# Implementing a NTP-Based Time Service within a Distributed Middleware System

*ACM International Conference on the Principles and Practice of Programming in Java (PPPJ `04)*

**Hasan Bulut**

# Motivation

- Collaboration environments make use of distributed middleware systems to achieve collaboration among people geographically distributed.
- Time ordering of events generated by entities within a distributed infrastructure is far more difficult than time ordering of events generated by a group of entities having access to the same underlying clock.
- The Network Time Protocol has been around for more than a decade and being used in the Internet to let people to synchronize their computer clocks with atomic time servers clocks.
- So we would like to achieve time-ordering of events generated at different entities by providing a time service to the distributed middleware system.

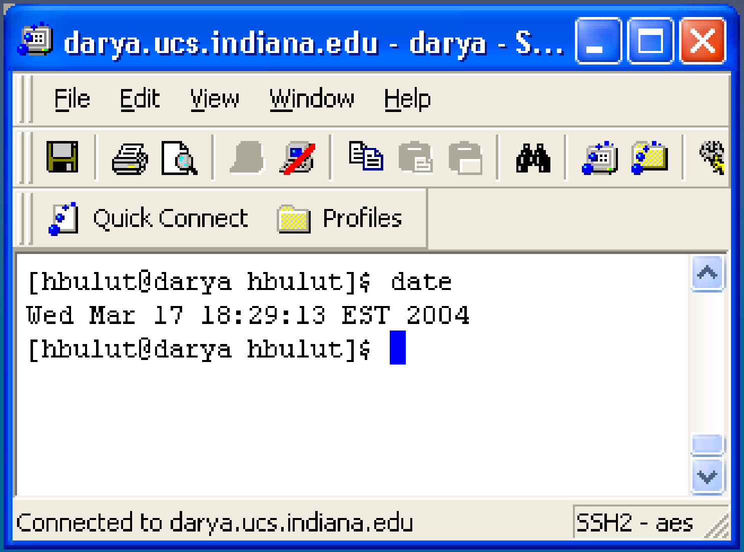
# Example Application

- An archiving system designed to archive the streams generated in a audio/video conferencing session(s)
- In a collaboration session, streams from these sources need to be synchronized for archival and replay purposes.
- Real-time constraints for audio/video conferencing applications can vary anytime between 30-100 msec, depending on the jitter and in inter-packet arrivals.
- In a distributed event brokering environment, RTP packets are considered as events.
- Time-ordering of these events are necessary to achieve an efficient archival/replay mechanism.

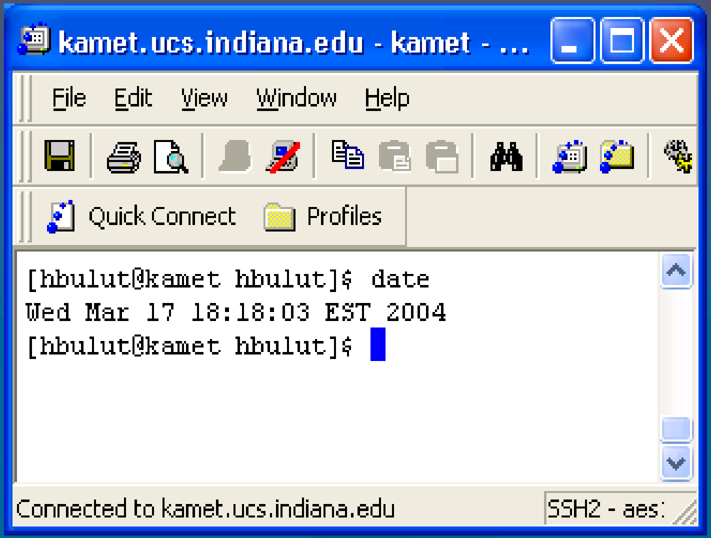
# Computer Clocks I

- Computer timers keep time by using a quartz crystal and a counter.
- Two types of clocks on a computer: hardware and software.
- What causes computer clocks get out of sync?
  - run at different rates
  - manufacturing defects
  - environment: temperature change, electric and magnetic interference
  - CPU load
  - ill-behaved program
- So how much do they get out of sync?

