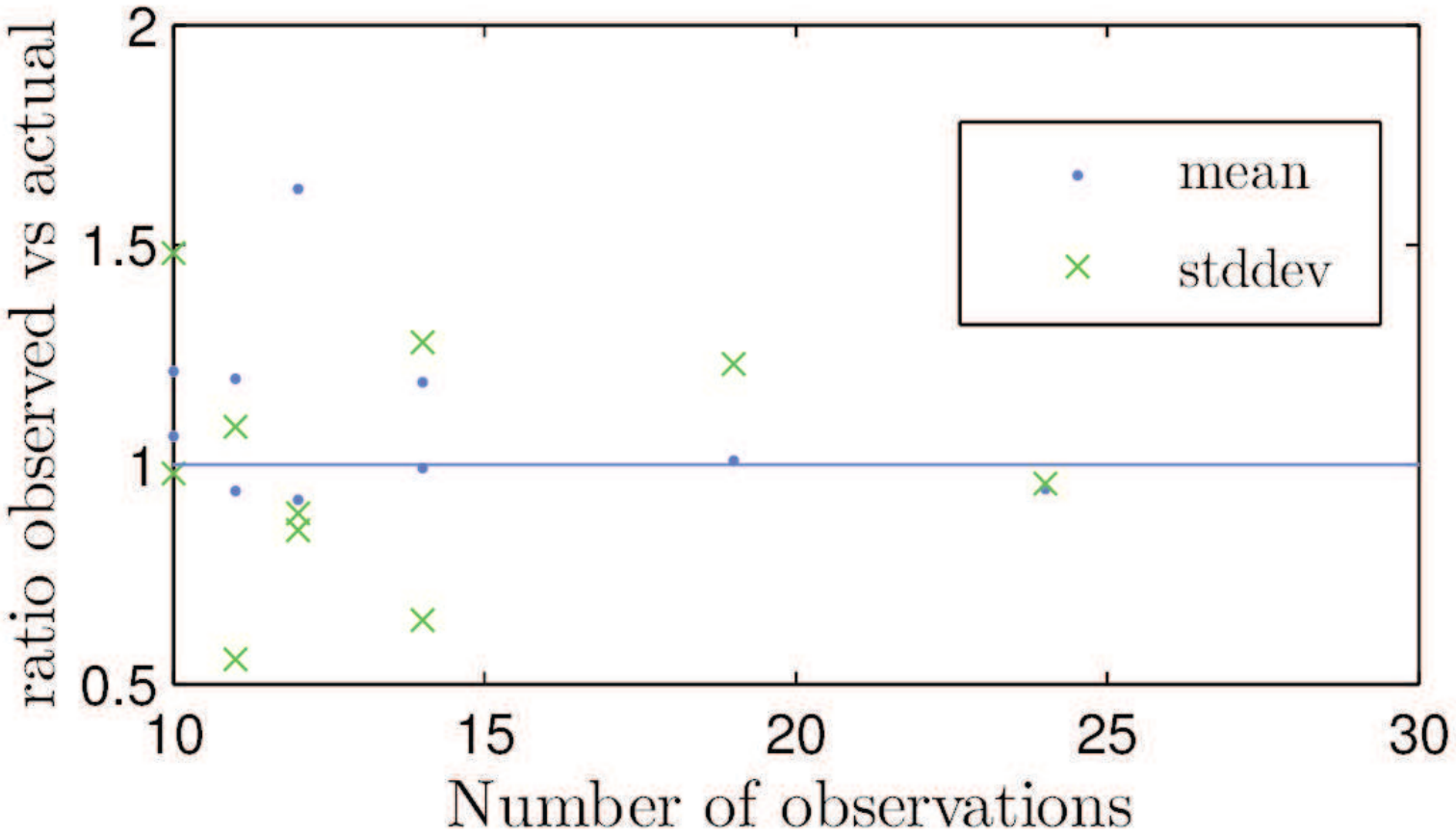
Results - Type 2 (median)



Results - Type 3 (curvefit)



Results - Type 4



Countermeasures: varying the timeout