



## A Brief Study of Time

*By*

**Florian Buchholz and Brett Tjaden**

*Presented At*

The Digital Forensic Research Conference

**DFRWS 2007 USA** Pittsburgh, PA (Aug 13<sup>th</sup> - 15<sup>th</sup>)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

**<http://dfrws.org>**

# A Brief Study of Time

Florian Buchholz

Brett Tjaden

Department of Computer Science

James Madison University



# Computer Clocks and Digital Forensics

- Clocks may not show the “real” time
  - Clock skew
  - Clocks can be set to any value
    - By a user or by a program
    - Intentionally, accidentally, or maliciously
- Timestamps on a computer are derived from its clock
- Problems when clocks are wrong:
  - Mapping timestamps back to “real” time
  - Handling evidence from multiple disparate sources
    - Each have their own clock
    - May warp the time line, leading to an incorrect forensic analysis

# Our Research

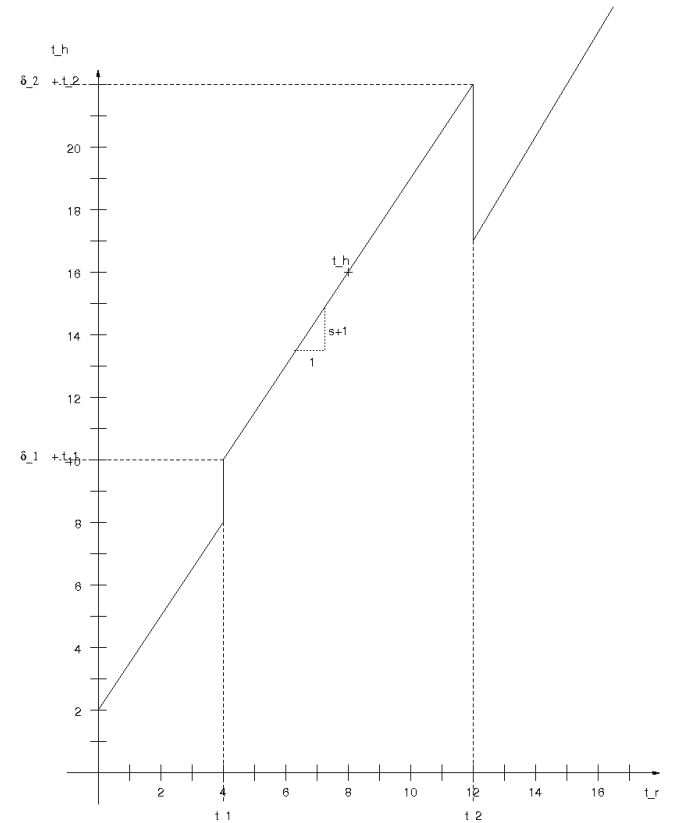
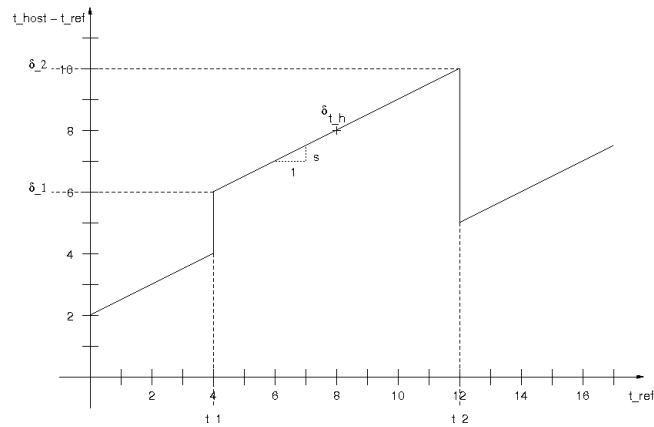
- Sampled clocks of 8410 web servers since October 2006
- Goals:
  - Find out if further research in this area is warranted
  - Gain an understanding how computer clocks behave
  - Do the clocks adhere to the clock model?
  - Can we use information from “external” clocks in a temporal analysis of the events of a local host?

# Previous studies of clocks

- Surveys of the NTP network (Minar)
- Identifying hosts by their clock behavior (Kohno et al.)
- Correlating timestamps within a small-scale network (Schatz et al.)
- Clock modeling (Buchholz)
- What is lacking is general data for how computer clocks behave.

