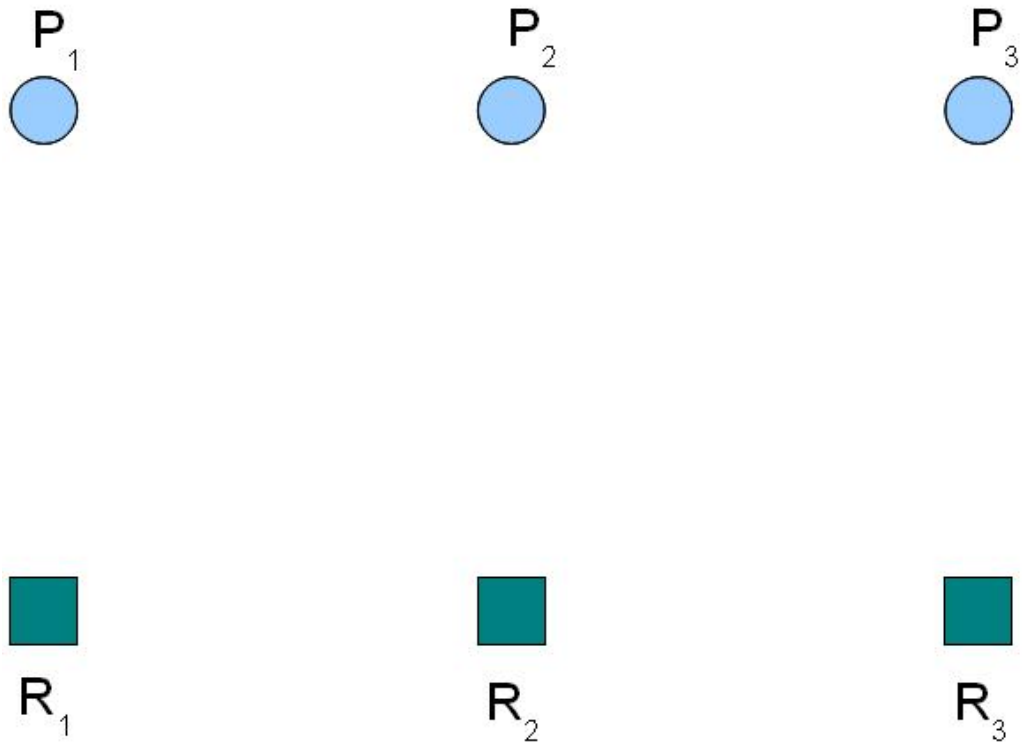
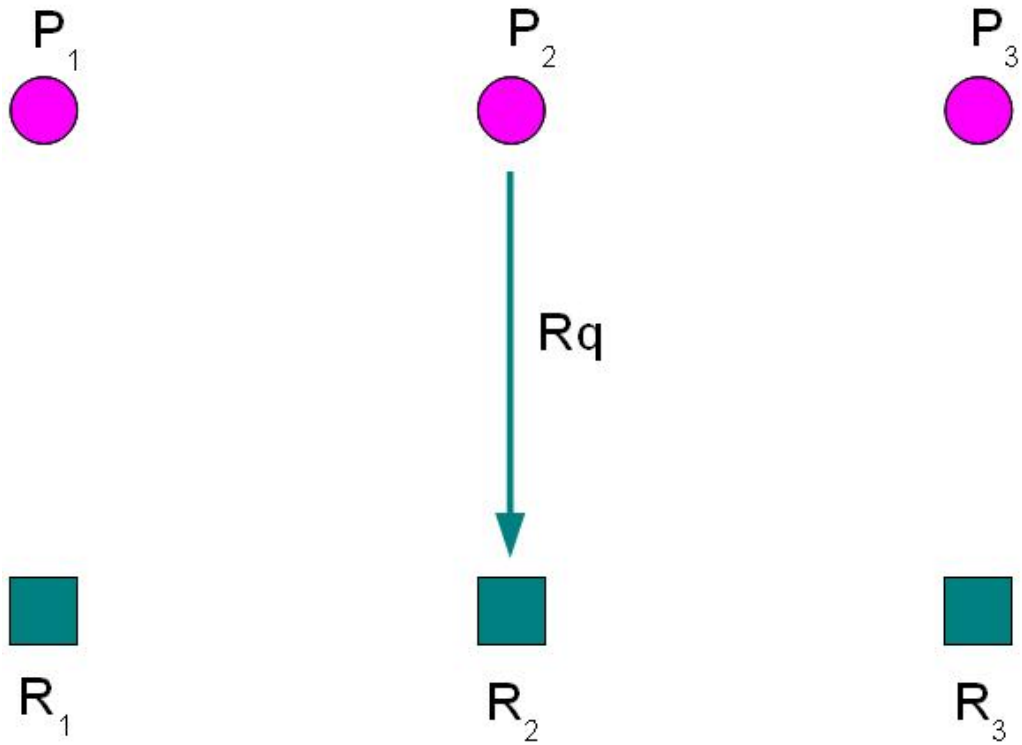
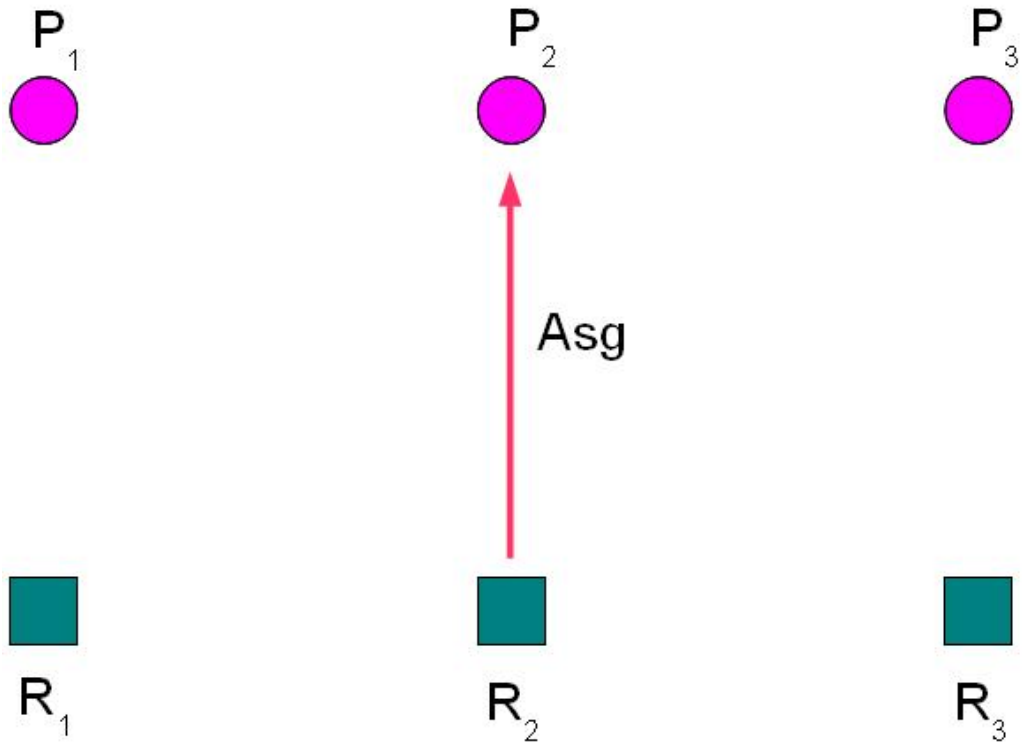
Deadlock on resources (devices/data)



