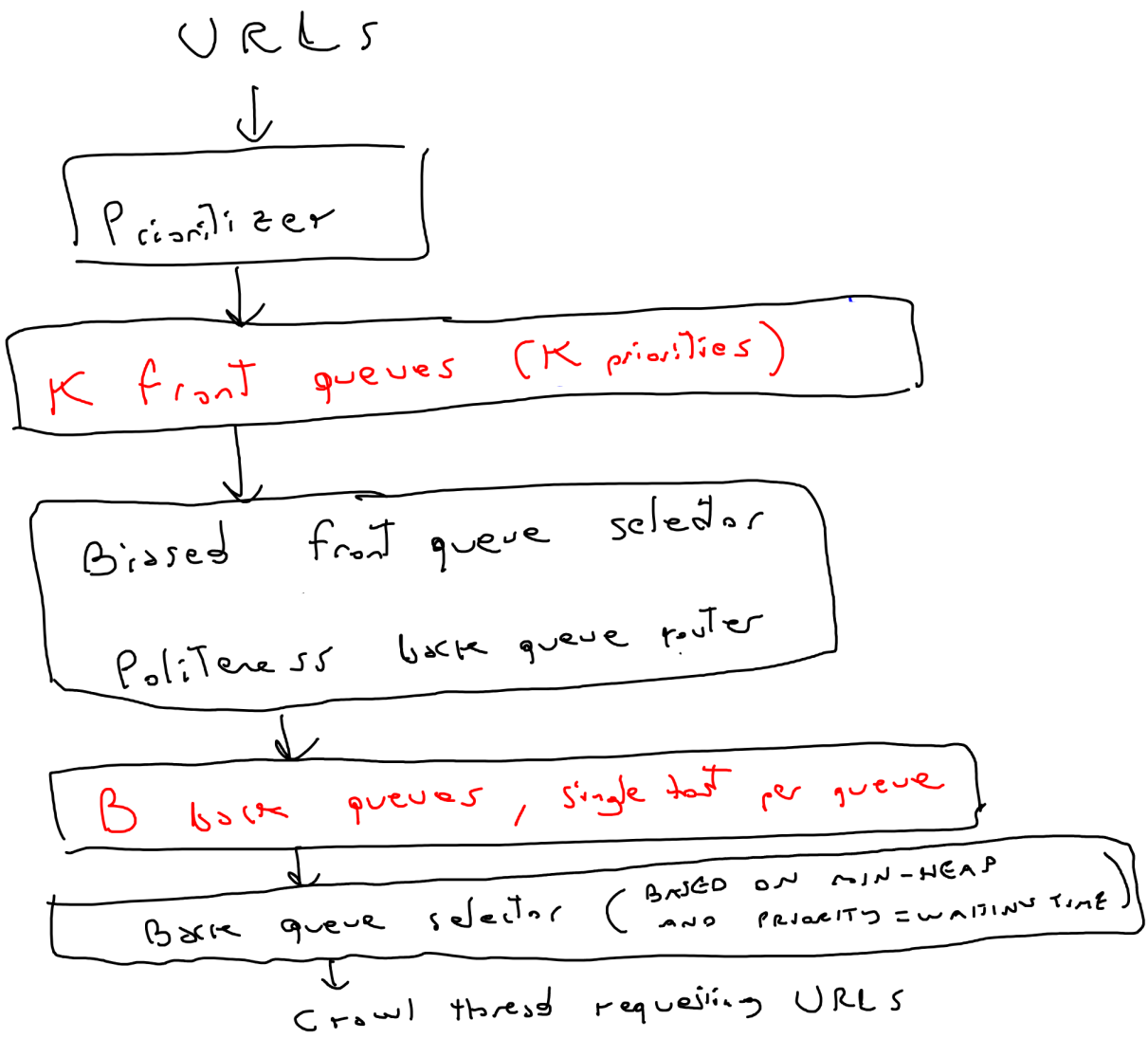
Mercator

A scalable web crawler

Characteristics

- Only one connection per host is open at a time
- Waiting time of a few second between successive requests to the same host
- High-priority pages are crawled preferentially

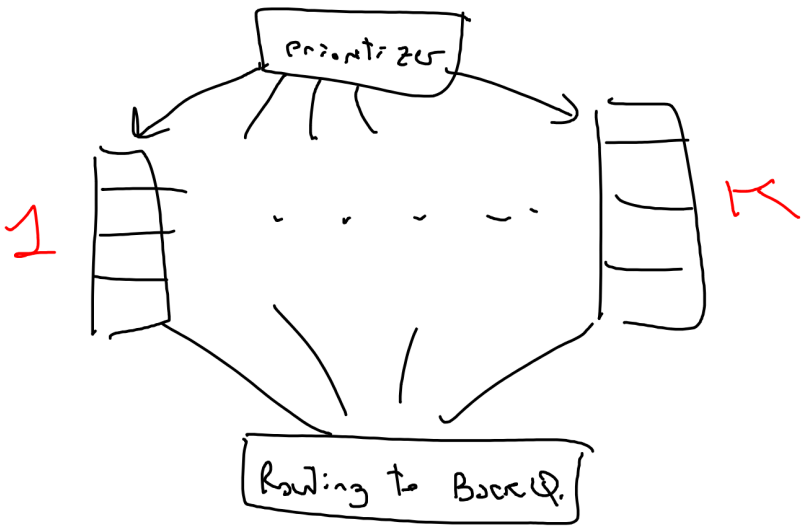


Prioritizer

Assign to an URL a priority between 1 and K and append it to the relative queue

K Front Queues

URLs divided in priorities

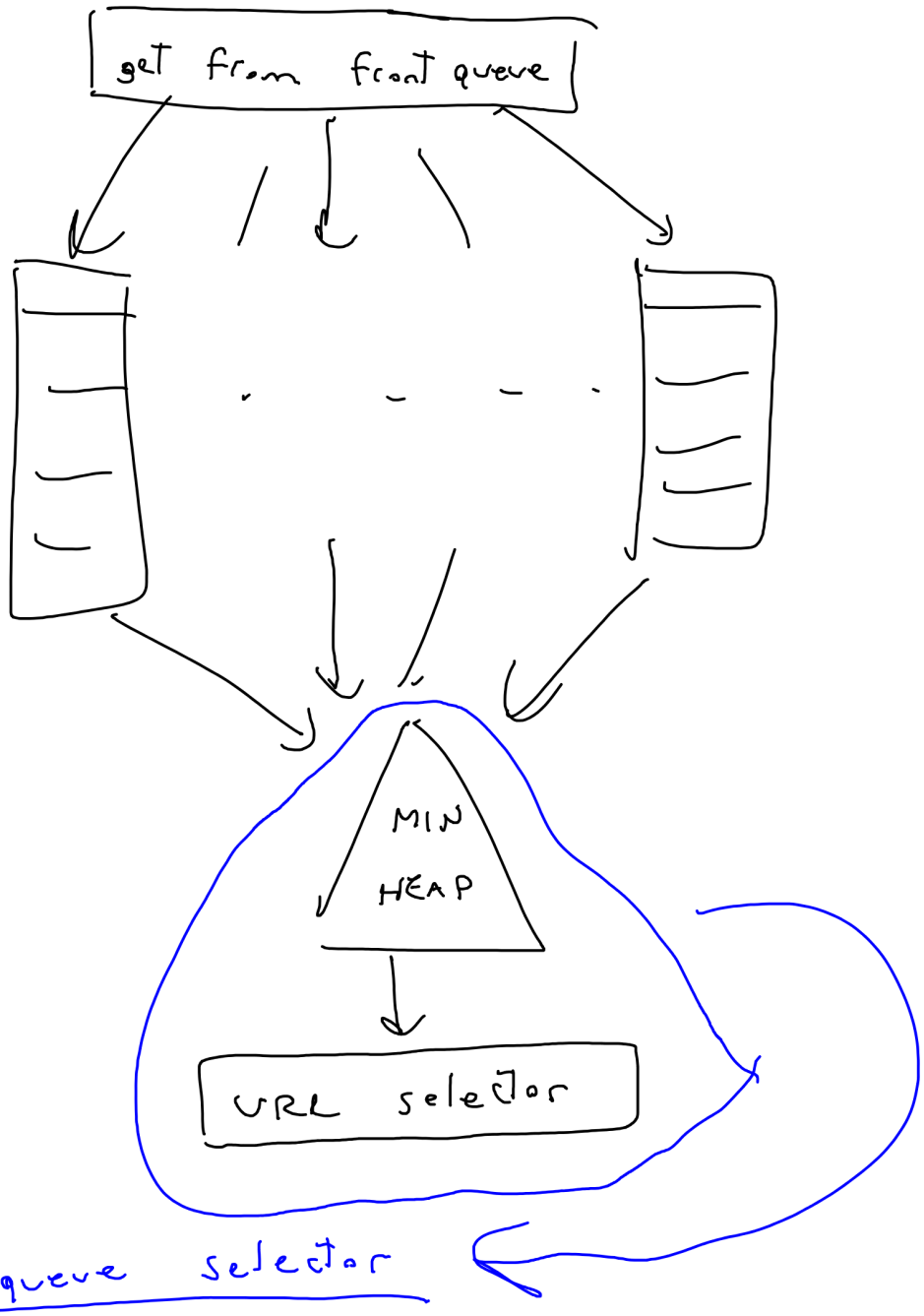


Biased front queue selector ....

Extract the high priority URLs with high probability and append to the back queues of its host.

B Back Queues

Used to enforce politeness: each queue is kept non-empty and contains only URLs from one host. USUALLY $B = 3K$



Back queue selector

A min heap contains one entry per back queue. The entries represent the earliest time t_e at which the corresponding host (back queue) can be crawled again.

This earliest time is determined from the last access of that host or from a time buffer heuristic.

CRAWL THREAD

Extract the root of the heap/
remove the URL from the head of a back queue,
waits the time te, parses the URL
and adds to the front queue again.

NOTE

Since the back queues are kept non-empty:

- if the extraction causes a back queue to go empty, it pulls a new URL v from the front queue:
  - if there is already a back queue for the host of v, append v to it and repeat until the empty back queue is nonempty again
  - if there is no back queue for the host of v, make the empty queue the new back queue for that host
- if the back queue is still non-empty, adds one of its URLs to the min-heap with priority = waiting time twe