CPU Scheduling in a Software-Based Network Packet Processor

Lothar Thiele
Samarjit Chakraborty

Software-Based NP

- How to Schedule the CPU cycles meaningfully?
  - Differentiating the level of service given to different flows
  - Each flow being processed by a different processing function

Our Model – Simple NP

- Real-Time Flows (RT)
- Best Effort Flows (BE)

- Real-time flows have deadlines which must be met
- Best effort flows may have several QoS classes and should be served to achieve maximum throughput

Task Model

- Packet processing functions may be represented by directed acyclic graphs
- End-to-end deadlines for RT packets
**Architecture**

```
Input ports  Classifier  Output ports
```

- **Packet Processing functions**
  - Real-time Flows
  - Best effort flows

**CPU Scheduling**

- First Schedule RT, then BE (background scheduling)
  - Overly pessimistic
- Use EDF Total Bandwidth Server
  - EDF for Real-Time tasks
  - Use the remaining bandwidth to server Best Effort Traffic
  - WFQ (weighted fair queuing) to determine which best effort flow to serve; not discussed here ...

**CPU Scheduling**

- As discussed, the basis is the TBS:
  - \[ d_k = \max\{r_k, d_{k-1}\} + c_k / U_k \]
  - computation demand of best effort packet
  - deadline of best effort packet
  - arrival of best effort packet
  - utilization by real-time flows
- But: utilization depends on time (packet streams)!
  - Just taking upper bound is too pessimistic
  - Solution with time dependent utilization is (much) more complex – BUT IT HELPS ...
CPU Scheduling

a) Before

- plain best effort + EDF scheme

end-to-end packet delay [sec]

b) plain best effort + EDF scheme

end-to-end packet delay [sec]

CPU Scheduling

a) After

- approximation with two segments

Simulation time [sec]

- approximation with two segments

Simulation time [sec]