Vector Clocks

- Each process maintains a vector $C$ of size $n$, where $n$ is the number of processes in the system.
- For process $i$, the $i^{th}$ entry of the vector is the local clock. The other entries represent its best guess of the clock at other processes.
  - When an event occurs at a process $i$, $C[i]$ is incremented.
  - When a message is sent, it is time-stamped (with the vector clock). Upon receipt by process $j$, $C_j$ is updated as
    - forall $k$, $C_j[k] = \max (C_j[k], \text{tmstamp}[k])$
- Every process has the most up to date knowledge of its clock (forall $i,j$, $C[i] \geq C[j]$)
• Two vector timestamps are equal iff all their components are equal, unequal if even one component differs.
• Less than or equal to iff each component is less than or equal to, not LTE if even one component is greater.
• Less than iff (LTE AND not EQ) => if at least one component is smaller
• Not less than iff not(LTE and NEQ)
• Concurrent iff ((a NLT b) AND (b NLT a))
• LTE specifies a partial order (but concurrency does not)
• Note that now, -- > iff (a LT b)
Causal Ordering of Messages

– If M1 is sent before M2, then every recipient of both messages must get M1 before M2
  • underlying network will not necessarily give this guarantee.
– Consider a replicated database system. Updates to the entries should be received in order!
– Basic idea -- buffer a later message
Birman-Schiper-Stephenson Protocol

- Assumes that communication is via broadcasts
- Pi stamps outgoing messages with a vector time
- Pj, upon receiving a message from Pi VTm buffers it till
  - VTpj[i] = VTm[i] - 1 AND forall k, k != i, VTpj[k] >= VTm[k]
- When Pj receives a message, it updates VTpj
Schipper-Eggli-Sandoz Protocol

– Each process maintains a vector VP of size N-1. The elements are tuples (Pj,t), where Pj is the destination of a message, and t the time the message was sent.

• Send:
  – Send message with timestamp tm and VP to Pk
  – insert (Pk, tm) into VP

• RECV:
  – If VM does not contain any tuple with Pk, OR tm <= tlocal then receive else buffer
  – Upon Receipt
    » Merge VM with VPk
    » Update P2’s logical Clock
    » Check for buffered messages that can be delivered.