



# Comparing 2 implementations of the IETF-IPPM One-Way Delay and Loss Metrics

Sunil Kalidindi, Matt Zekauskas

Advanced Network & Services

Armonk, NY, USA

Henk Uijterwaal, René Wilhelm

RIPE-NCC

Amsterdam, The Netherlands

# Outline

- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions

# The Problem

- The IETF IPPM WG has defined metrics for (type-P) one-way delay and packet losses
  - RFC's 2330, 2679, 2680
- It is the goal of the IPPM-WG to turn these metrics into Internet standards
- This requires 2 independent implementations that are interoperable
- There are 2 implementations of these metrics
- *So what is the problem then?*

## The Problem (2)

- One has to show that the implementations are interoperable
- For metrics, this means that both implementations, measuring along the same path, give the same results
- The results of individual delay and loss measurements depend on the instantaneous condition of the network

## The Problem (3)

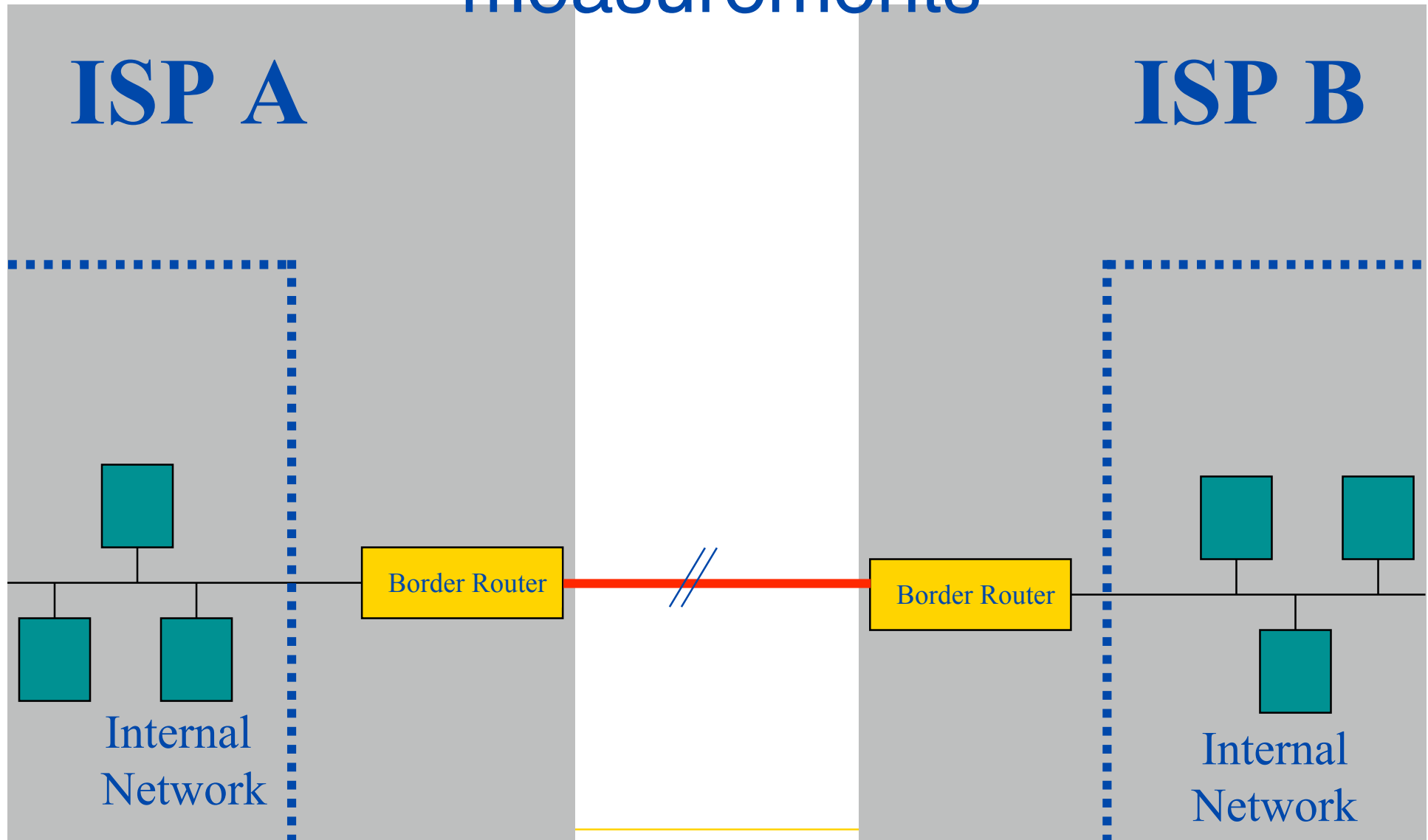
- No direct comparison of individual measurements is possible
- One has to look at distributions instead
  - Distribution of delays and losses over time
  - Patterns of the delays and losses over time
  - Statistical methods
- This presentation is a first attempt at such a comparison

# Outline

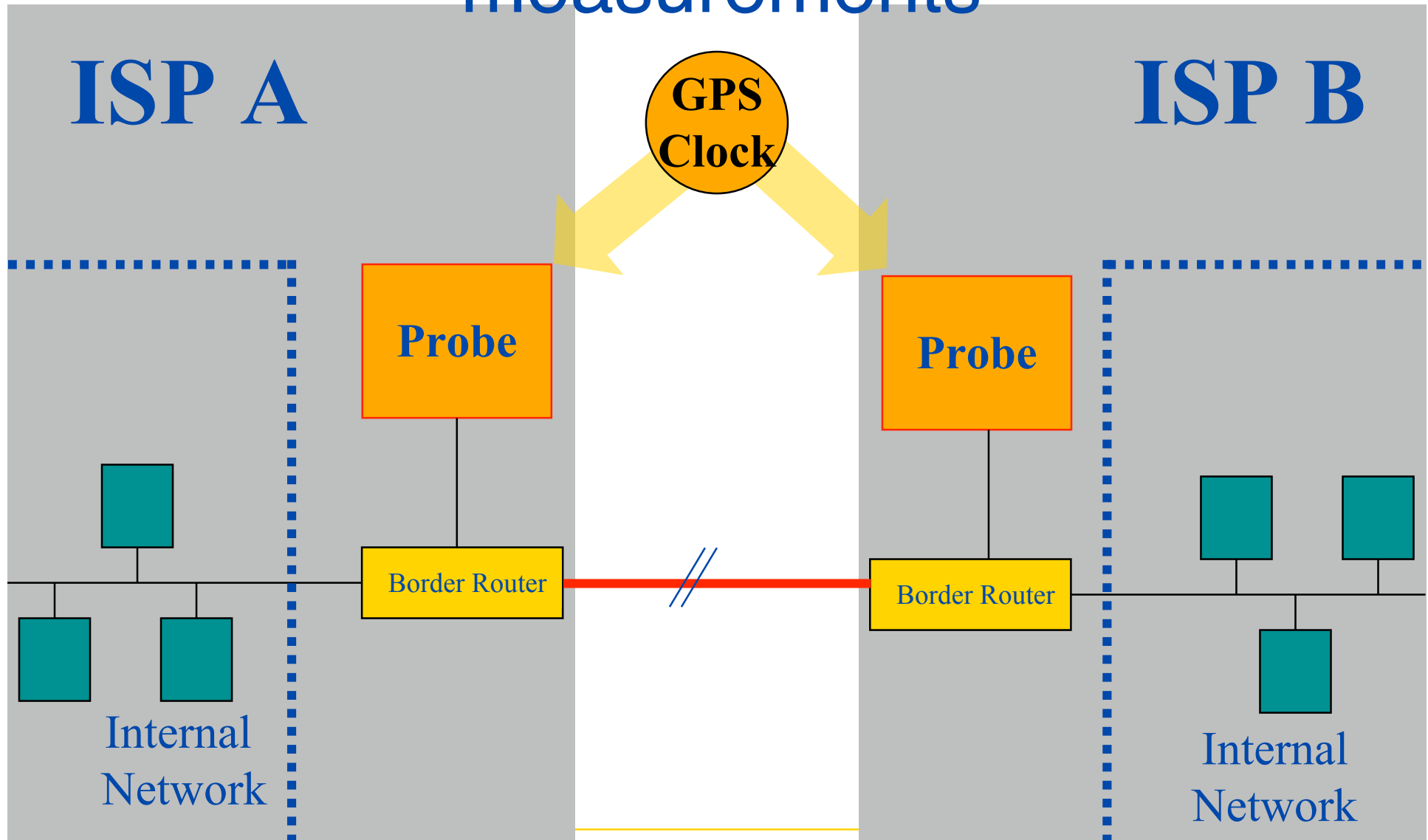
- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions



# One-way delay and loss measurements

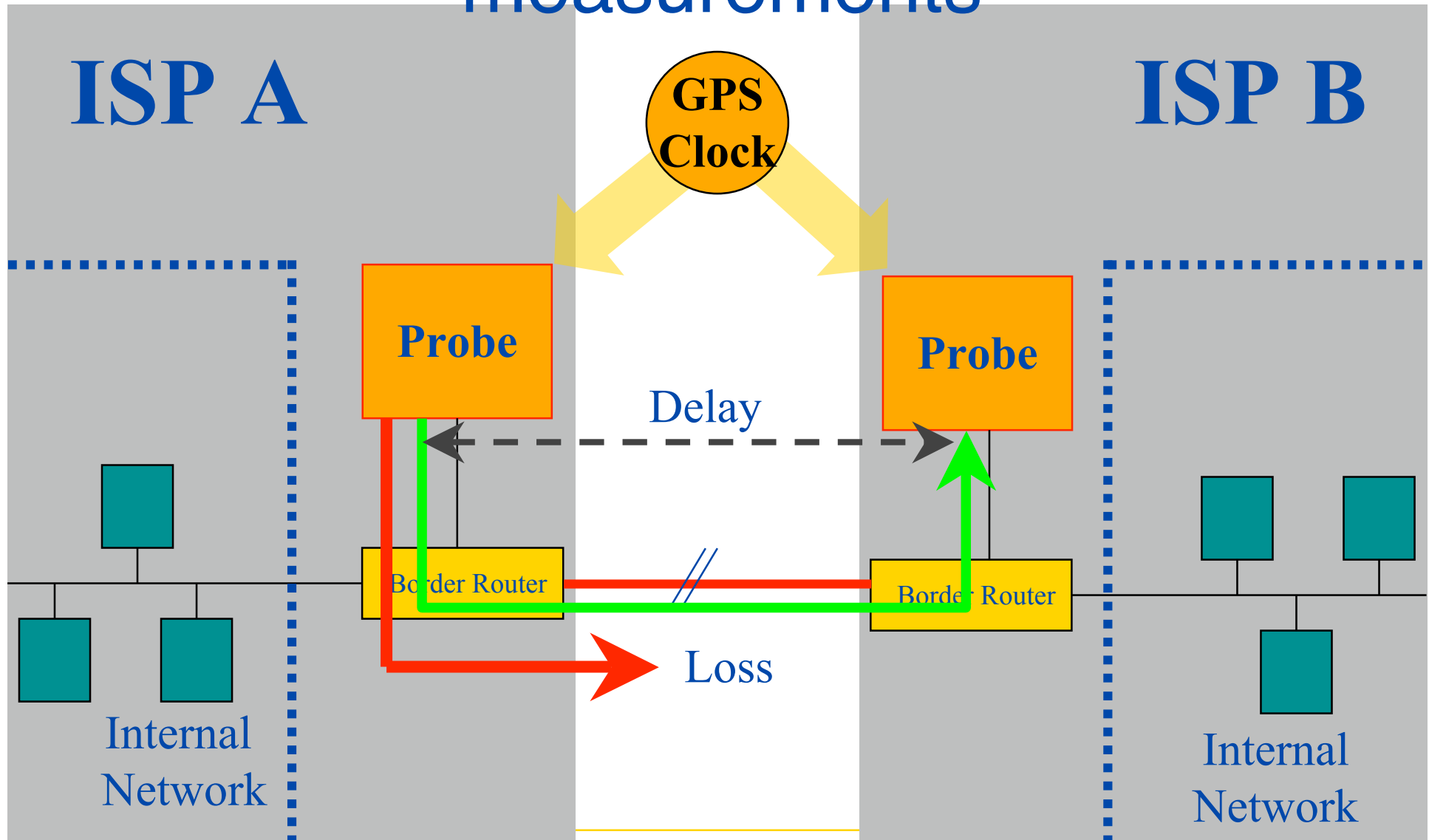


# One-way delay and loss measurements





# One-way delay and loss measurements



# Outline

- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions



# The two implementations

- Advanced Network & Services: Surveyor
  - <http://www.advanced.org/surveyor>
  - Measurement machine: surveyor box
- RIPE-NCC: TTM or Test-Traffic Measurements
  - <http://www.ripe.net/test-traffic>
  - Measurement machines: test-box

# Common features

- Active tests of type-P one-way delay and loss
  - Test packets time-stamped with GPS time
  - UDP packets
    - 40 bytes (total), 2/second: Surveyor
    - 100 bytes, 3/minute: TTM
      - *Later slide*
  - Scheduled according to a poisson distribution
  - Accuracy:
    - Surveyor: Back-to-back calibration: 95% of measurements  $\pm 100 \mu\text{s} \rightarrow 10 \mu\text{s}$  “soon” (in-kernel packet timestamping)
    - RIPE-NCC:  $10 \mu\text{s}$



## Common features (2)

- Concurrent routing measurements
  - Traceroute
  - Only look at the IP-addresses of the intermediate points
- Measurements centrally managed
- Reports on the web

# Common features (3)

## Measurement machines

### Surveyor

- Dell 400 MHz Pentium Pro
- 128 MBytes RAM
- 8 GBytes disk
- BSDI Unix
- TrueTime GPS card and antenna (coax)
- Network Interface (10/100bT, FDDI, OC3 ATM)
- Special driver for the GPS card

### TTM

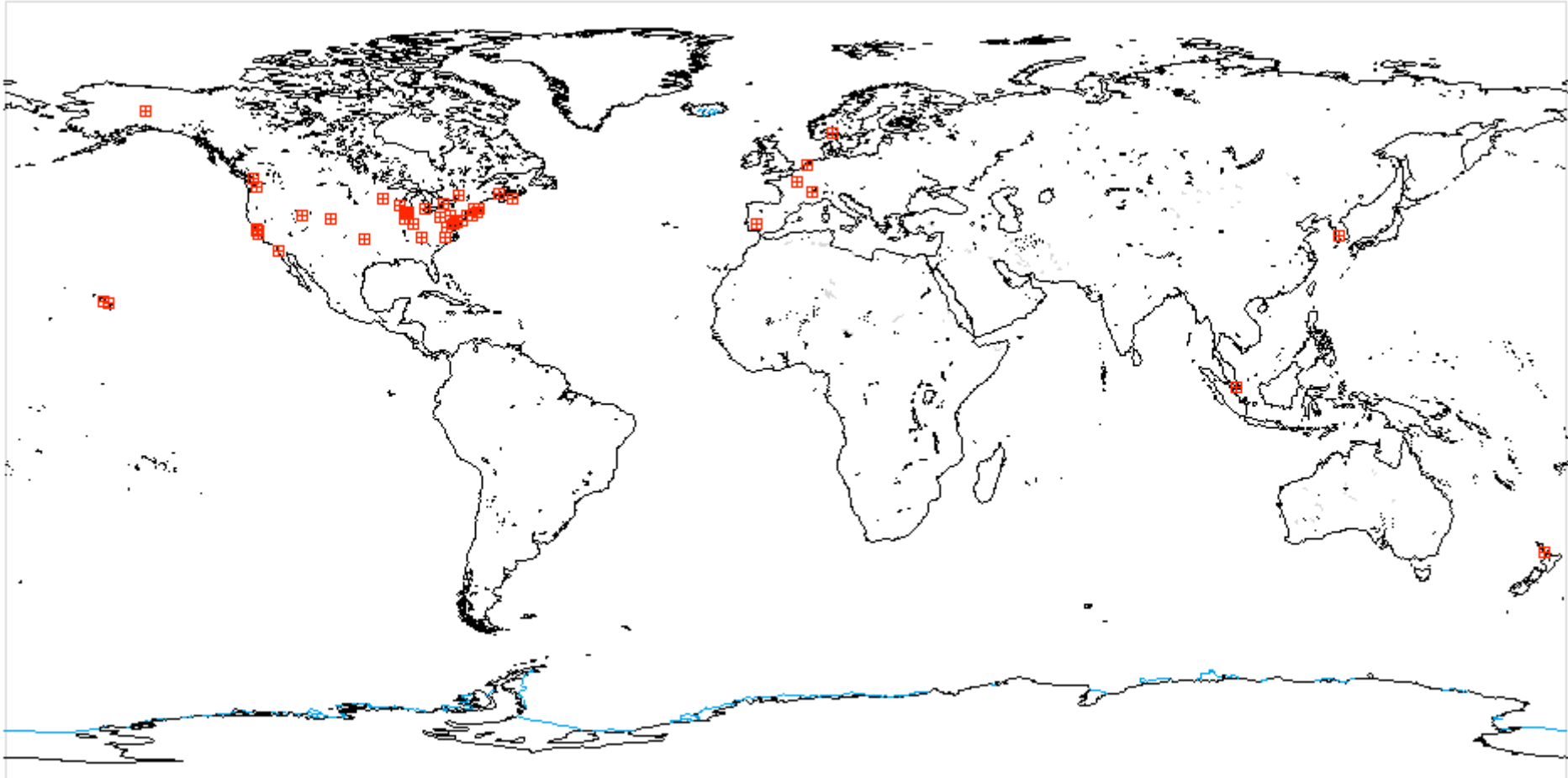
- Pentium, Pentium II, 200...466 MHz
- 32...64 MBytes RAM
- 4...8 GBytes disk
- FreeBSD Unix
- Motorola Oncore GPS receiver and antenna
- Network Interface: 10/100bT
- Special kernel for time-keeping



# Current Surveyor Deployment

- Measurement machines at campuses and at other interesting places along paths (e.g., gigaPoPs, interconnects)
- 71 machines
  - Universities
  - Tele-Immersion Labs
  - National Labs
  - Auckland, NZ
  - ...others
- 2741 paths
  - NASA Ames XP
  - I2 gigaPoPs (some)
  - CA\*net2 gigaPoPs
  - APAN sites
  - Abilene router nodes up with NTP, awaiting GPS