# Clock Model

- Linear skew
- Discrete “jumps”
- Relative view shows offset to reference time
- Absolute view is used to map timestamps



# Clock Study

- Identified 8,410 popular Internet servers located all over the world
- Take clock readings on each server once a night since October 2006
  - Used two different methods for measurement
  - Some clocks that stood out were sampled hourly after they were identified
- Statistical analysis and visual inspection for our preliminary results

# Sampling methods

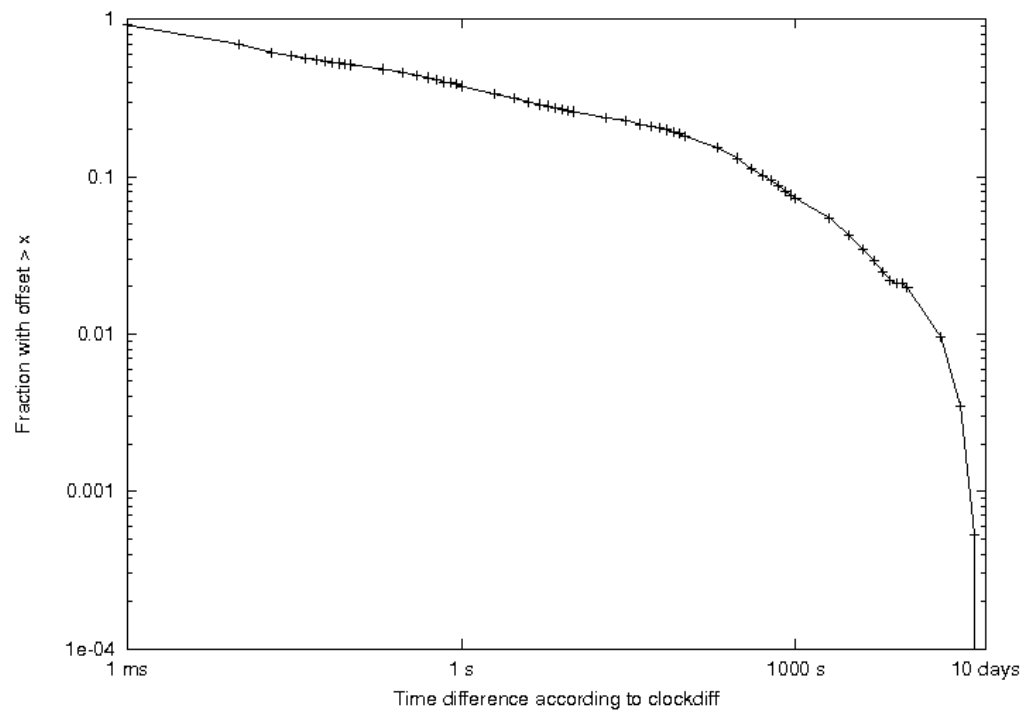
- HTTP requests
  - The HTTP 1.1 protocol requires servers to include a timestamp in UTC in their replies.
  - An HTTP “GET” request is sent to each client.
  - Midpoint between request and reply is used as reference point
  - The reference point and its difference to the timestamp are recorded
- Data from ICMP and IP options
  - Clockdiff tool (part of iputils package)
  - Reports milliseconds offset modulo 24 days between monitor and target hosts



# Study Results

- About 74% of hosts synchronized within +/- 10 seconds
  - We played it safe, because we do not have established measurement errors
- About 26% not well synchronized
  - Some hosts off by hours, days, weeks, months, years, a century
- 3767 of the 8140 hosts responded to clockdiff
- Measurements obtained by different methods usually agree (95% are within +/- 5s)

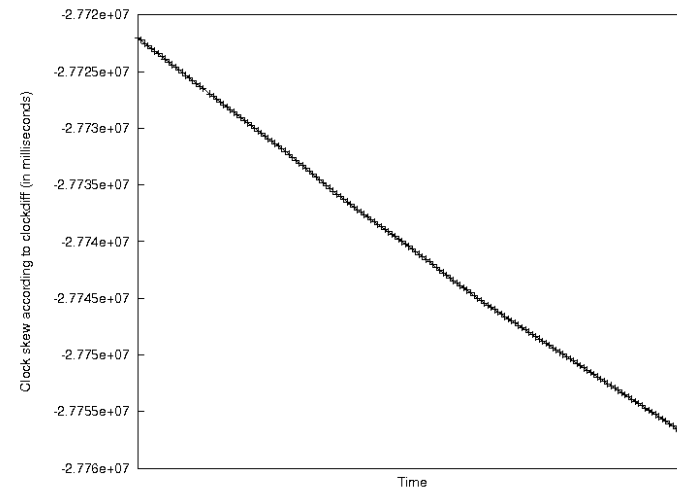
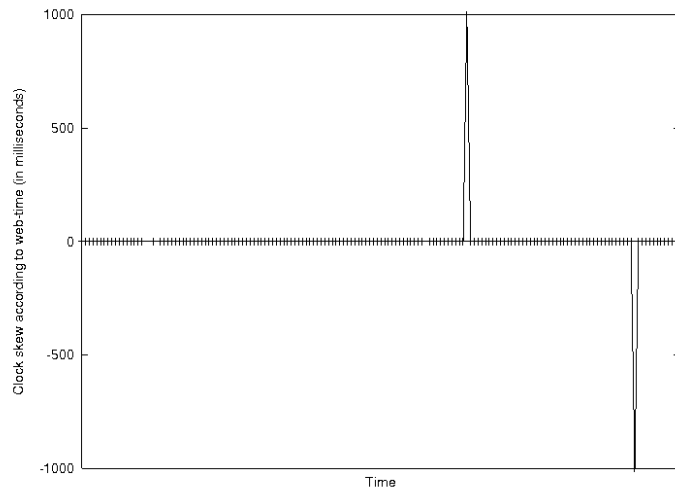
# Distribution of time offsets (clockdiff)