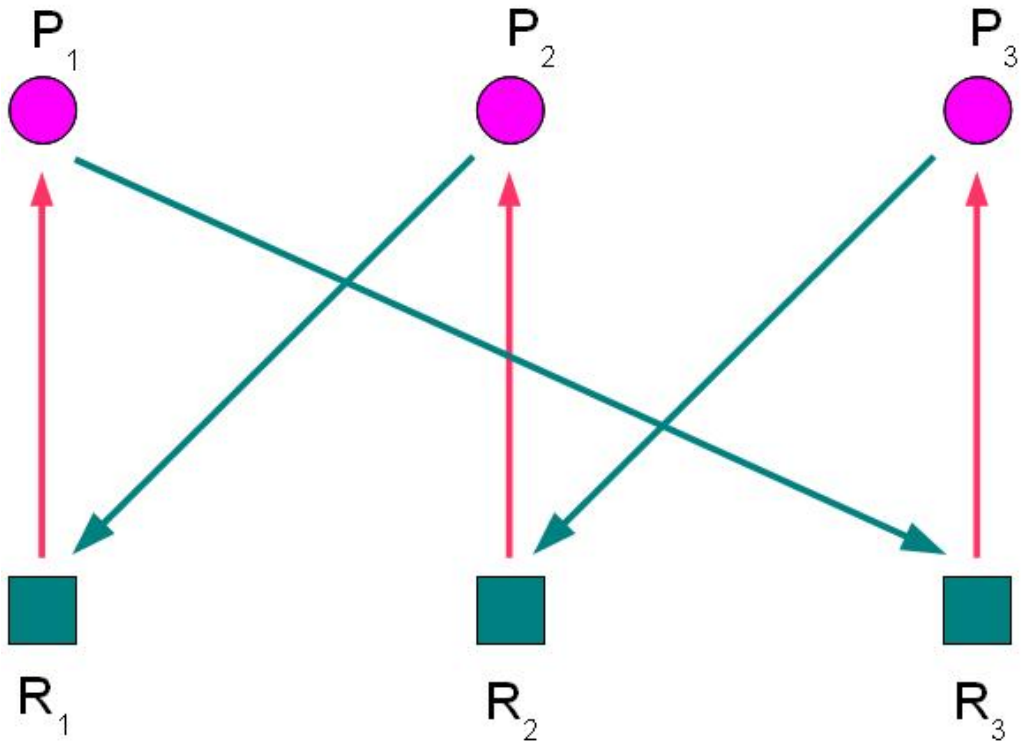
Deadlock on resources (devices/data)



Deadlock on resources (devices/data)



Deadlock on resources (devices/data)



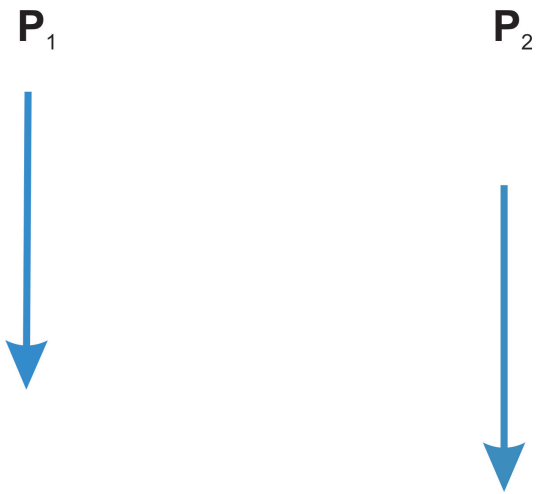
deadlock !

Deadlock on resources (devices/data)

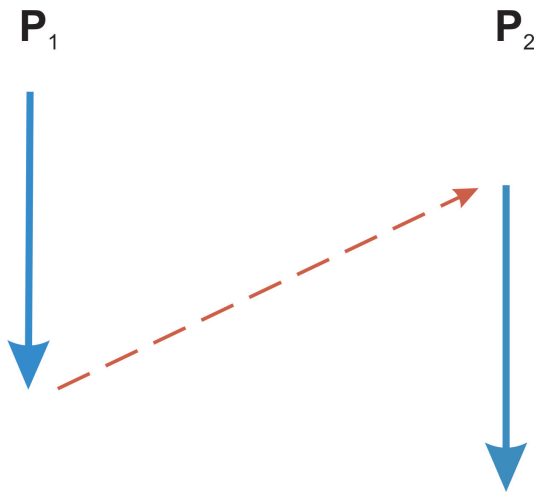
Deadlock conditions

1. Resource cannot be assigned to more processes (assignement is exclusive in the time).
2. Exists a situation when process waits for a resource assignement assigned to another process.
3. Process can unlock the resource by its own activity only.
4. There can arise a cycle in the graph of dependences.

Exclusive access

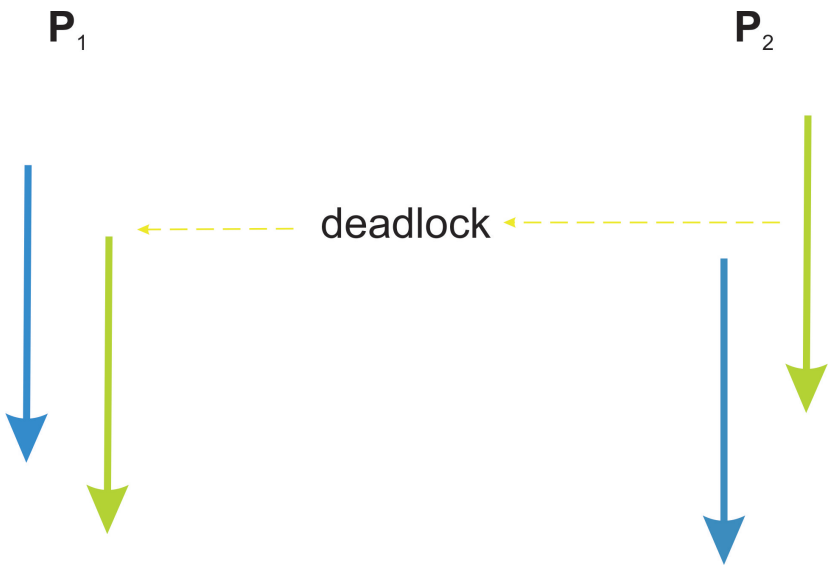


Exclusive access

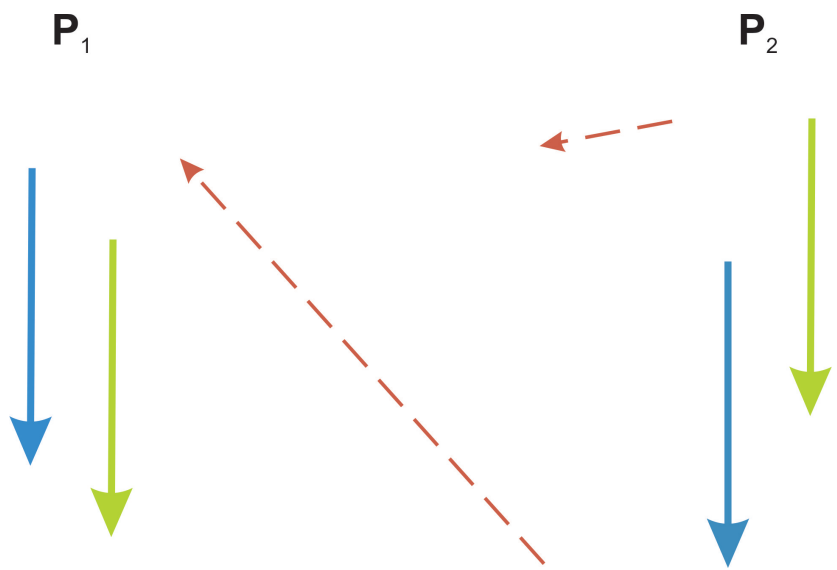


Deadlock on resources (devices/data)

Deadlock on resources (devices/data)



Deadlock on resources (devices/data)



