ECEN 5623

Dynamic Priorities

Dynamic Priority

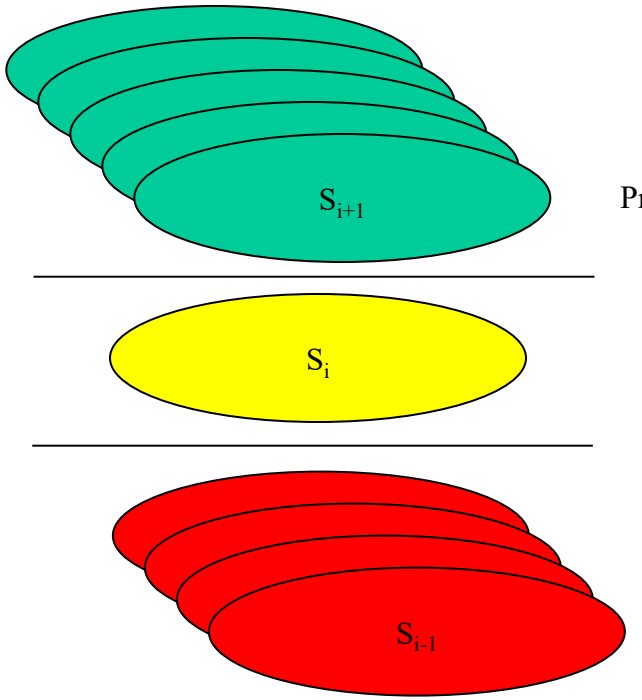
- EDF [Earliest Deadline First] – When Ready Queue is Updated, the Scheduler Must Re-assign all Priorities According to Each Service's Time Remaining to Deadline

- LLF [Least Laxity First] - Ready Queue is Updated, the Scheduler Must Re-assign all Priorities According to Each Service's (Time Remaining to Deadline - Execution Time Remaining)

Dynamic Priority Issues

- Earliest Deadline First Policy (aka Deadline Driven Scheduling)
  - ***Can Be High Overhead*** - Every time the Ready Queue is Updated, the Scheduler Must Re-assign all Priorities According to Each Service's Time Remaining to Deadline and Perform a Priority Preemption.
  - ***Overload Failure Modes Non-Deterministic*** - In an Overload Scenario, EDF Can Have Difficult to Predict and Cascading Multi-Service Failures
    - Overload with RM Policy is Deterministic - the Lowest Priority Service Will Fail First and The Number of Services Failing From Lowest Priority Up Can be Determined by Duration of Over-run and Priority of Over-running Service
      - Over-run Interference Has Known Impact on Lower Priority Services
      - Over-runs Will Be Preempted By Higher Priority Services if Allowed to Continue
    - Overload with EDF Policy is Non-Deterministic
      - Time to Deadline for an Over-running Service is Zero or Negative?
      - Over-running Service Will Maintain High Priority Based on Time to Deadline
      - Additional Services that Over-run Due to Interference Also Have Elevated Priority
      - Chaining Priority Amplification and Over-runs Can Cause Cascading Failure
    - Over-run Termination Policy May Help (Worst Case Interference Margin)
  - ***Difficult to Debug Dynamic Priority When Deadlines Are Missed***