# More results - October 18

- HTTP
  - 2313 clocks were slow
    - 12,029 seconds (3.34 hours) on average
    - Clock farthest behind was 152 days (after taking out 100 years and January 1st, 1970)
  - Roughly 50% showed a difference of 0 s
  - Clock farthest ahead was 24h off
- Clockdiff
  - 2028 of 3767 clocks were slow
  - 1610 were fast
  - 129 showed a difference of 0 ms
  - 67% of the clocks were within +/-2s
  - 57% within +/- 500 ms

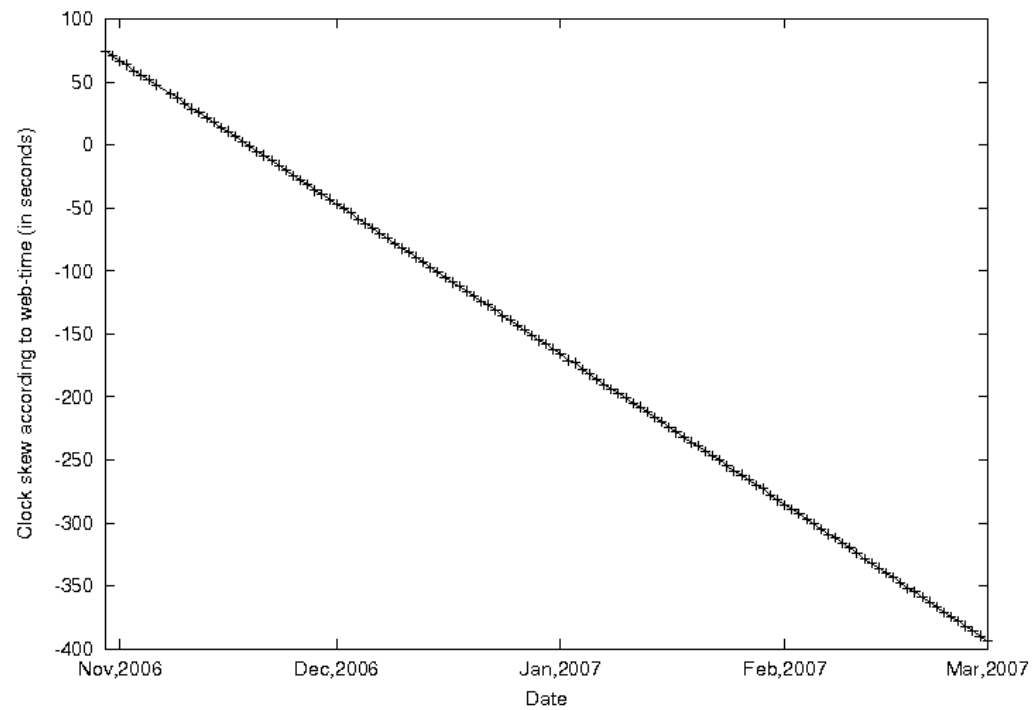
# Discrepancies Between Sampling Methods