# Computer Clocks II


darya.ucs.indiana.edu - darya - S...
File Edit View Window Help
Quick Connect
Profiles
Connected to darya.ucs.indiana.edu
SSH2 - aes

```
[hbulut@darya hbulut]$ date
Wed Mar 17 18:29:13 EST 2004
[hbulut@darya hbulut]$
```


kamet.ucs.indiana.edu - kamet - ...
File Edit View Window Help
Quick Connect
Profiles
Connected to kamet.ucs.indiana.edu
SSH2 - aes:

```
[hbulut@kamet hbulut]$ date
Wed Mar 17 18:18:03 EST 2004
[hbulut@kamet hbulut]$
```

- darya.ucs.indiana.edu - Wed Mar 17 18:29:13 EST 2004
- kamet.ucs.indiana.edu - Wed Mar 17 18:18:03 EST 2004

# Approaches to Order Events and Synchronize Clocks

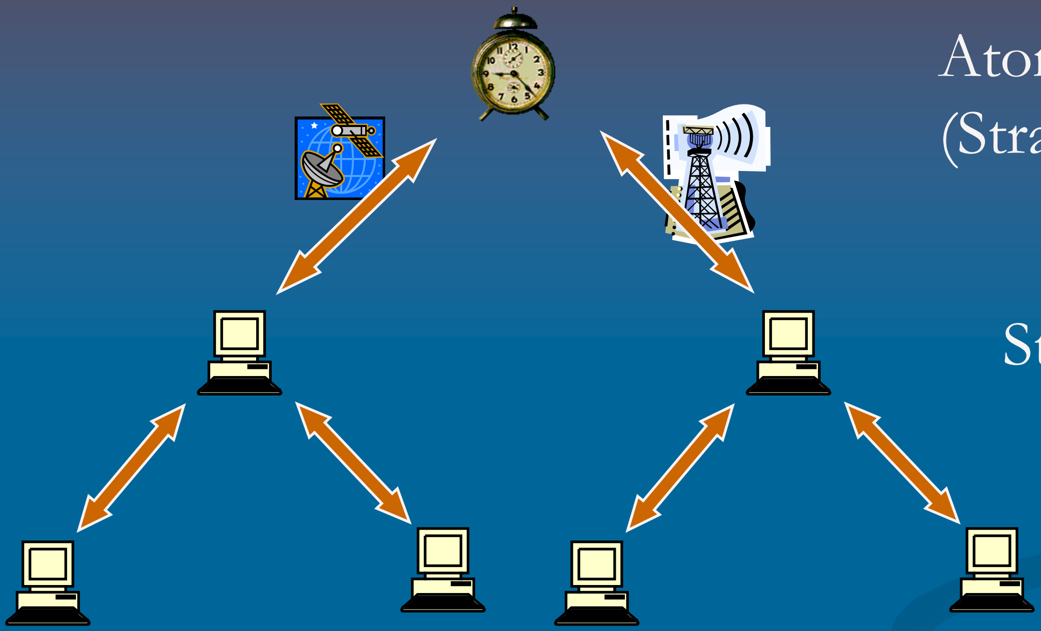
- Approaches can be divided into two categories: software (SW) and hardware (HW)
- Logical timestamps (SW)
  - Lamport timestamps
  - Vector clocks
- Synchronizing system clocks (SW)
  - Cristian’s algorithm
  - Berkeley algorithm
  - Averaging algorithms (decentralized algorithms)
- Network Time Interface (NTI) M-Module – custom VLSI chip with interfaces to GPS receivers. (HW)

# Network Time Protocol (NTP)

- NTP is used to synchronize timekeeping among a set of distributed time servers and clients.
- Formal specification for version 3 can be found in RFC-1305

  http://www.ietf.org/rfc/rfc1305.txt
- It defines the architectures, algorithms, entities and protocols used by NTP.
- NTP is built on UDP/IP, which provides a connectionless mechanism.

# NTP (con't)



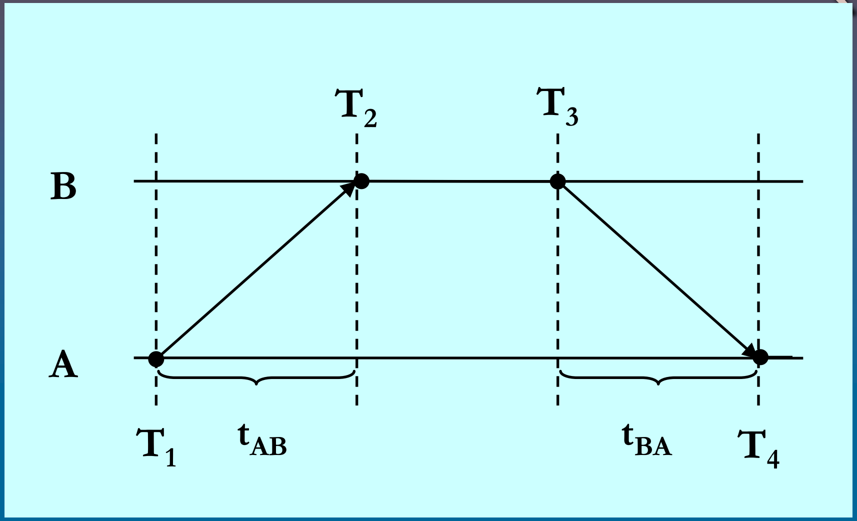
Atomic time clock (Stratum-0)

