Lecture 24

- Administration

Chapter 4: Global State and snapshot recording algorithms

Chandy and Lamport Snapshot

Marker-Sending Rule for a Process p:

p records its state;

for (each channel C directed away from p, with a marker not sent)

{ p sends one marker along C before p sends any further messages along C; }

Marker-Receiving Rule for a Process q:

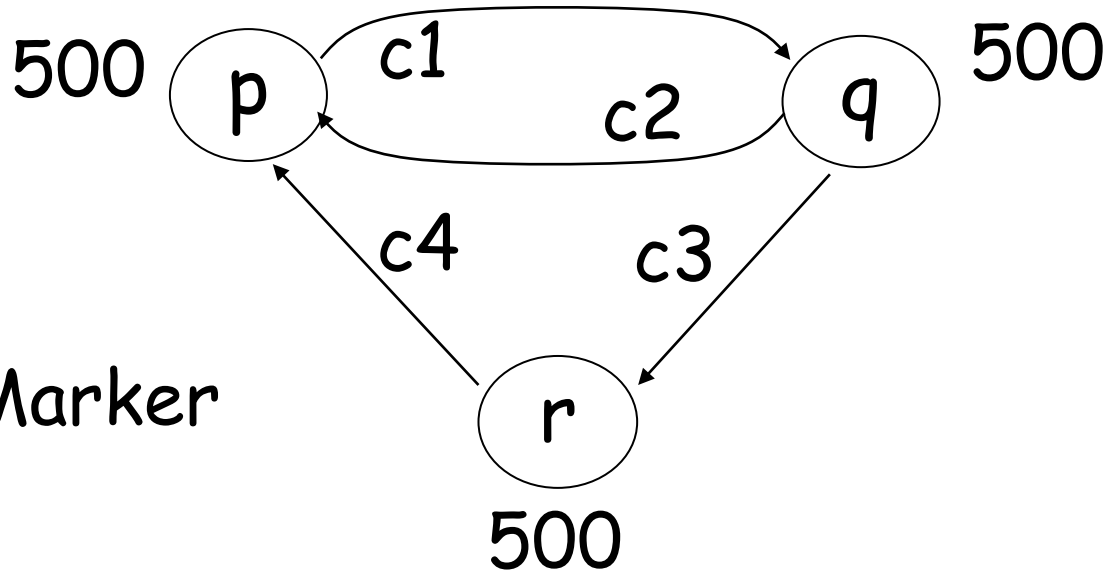
if (q has not recorded its state) then

{ q records the state of C as the empty sequence;
execute marker-sending rule.
}

else { q records the state of C as the sequence of message received along C after q's state was recorded and before q received the marker along C.
}

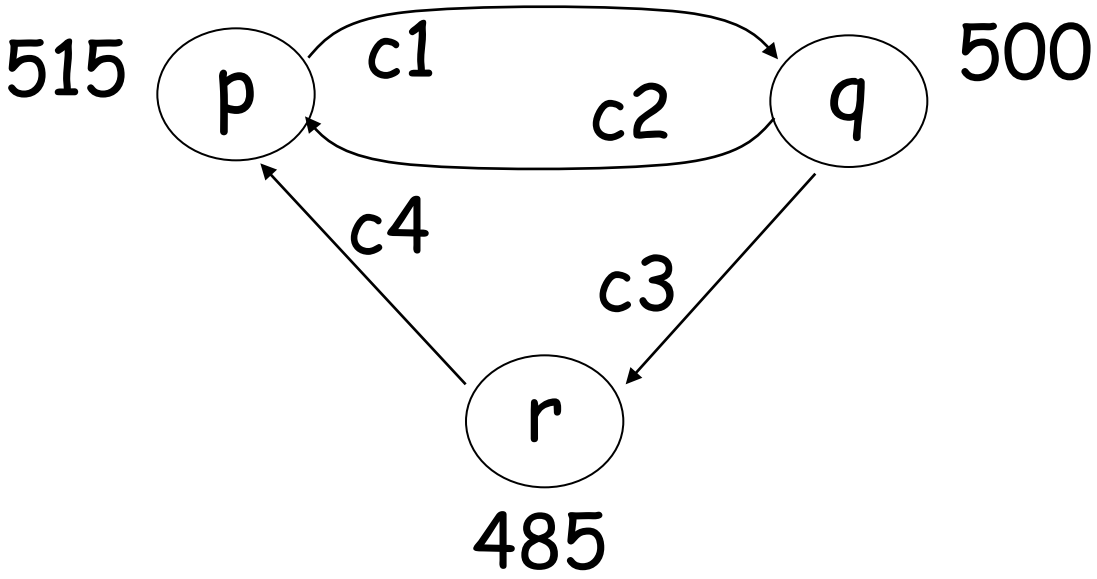
- When a process receives a mark, it knows that a snapshot is in process.
- An individual node knows that it is done when it records its own state and all the states in my incoming channels.

