A logical clock (LC) is an abstraction necessary to ensure the clock condition in the absence of a global

clock. Each process pi maps events to positive integers. Call LC(e) the local variable associated with

event e. Each process time stamps each message m sent with the value of the logical clock at the time of

sending, T S(m) = LC(send(m)). The rules to update the logical clock are specified by the following

relationship:

LC(e) =

LC +1 if e is a local event or a send(m) event

max (LC, T S(m) + 1) if e = receive(m).

(2.23)

The concept of logical clocks is illustrated in Figure 2.5 using a modified space-time diagram in

which the events are labeled with the logical clock value. Messages exchanged between processes are

shown as lines from the sender to the receiver; the communication events corresponding to sending and

receiving messages are marked on these diagrams.

Each process labels local events and sends events sequentially until it receives a message marked

with a logical clock value larger than the next local logical clock value, as shown in Equation 2.23.

It follows that logical clocks do not allow a global ordering of all events. For example, there is no way to

establish the ordering of events e1

1, e1

2, and e1

3 in Figure 2.5. Nevertheless, communication events allow

different processes to coordinate their logical clocks; for example, process p2 labels the event e3

2 as 6

because of message m2, which carries the information about the logical clock value as 5 at the time

message m2 was sent. Recall that e j

i is the j -th event in process pi .

Logical clocks lack an important property, gap detection; given two events e and e  and their logical

clock values, LC(e) and LC(e

), it is impossible to establish if an event e   exists such that

LC(e) < LC(e

) < LC(e

). (2.24)

For example, for process p1 there is an event, e4

1, between the events e3

1 and e5

1 in Figure 2.5; indeed,

LC(e3

1) = 3, LC(e5

1) = 5, LC(e4

1) = 4, and LC(e3

1) < LC(e4

1) < LC(e5

1). However, for process

p3, the events e3

3 and e4

3 are consecutive, though LC(e3

3) = 3 and LC(e4

3) = 10.