# Surveyor locations







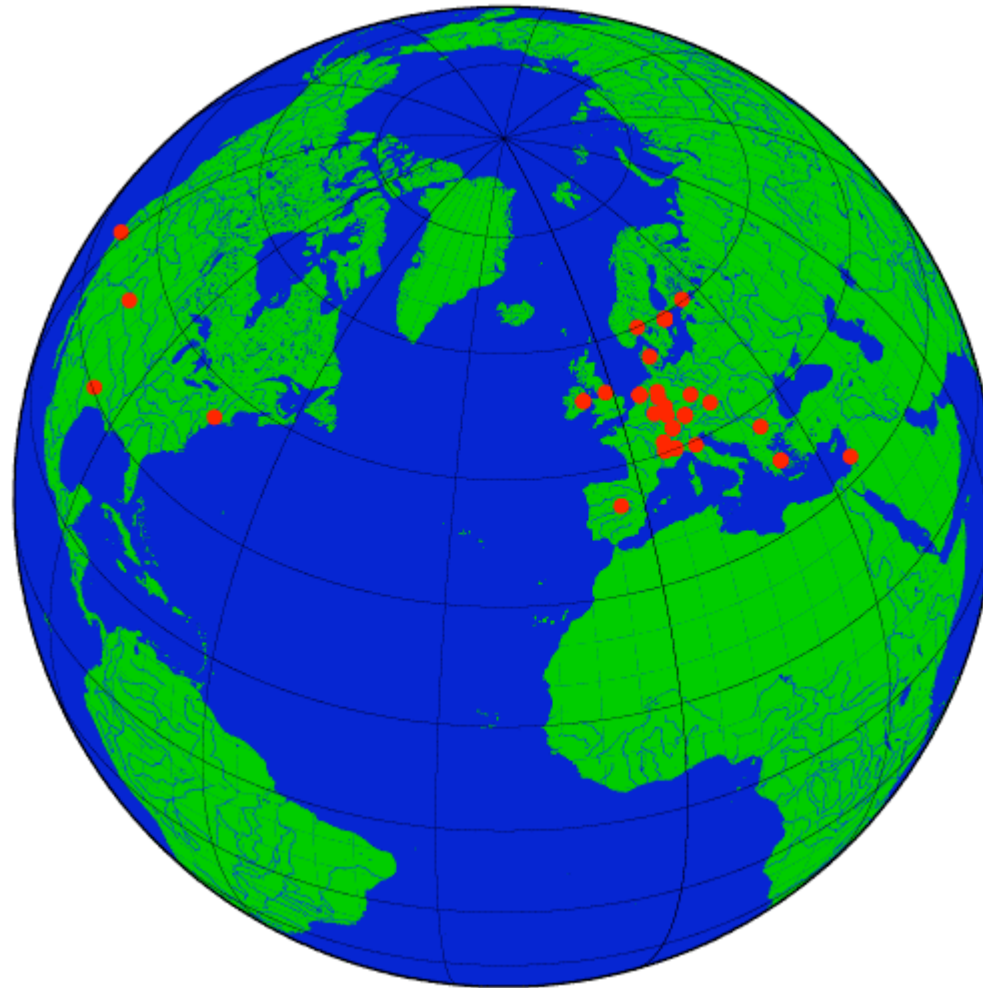
# RIPE-NCC

## Test-Traffic Measurements

- 43 machines
  - RIPE-Membership: ISP's, research networks, etc in Europe and surrounding areas
  - A few sites interested in One-Way Delay measurements outside Europe
  - Common locations with Surveyor:
    - Advanced Network & Systems
    - SLAC (Menlo Park, USA)
    - CERN (Geneva, CH)
- Full mesh with approximately 1600 paths



# Location of the RIPE-NCC Test-boxes

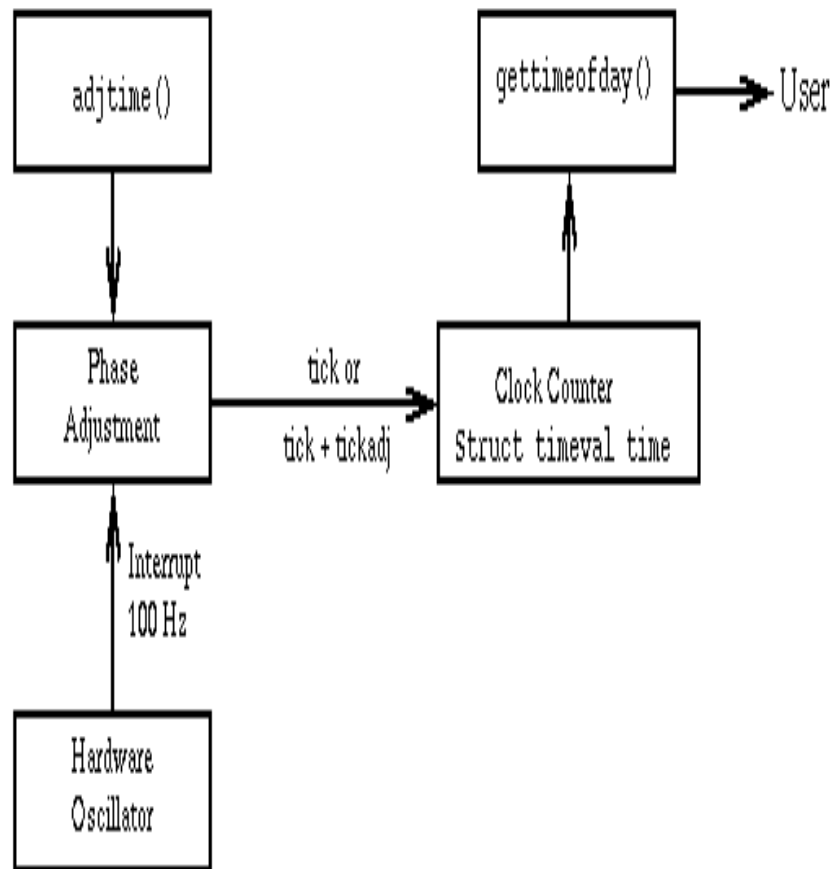


# Outline

- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
  - The key issue to make this work
  - Different approaches
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions

# RIPE-NCC approach

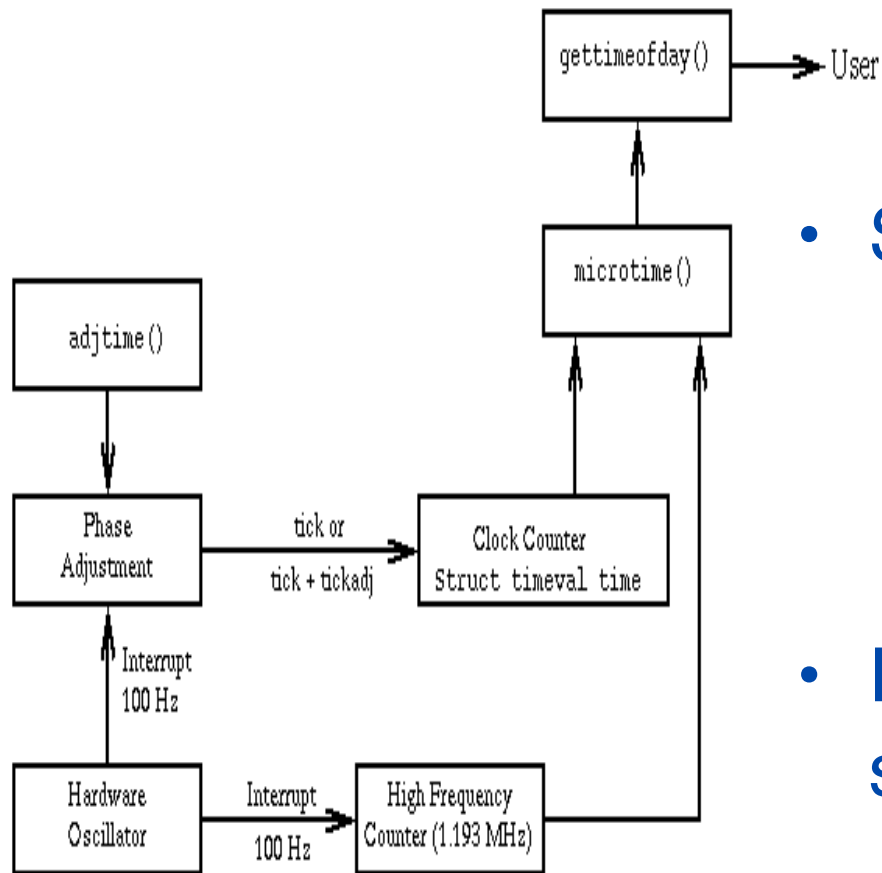
## Unix timekeeping



- Hardware oscillator
  - Interrupt every 10ms
- Software counter
  - Counts # interrupts since 1/1/70
- User access to time
  - `gettimeofday()`, `adjtime()`
- Resolution only 10ms
  - same order of magnitude as typical network delays