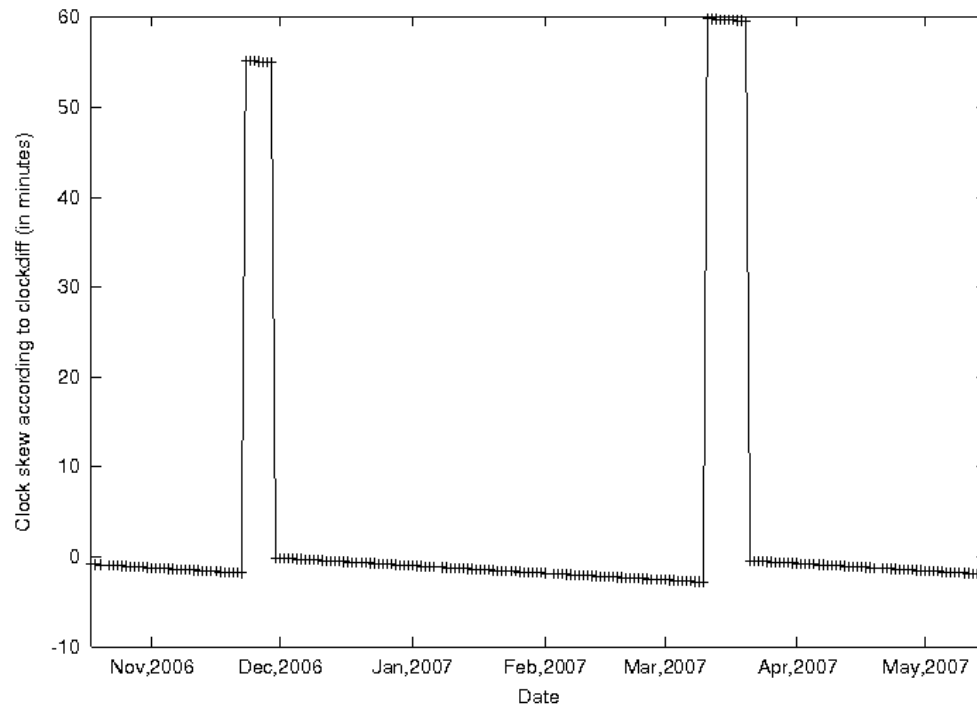
# Observations

- Clock skew
- Clock jumps
- Clocks set well in the past or future
- Clocks conforming to the model
- Unexplained phenomena
- Synchronized clocks usually stay synchronized

