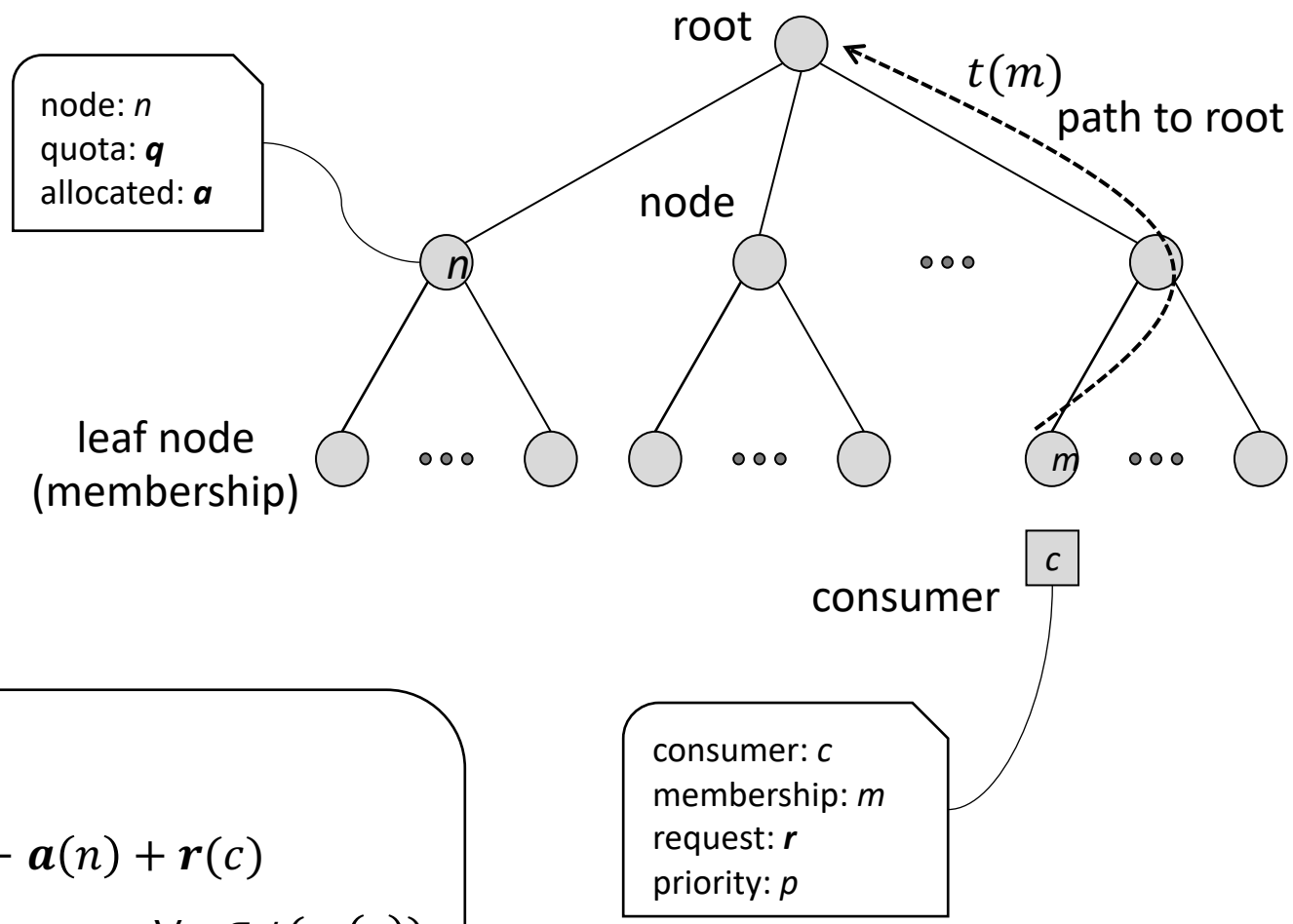
Hierarchical Quota Management

abstract modeling, implementation, and algorithms

Definitions



set of all nodes: N
set of leaf nodes: M
height of tree: $h \geq 1$

q, ***a***, ***r*** = [resources]
y = available capacity

allocate *c:*

$$\boldsymbol{a}(n) \leftarrow \boldsymbol{a}(n) + \boldsymbol{r}(c)$$

$$\forall n \in t(m(c))$$

such that

$$\boldsymbol{a}(n) \leq \boldsymbol{q}(n)$$

slack:

$$\boldsymbol{s}(n) = \boldsymbol{q}(n) - \boldsymbol{Q}(n)$$

where

$$\boldsymbol{Q}(n) = \sum_{v \in children(n)} \boldsymbol{q}(v)$$

$$\boldsymbol{s}(n) \lneqq \boldsymbol{0}$$

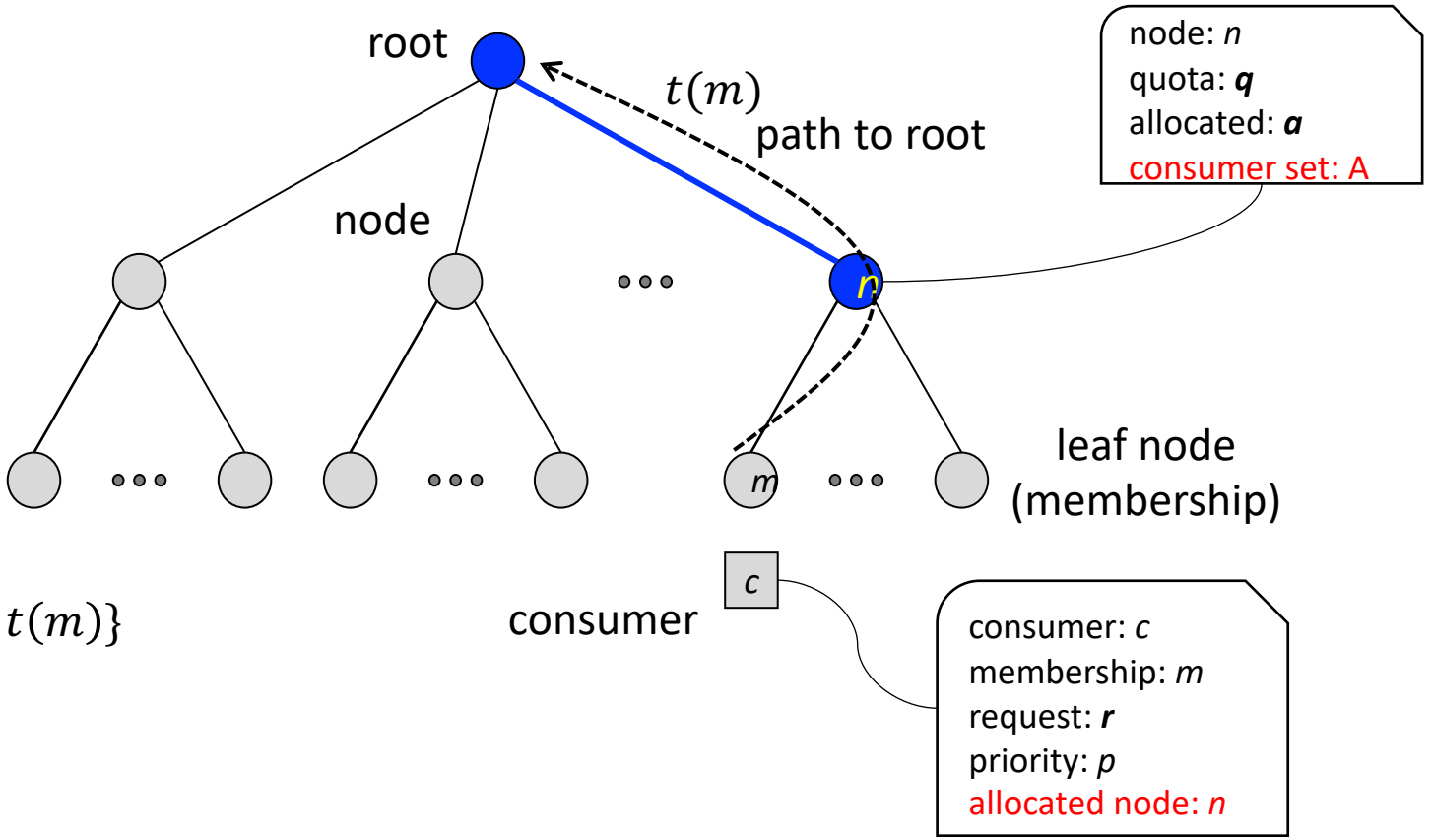
no slack (tight quota) ⇒

$$s\,(n) = 0, \forall n \in N$$

Algorithm Highlights

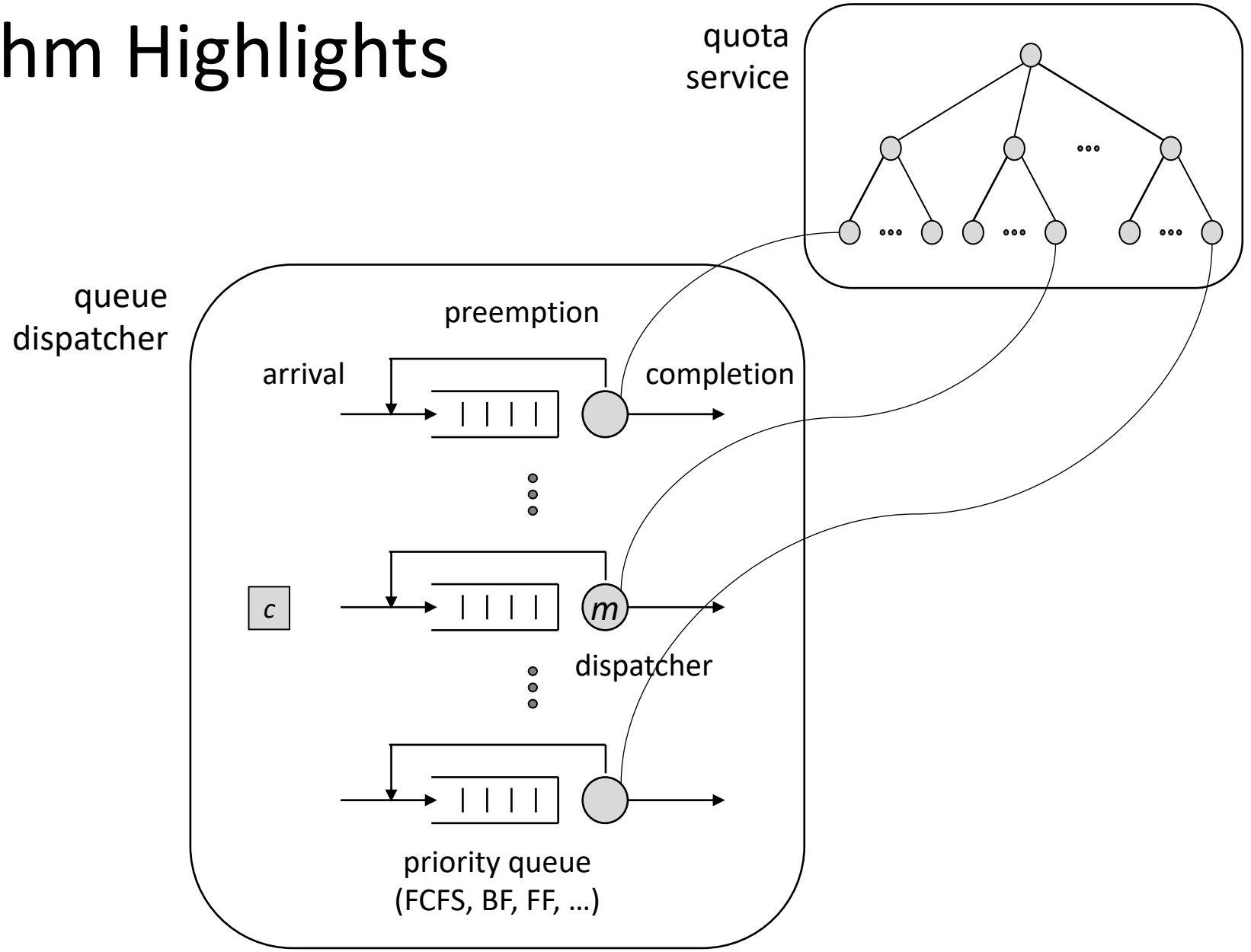
$$\boldsymbol{q}(root) = \boldsymbol{y}$$