# Unix timekeeping (2)

## BSD Clock Implementation

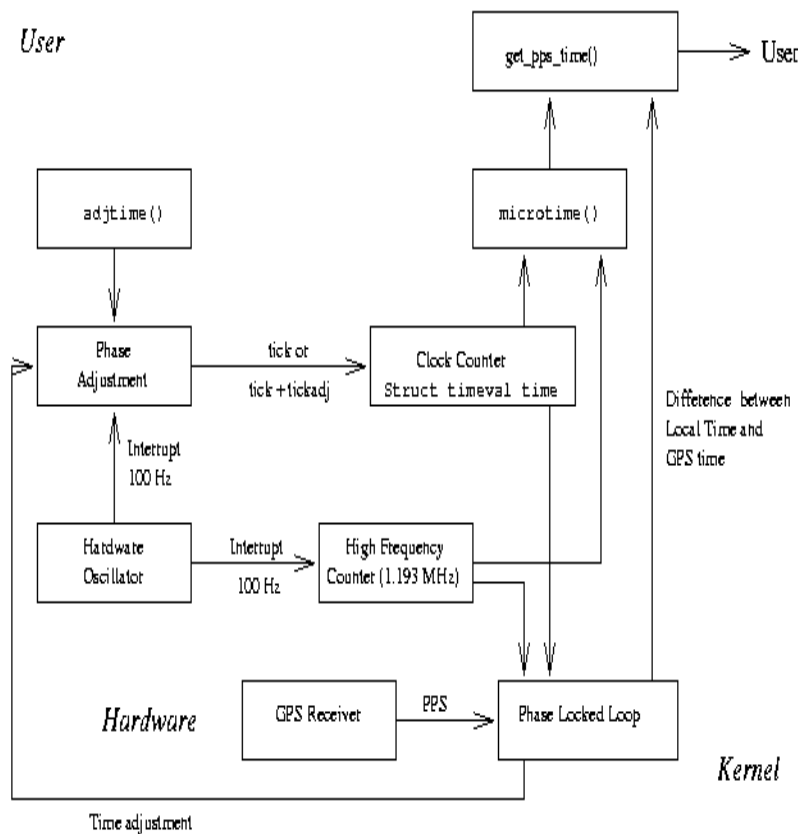


- Second counter
  - Counts at a rate of 1.193 MHz (0.84  $\mu$ s steps)
  - Provides time inside a 10 ms interval
- Resolution increases to 1  $\mu$ s

# Unix timekeeping (3)

- A resolution of 1  $\mu$ s is several orders of magnitude better than the typical delays on the Internet
- But the clocks on two machines will run completely independent of each other
- We have to synchronize our clocks
  - Set the clock to the right initial value
  - Tune it to run at the right speed
  - Correct for experimental effects
- To do that, we need
  - An external time reference source
  - “Flywheel” to keep the clock running at right speed

# Flywheel/Phase Locked Loop



- External time source: GPS
- PLL
  - Determine the difference between internal and external clock
  - Make the internal clock run faster/slower
  - Correct for variations over time
- Kernel level code
- NTP
- Internal clock synchronized to a few  $\mu\text{s}$



# Time-keeping Advanced N&S solution: Hardware

- Wanted off-the-shelf solution
- TrueTime PC[I]-SG “bus-level” card
  - Bancom/Datum has similar product
- Synchronize using GPS satellites
- “Dumb” antenna (receiver on card)
- Oscillator & time of day clock on-board
- Claim: within 1  $\mu$ s of UTC
- Major disadvantage: cost (\$2500 US)



# Time of Day: Software

- System clock ignored
- Must access card for time-of-day
- Deployed software
  - timestamp at user-level
  - read via `ioctl()` (implies bus transaction)
  - Calibration error of  $10\ \mu\text{s}$  (loose), if there is no other load
  - $100\ \mu\text{s}$  is a loose bound for 80 peers

# Outline

- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions

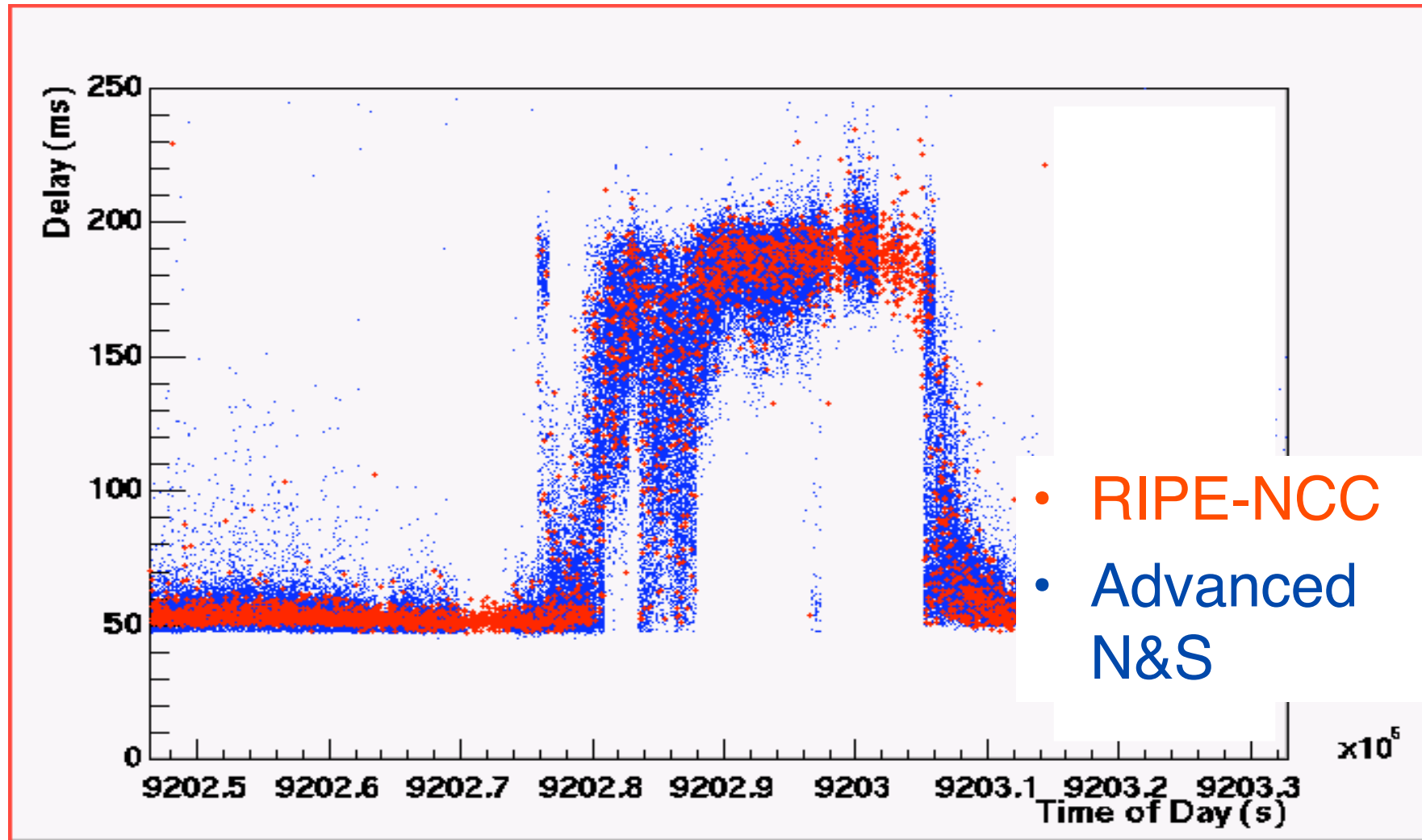


# Comparing the data

- RIPE-NCC and Advanced N&S exchanged boxes in October 1998.
- Boxes are on the same network segments at both sides
- Data taking since October 1998.
- Other sites with both a Surveyor and TTM box:
  - CERN (Spring '99)
  - SLAC (Fall '99)