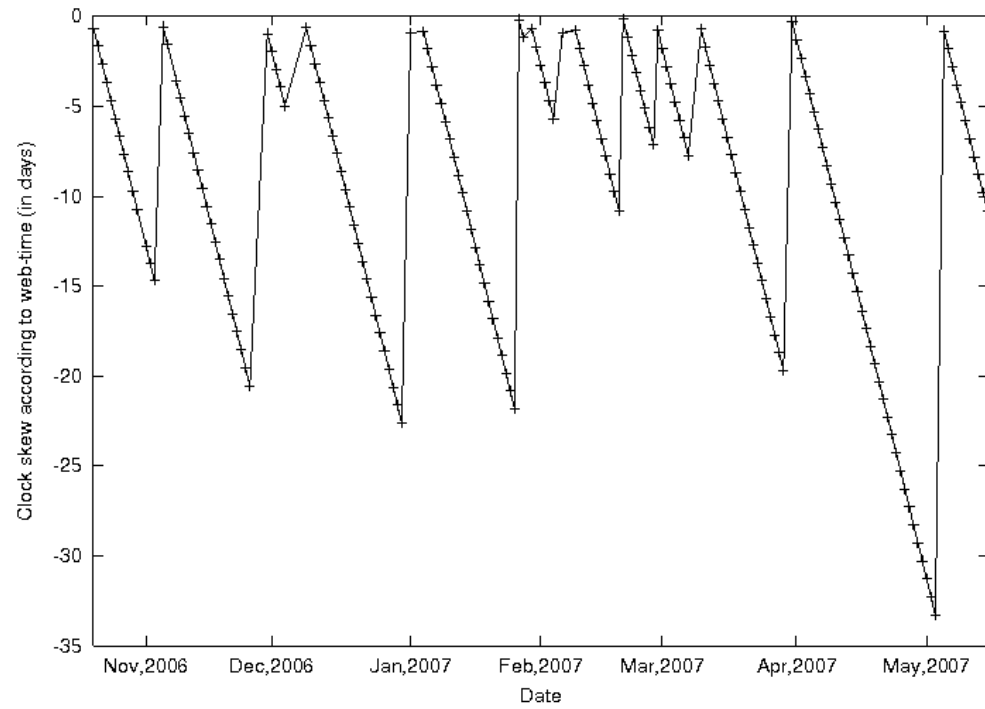
# Clock with linear skew



# A Clock conforming to the model

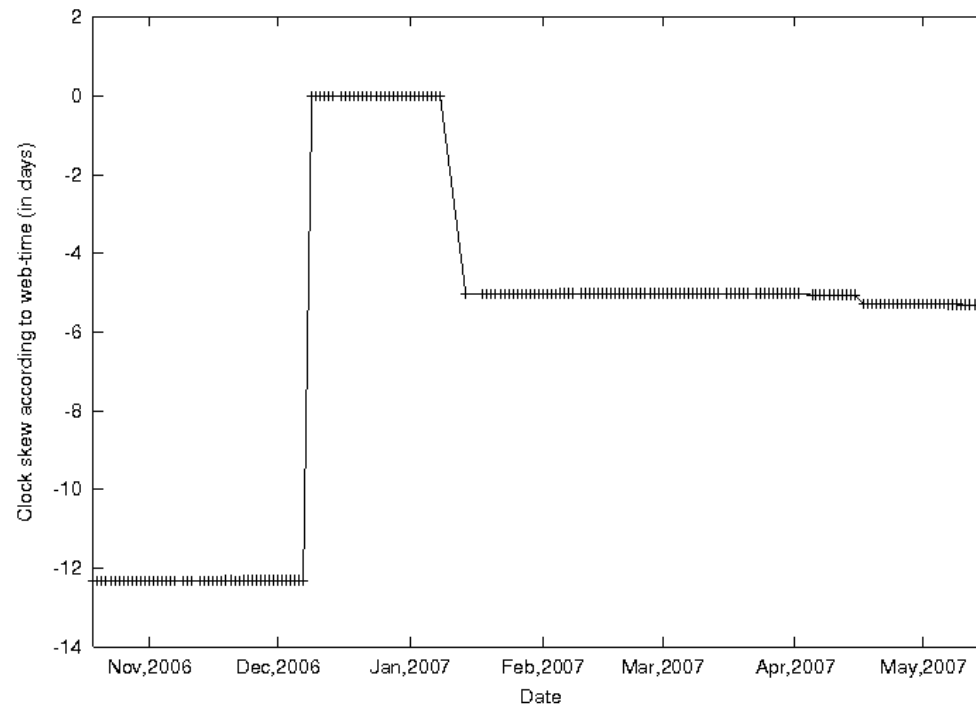


# “Stuck” clocks

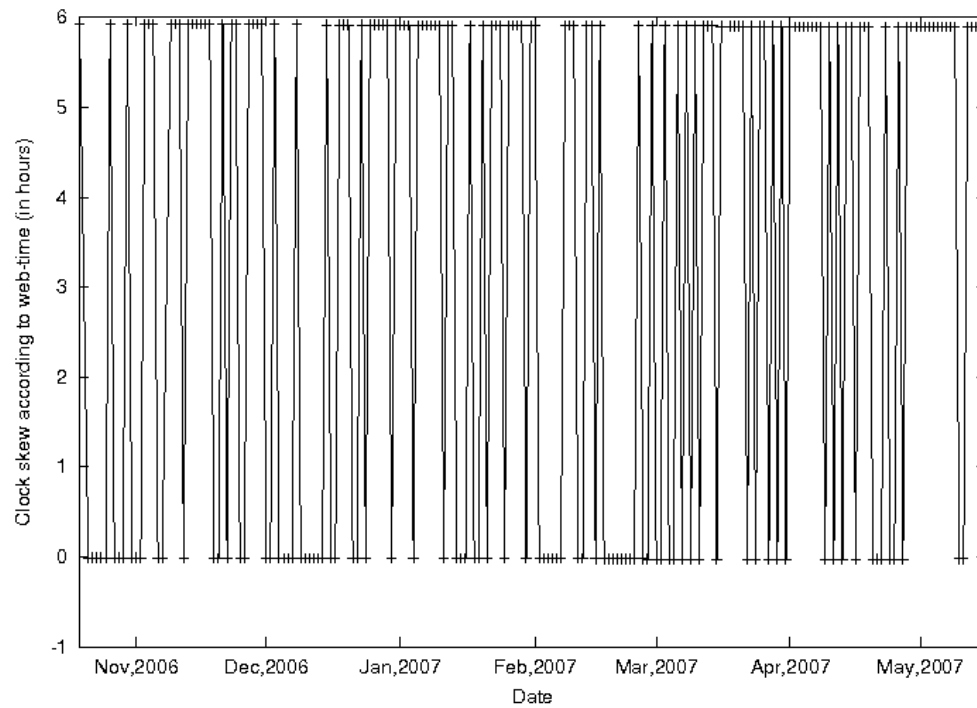




# Large jumps



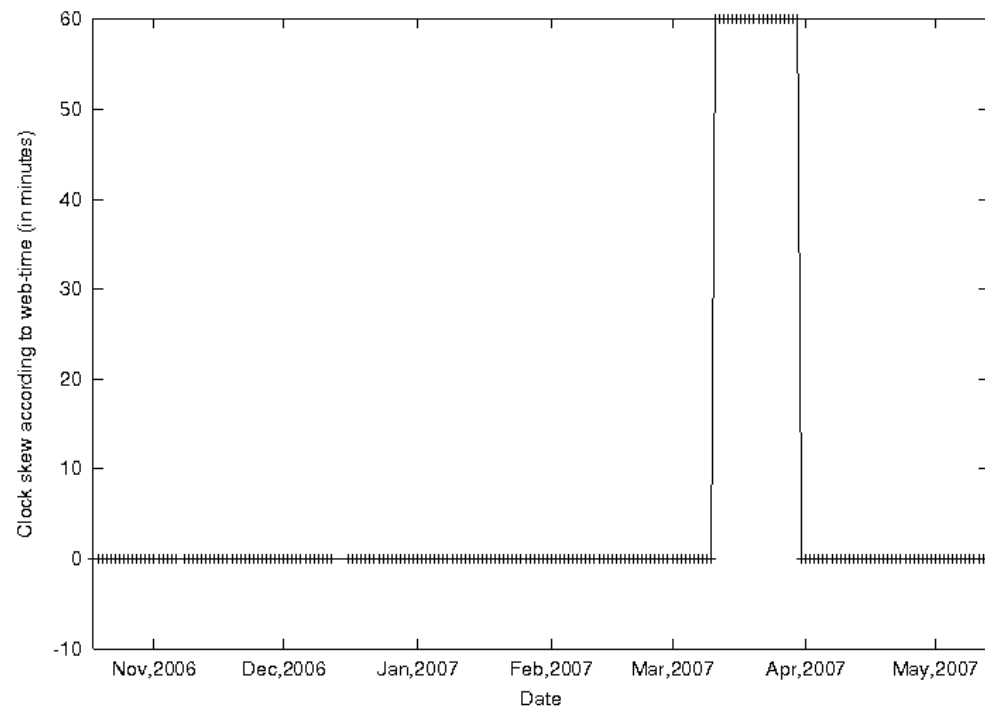
# Frequent jumps



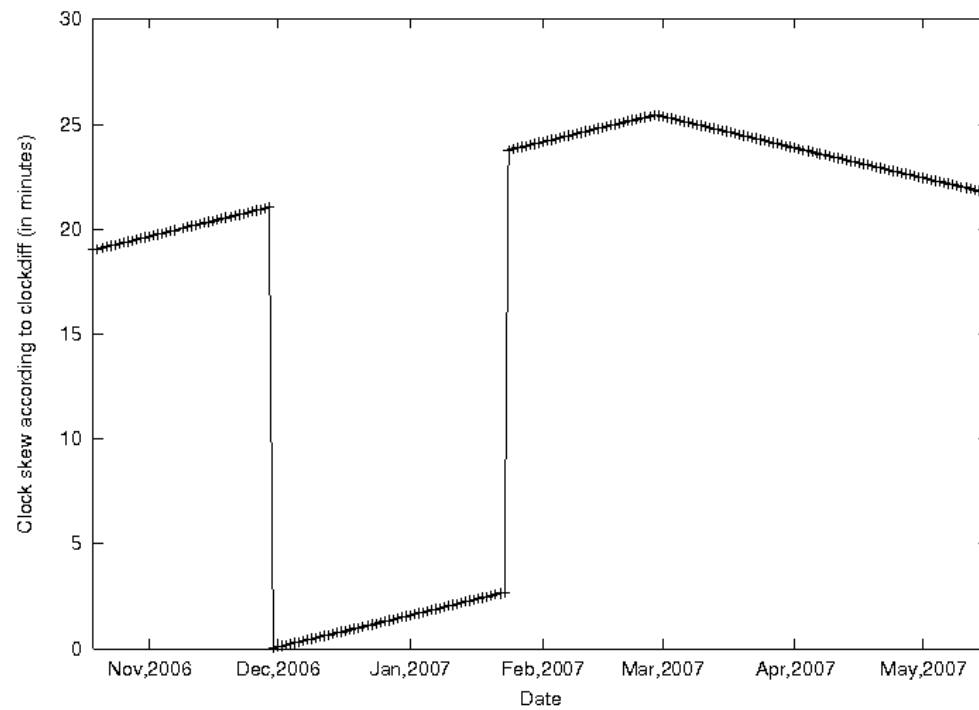
# Daylight Saving Time

- UTC is not aware of DST
  - Nothing should happen with the clocks when the computer “switches” to DST
- New DST rules for the USA in 2007
  - Second Sunday in March (from first Sunday in April)
  - First Sunday in November (from last Sunday in October)
- Compared number of hosts that were off by more than one hour during the relevant period in March and April
  - March 11: number increased from roughly 185 to roughly 230