Stratum-1 time servers

Stratum-2 time servers or clients

- Stratum number is an integer indicating the distance from the reference clock.
- Primary reference sources are synchronized by wire or radio.

# NTP (con't)



- A: Client B: Time Server
- Transmission time from A to B ($t_{AB}$) is $t_{AB} = T_2 – T_1$
- Transmission time from A to B ($t_{BA}$) is $t_{BA} = T_4 – T_3$

Roundtrip delay (δ) and offset (Θ ) can be computed as follows

- Roundtrip delay (δ) $\boldsymbol{\delta = (T_4 - T_1) - (T_3 - T_2)}$
- Offset (Θ) $\boldsymbol{\Theta = [(T_2 - T_1) + (T_3 - T_4)] /2}$

# NaradaBrokering (NB)

- An event brokering middleware. Java Message Service (JMS) compliant
- Supports publish-subscribe messaging model with a dynamic collection of brokers.
- Provides services for TCP, UDP, Multicast, SSL, HTTP, HTTPS, raw RTP clients and storage services - SQL and file based.
- Supports audio-video conferencing, communication through firewalls and proxies, software multicast.
- Publish/subscribe of A/V
  - A A/V stream is regarded as a "topic" and each RTP packet from this stream is regarded as an "event" for this topic.
  - Only the sender of this stream can "publish" A/V events to this topic. Other endpoints need to subscribe to this topic in order to receive the stream.
- http://www.naradabrokering.org

