LECTURE -2

VIRTUAL TIME AND GLOBAL STATE

Prof. D. S. Yadav
Department of Computer Science
IET Lucknow

THE CONCEPT OF TIME

- **The Concept of Time**

- A standard time is a set of instants with a temporal precedence order < satisfying certain conditions [Van Benthem 83]:
  - Transitivity
  - Irreflexivity
  - Linearity
  - Eternity ($\forall x \exists y: x<y$)
  - Density ($\forall x,y: x<y \rightarrow \exists z: x<z<y$)

- Transitivity and Irreflexivity imply asymmetry

Note :

REFLEXIVE RELATION: A relation *R* is said to be reflexive over a set *A* if (a,a) € *R* for every *a* € *A*.

- SYMMETRIC RELATION: A relation *R* is said to be symmetric if (a,b) € R => (b,a) € R
- TRANSITIVE RELATION: A relation *R* is said to be Transitive if (a,b) € R, (b,c) € R => (a,c) € R.

TIME AS A PARTIAL ORDER

- **A linearly ordered structure of time is not always adequate for distributed systems**
  - Captures dependence, not independence of distributed activities

- **A partially ordered system of *vectors* forming a *lattice* structure is a natural representation of time in a distributed system**

- **Resembles Einstein-Minkowski's relativistic space-time**

GLOBAL TIME & GLOBAL STATE OF DISTRIBUTED SYSTEMS

- **Asynchronous distributed systems consist of several *processes* without common memory which communicate (solely) via *messages* with unpredictable transmission delays**
- **Global time & Global State are hard to realize in distributed systems**
  - Processes are distributed geographically
  - Rate of event occurrence can be high (unpredictable)
  - Event execution times can be small
- ***We can only approximate the global view***
  - *Simulate synchronous* distributed system on given asynchronous systems
  - ***Simulate a global time*** – Logical Clocks
  - ***Simulate a global state*** – Global Snapshots

SIMULATE SYNCHRONOUS DISTRIBUTED SYSTEMS

- ***Synchronizers*** [**Awerbuch 85**]
  - Simulate clock pulses in such a way that a message is only generated at a clock pulse and will be received before the next pulse
  - Drawback
    - Very high message overhead

SIMULATING GLOBAL TIME : CLOCK SKEW & CLOCK DRIFT

- **An accurate notion of global time is difficult to achieve in distributed systems.**
  - We often derive "causality" from loosely synchronized clocks

- **Clocks in a distributed system drift**
  - Relative to each other
  - Relative to a real world clock
    - Determination of this real world clock itself may be an issue
  - **Clock Skew versus Drift**
    - Clock Skew = Relative Difference in clock values of two processes
    - Clock Drift = Relative Difference in clock frequencies (rates) of two processes

- .

CLOCK SYNCHRONIZATION

- **A non-zero clock drift will cause skew to continuously increase**
- **Maximum Drift Rate (MDR) of a clock**
  - Absolute MDR is defined relative to a Coordinated Universal Time (UTC)
  - MDR of a process depends on the environment.
  - Max drift rate between two clocks with similar MDR is 2 * MDR

  Max-Synch-Interval = (MaxAcceptableSkew — CurrentSkew) / (MDR * 2)
- **Clock synchronization is needed to simulate global time**

  Correctness – consistency, fairness
- **Physical Clocks vs. Logical clocks**

  Physical clocks - must not deviate from the real-time by more than a certain amount.

PHYSICAL CLOCK SYNCHRONIZATION

PHYSICAL CLOCKS

- **How do we measure real time?**
  - 17th century - Mechanical clocks based on astronomical measurements
    - Solar Day - Transit of the sun
    - Solar Seconds - Solar Day/(3600*24)
  - Problem (1940) - Rotation of the earth varies (gets slower)
  - Mean solar second - average over many days

ATOMIC CLOCKS

- **1948**
  - counting transitions of a crystal (Cesium 133) used as atomic clock
  - TAI - International Atomic Time
    - 9192631779 transitions = 1 mean solar second in 1948
  - UTC (Universal Coordinated Time)
    - From time to time, we skip a solar second to stay in phase with the sun (30+ times since 1958)
    - UTC is broadcast by several sources (satellites…)

ACCURACY OF COMPUTER CLOCKS

- **Modern timer chips have a relative error of 1/100,000 - 0.86 seconds a day**
- **To maintain synchronized clocks**
  - Can use UTC source (time server) to obtain current notion of time
  - Use solutions without UTC.