# Raw Data

## 20 hours

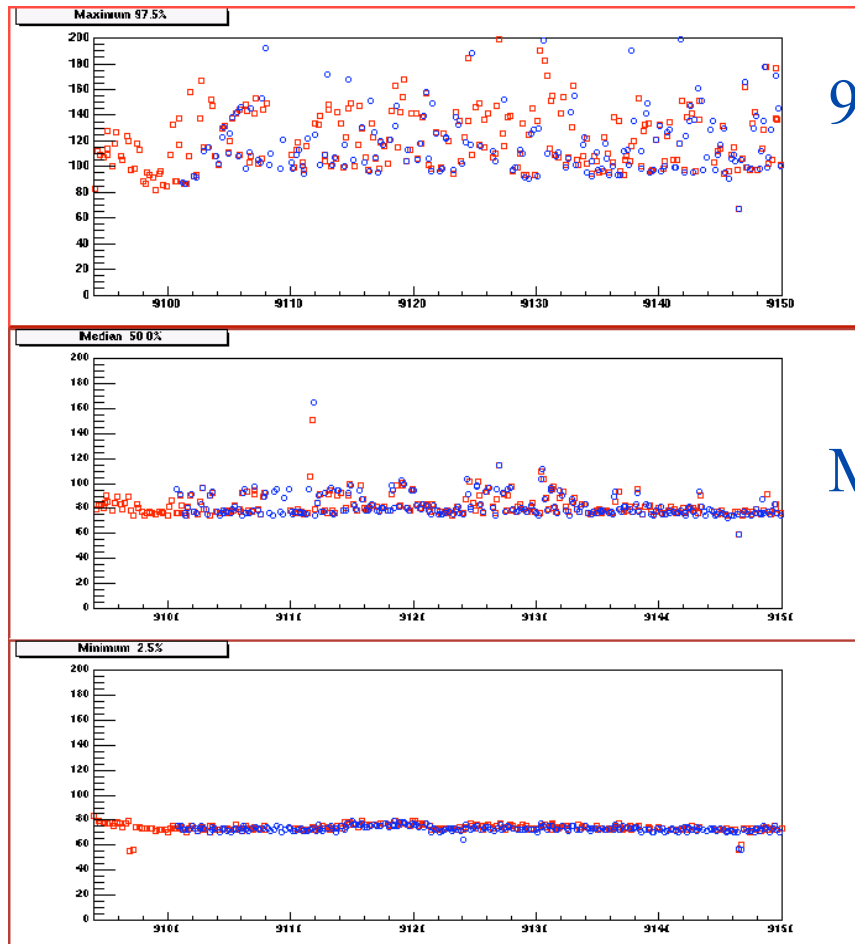


# Percentile delays over a 2 month period



Advanced N&S-data

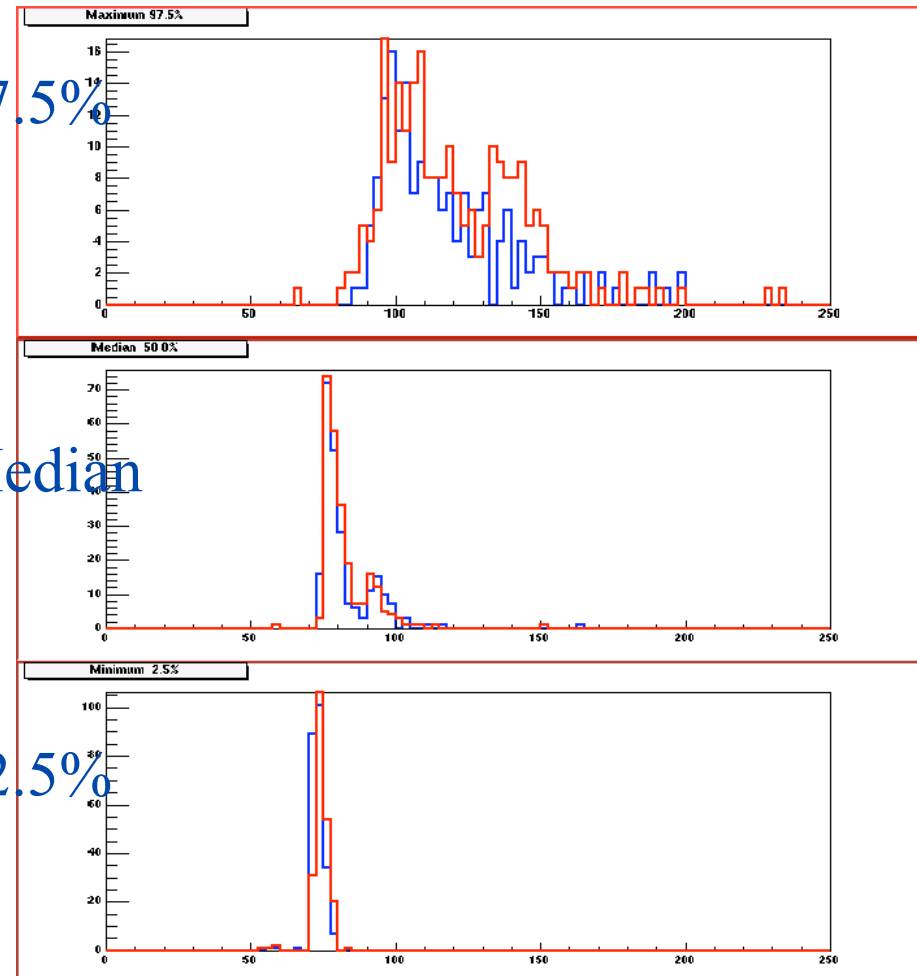
RIPE-NCC-data



97.5%

Median

2.5%



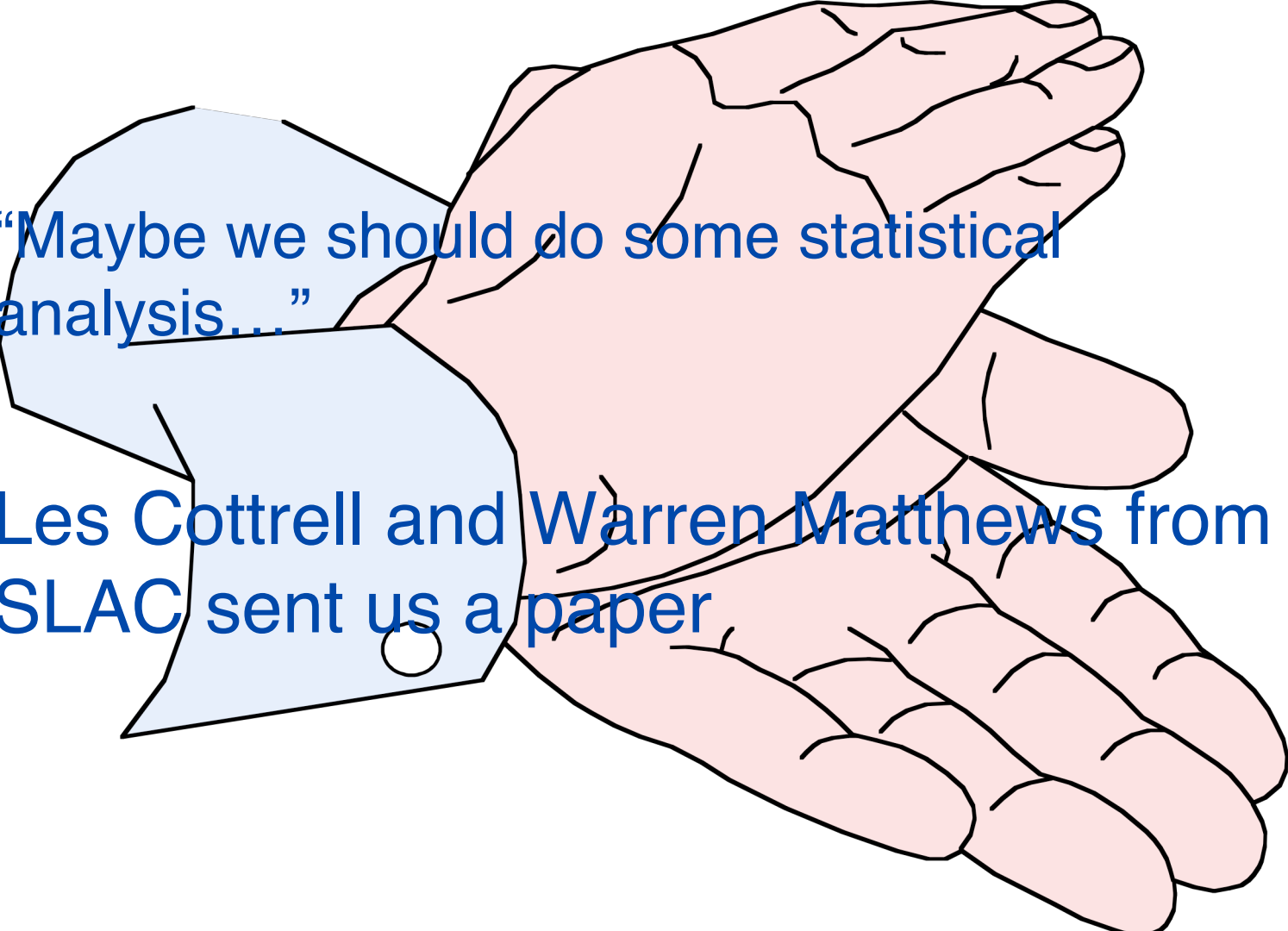
# Outline

- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions

# Statistical approach

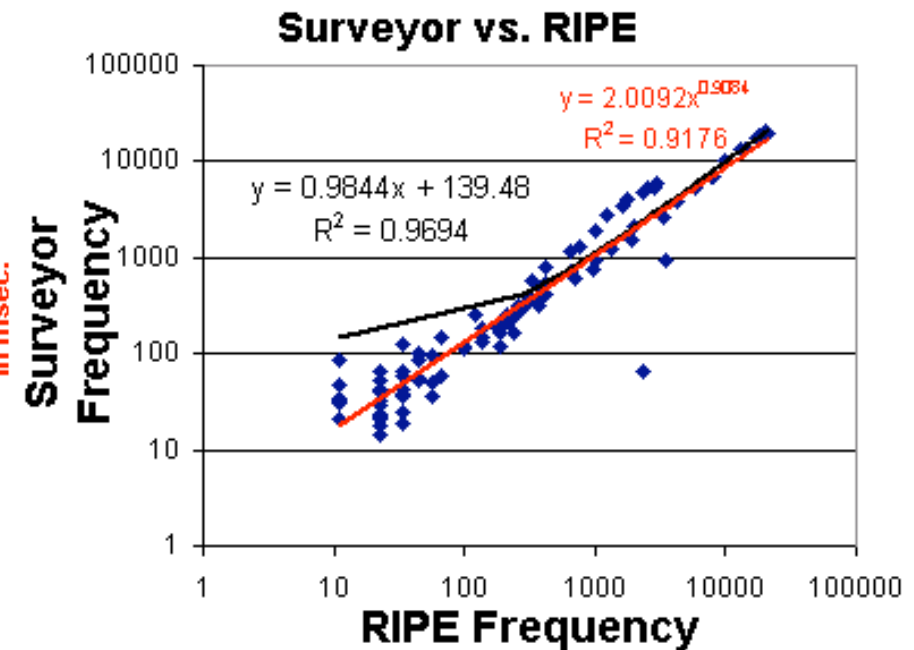
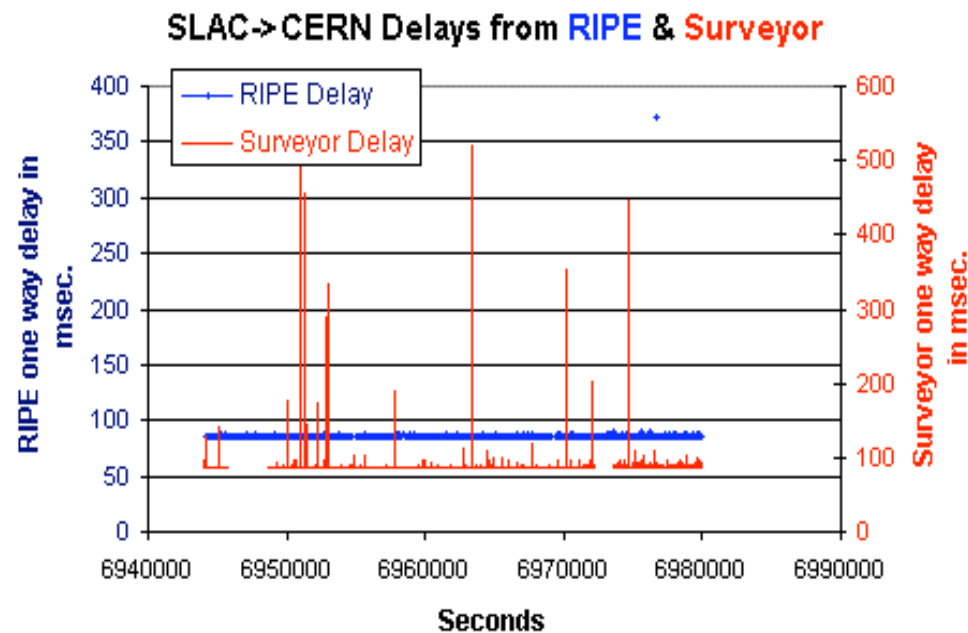
- “Maybe we should do some statistical analysis...”

# Statistical approach

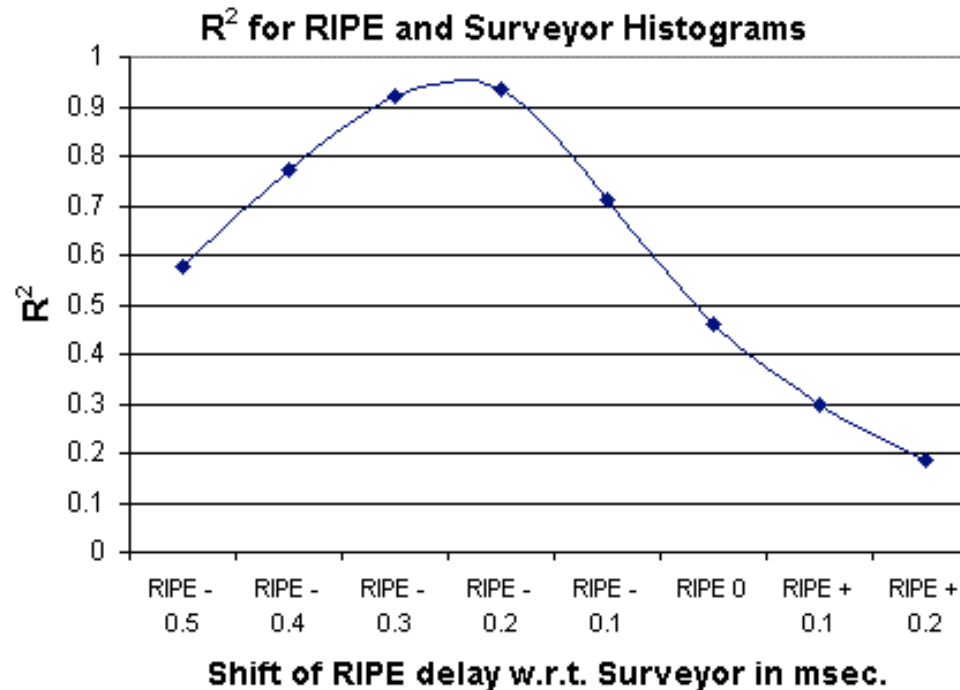
- 
- A large, pink, cartoonish hand is shown with its palm facing up. A light blue speech bubble with a black outline is positioned to the left of the hand, with its tail pointing towards the palm. The speech bubble contains two bullet points.
- “Maybe we should do some statistical analysis...”
  - Les Cottrell and Warren Matthews from SLAC sent us a paper



# SLAC $\Rightarrow$ CERN



# Matching the delays?



- Vary RIPE-NCC delays in the histograms
- Find the value where the 2 sets agree best
- Decrease RIPE-NCC delays by 0.2 ms
- Why?