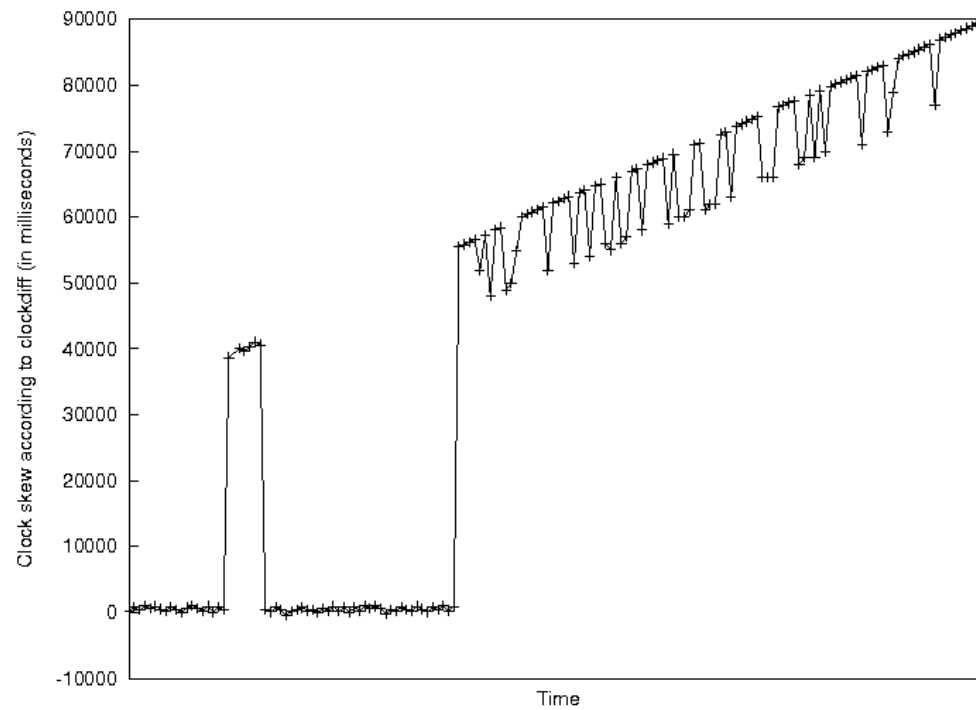
# Daylight Saving Time (cont.)



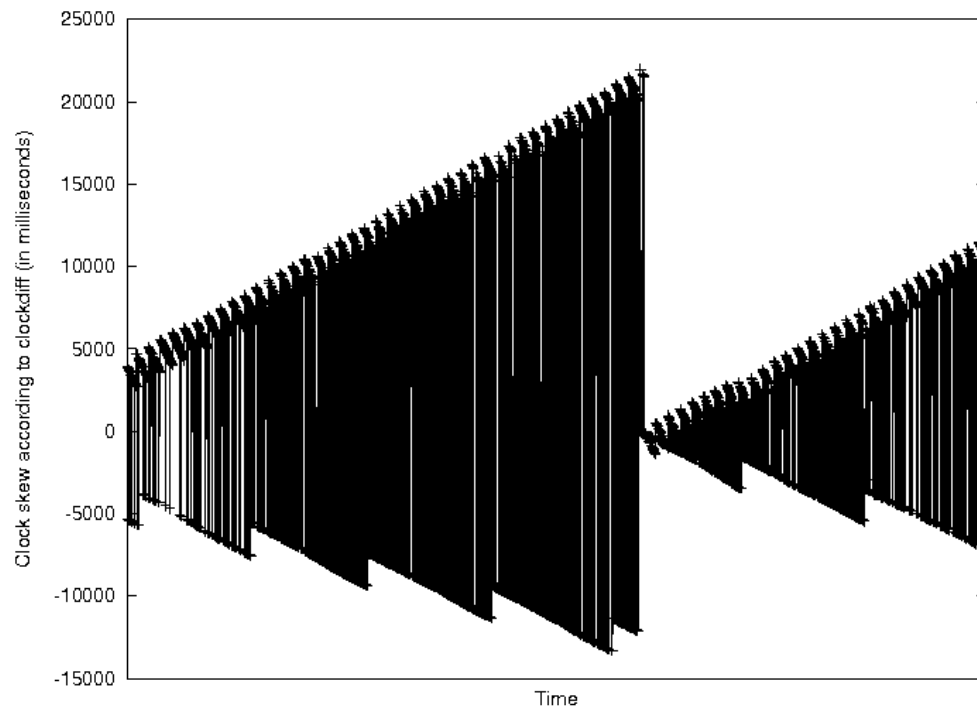
# Did the hardware change?



