Chapter 6 Synchronization

• Clock synchronization
• Logical clocks
• Distributed snapshot
• Election algorithms
• Mutual exclusion
• Transactions

Clock Synchronization

• Time is unambiguous in centralized systems
  – System clock keeps time, all entities use this for time
• Distributed systems: each node has own system clock
  – Problem: An event that occurred after another may be assigned an earlier time
Physical Clocks

• Use atomic clocks to keep time
  – One second is the time it takes the cesium 133 atom to make 9,192,631,770 transitions.
• Coordinated Universal Time (UTC) – international standard based on atomic time
  – Introduces a leap second from time to time to compensate that days are getting longer
  – UTC is broadcast by shortwave radio stations and satellites
• Computers use crystal-based clocks
  – Not accurate, results in clock drift

Clock Synchronization

• Need to synchronize machines with a UTC source or with one another
• Each clock has a maximum drift rate $\rho$
  
  $1-\rho \leq \frac{dC}{dt} \leq 1+\rho$
  
  – Two clocks may drift by $2\rho \Delta t$ in time $\Delta t$
  – To limit drift to $\delta \Rightarrow$ resynchronize every $\delta/2\rho$ seconds
Cristian’s Algorithm

- Synchronize machines to a *time server* with a UTC receiver
- Machine A periodically requests time from server B
  - A receives time $T_2$ and $T_3$ from server, sets clock to $T_3 + T_{res}$ where $T_{res}$ is the time to send reply from B to A
  - Use $(T_{req} + T_{res})/2$ as an estimate of $T_{res}$: $T_{res} = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$

Network Time Protocol (NTP)

- Widely used standard based on Cristian’s algorithm
- Improve accuracy by making 8 measurements, take the minimum value for $T_{res}$ as the best estimate for the delay between the two machines.
- A hierarchy of time servers: a time server in level $k$ synchronizes with a server in level $\leq k-1$
  - Level 0 is the atomic clock
- Clock cannot go backwards: If time needs to be adjusted backward, slow down the clock until time catches up.
Berkeley Algorithm

- Keep clocks synchronized with one another
- One computer is *master*, others are *slaves*
- Master periodically polls slaves for their times
  - Average times and return differences to slaves
- Failure of master => election of a new master

Global Positioning System (GPS)

- 29 satellites, each has up to 4 atomic clocks
- Each satellite continuously broadcasts its position and its local time
- Using 4 satellites, a receiver can compute its own position and the actual time!
- Facts to consider when computing position and time:
  - It takes a while before message from a satellite reaches the receiver.
  - The receiver’s clock is generally not in synch with that of a satellite.
GPS (continued)

• Dr – deviation of receiver from actual time
• Message with timestamp Ti received at T_{now}
  – Measured delay D_i = (T_{now} - T_i) + D_r
  – Measured distance = cD_i = d_i + cD_r (c is speed of light)
  – d_i = sqrt[(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2] is the real distance
• Four unknowns (x_r, y_r, z_r, D_r), need 4 satellites.
• You can get an accurate account of time as a side-effect of GPS!