CRISTIAN'S (TIME SERVER) ALGORITHM

- Uses a *time server* to synchronize clocks
  - Time server keeps the reference time (say UTC)
  - A client asks the time server for time, the server responds with its current time, and the client uses the received value *T* to set its clock
- But network round-trip time introduces errors…
  - Let **RTT = response-received-time – request-sent-time** (measurable at client),
  - If we know (a) min = minimum client-server one-way transmission time and (b) that the server timestamped the message at the last possible instant before sending it back
  - Then, the actual time could be between **[T+min,T+RTT— min]**

CRISTIAN'S ALGORITHM

- **Client sets its clock to halfway between T+min and T+RTT—min i.e., at T+RTT/2**

☹Expected (i.e., average) skew in client clock time = (RTT/2 – *min*)

- **Can increase clock value, should never decrease it.**
- **Can adjust speed of clock too (either up or down is ok)**
- **Multiple requests to increase accuracy**
  - For unusually long RTTs, repeat the time request
  - For non-uniform RTTs
    - Drop values beyond threshold; Use averages (or weighted average)

BERKELEY UNIX ALGORITHM

- One daemon without UTC
- Periodically, this daemon polls and asks all the machines for their time
- The machines respond.
- The daemon computes an average time and then broadcasts this average time.

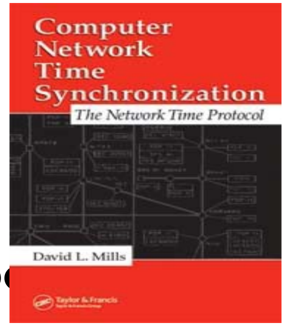
DECENTRALIZED AVERAGING ALGORITHM

- Each machine has a daemon without UTC
- Periodically, at fixed agreed-upon times, each machine broadcasts its local time.
- Each of them calculates the average time by averaging all the received local times.

CLOCK SYNCHRONIZATION IN DCE

- **DCE's time model is actually in an interval**
  - I.e. time in DCE is actually an interval
  - Comparing 2 times may yield 3 answers
    - t1 < t2
    - t2 < t1
    - not determined
  - Each machine is either a time server or a clerk
  - Periodically a clerk contacts all the time servers on its LAN
  - Based on their answers, it computes a new time and gradually converges to it.

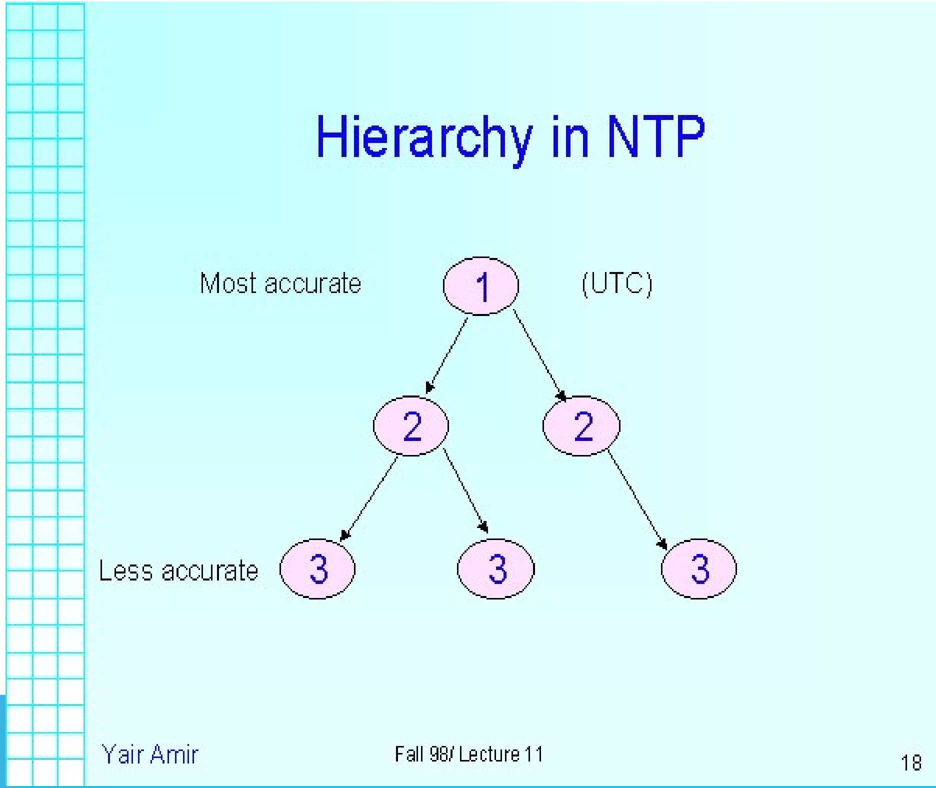
NETWORK TIME PROTOCOL (NTP)