# Unexplained clock behavior



# Multiple hosts behind one network address?



# Conclusions

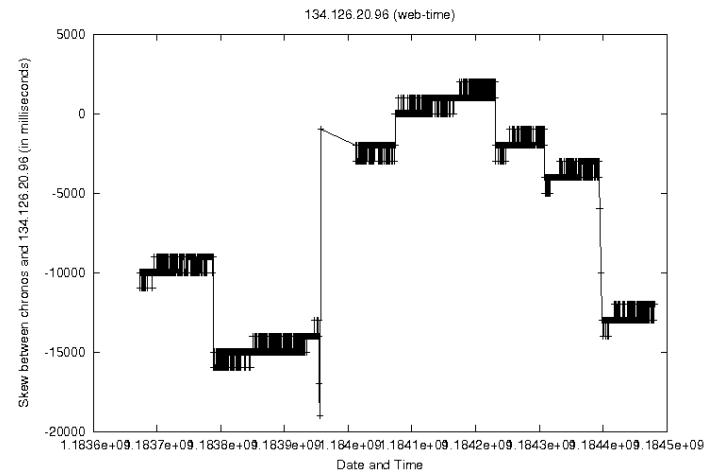
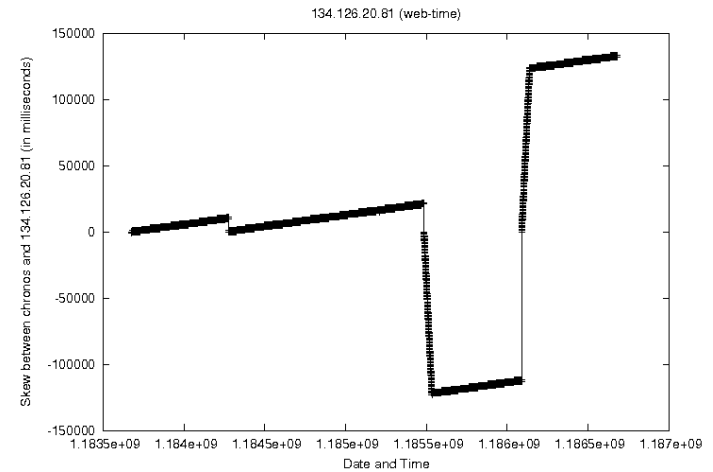
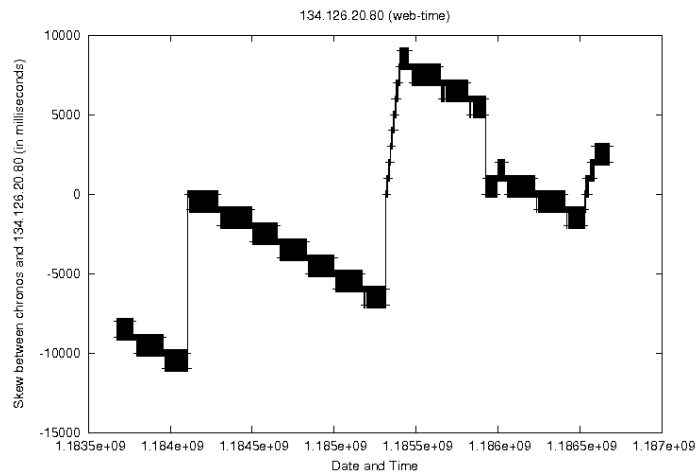
- Only about 74% of the servers synchronized their clocks
  - Makes time correlation problems in forensics investigations likely
- Many “interesting” clock descriptions
  - Causes of clock jumps need to be investigated
  - Clock model probably still holds, but rate of jumps can be very high
- Mixed correlation between different sampling methods
  - Remote measuring of a clock is not reliable
  - Additional methods needed to measure



# The next steps

- We are currently classifying the clock observations
  - Once we have classes, determine (if possible) common traits among hosts (web server, OS)
- Back to the lab:
  - Determine default clock behavior for common operating systems
  - Increase complexity (networking)

# A look ahead: Default Windows clocks



# Future Work

- Determine error rates and certainty of remote measurements
- Establish what sampling rate to use for remote measurements
- Determine more ways to measure remote clocks (NTP, e-mail, ...?)
- Develop ways to filter out individual clocks
- Find causes for why clocks jump
- Can we predict how clocks will behave or tell how they behaved in the past?