RM Fixed Prio Overload

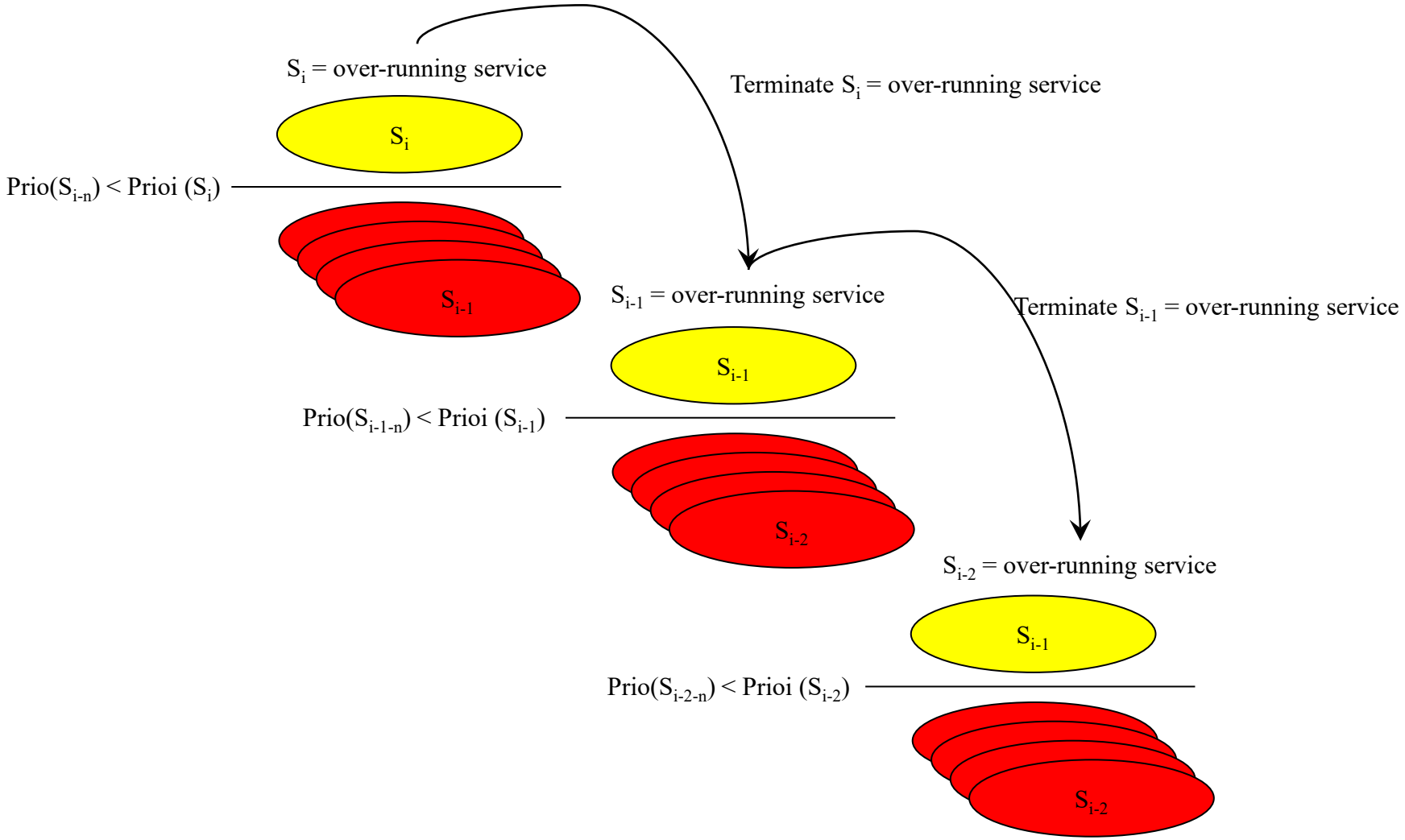


$Prio(S_{i+n}) > Prioi\ (S_i)$

S_i = over-running service

$Prio(S_{i-n}) < Prioi\ (S_i)$

EDF Dynamic Prio Overload

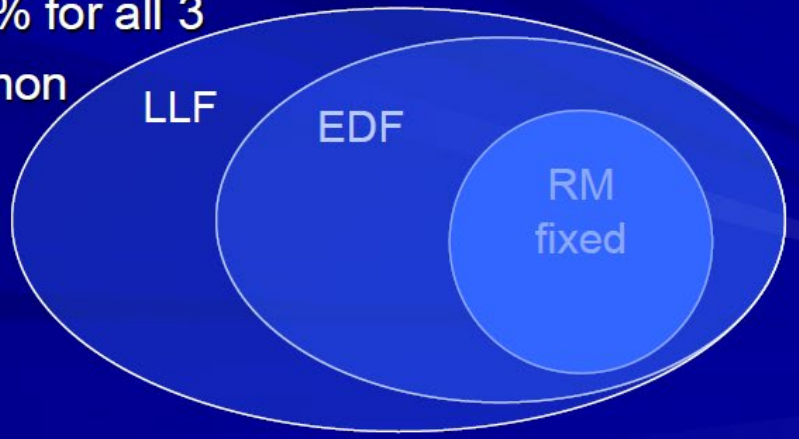


Dynamic Priority Issues Cont'd

- Least Laxity First Policy
  - ***Can Be High Overhead*** - Every time the Ready Queue is Updated, the Scheduler Must Re-assign all Priorities According to Each Service's (Time Remaining to Deadline - Execution Time Remaining) and Perform a Priority Preemption.
  - ***Execution Time Remaining Hard to Estimate*** - Compared to EDF, Least Laxity First is More Difficult to Implement Since Execution Time Remaining for All Services Must Be Estimated to Re-assign Priorities
  - ***Overload Failure Modes Better, But Still Non-Deterministic*** - In an Overload Scenario, LLF May Behave More Favorably Since it Encodes the Concept of Dispatching the Service with the Most Pressing Need for the CPU Resource (I.e. Remaining Execution Time is Factored In)