Example -- initial

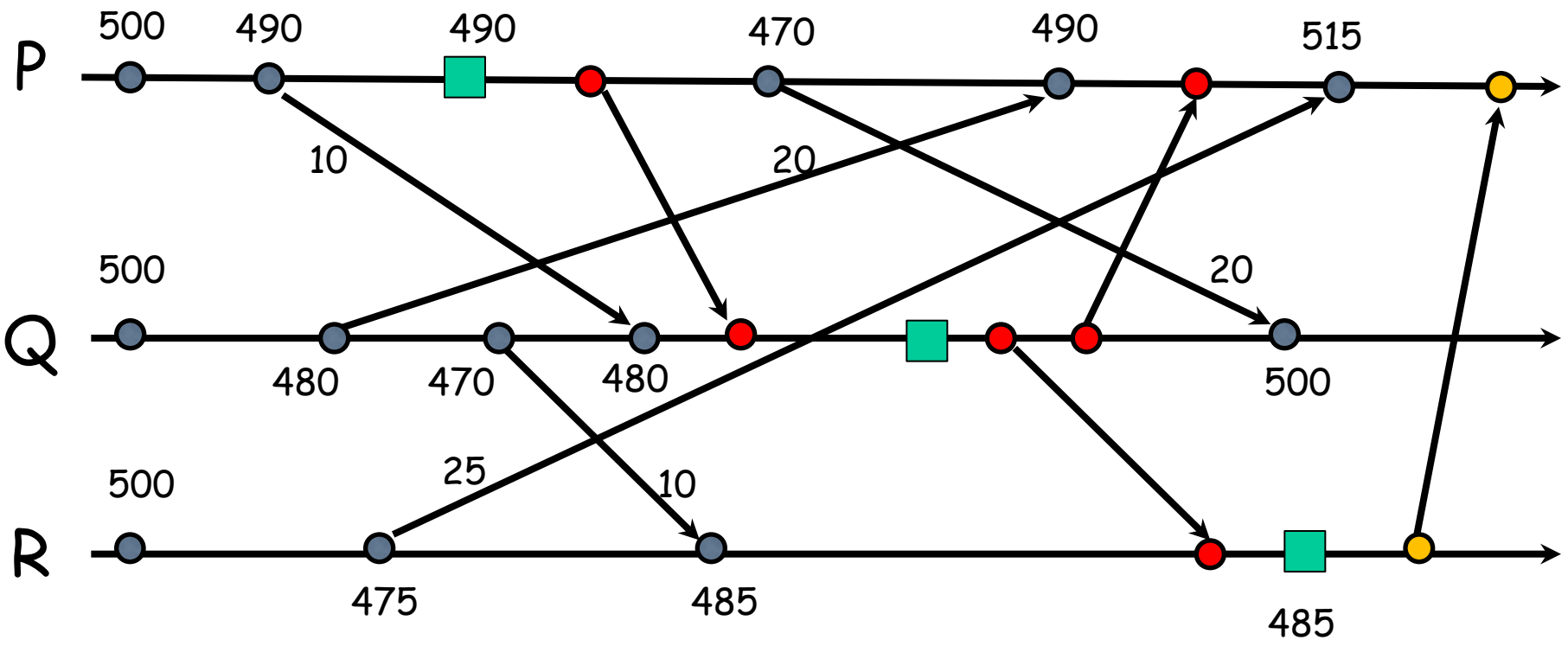


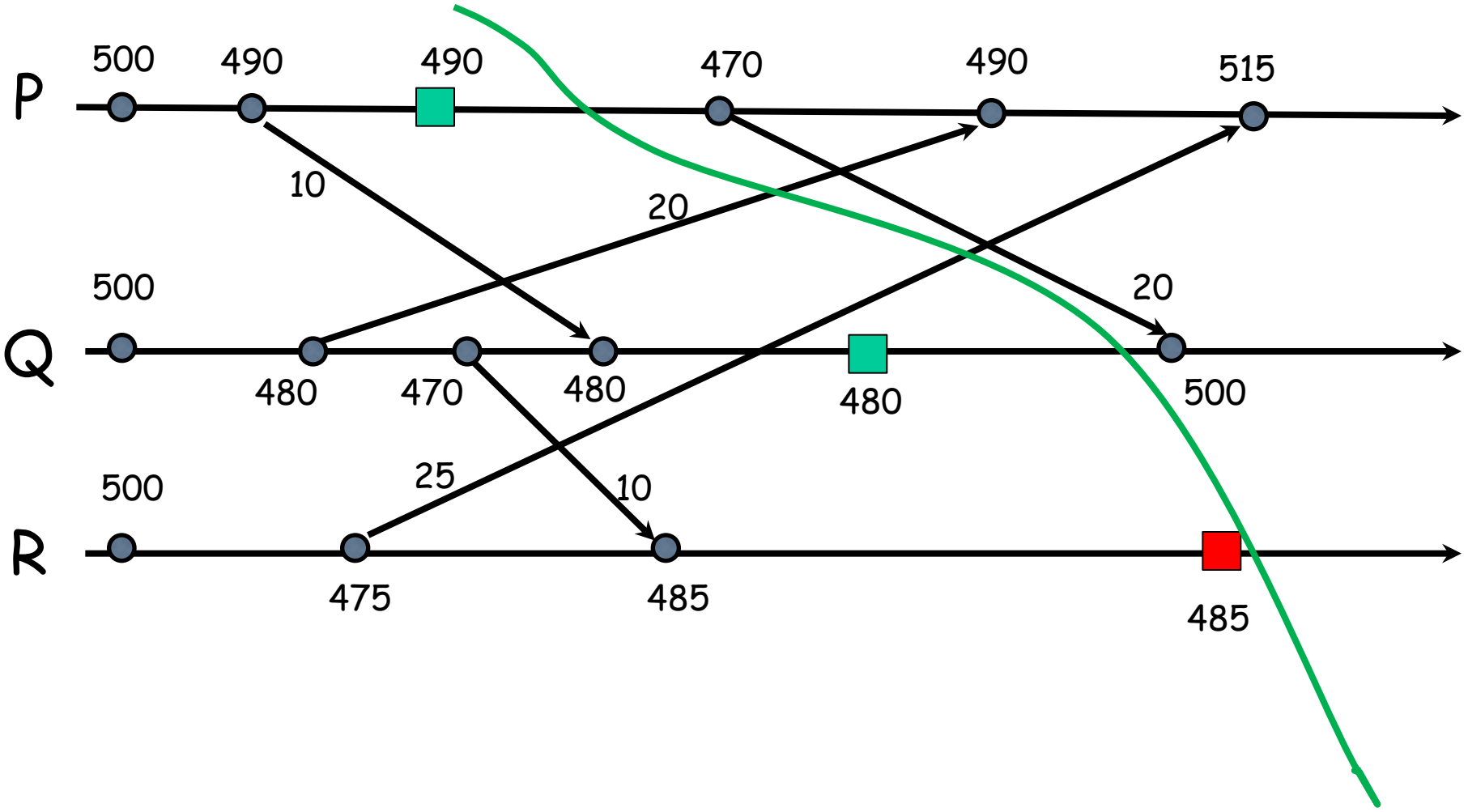
Node	Recorded state				
		c1	c2	c3	c4
p			{}		{}
q		{}			
r				{}	