# Outline

- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions

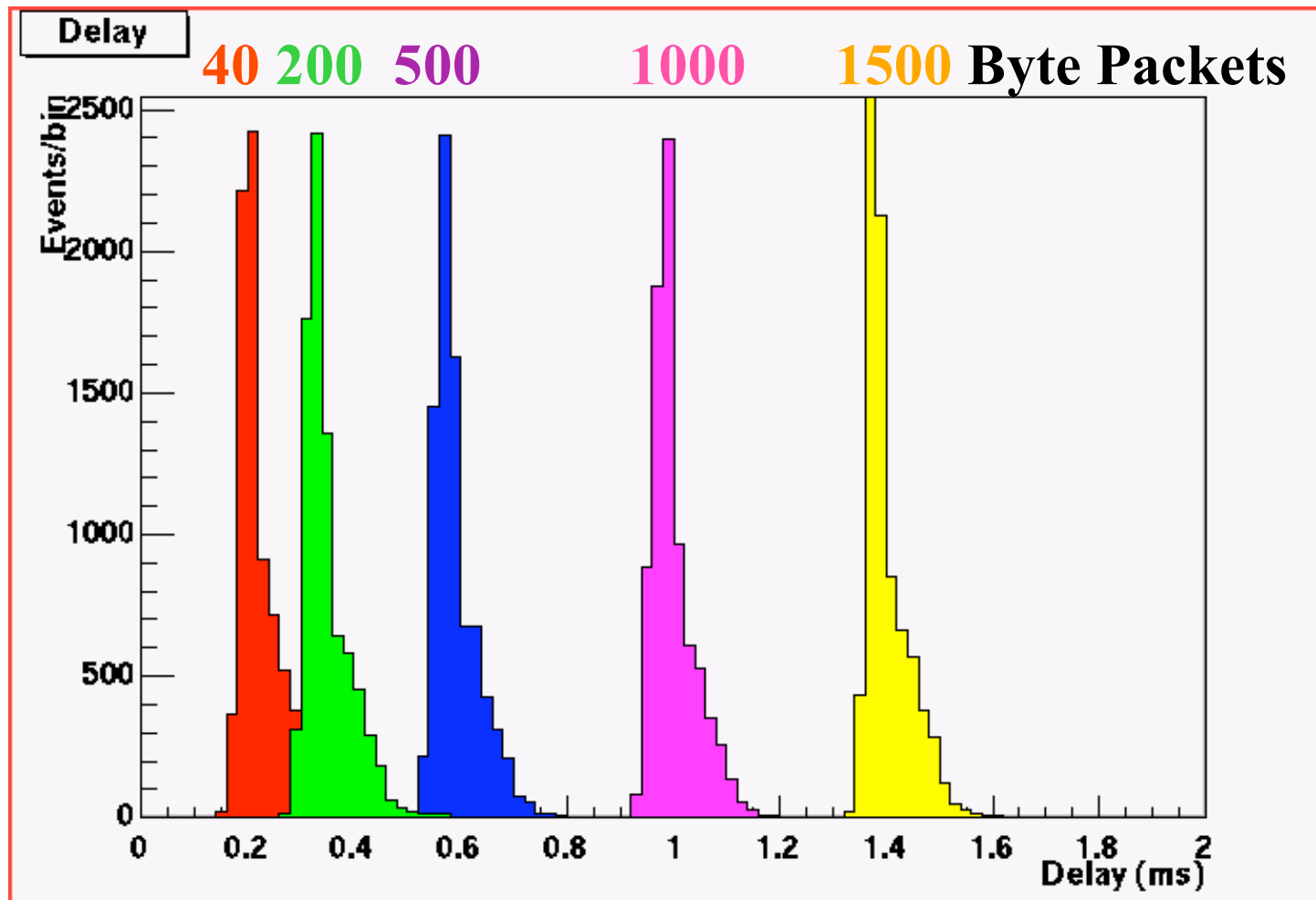


# Effects of the packet-size on delays



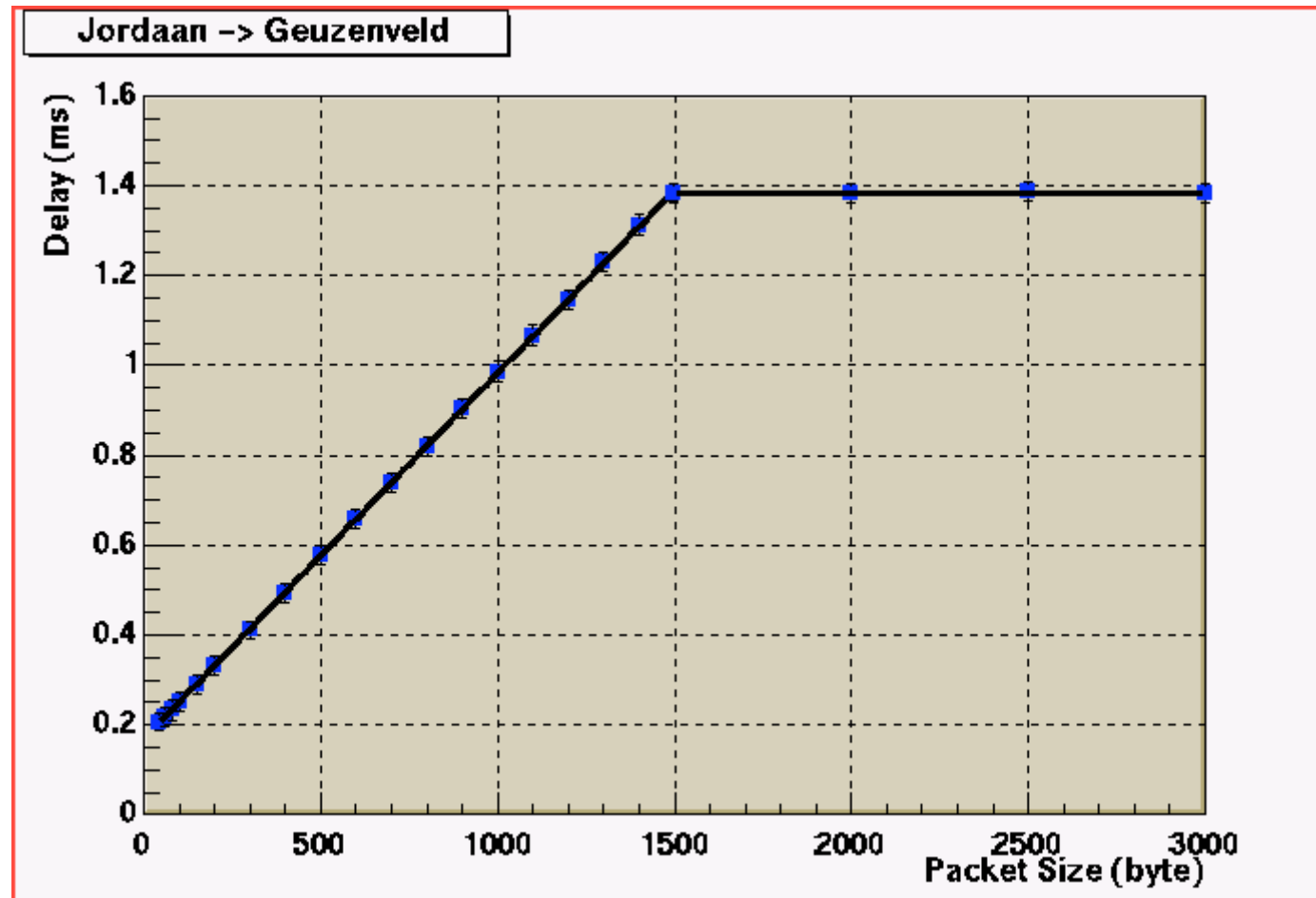
- Obviously, larger packets take longer to transmit
- But are packets treated differently?
- 3 experiments:
  - Local network (1999)
  - Transatlantic network
    - Advanced-RIPE (1999)
    - SLAC-CERN (2000)

