Algorithms for Clock Synchronisation

Cristian's algorithm

(1) Client C with time t_c sends a time request to server S.

(2) S sends C his time t_s.

(3) C receives t_s at time t_r

round trip time

(4) C sets his time to $t_s + \frac{RTT}{2}$, $RTT = t_r - t_c$

Berkeley algorithm

(1) There is one master node that polls its slave nodes.

(2) Each slave replies to the master with its time.

(3) The master <u>averages</u> the slaves' and its own time and eliminates any times with excessive RTTs. to compute one final time.

(4) The master sends each slave a <u>delta</u> which tells them how much to add or take off to/from their clocks.

Network time protocol (NTP)

- Works for networks of a larger scale (unlike Cristian, Berkeley)
- There are three methods of synchronization

Multicast mode
- Time server sends his time to all servers on LAN at once.
- Each server receives and resets its clock (assuming little delay)

Procedure call mode
- Effectively Cristian's algorithm
- Requesting node sets its time to $t_s + \frac{RTT}{2}$

Symmetric mode
- Most accurate, messages are exchanged and data is built up to improve accuracy of synchronization over time. Messages contain timing info about the previous message received (time sent, time received, etc.)]