Deadlock on resources problem solving

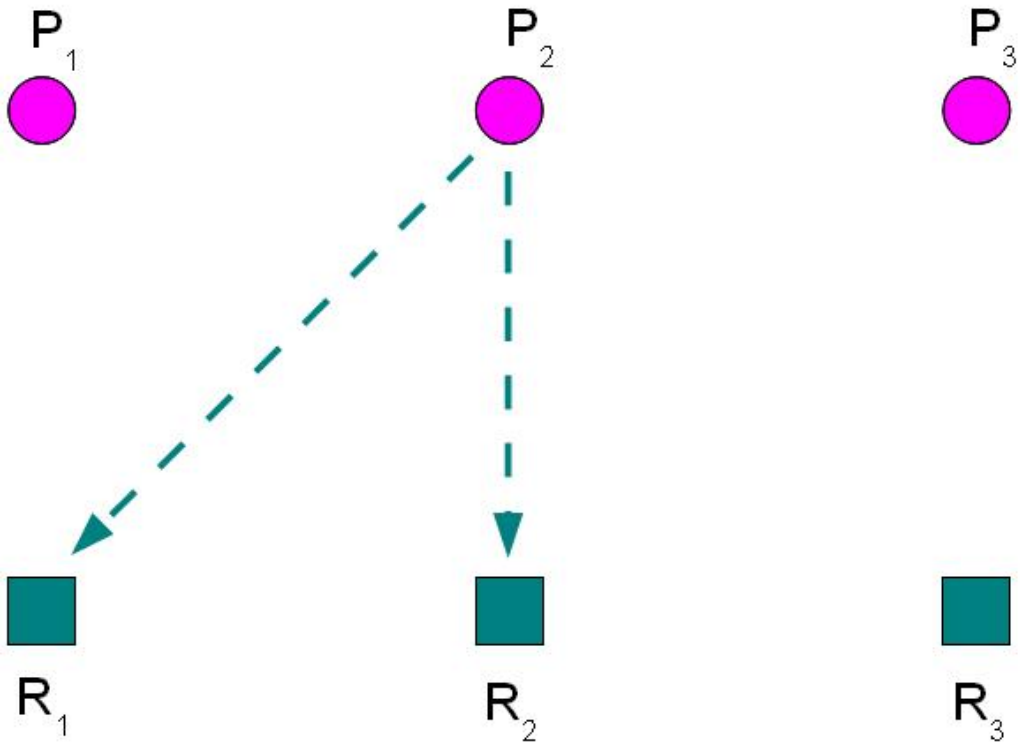
A priori (pessimistic) methods

Lomets algorithms

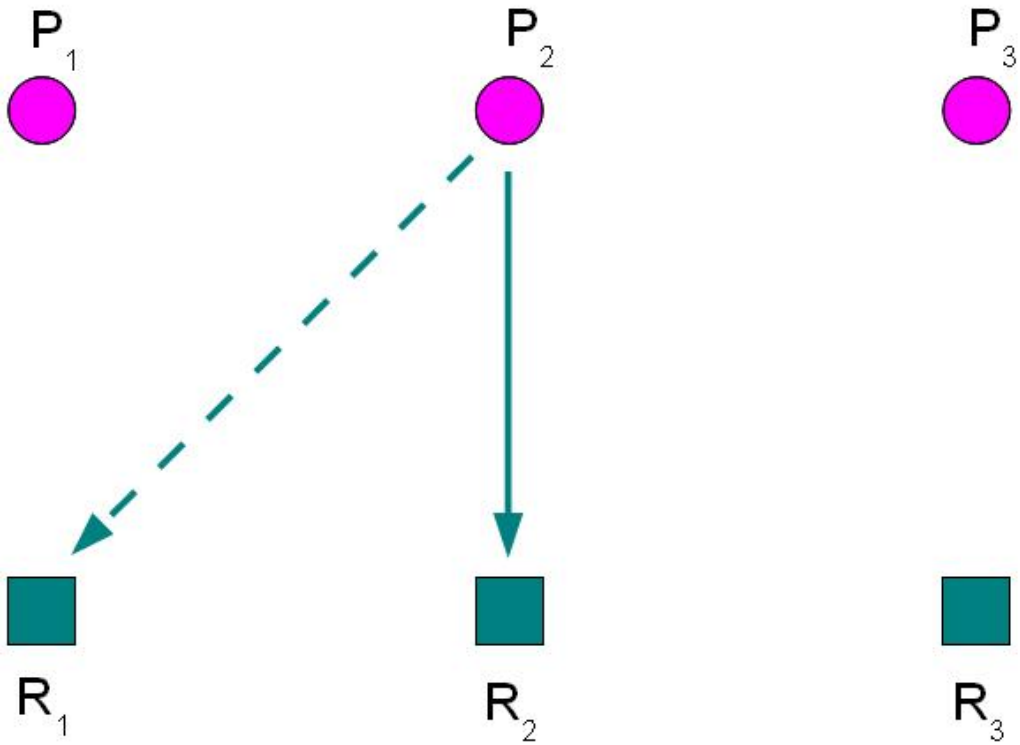
A posteriori (optimistic) methods

working with transactions