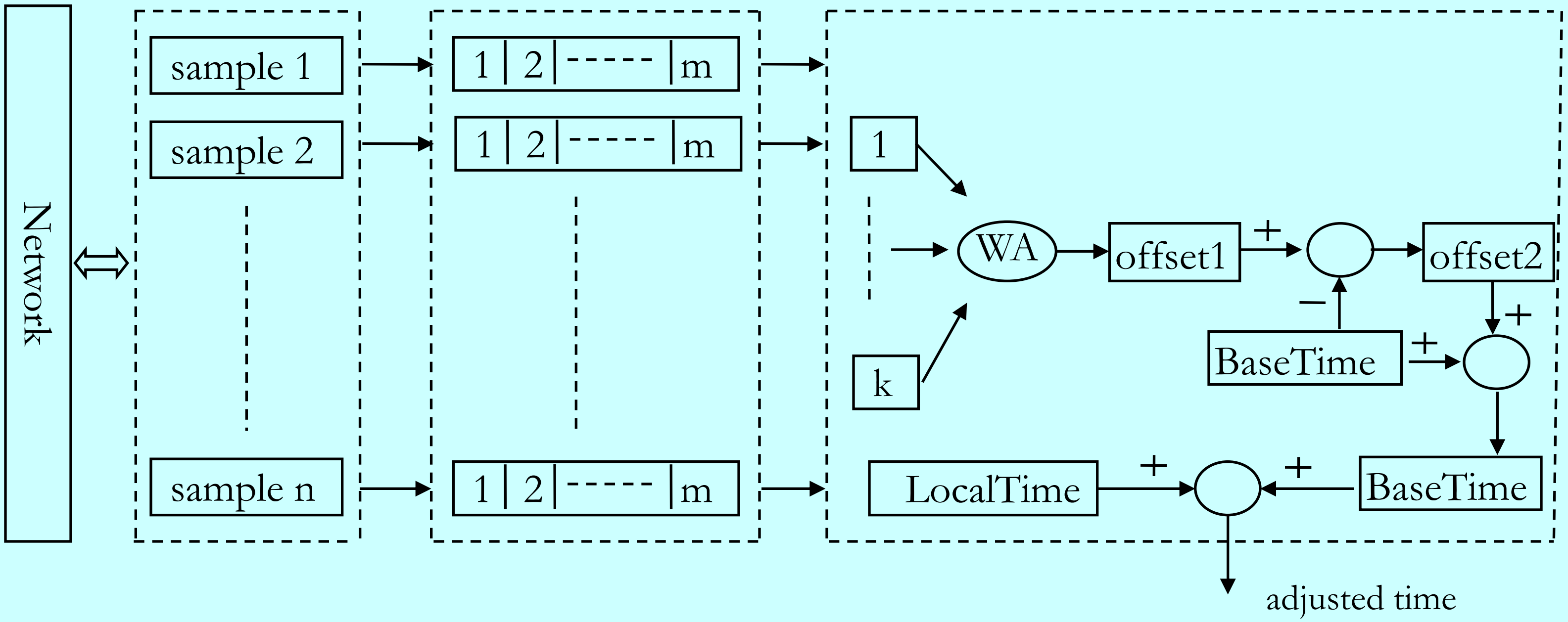
# NB Time Service

- NB Time Service runs in the kernel of NaradaBrokering.
- There are two important parameters
  - Number of time servers
  - Time interval that the NB Time Service should run
- A number of time servers can be specified. There is no restriction.
- By default, NB Time Service uses 8 stratum-1 time servers and time interval for the service is set to 30 seconds, which means NB Time Service updates the offset every 30 seconds.

# NB Time Service (con't)

- Unlike NTP daemons, NB Time Service does not change underlying system clock. Because
  - changing underlying system clock requires administrator privileges
  - the objective of this time service is to provide a mechanism to be able to time-order the events generated within NaradaBrokering.
- Provides an interface, getTimestamp(), to get updated time in milliseconds.
- Entities generating events in the system should utilize Time Service to timestamp the events.

# NB Time Service (con't)



Phase 1:
collecting
samples

Phase 2:
filtering & adding
to the queue

Phase 3:
establishing candidate list
& computing offset

offset1: Offset computed with regard to local computer time
offset2: Offset computed with regard to adjusted time

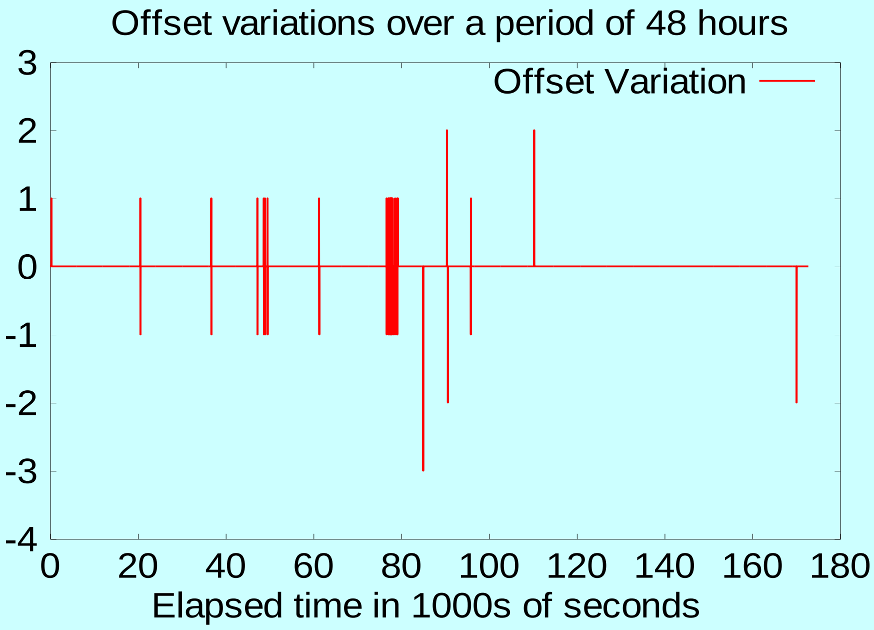
# NB Time Service (con't)

- Phase 1: Getting samples from NTP time servers.
  - NTP messages are sent to the time servers provided one by one.
  - UDP connection is used to send these messages and the timeout is set to 500 ms.
- Phase 2: Filtering and adding to the queue.
  - Timestamps are checked to validate the received NTP reply message.
  - For each time server a separate queue is provided to keep the previous NTP samples. Queue length is finite. First-In-First-Out (FIFO) scheme is used to accommodate new samples when this queue is full.