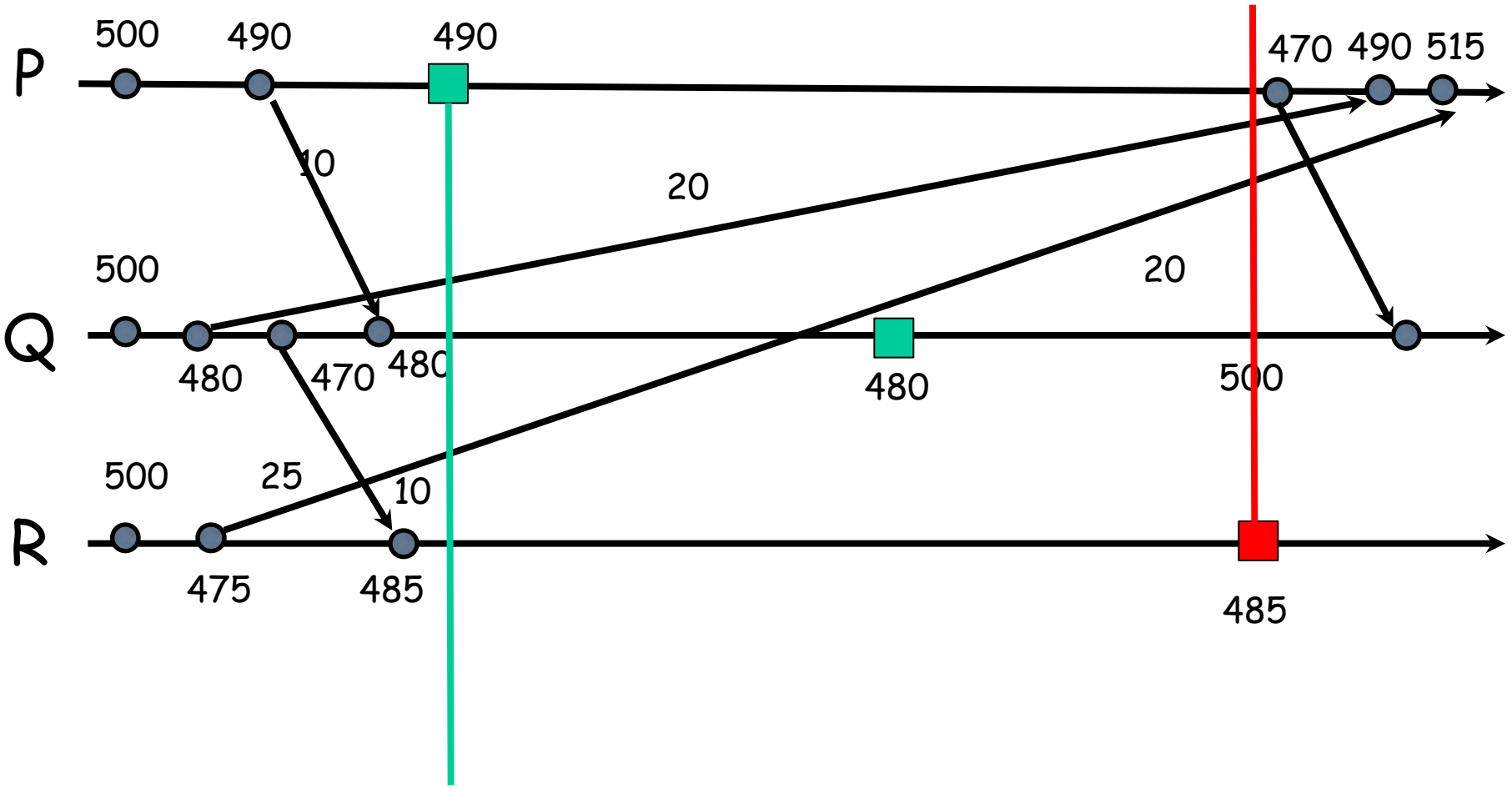
Example – step 5

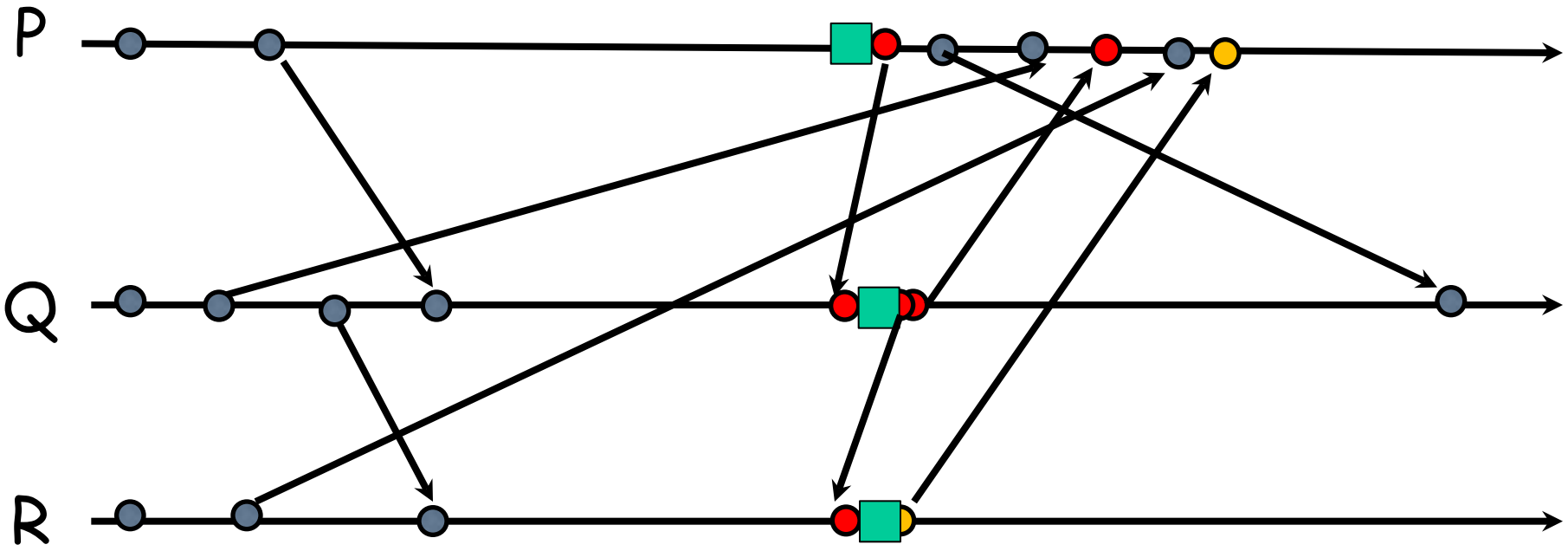


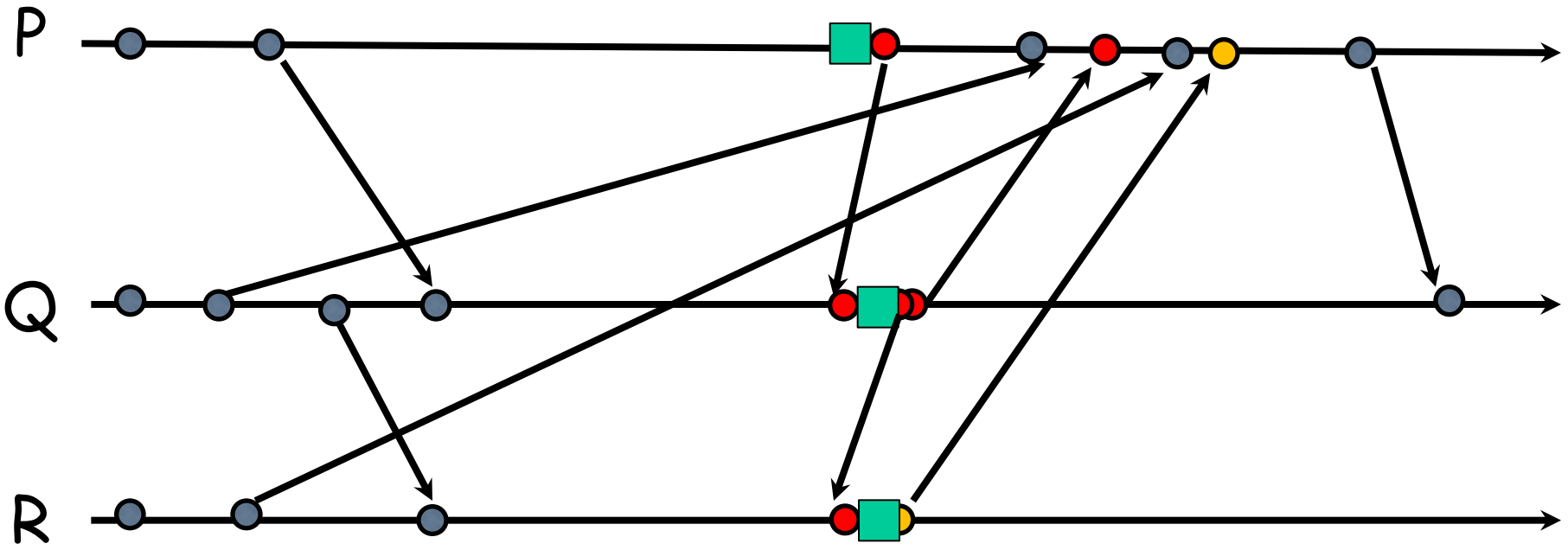
Node	Recorded state				
	state	c1	c2	c3	c4
p	490		{20}		{25}
q	480	{empty}			
r	485			{empty}	