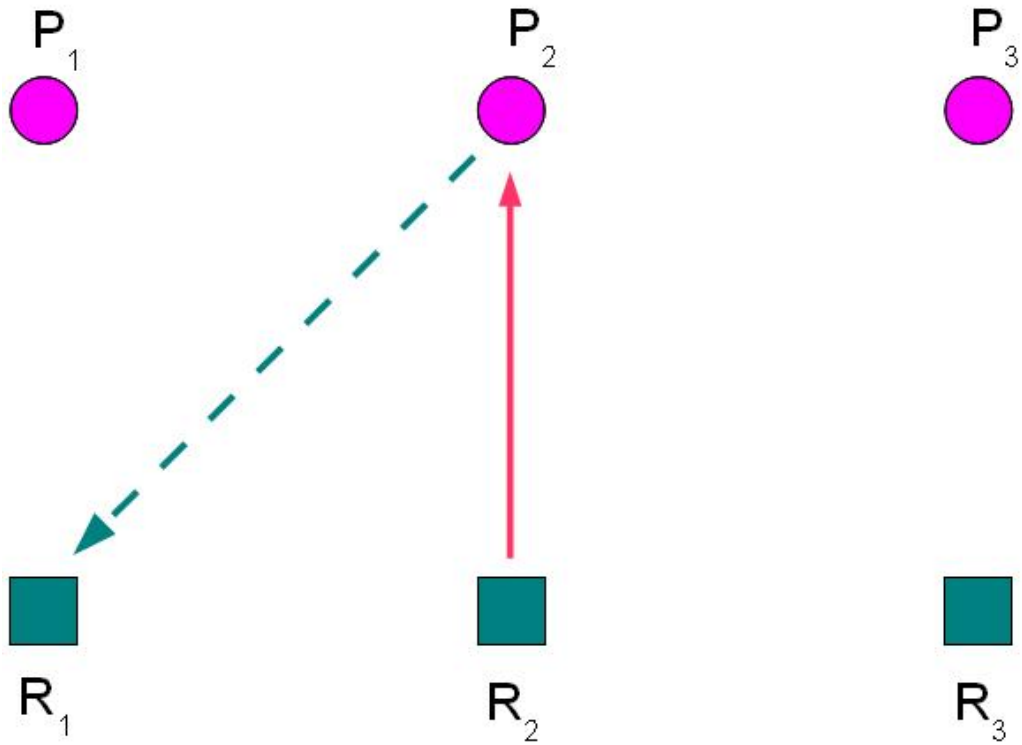
Preemptive algorithm - Lomet



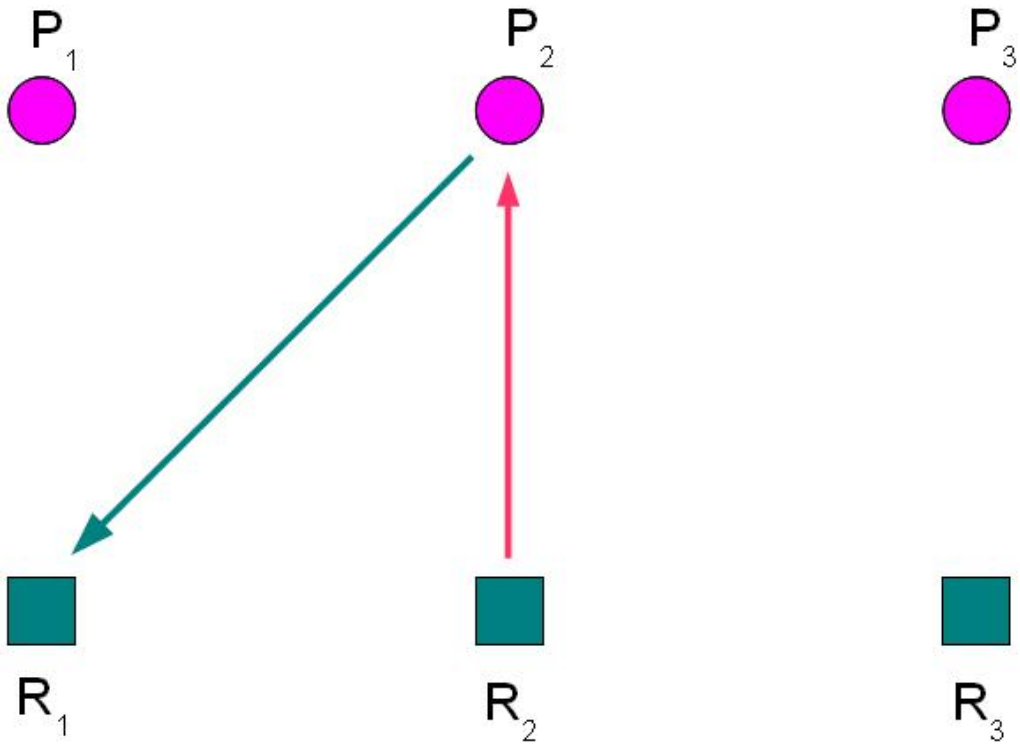
Preemptive algorithm - Lomet



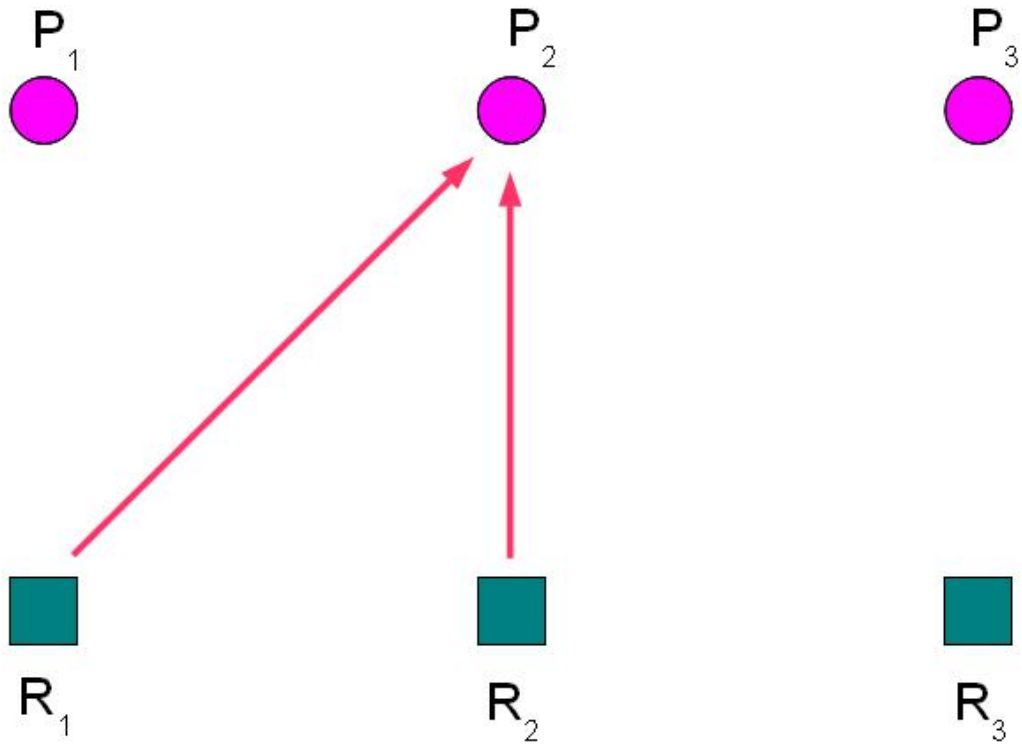
Preemptive algorithm - Lomet



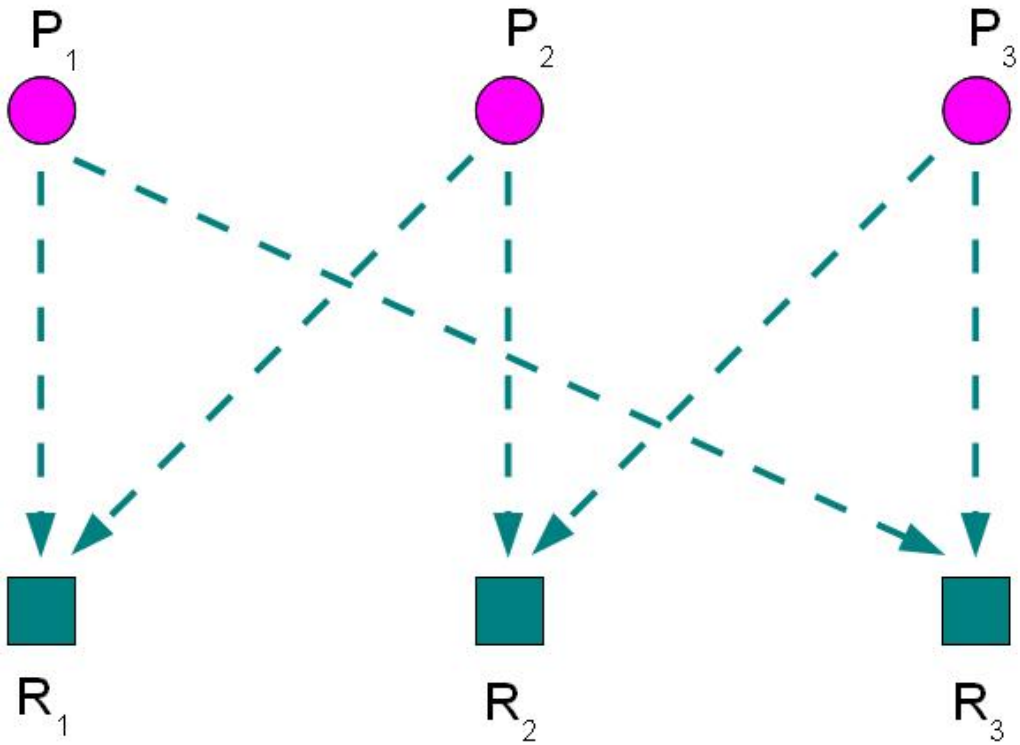