    - Over-run Invalidates Estimates of Execution Time Remaining
    - Suffers Same Potential Cascading Priority Amplification and Interference on Over-runs
    - Over-run Termination Policy May Help
  - ***Difficult to Debug Dynamic Priority When Deadlines Are Missed***

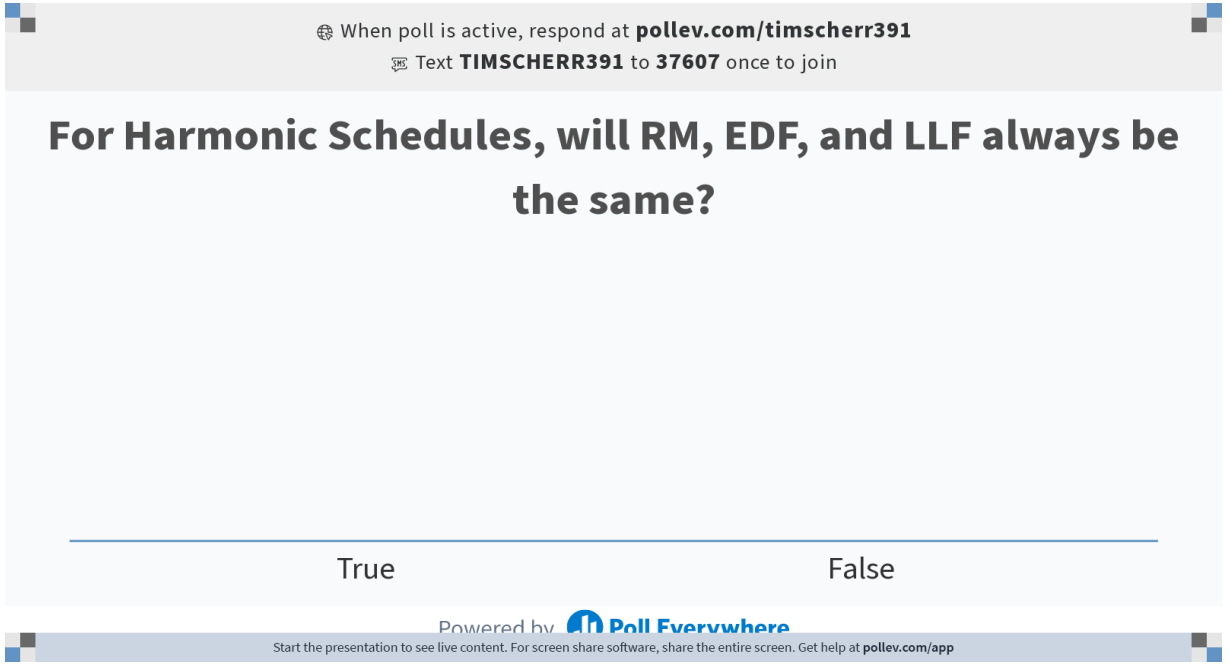
EDF and LLF

- Generally Higher Overhead than Fixed Priority
  - EDF – Update priorities based on TTD (Time-To-Deadline) on every Ready Queue change
  - LLF – Update priorities based on Laxity (TTD – Time-to-Complete)
  - Encodes Urgency, so Distinct Advantage
  - Failure Modes on Over-runs harder to DEBUG (less deterministic)
  - Can Schedule More Cases
  - Harmonic Cases, U=100% for all 3
  - RM and EDF Most Common
  - LLF – Hard to Implement



