Algorithm 1 FlowStatisticsCollectionScheduling(*Event e*)

globals: $active\_flows$ //Currently Active Flows
$schedule\_table$ //Associative table of active flows
// indexed by poll frequency
U // Utilization Statistics. Output of this algorithm

if e is *Initialization* event **then**
  $active\_flows \leftarrow \phi$, $schedule\_table \leftarrow \phi$, $U \leftarrow \phi$
end if
if e is a `PacketIn` event **then**
  $f \leftarrow \langle e.switch, e.port, \mathcal{T}_{min}, 0 \rangle$
  $schedule\_table[\mathcal{T}_{min}] \leftarrow schedule\_table[\mathcal{T}_{min}] \cup f$
else if e is timeout τ in $schedule\_table$ **then**
  for all flows $f \in schedule\_table[\tau]$ **do**
    send a `FlowStatisticsRequest` to $f.switch$
  end for
else if e is a `FlowStatisticsReply` event for flow f **then**
  $diff\_byte\_count \leftarrow e.byte\_count - f.byte\_count$
  $diff\_duration \leftarrow e.duration - f.duration$
  $checkpoint \leftarrow current\_time\_stamp$
  $U[f.port][f.switch][checkpoint] \leftarrow \langle diff\_byte\_count, diff\_duration \rangle$
  if $diff\_byte\_count < \Delta_1$ **then**
    $f.\tau \leftarrow \min(f.\tau\alpha, \mathcal{T}_{max})$
    Move f to $schedule\_table[f.\tau]$
  else if $diff\_byte\_count > \Delta_2$ **then**
    $f.\tau \leftarrow \max(f.\tau/\beta, \mathcal{T}_{min})$
    Move f to $schedule\_table[f.\tau]$
  end if
end if