Global State

Global state recording

local state
Global state
Chandy-Lamport's global state (snapshot) recording algorithm

Local state

- **local state** *LSi* of a site (process) *Si* is an assignment of values to variables of *Si*

sending *send*(*mij*) and receiving *rec*(*mij*) of message *mij* from *Si* to *Sj* may influence *Lsi*

- we denote
  - *time*(*send*(*mij*) or *rec*(*mij*)) the time (physical or point in the computation) the send or receive occurs
  - *time*(*LSi*) time the local state of *Si* was recorded
- to aid the reasoning we consider the messages sent/received by the site as belonging to local state

$$\bullet\ send(m_{ij}) \in LS_i \quad \text{iff} \quad time(send(m_{ij})) < time(LS_i)$$
$$\bullet\ rec(m_{ij}) \in LS_j \quad \text{iff} \quad time(rec(m_{ij})) < time(LS_j).$$

that is the message is in **transit** if it was sent but not received, the message is **inconsistent** if it was received but never sent

Global state

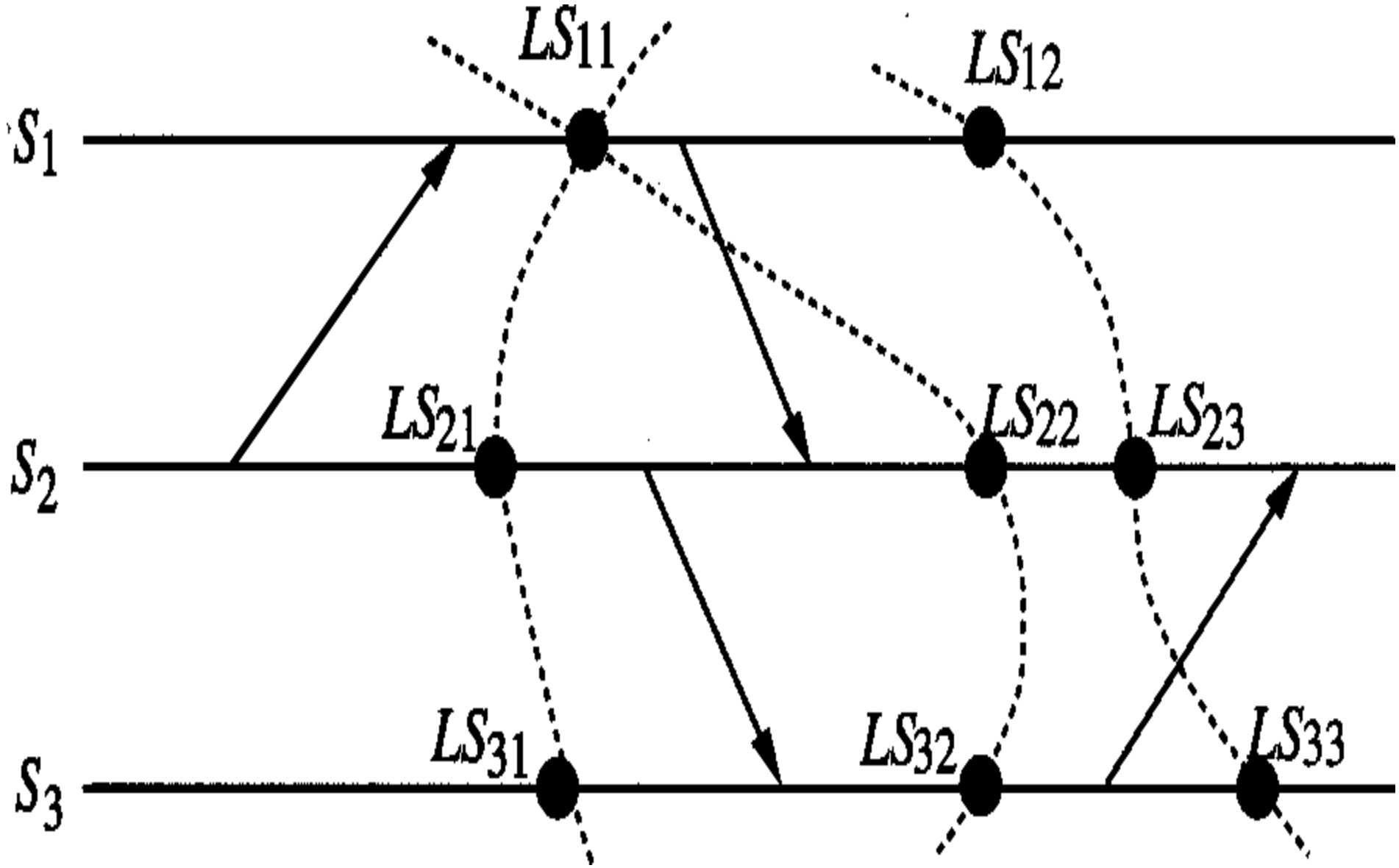
•**Global state** is a collection of local states of all sites and set of messages in the Channels

global state is **consistent** if it does not have any inconsistent messages,
that is:

$$\forall i, \forall j : 1 \leq i, j \leq n :: inconsistent(LS_i, LS_j) = \Phi$$

global state is **transit less(Strong)** if there are no messages in transition, that is

$$\forall i, \forall j : 1 \leq i, j \leq n :: transit(LS_i, LS_j) = \Phi$$


LS_{11}
LS_{12}
S_1
LS_{21}
LS_{22}
LS_{23}
S_2
LS_{31}
LS_{32}
LS_{33}
S_3

Chandy-Lamport's global state recording algorithm (snapshot algorithm)

Marker Sending Rule for a process P

- P records its state.
- For each outgoing channel C from P on which a marker has not been already sent, P sends a marker along C before P sends further messages along C.

Marker Receiving Rule for a process Q. On receipt of a marker along a channel C:

```
If Q has not recorded its state
    then
        begin
            Record the state of C as an empty sequence.
            Follow the "Marker Sending Rule."
        end
    else
        Record the state of C as the sequence of messages received
        along C after Q's state was recorded and before Q received
        the marker along C.
```