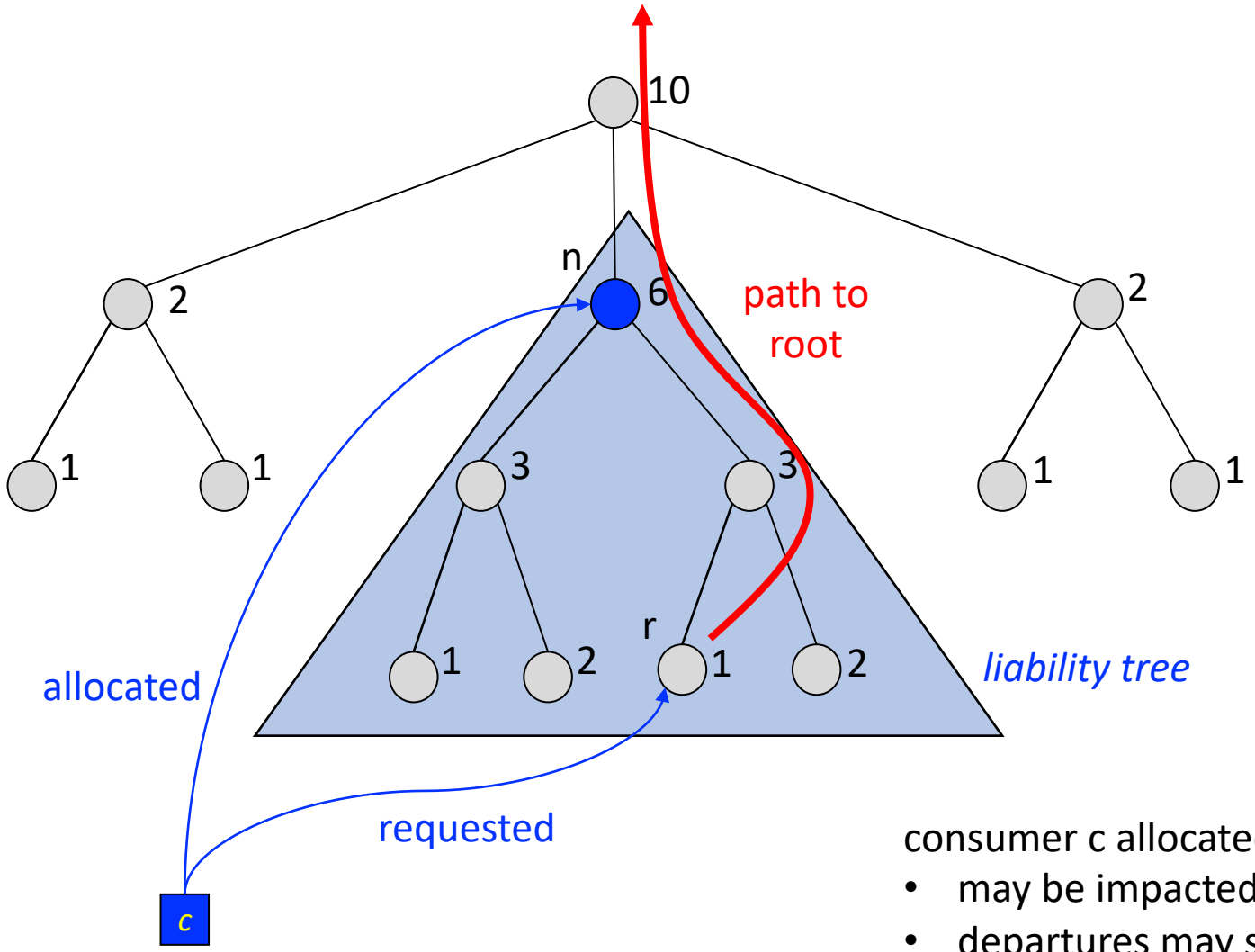
- allocate *c*
  - traverse path *t(m)* from *m*
  - find lowest level node *n* s.t.
    $\boldsymbol{a}(v) + \boldsymbol{r}(c) \leq \boldsymbol{q}(v), \forall v \in t(n)$
  - update allocations along path
    $\boldsymbol{a}(v) \leftarrow \boldsymbol{a}(v) + \boldsymbol{r}(c), \forall v \in t(n)$
    $A(n) \leftarrow A(n) \cup \{c\}$
  - if not found, attempt preemption
- preempt
  - create candidates set *D*
    $D = \{d | d \in A(n) \wedge p(d) < p(c) \wedge n \in t(m)\}$
  - select victims set *V* using some criterion
    $V = \{v | v \in D \wedge \sum_{u \in D} \boldsymbol{r}(u) \geq \boldsymbol{r}(c)\}$
- deallocate
  - starting from node $n(c)$ along the path $t(n(c))$
  - release $\boldsymbol{r}(c)$
  - slide down consumers as they fit



Algorithm Highlights

$$\boldsymbol{q}(root) = \boldsymbol{y}$$

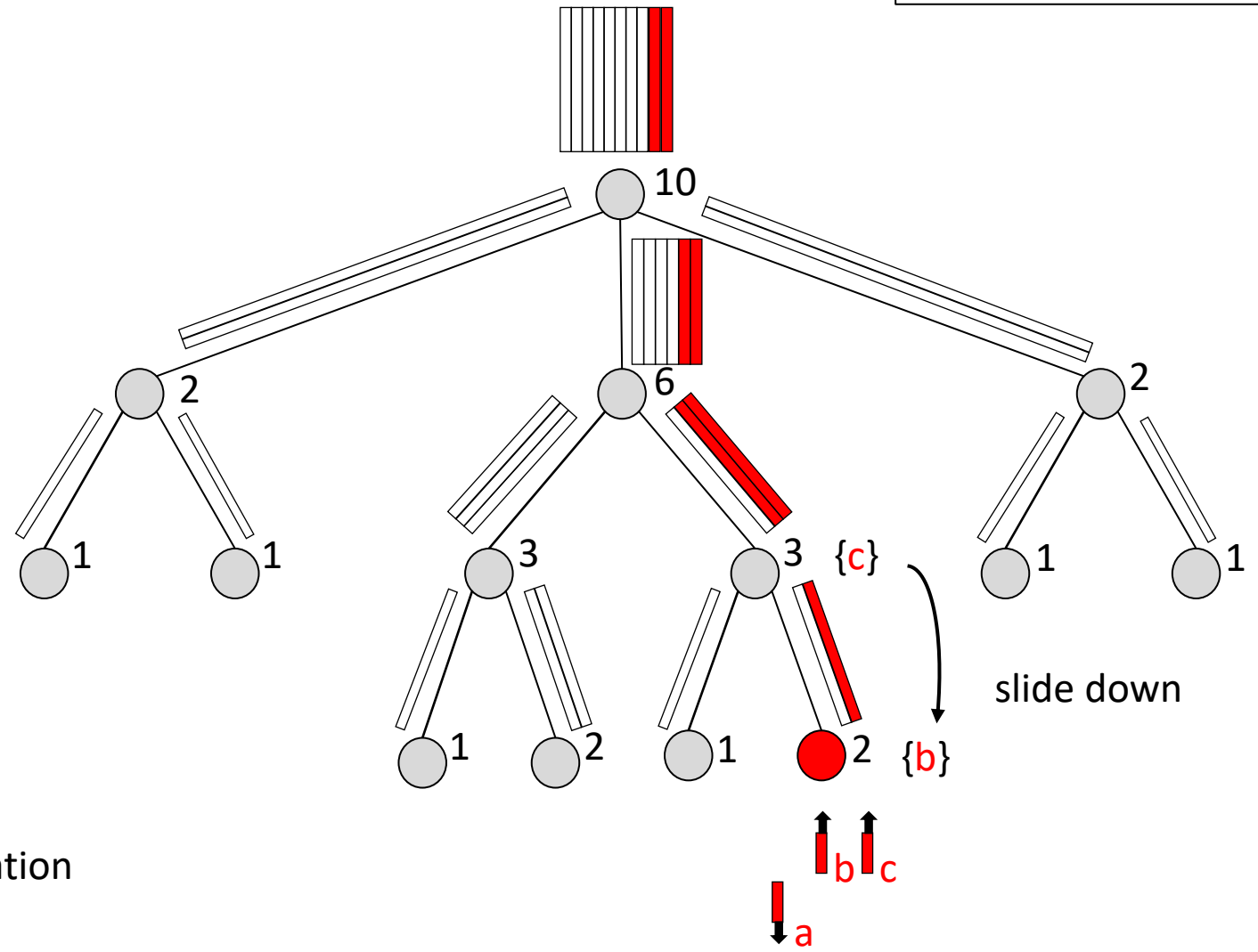




consumer c allocated at node n

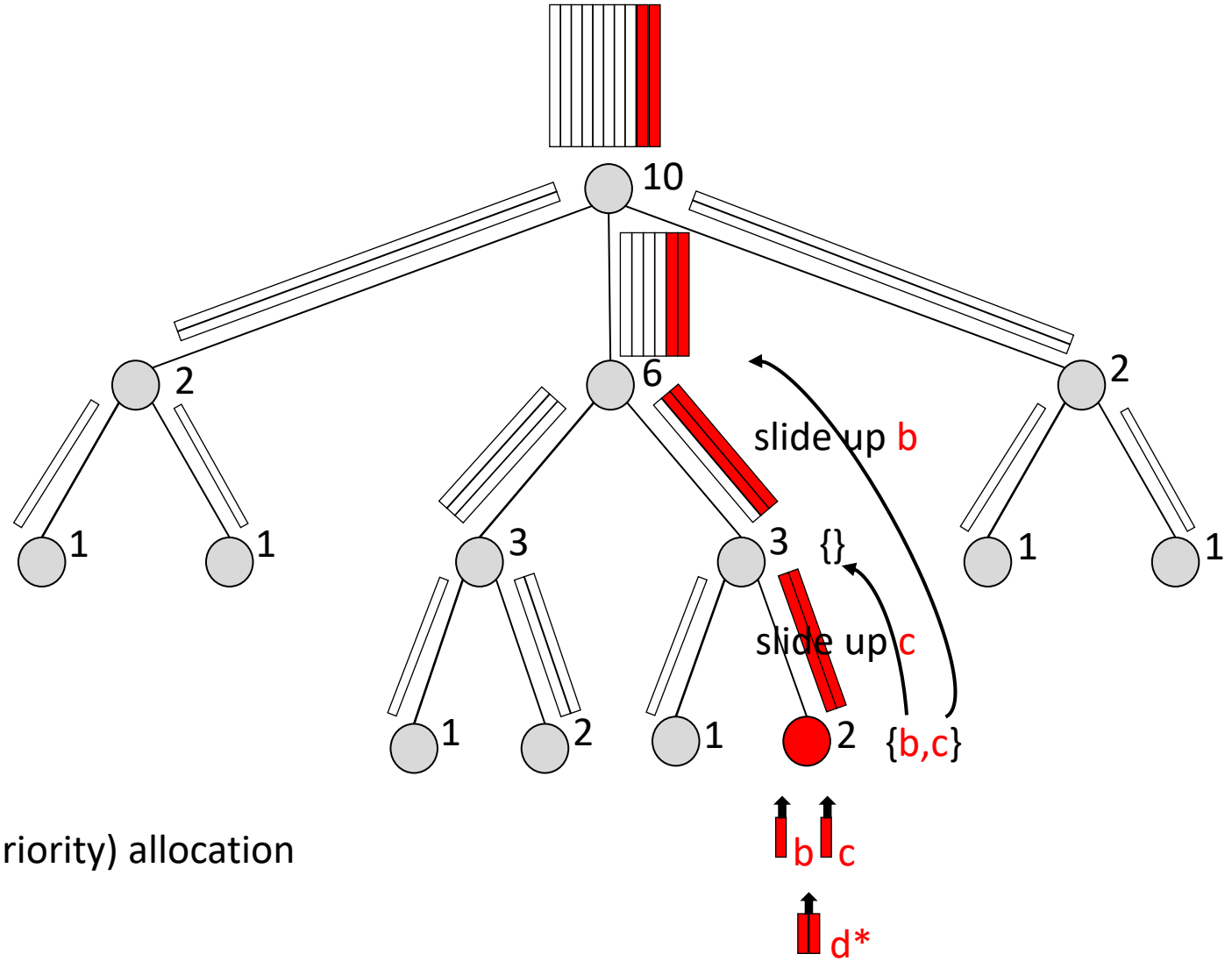
- may be impacted by consumers at subtree n
- departures may slide c down (less liability)
- arrivals may slide c up (more liability)
- arrivals may preempt c if node n is root
- higher priority arrivals may preempt c at any node n

- departures may slide c down (less liability)



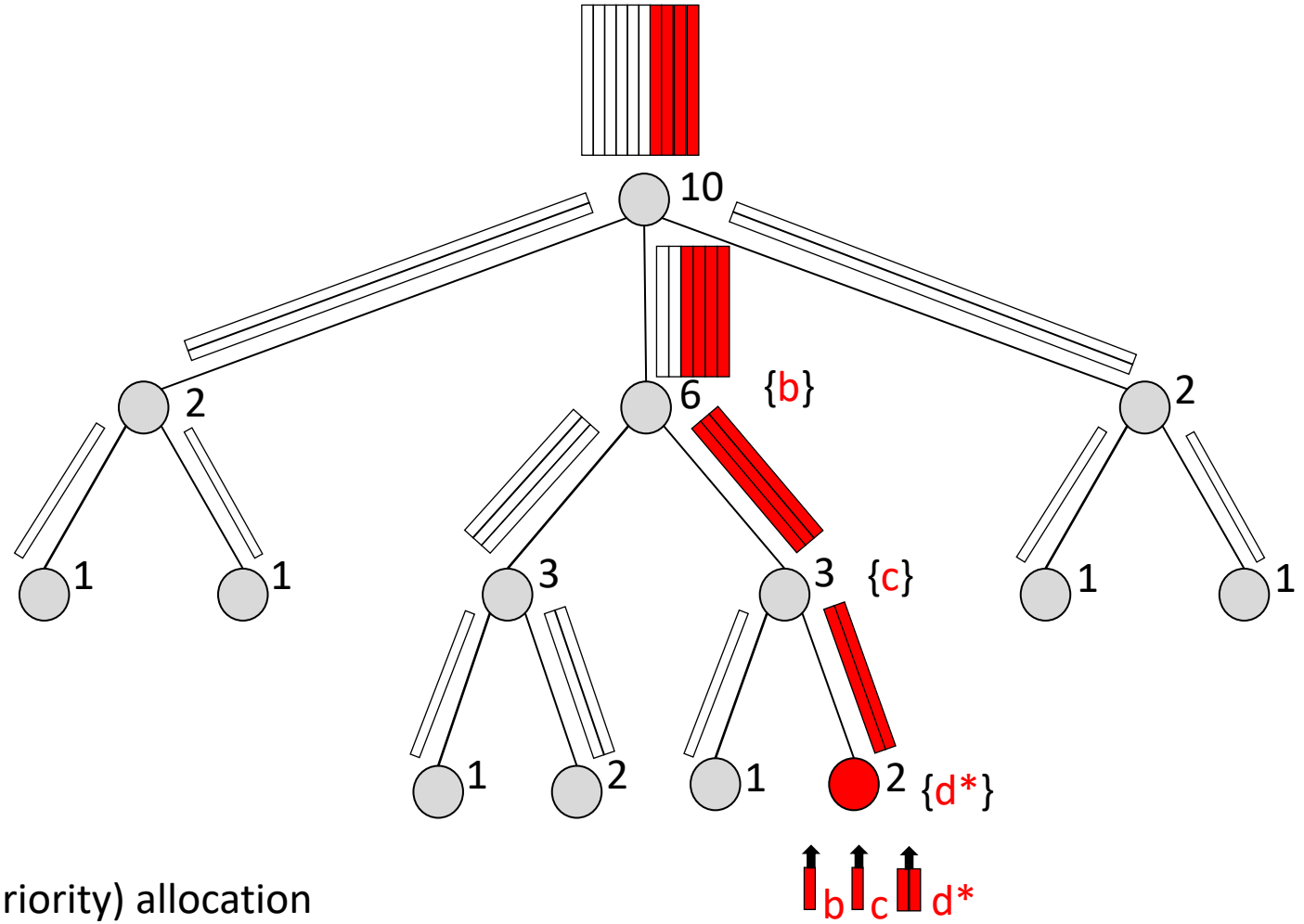
consumer a deallocation

- arrivals may slide c up (more liability)

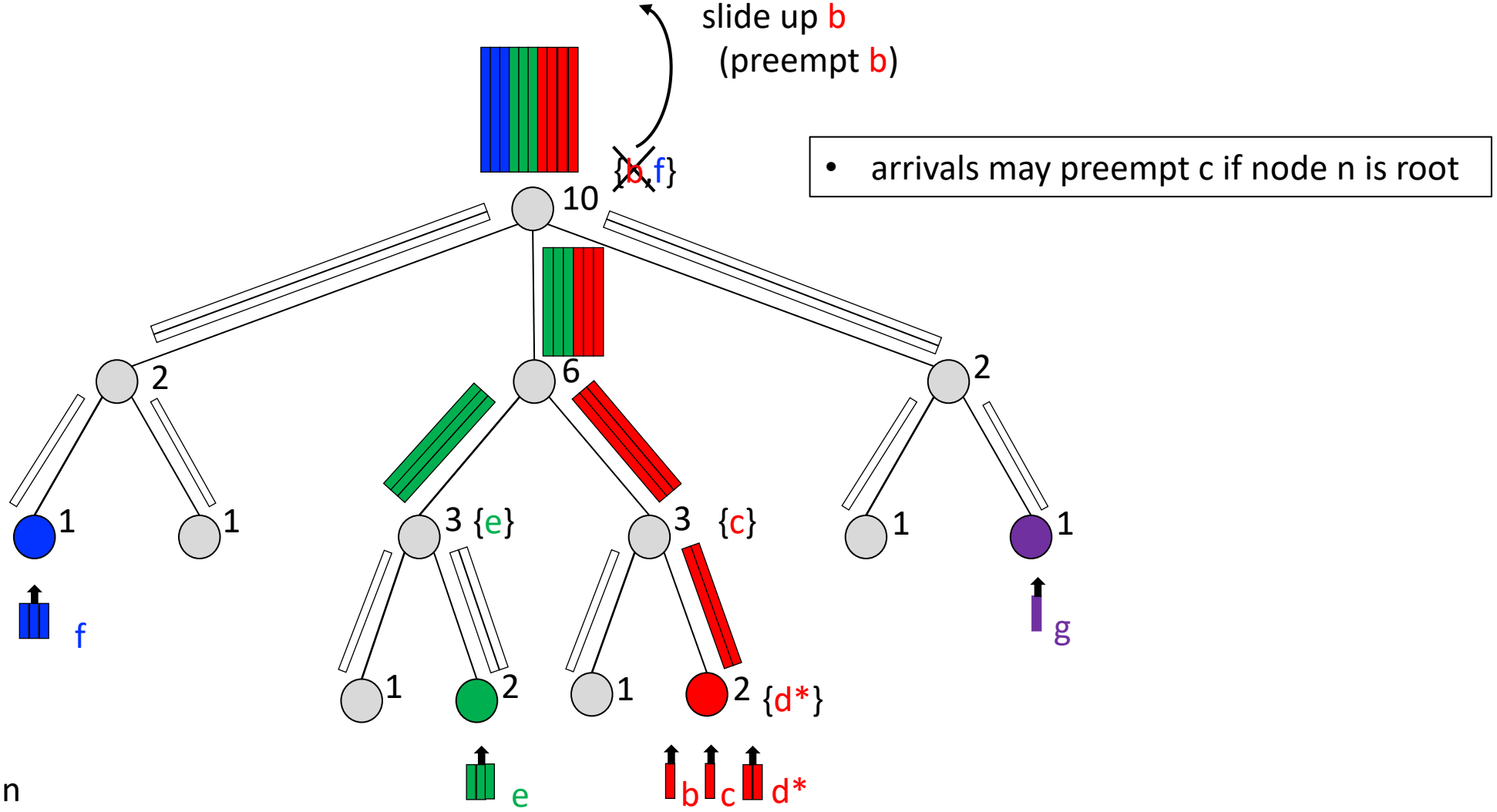


consumer d* (high priority) allocation

- arrivals may slide c up (more liability)

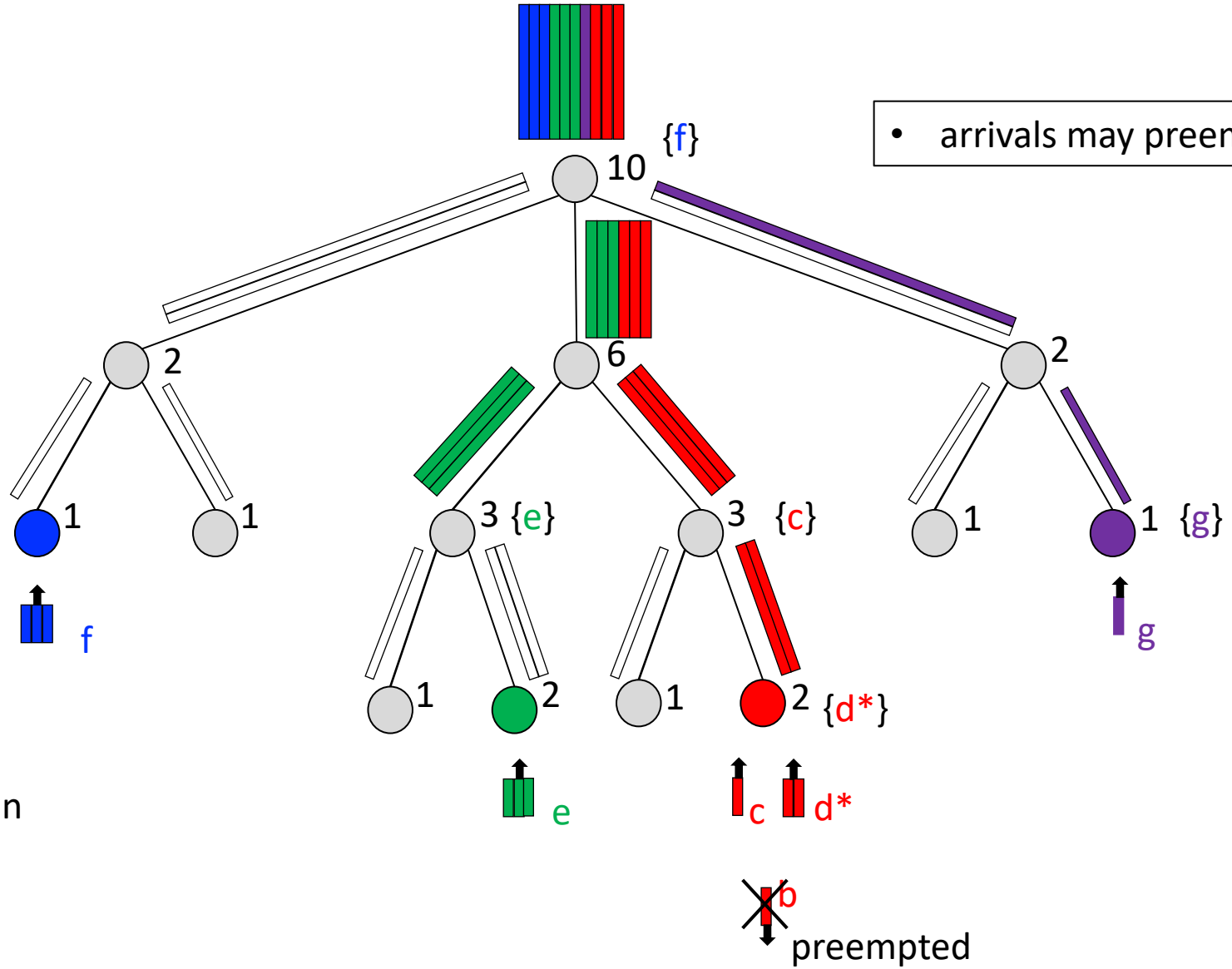


consumer d* (high priority) allocation



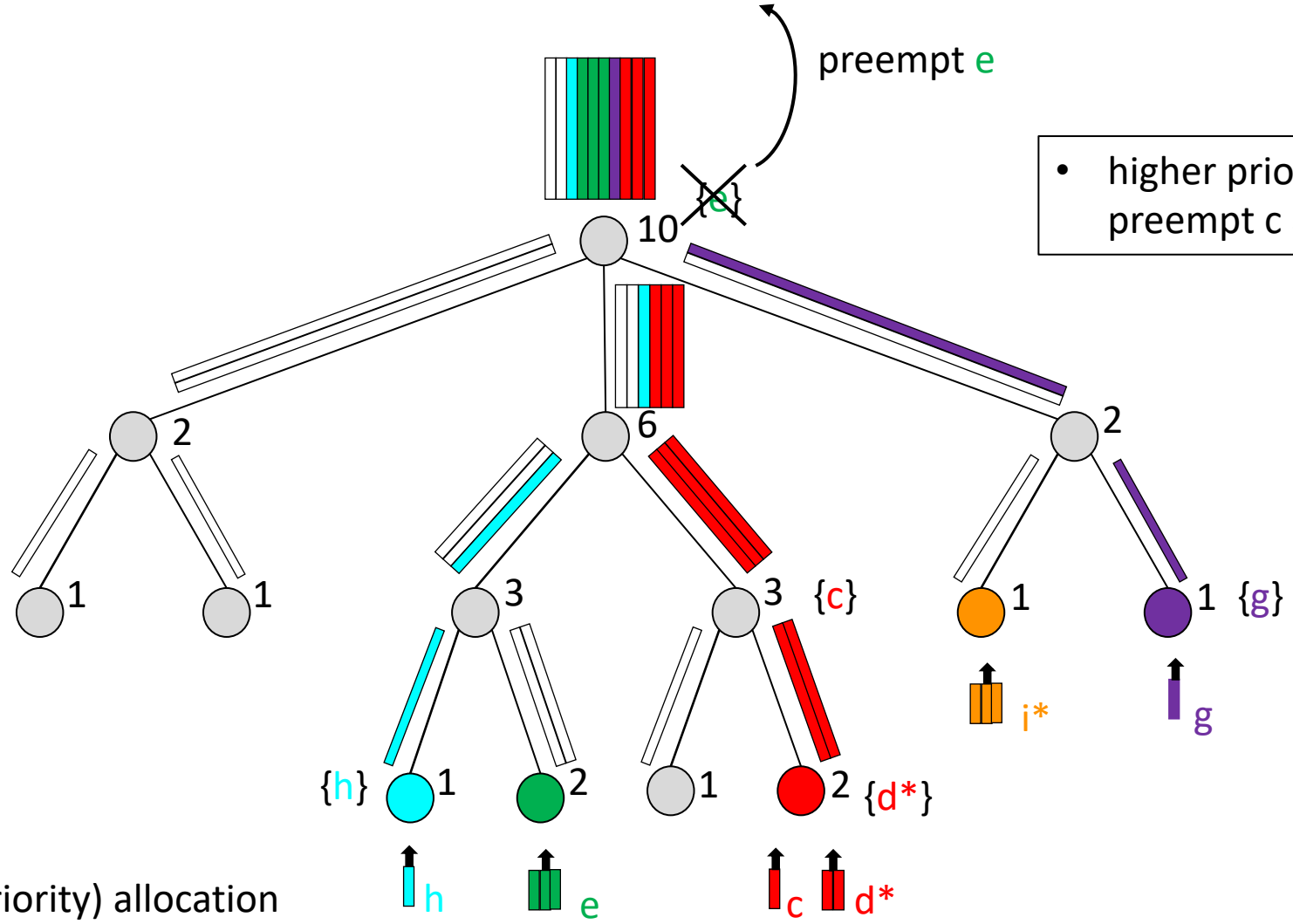
- arrivals may preempt c if node n is root

consumer g allocation



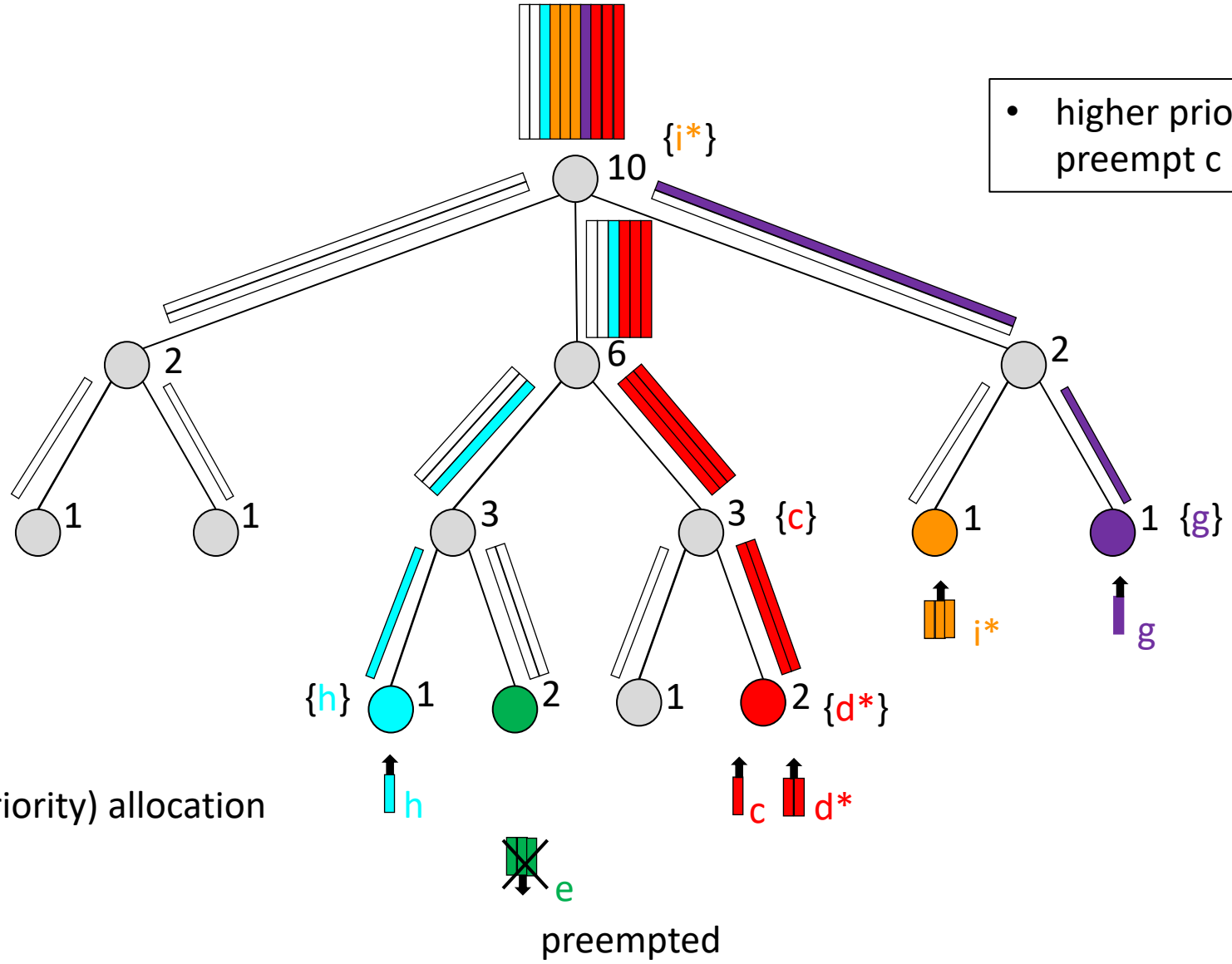
- arrivals may preempt c if node n is root

consumer g allocation



- higher priority arrivals may preempt c at any node n

consumer i* (high priority) allocation



- higher priority arrivals may preempt c at any node n

consumer i* (high priority) allocation