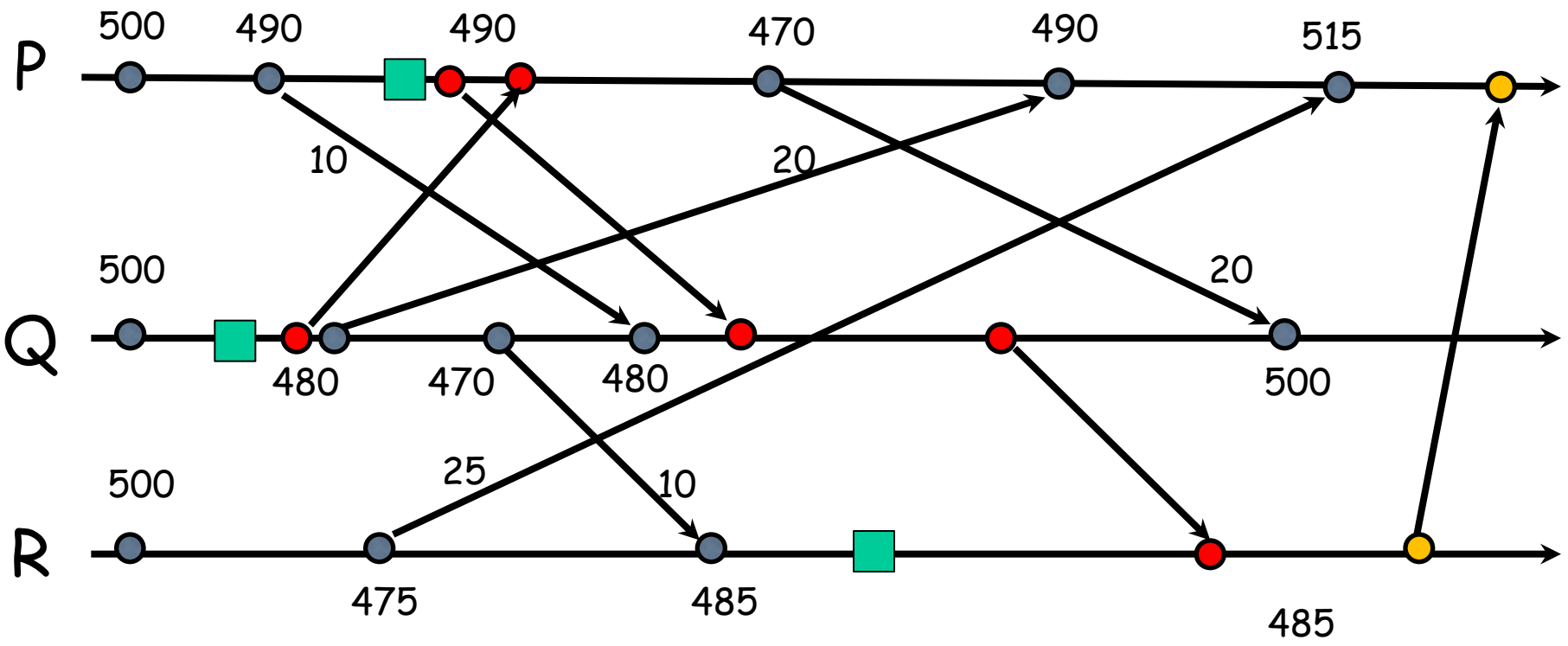

P
Q
R
500
490
490
470
490
515
10
20
500
20
480
470
480
500
500
25
10
475
485
485


P
Q
R
500
490
490
470
490
515
10
20
500
20
480
470
480
480
500
500
25
10
475
485
485


P
Q
R
500
490
490
470
490
515
10
20
500
20
480
470
480
480
500
500
25
10
475
485
485


P
Q
R


P
Q
R


P
Q
R
500
490
490
470
490
515
10
20
500
20
480
470
480
500
500
25
10
475
485
485

What if more than one initiate?

Table 4.1 A comparison of snapshot algorithms.

Algorithms	Features
Chandy–Lamport [7]	Baseline algorithm. Requires FIFO channels. $O(e)$ messages to record snapshot and $O(d)$ time.
Spezialetti–Kearns [29]	Improvements over [7]: supports concurrent initiators, efficient assembly and distribution of a snapshot. Assumes bidirectional channels. $O(e)$ messages to record, $O(rn^2)$ messages to assemble and distribute snapshot.
Venkatesan [32]	Based on [7]. Selective sending of markers. Provides message-optimal incremental snapshots. $\Omega(n+u)$ messages to record snapshot.
Helary [12]	Based on [7]. Uses wave synchronization. Evaluates function over recorded global state. Adaptable to non-FIFO systems but requires inhibition.
Lai–Yang [18]	Works for non-FIFO channels. Markers piggybacked on computation messages. Message history required to compute channel states.
Li *et al.* [20]	Similar to [18]. Small message history needed as channel states are computed incrementally.
Mattern [23]	Similar to [18]. No message history required. Termination detection (e.g., a message counter per channel) required to compute channel states.
Acharya–Badrinath [1]	Requires causal delivery support. Centralized computation of channel states. Channel message contents need not be known. Requires $2n$ messages, 2 time units.
Alagar-Venkatesan [2]	Requires causal delivery support. Distributed computation of channel states. Requires $3n$ messages, 3 time units, small messages.

n = # processes, u = # edges on which messages were sent after previous snapshot, e = # channels, d = diameter of the network, r = # concurrent initiators.