- **Most widely used physical clock synchronization protoc on the Internet (http://www.ntp.org)**
  - Currently used: NTP V3 and V4
- **10-20 million NTP servers and clients in the Internet**
- **Claimed Accuracy (Varies)**
  - milliseconds on WANs, submilliseconds on LANs, submicroseconds using a precision timesource
  - Nanosecond NTP in progress

NTP DESIGN

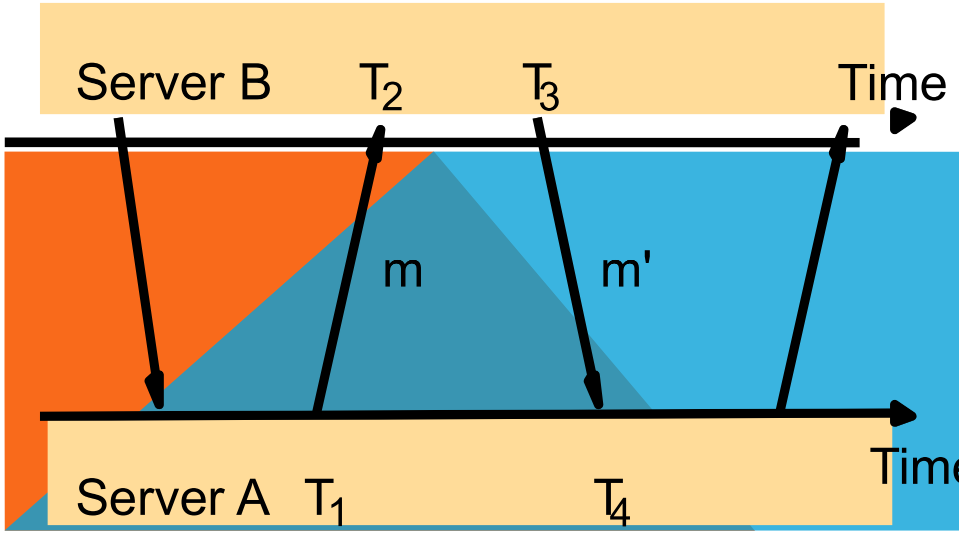
- Hierarchical tree of time servers.
- The primary server at the root synchronizes with the UTC.
- The next level contains secondary servers, which act as a backup to the primary server.
- At the lowest level is the synchronization subnet which has the clients.



NTPS OFFSET DELAY ESTIMATION METHOD

- Source cannot accurately estimate local time on target
  - varying message delays
- NTP performs several trials and chooses trial with minimum delay
  - Let a = T1−T3 and b = T2−T4.
  - If differential delay is small, the clock offset Θ and roundtrip delay δ of B relative to A at time T4 are approximately given by