# NB Time Service (con't)

- Phase 3: Selection and combine algorithms.
  - Using clustering algorithms a candidate list is constructed.
  - Clustering algorithm uses synchronization distance related to each time server to find the candidate list. It is a parameter computed from roundtrip delay and dispersion.
- Updating Offset
  - Calculating offset as described is not sufficient to achieve time-ordering of events.
  - Offset is updated unless adjusted time goes backwards.

# Test Results I



Change of offset with time for darya.ucs.indiana.edu

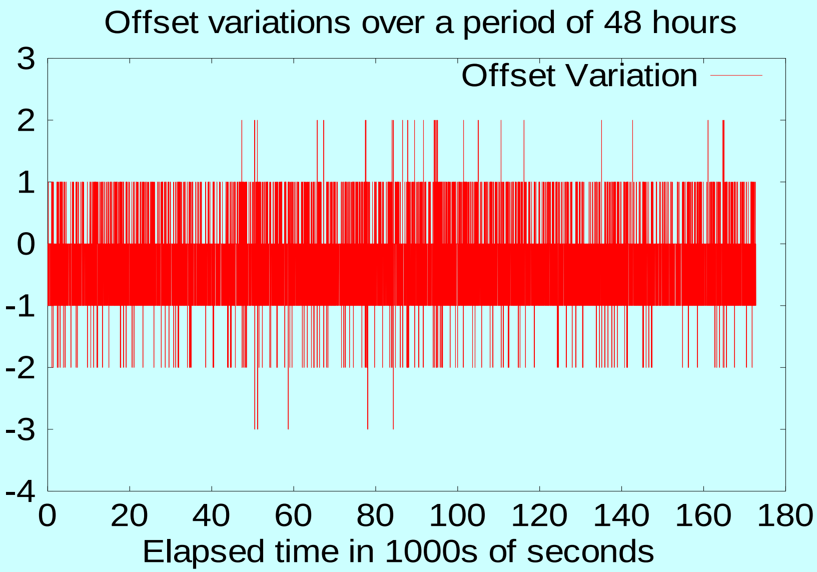
OS: Red Hat Linux release 7.3 (Valhalla)
CPU: AMD Athlon(tm)MP 1800+, 1533.42 MHz
Memory: 512 MB
JVM version: 1.4.1_03

| | |
|---|---|
| initialization offset value | 0 ms |
| standard deviation | 0.11 |
| average | -0.00018 ms |
| min value | -2 ms |
| max value | 3 ms |
| total change | -1 ms |
| number of data | 5690 |
| total test duration | 172800 sec |

Numeric values for darya.ucs.indiana.edu

- ntpd daemon is running and it synchronizes its time with “time.nist.gov” time server.
- Out of 5690 samples only 24 of them are different than zero for this experiment.

# Test Results II



Change offset with time for murray.ucs.indiana.edu

OS: Red Hat Linux release 7.2 (Enigma)

CPU: Intel Intel(R) Pentium(R) III CPU family 1266MHz

Memory:1 GB

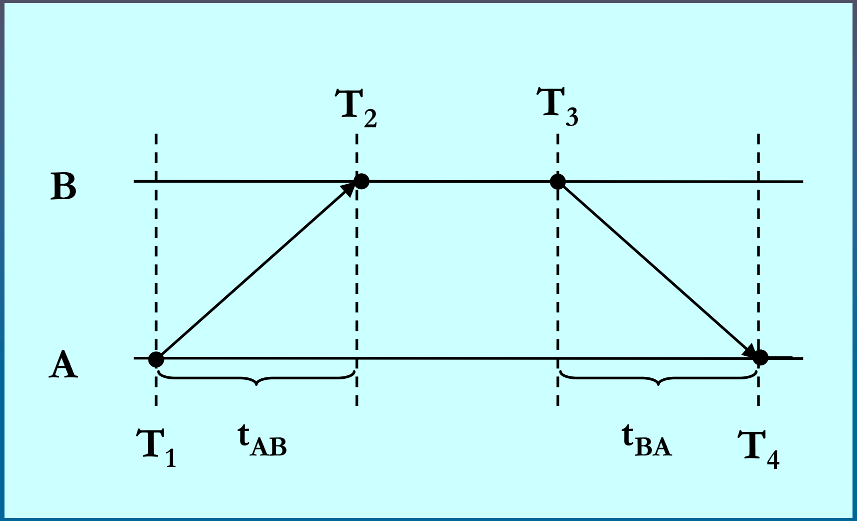
JVM version: 1.4.1-rc

| | |
|---|---|
| initialization offset value | -139895 ms |
| standard deviation | 0.71 |
| average | -0.19 ms |
| min value | -3 ms |
| max value | 2 ms |
| total change | -1060 ms |
| number of data | 5690 |
| total test duration | 172800 sec |

Numeric values for murray.ucs.indiana.edu

- The first offset value is -139895 ms, which shows how much the clock in that machine is ahead of the real time.
- The change of offsets is between (-3) - (2) ms.