# Local Network



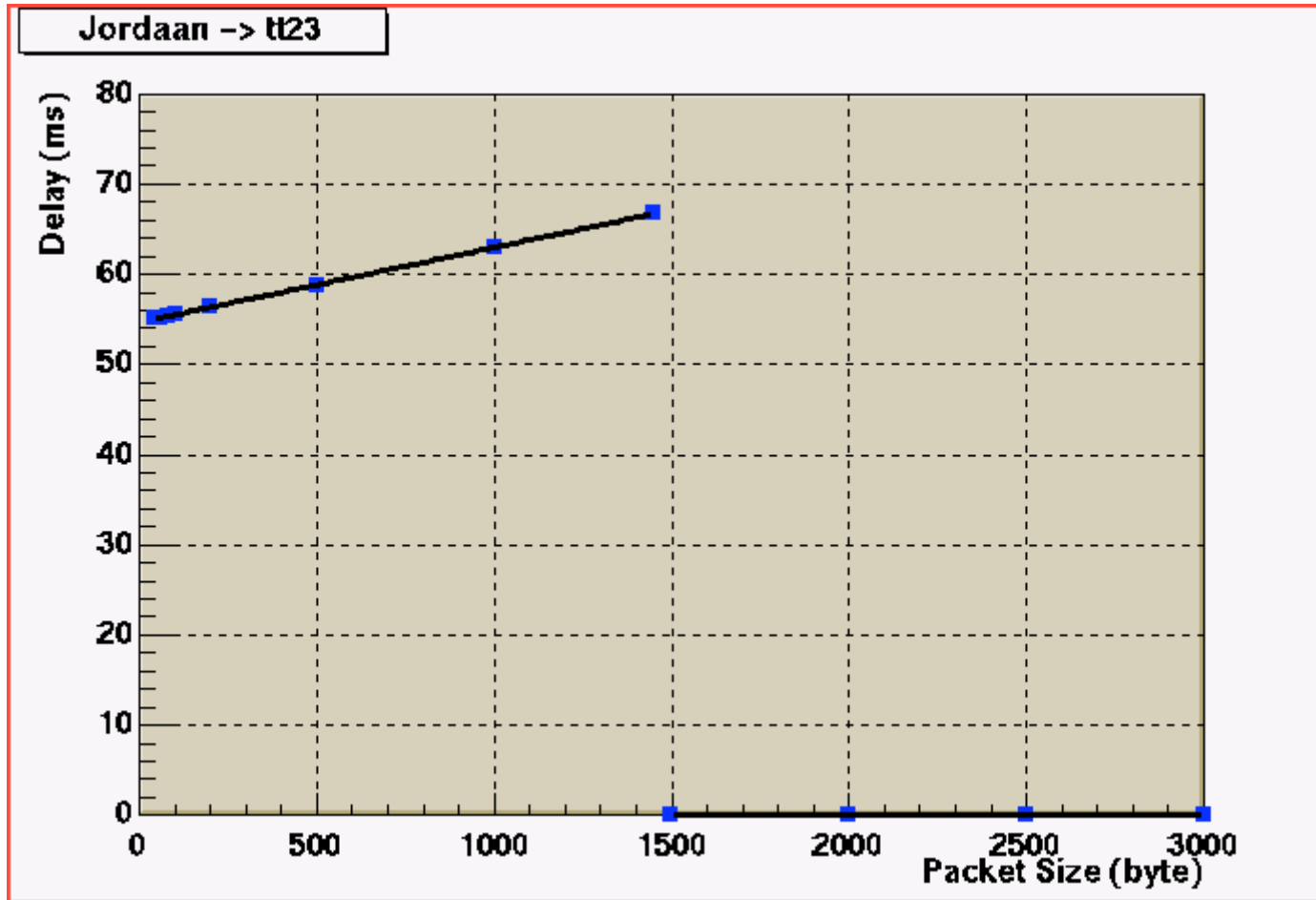
- Similar shapes but shifted in time

# Local Network



- Linear up to MTU, then fragmentation

# Trans-Atlantic connection



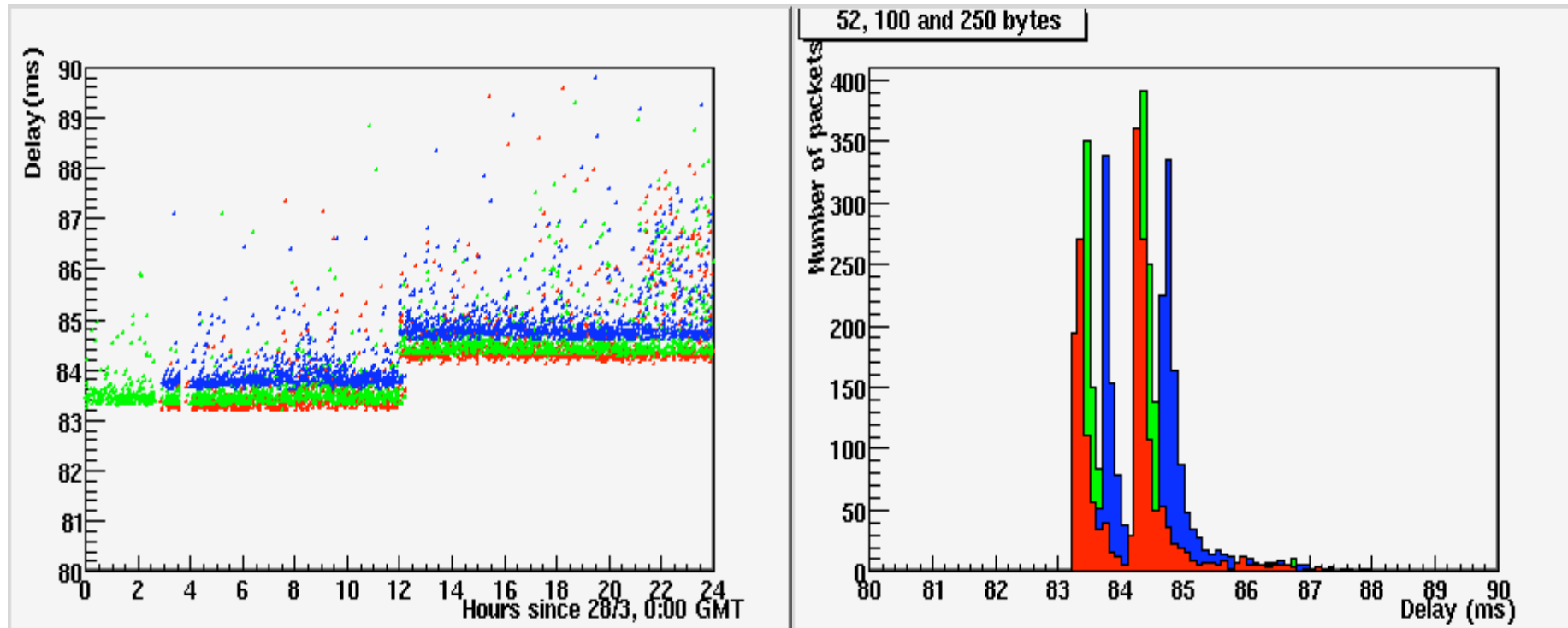
- Linear up to MTU, larger packets dropped

# Delays versus packet-size

- Model  $D = a_0 + a_1 B$ , for  $B < MTU$
- Local throughput:  
 $a_1 = (8.09 \pm 0.10) 10^{-4} \text{ byte/ms} \Rightarrow \text{throughput} = (1.235 \pm 0.015) \text{ Mbyte/s}$
- Transatlantic connection throughput:  
 $a_1 = (8.47 \pm 0.05) 10^{-3} \text{ byte/ms} \Rightarrow \text{throughput} = (118 \pm 2) \text{ kbyte/s}$
- Does this explain the difference observed in the CERN-SLAC data?

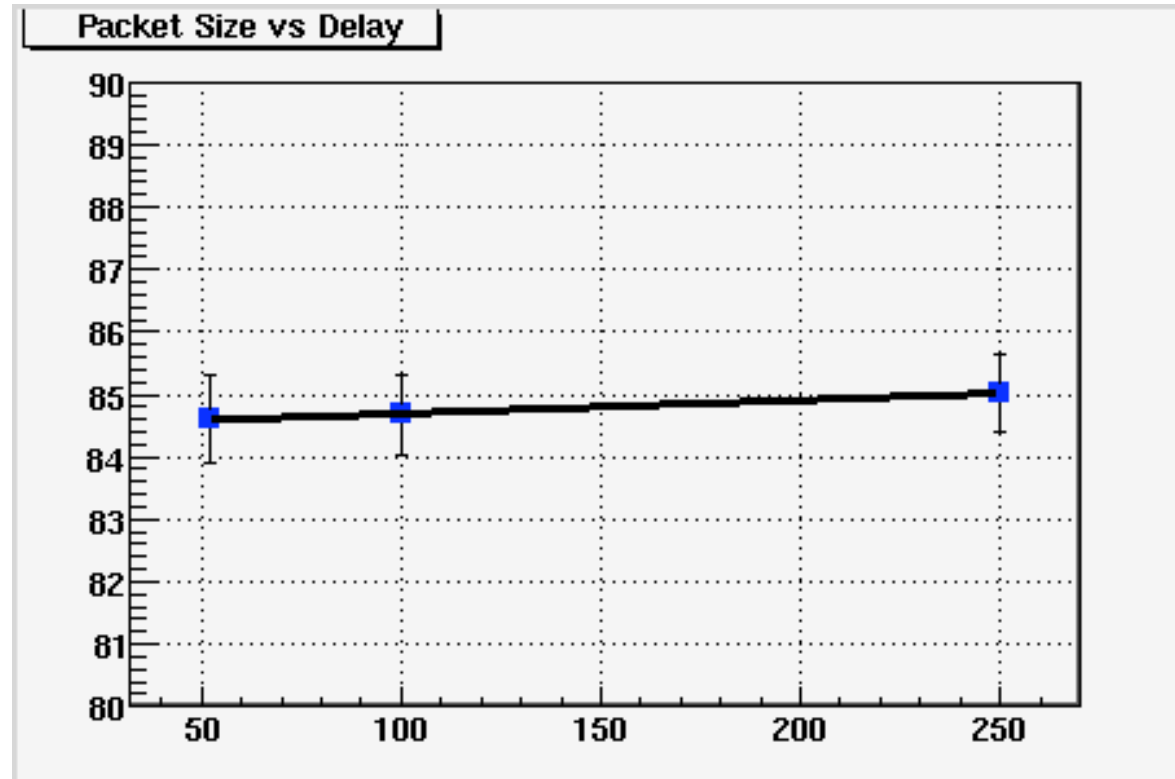


# SLAC $\Rightarrow$ CERN data



- SLAC  $\rightarrow$  CERN, March 28, 2000
- Split data into 2 sub-samples

# SLAC $\Rightarrow$ CERN data



- Extrapolate to 60 bytes difference: 0.14 ms



# SLAC $\Rightarrow$ CERN data

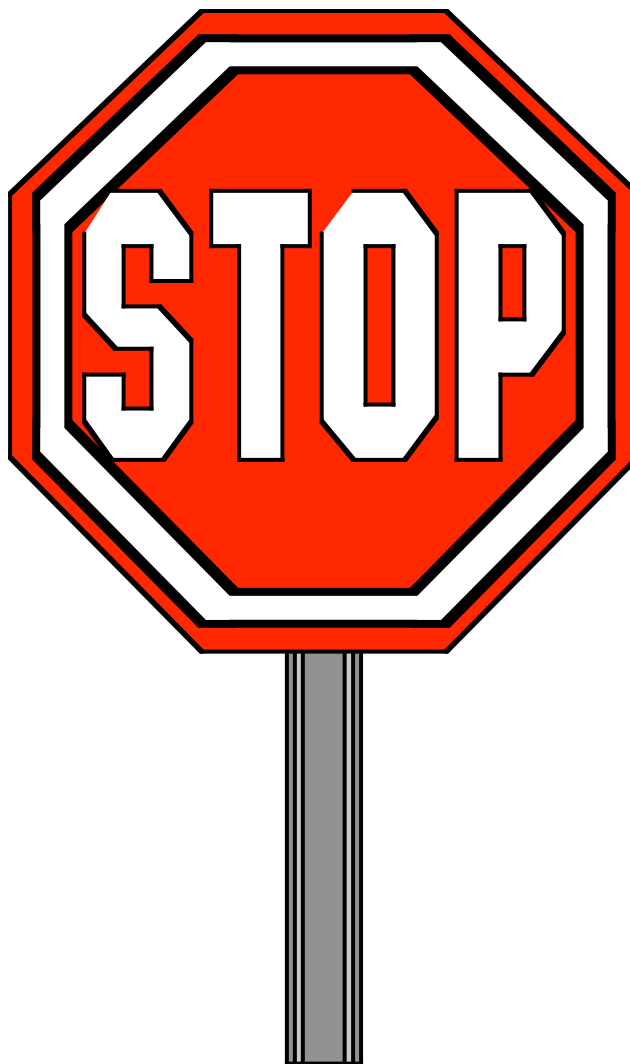
- 0.2 ms difference
- 0.14 ms can be explained by differences in packet-size
- Further investigation needed on the remaining 0.06 ms
- But this is less than 0.1% of the observed delay
- Experimental errors  $O(0.02)$  ms.

# Outline

- The problem
- Theory behind one-way delay and loss measurements
- The two experiments
- Time-keeping
- Comparing raw-data
- Statistical approach to comparing data
- Effect of packet-sizes on delays
- Outlook and conclusions

# Conclusion and outlook

- All tests seem to indicate that the 2 setups measure the same delays and losses
- Is this sufficient to meet the two independent implementations requirement?
  - Look at more paths, look for more unusual occurrences
  - Any other statistical tests that people consider useful?
- Look at the effects of different sampling frequencies
- These slides will be at <http://www.ripe.net/test-traffic> on Monday April 10



# Phase Locked Loop

- A PLL maintains a sense of time over a long period
  - Advantage: small glitches will not immediately affect the clock
  - Disadvantage: it takes a while before the clock is synchronized
- The time difference between a *pair of clocks* will drift around a constant
  - Our software has a correction for this effect

# Implementation

- NTP
- Kernel level implementation of the PLL
- Home-built GPS receiver
  - Based on Motorola's Oncore-VP