NTP - Network Time Protocol

Michael Höfling michael.hoefling@gmx.net

Florian Jäckel me@webstor.de

5DV020 Distributed System

University of Umeå

Fall 2007

Historical facts

- Developed in 1985 by David Mills at the University of Delaware, USA.
- David is still maintaining NTP.
- NTP is one of the oldest Internet protocols still in use.
- Latest version is NTP version 4 (version 5 is under development)

Differences to Cristian‘s method and the Berkley algorithm

- Cristian‘s method(CM) and the Berkley algorithm(BA) are both designed for primarily use in intranets.
- The Network Time Protocol was designed for use in the Internet (or other unreliable networks) right up from the start.
- CM and BA both synchronize against one time server.
- NTP synchronizes against many time servers.

Technical facts

- NTP is an application layer protocol
  - Default port number 123
  - Uses standard UDP Internet transport protocol
- Timestamps are used in messages
  - Timestamps counted as seconds from 1900-01-01 (year 2036 bug in NTP version 3 due to 32bit-wide seconds field)
  - 64bit-wide seconds field (NTP version 4)
  - NTP version 5 might have 128bit-wide seconds field
- Accuracy
  - approx. 200µs over LAN
  - approx. 10ms over Internet

Design aims of NTP

1. Adjust system clock as close to UTC as possible over the Internet. (statistic techniques for filtering timing data)
2. Handle bad connectivity. (redundant paths, reconfigurable servers)
3. Enable sufficiently frequently resynchronizations. (scaling well on large numbers of clients and servers)
4. Security. (distrust server's time, authentication)

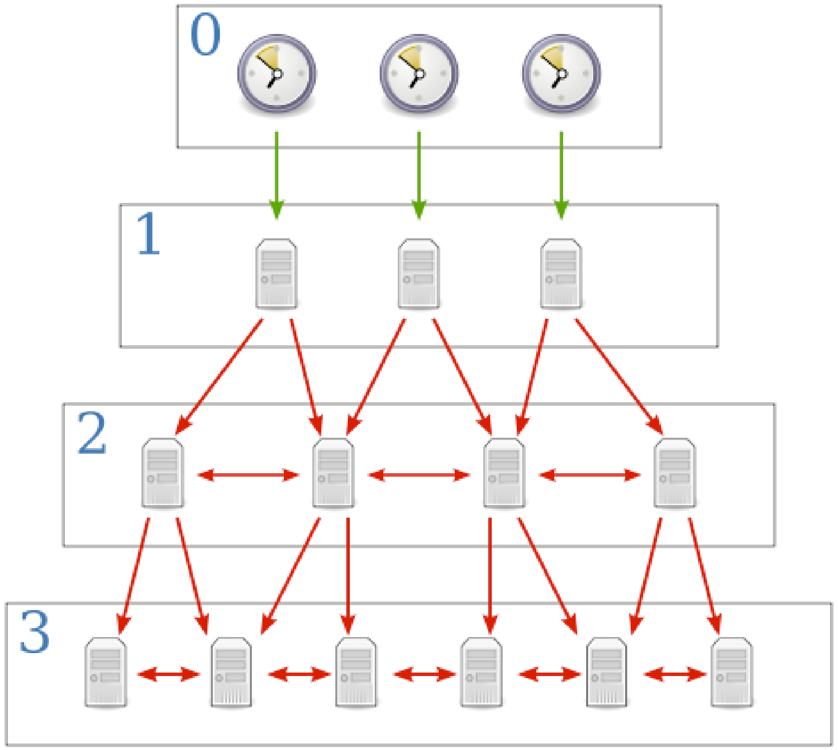
Hierarchical structure

- The NTP service is provided by a network of servers located across the Internet.
- Primary servers are connected directly to a time source. (e.g. a radio clock receiving UTC, GPS).
- Secondary servers are synchronized with primary servers.
- The servers are connected in a logical hierarchy called a *synchronization subnet*.

Synchronization subnet

- Each level of the synchronization subnet is called *stratum*. (e.g. primary servers are stratum 1, secondary stratum 2 and so on)
- Lowest-level (leaf) servers execute in users‘ workstations. (e.g. *ntp* under Linux)
- Servers with high stratum numbers are liable to have less accurate clocks than those with lower stratum numbers.
- The synchronization subnet can reconfigure as servers become unreachable or failures occur.

Example: strata



Source: http://upload.wikimedia.org/wikipedia/commons/c/c9/Network_Time_Protocol_servers_and_clients.svg

Server synchronization

- There are three possible synchronization modes:
  - multicast (sometimes called *broadcast*)
  - procedure-call (sometimes called *client*)
  - symmetric (sometimes called *peer*)

multicast mode

- One or more servers periodically multicasts the time to the servers in the network.
- Receivers set their clock assuming a small delay.
- Receivers don‘t reply.
- Problem:
  - Relative low accuracy.
  - Due to hardware limitations this mode only works in IP multicast enabled networks like LAN.

procedure-call mode

- One server accepts requests from other computers.
- The server replies with its timestamp.
- Higher accuracy.