Preemptive algorithm - Lomet



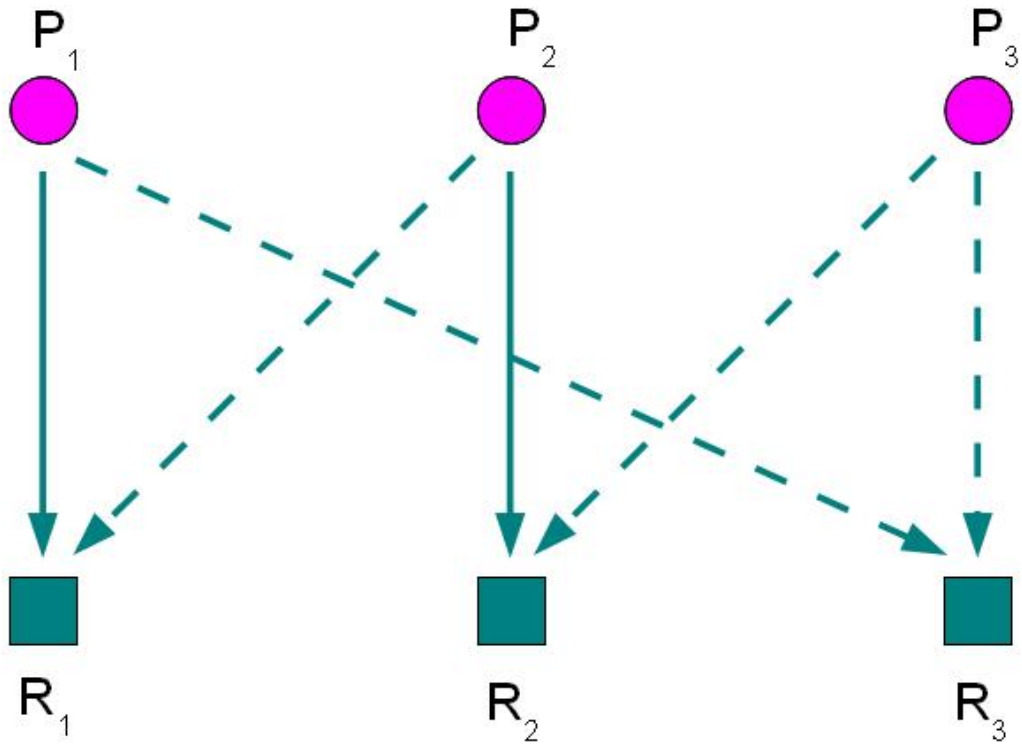
Preemptive algorithm - Lomet



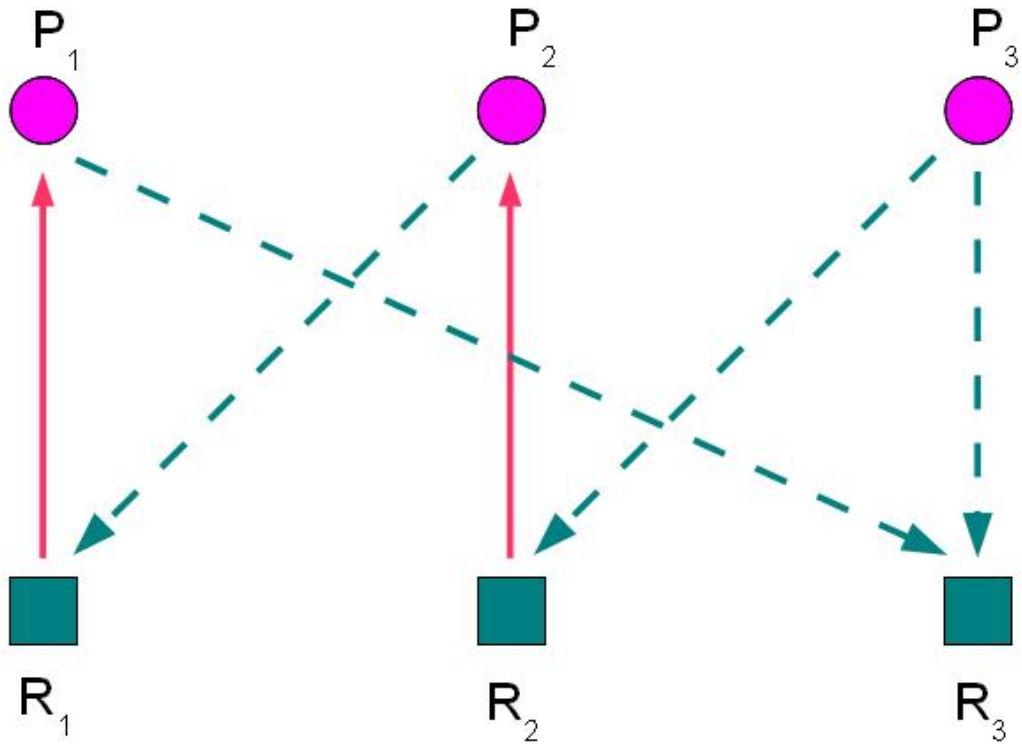
Preemptive algorithm - Lomet



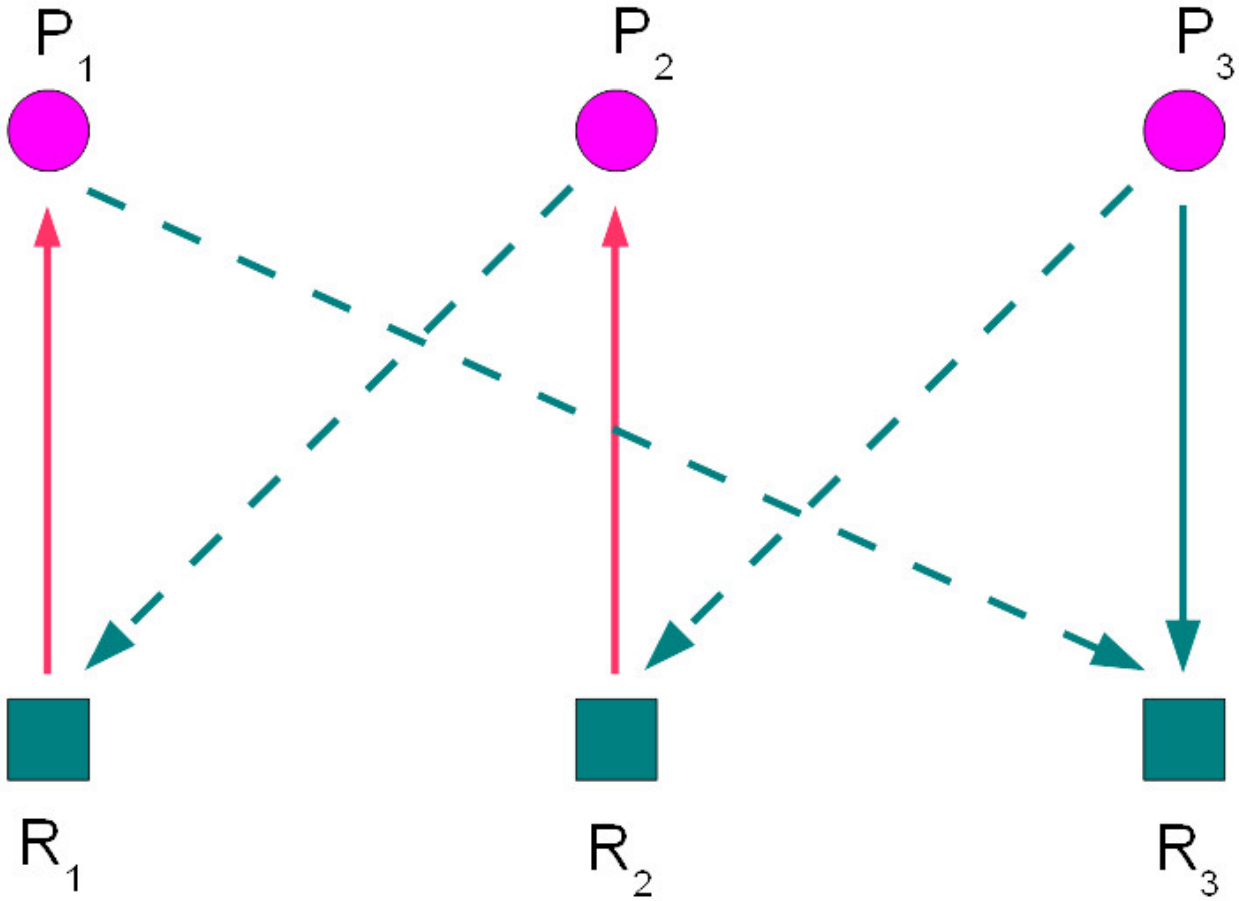
