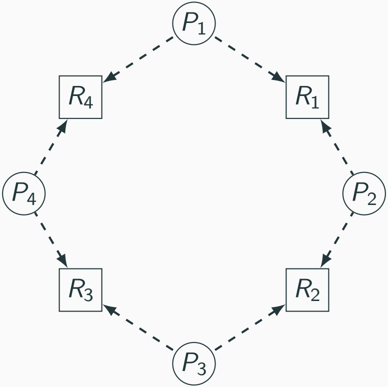
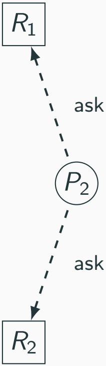
Dining Philosophers



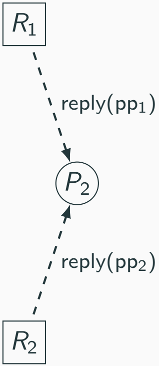
Dining Philosophers - Assumptions

	r_1	r_2	r_3	r_4
1				
2				
3				
⋮	⋮	⋮	⋮	⋮
n				

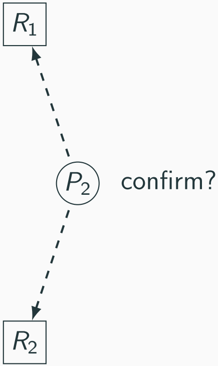
Dining Philosophers - Protocol



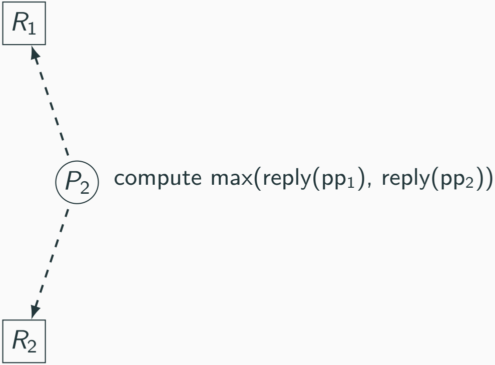
Dining Philosophers - Protocol



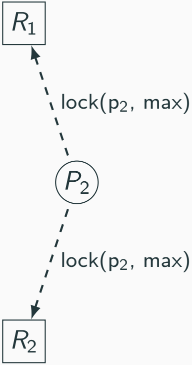
Dining Philosophers - Protocol



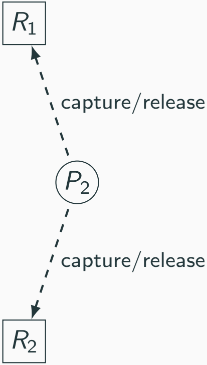
Dining Philosophers - Protocol



Dining Philosophers - Protocol



Dining Philosophers - Protocol



Dining Philosophers - Problems

Problem$_1$: qpp(r_n) is incremented (at reply) but if not used (at confirm) points to the wrong location. Incrementing qpp(r_n) at confirm would result in multiple request receiving the same qpp(r_n).

Fix: update qpp(r_n) again at locking;

Dining Philosophers - Problems

Problem$_2$: Suppose, p_1 requested $\{r_1, r_2\}$.
r_1 replies p_1 with n_1 and before r_2 replies with n_2 ($n_1 < n_2$);
Suppose $pp(r_1)$ has been incremented few times (because someone else also asked for the same resource). Thus when eventually he gets both replies and computes $\max(n_1, n_2) = n_2$ for p_1 cannot write to r_1 at $r_1(n_2)$.
Fix:

Dining Philosophers - Problems

Problem$_3$: Confirming that all requests and reply messages have been received, and a maximum number can be returned.

Fix: