Eventually Perfect Failure Detector
Interface of EPFD

- **Module:**
  - Name: EventuallyPerfectFailureDetector, instance $\diamond P$

- **Events:**
  - Indication: $\langle \diamond P, \text{suspect} \mid p_i \rangle$
    - Notifies that node $p_i$ is suspected to have crashed
  - Indication: $\langle \diamond P, \text{restore} \mid p_i \rangle$
    - Notifies that node $p_i$ is not suspected anymore

- **Properties:**
  - $PFD1$ (*strong completeness*)
  - $PFD2$ (*eventual strong accuracy*). Eventually, no correct node is suspected by any correct node.
Implementing $\diamond$ P

- Assume partially synchronous system
  - Eventually some bounds exist

- Each node every $\gamma$ time units
  - Send <heartbeat> to all nodes

- Each node waits $T$ time units
  - If did not get <heartbeat> from $p_i$
    - Indicate <suspect | $p_i$> if $p_i$ is not in suspected
    - Put $p_i$ in suspected set
  - If get HB from $p_i$, and $p_i$ is in suspected
    - Indicate <restore | $p_i$> and remove $p_i$ from suspected
    - Increase timeout $T$
Correctness of ◊P

- **EPFD1 (strong completeness)**
  - Same as before

- **EPFD2 (eventual strong accuracy)**
  - Each time p is inaccurately suspected by a correct q
    - Timeout T is increased at q
    - Eventually system becomes synchronous, and T becomes larger than the unknown bound $\delta$ ($T > \gamma + \delta$)
    - q will receive HB on time, and never suspect p again