Preemptive algorithm - Lomet



Preemptive algorithm - Lomet

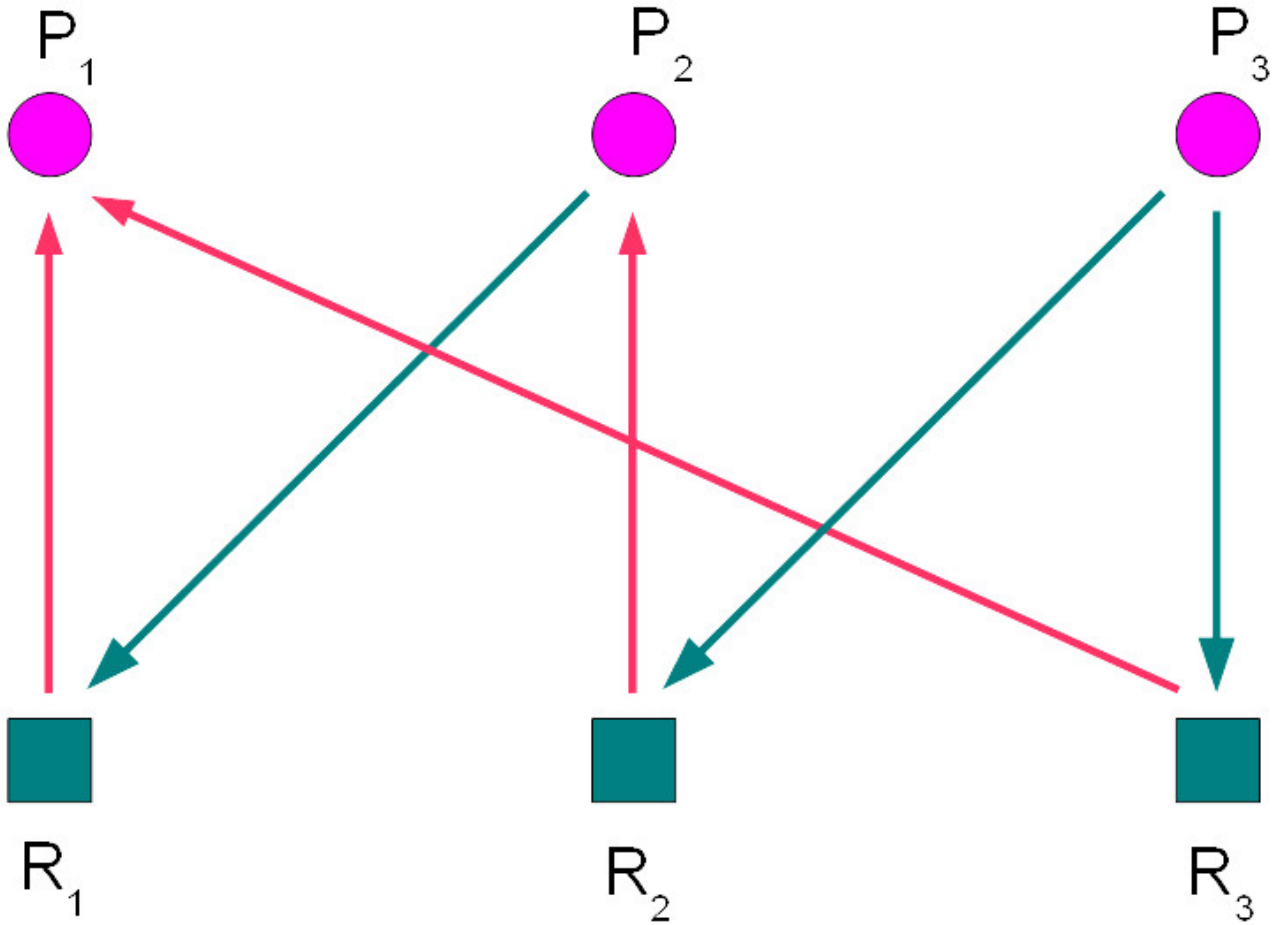


Preemptive algorithm - Lomet



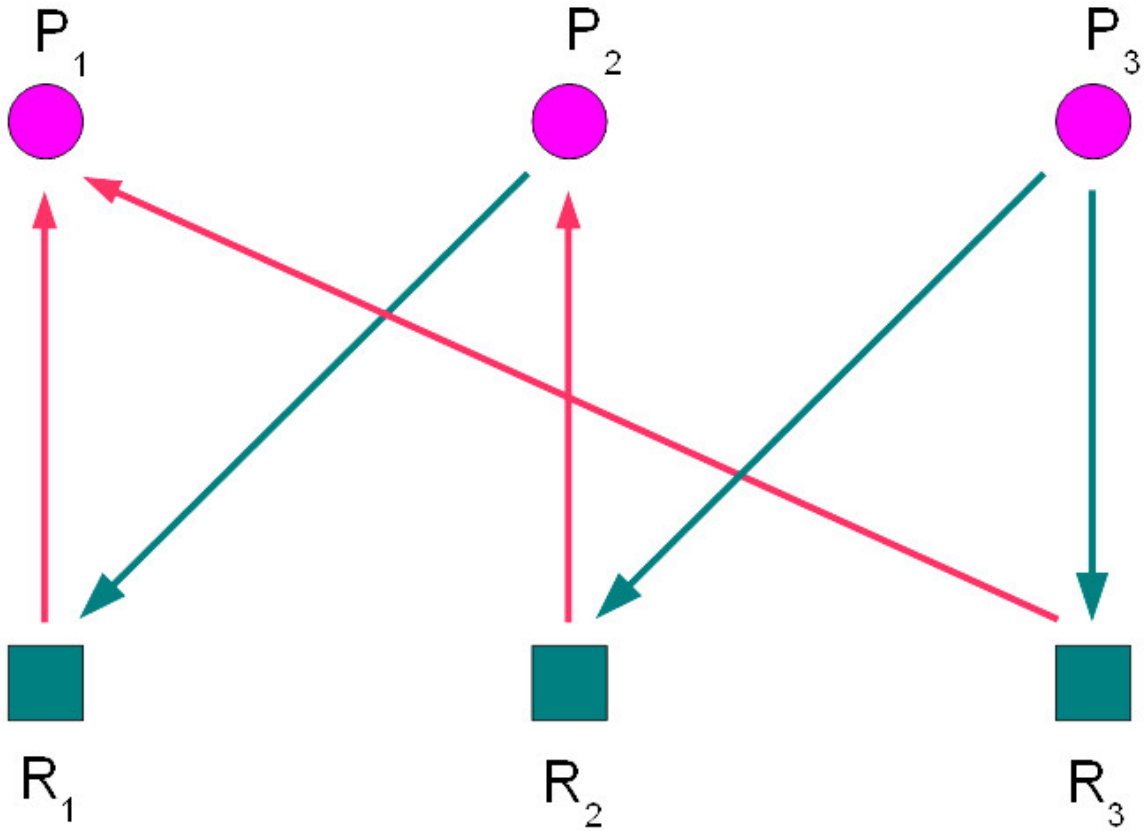
I have to wait here ...

Preemptive algorithm - Lomet



how to solve it ?