# Inter Client Discrepancy

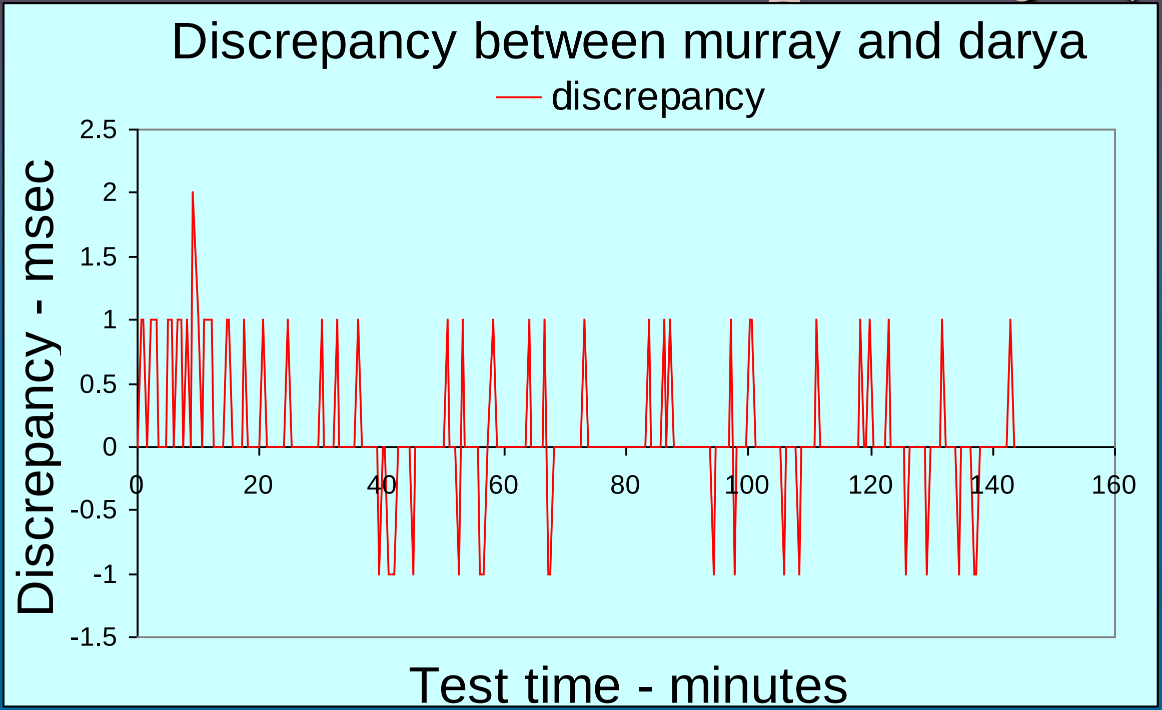


- Client discrepancy measures how much clocks of these clients are consistent with each other.

- $t_{AB}$ and $t_{BA}$ can be ignored if roundtrip delay is very small.
- If $t_{AB}$ and $t_{BA}$ are ignored (or if they are approximately equal to each other), the discrepancy ($\Delta T$) between these two clocks can be approximated as

$$\Delta T=0.5*(T_2+T_3-T_1-T_4)$$

# Inter Client Discrepancy (con't)



Test duration is 143 minutes

Absolute maximum discrepancy is 2 msec

Absolute minimum discrepancy is 0 msec

The roundtrip delay between two machines is around 2 – 4 msec.

Average discrepancy for this test is 0.082 msec.

# Conclusion

- Different machines with different platforms and with different loads may have different clock rates which in time would get out of sync.
- One cannot rely on the underlying clock and use the system clock to timestamp the events generated in messaging systems.
- Real-time constraints for collaboration environments is around 100 msec. NTP provides a sufficient synchronization range for such a  collaboration environment.
- Stratum-1 servers should be chosen so that the roundtrip time between time servers and the local machine is minimized.
- A Buffering Service has been implemented for NaradaBrokering which enables total order and time-ordered delivery.