$$\Theta = (a + b)/2, \ \delta = a - b$$



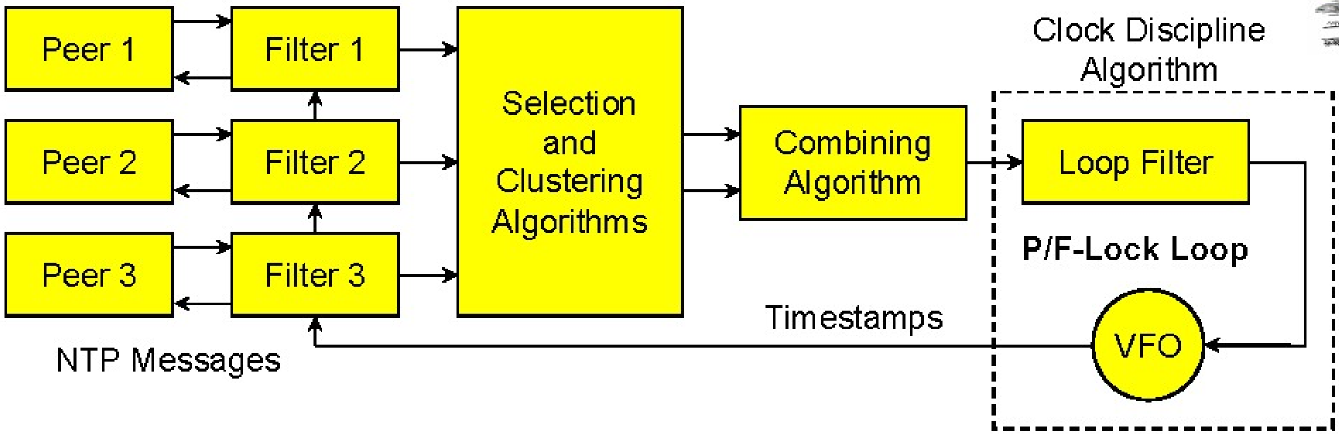
•A pair of servers in symmetric mode exchange pairs of timing messages.

•A store of data is then built up about the relationship between the two servers (pairs of offset and delay). Specifically, assume that each peer maintains pairs (Oi ,Di), where Oi - measure of offset; Di - transmission delay of two messages.

•The eight most recent pairs of (O, D_i) are retained.

•The value of O_i that corresponds to minimum D_i is chosen to estimate O.

NTP architecture overview



- Multiple servers/peers provide redundancy and diversity.
- Clock filters select best from a window of eight time offset samples.
- Intersection and clustering algorithms pick best *truechimers* and discard *falsetickers.*
- Combining algorithm computes weighted average of time offsets.
- Loop filter and variable frequency oscillator (VFO) implement hybrid phase/frequency-lock (P/F) feedback loop to minimize jitter and wander.

From (http://www.ece.udel.edu/~mills/database/brief/seminar/ntp.pdf)

NTP protocol header and timestamp formats

NTP Protocol Header Format (32 bits)

LI	VN	Mode	Strat	Poll	Prec
Root Delay					
Root Dispersion					
Reference Identifier					
Reference Timestamp (64)					
Originate Timestamp (64)					
Receive Timestamp (64)					
Transmit Timestamp (64)					
Extension Field 1 (optional)					
Extension Field 2… (optional)					
Key/Algorithm Identifier					
Message Digest (128)					

Cryptosum

Authenticator (Optional)

LI	leap warning indicator
VN	version number (4)
Strat	stratum (0-15)
Poll	poll interval (log2)
Prec	precision (log2)

NTP Timestamp Format (64 bits)

Seconds (32)	Fraction (32)

Value is in seconds and fraction since 0^h 1 January 1900

NTP v4 Extension Field

Field Type	Length
Extension Field (padded to 32-bit boundary)	

Last field padded to 64-bit boundary

NTP v3 and v4
NTP v4 only
authentication only

Authenticator uses MD5 cryptosum of NTP header plus extension fields (NTPv4)

From (http://www.ece.udel.edu/~mills/database/brief/seminar/ntp.pdf)

ASSIGNMENT 1

WRITE A TERM PAPER ON

ADVANCEMENT IN VIRTUAL TIME AND CLOCK SYNCHRONIZATION : TOOLS & TECHNIQUES

LAST DATE OF SUBMISSION : 08 SEPT 2019
DATE OF PRESENTATION : WILL BE ANNOUNCED LATER

END OF LECTURE 2