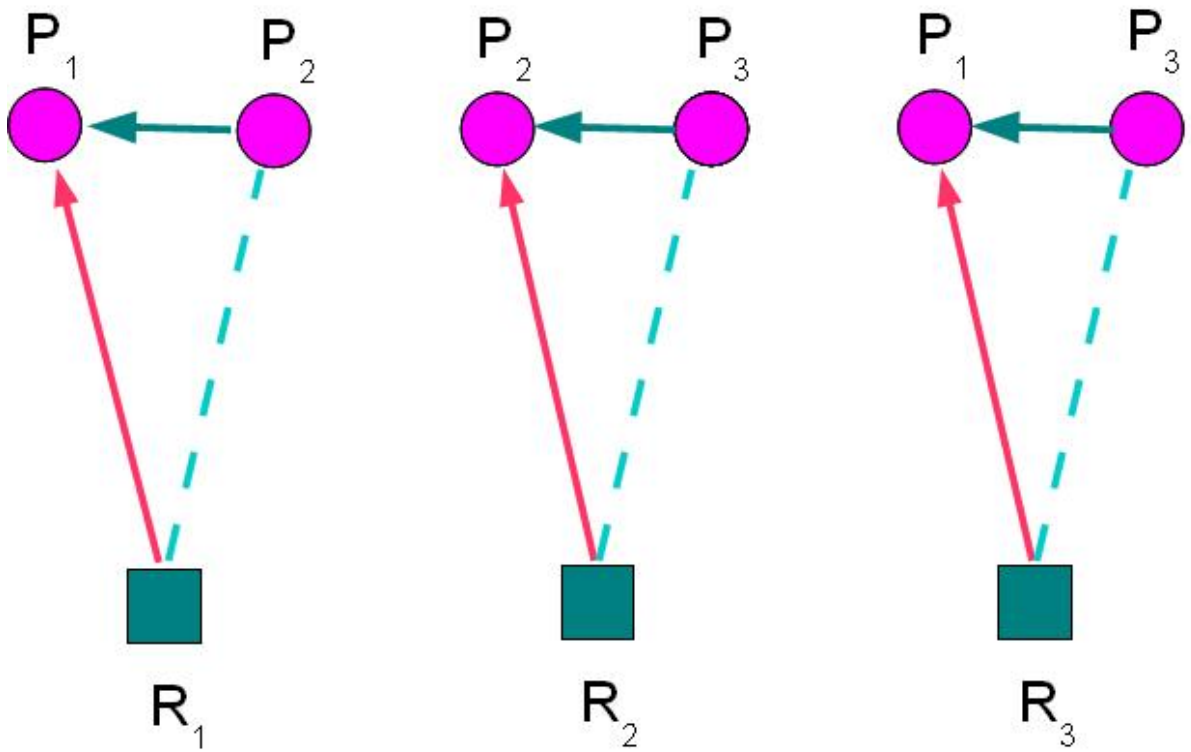
Preemptive algorithm - Lomet



how to solve it ?

... time stamps

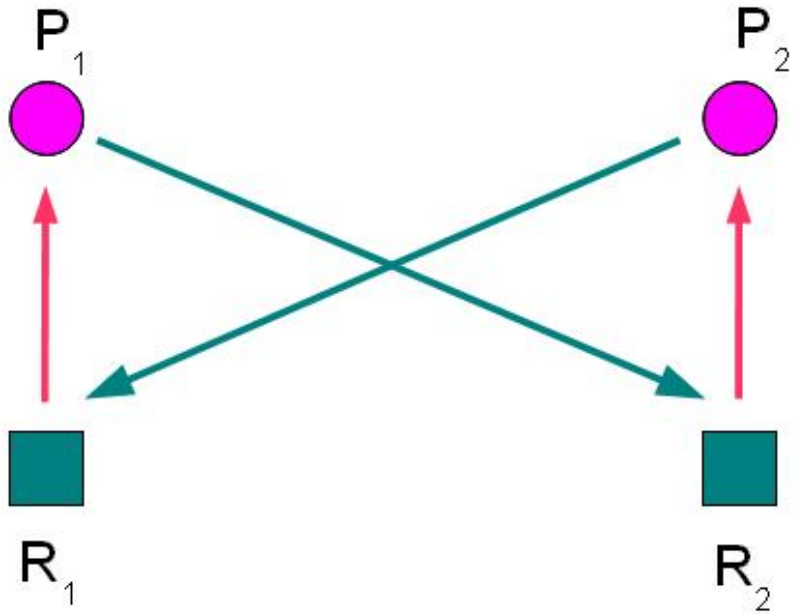
Preemptive algorithm - Lomet (local)



Detection

a posteriori algorithms

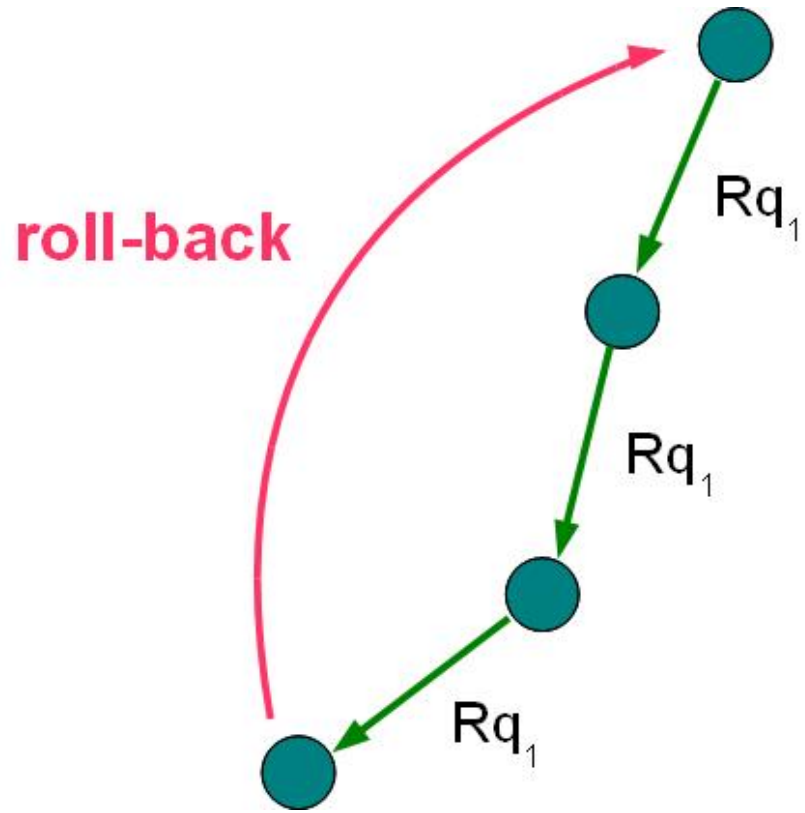
optimistic strategy



deadlock solution

- selection of a offering process
- release of resources
- returning transactions in processes