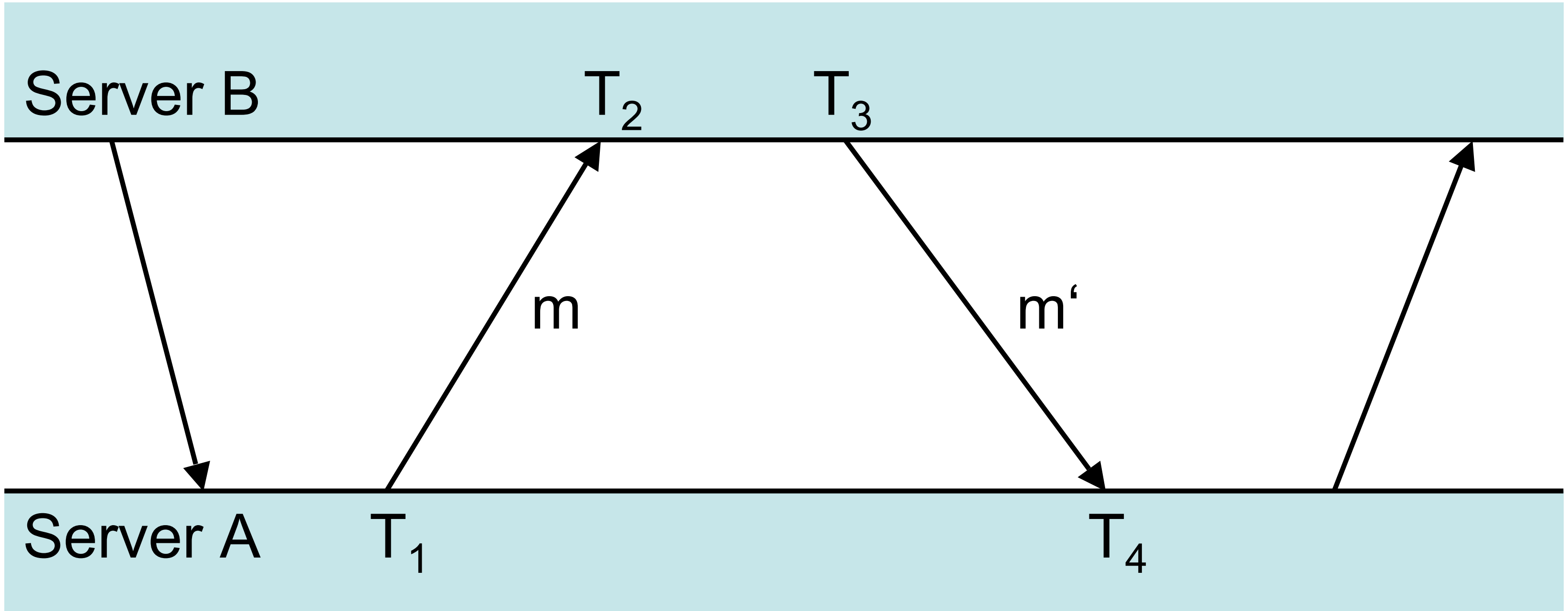
symmetric mode

- Intended to be used on higher levels of the synchronization subnet.
- A pair of servers exchange messages to improve the accuracy of their synchronization over time. (reduction of the *synchronization dispersion*)
- Highest accuracy.

Example: message exchange

- Let A and B be NTP servers exchanging messages with each other.
- m is the message A sends to B.
- m‘ is the message B replies to A.
- m and m‘ form a pair of messages.
- T_1 to T_4 are timestamps

Example: message exchange ff



How NTP works

- NTP calculates an offset o_j and a delay d_j
  - $o_j = T(B) - T(A) = [(T_2 - T_1) + (T_4 - T_3)]/2$
  - $d_j = T(ABA) = (T_4 - T_1) - (T_3 - T_2)$
- As the transmission times of m and m‘ are allways ≥ 0 the real offset o is
  - $o_j - d_j/2 \le o \le o_j + d_j/2$

How NTP works ff

- NTP uses Marzullo's algorithm to filter the offset o out of successive pairs $<o_j,d_j>$.
- NTP calculates the filter dispersion ε.
- High ε indicates unreliable data.
- At the end $<o_j,d_j>$ with smallest d_j is choosen.

How NTP works ff

- To achieve higher accuracy an NTP server contacts several peers.
- NTP applies a peer-selection algorithm.
- This examines which peer has the most reliable value.
- The server changes its primary synchronization peer to the most reliable one.

How NTP works ff

- Lower stratum numbers are more favoured as they are closer to the primary time sources.
- NTP modifies the local clock frequency to reduce its drift rate.

More details

- For more details (especially about the filters used in NTP) take a look at the Network Time Protocol Version 4 Reference and Implementation Guide.

Bibliography

- Course book chapter 11.3.4
- Wikipedia:
  - http://en.wikipedia.org/wiki/Network_Time_Protocol
  - http://en.wikipedia.org/wiki/Marzullo%27s_algorithm
- Network Time Protocol Version 4 Reference and Implementation Guide:
  - http://www.ee.udel.edu/~mills/database/reports/ntp4/ntp4.pdf