Detection


it is possible to use return in the selected transaction
roll-back
Rq1
Rq1
Rq1

Detection

assuming that device/data is used by transaction T_1 we can

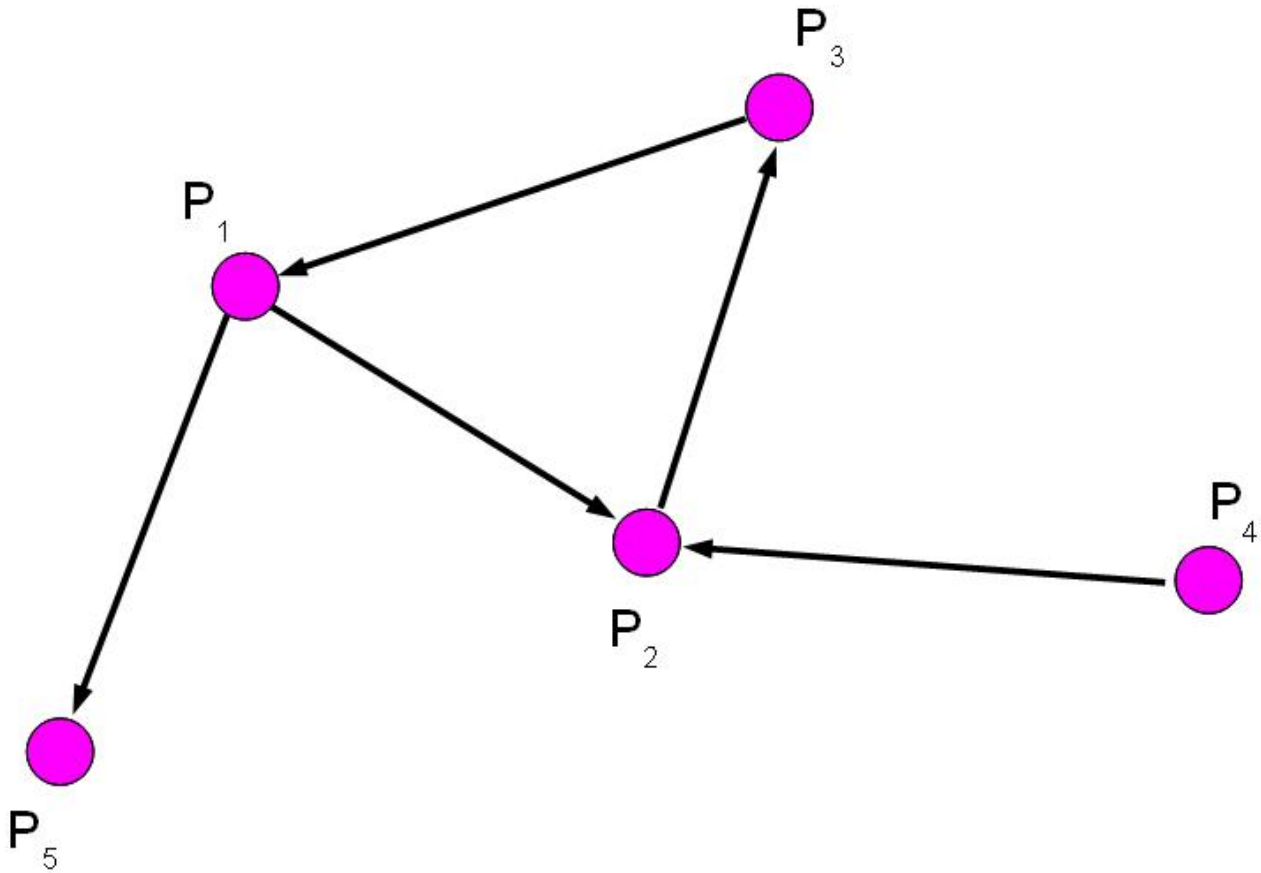
- kill newer transaction T_2 requiring device/data

if $e(T_2) < e(T_1)$ **then** halt T_2
else kill T_2

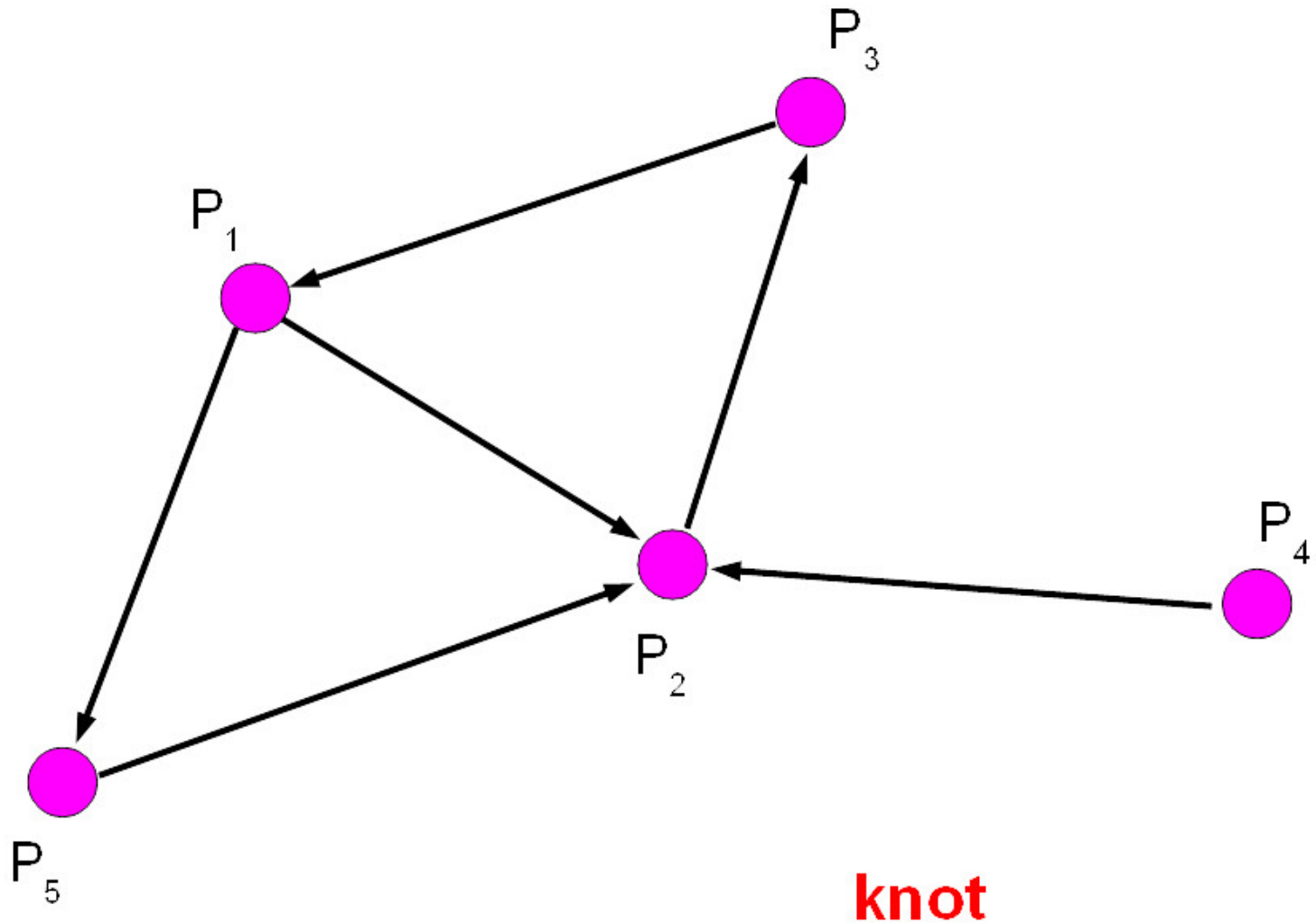
- kill newer transaction T_1 utilizing device/data

if $e(T_2) < e(T_1)$ **then** kill T_1
else halt T_2

Deadlock in communication



Deadlock in communication



Deadlock in communication

Chandy-Misra-Hass

```
when decision START DETECTION and State=PASSIVE do
 begin
  Last[i] := Last[i]+1;
  Wait[i] := T;
  for j in DSet do
    send QUESTION(i,Last[i],i,j) to j
  Number[i] := card(DSet)
 end

when received ANY OTHER MESSAGE do                    zpráva aplikace
 begin
  State := ACTIVE;
  for i:=1 to N do
    Wait[i] := F
 end
```

Deadlock in communication

Chandy-Misra-Hass

příjem dotazu

```
when received QUESTION(k,m,j,i) and State=PASSIVE do
 if m$>$Last[k] then
   begin
     Last[k] := m;
     Parent[k] := j;
     Wait[k] := T;
     for r in DSet do
       send QUESTION(k,m,i,r) to r;
     Number[k] := card(DSet)
   end
 else
 if Wait[k] and m=Last[k] then
   send ANSWER(k,m,i,j) to j
```

Deadlock in communication

Chandy-Misra-Hass

```
when received ANSWER(k,m,r,i) and State=PASSIVE do
  if m=Last[k] and Wait[k] then
    begin
      Number[k] := Number[k]-1;
      if Number[k]=0 then
        if k=i then
            { Pi je zablokován }
        else
        send ANSWER(k,m,i,Parent[k]) to Parent[i]
        Wait[k] := F
    end

begin                                          inicializace
  for i:=1 to n do
    begin
     Last[i] := 0; Wait[i] := F
    end
end
```