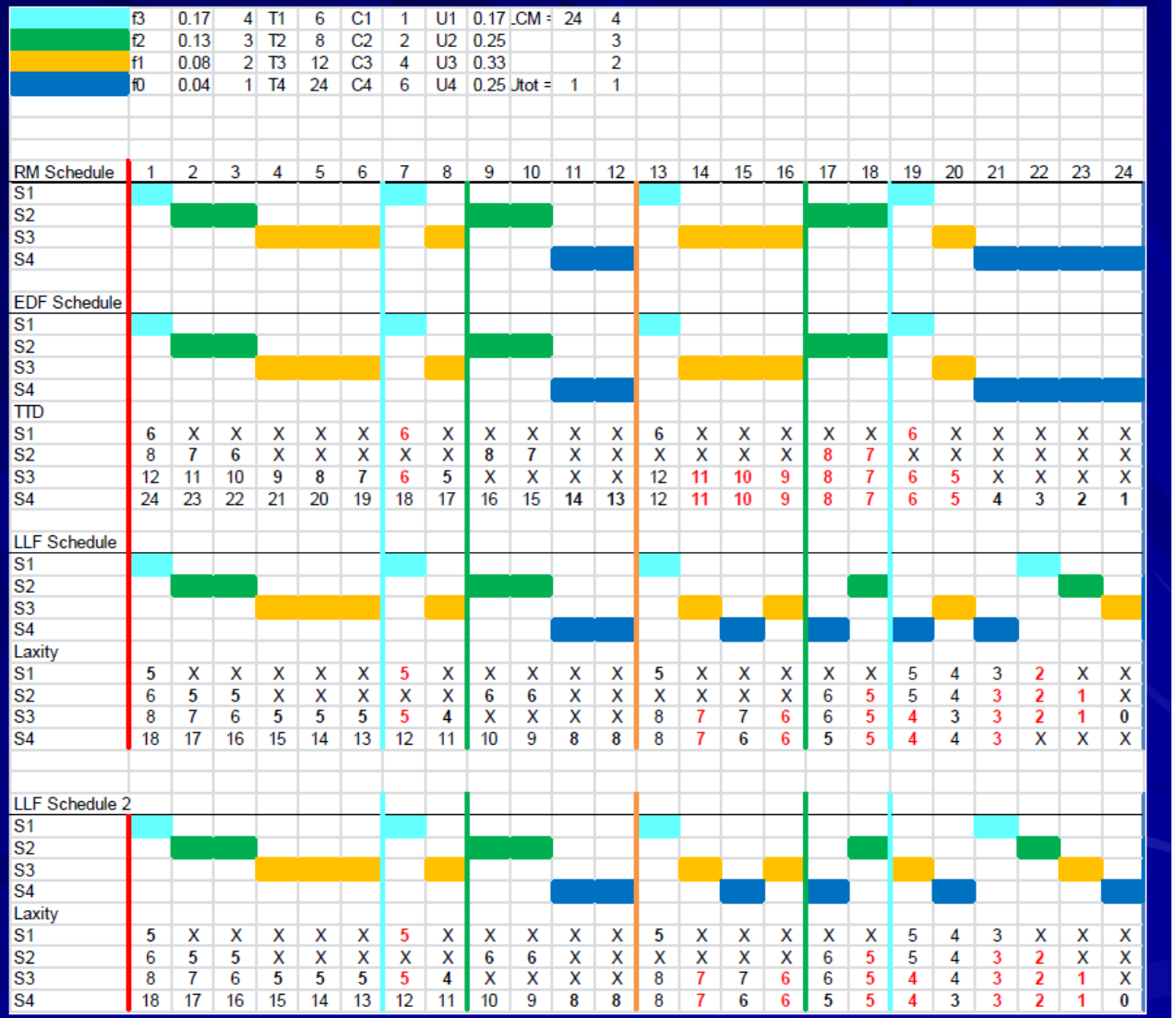
For Harmonic Schedules, will RM, EDF, and LLF always be the same?

A. Yes
B. No


When poll is active, respond at pollev.com/timscherr391
Text TIMSCHERR391 to 37607 once to join
For Harmonic Schedules, will RM, EDF, and LLF always be the same?
True
False
Powered by Poll Everywhere
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

Comparison

- EDF and LLF Can have Priority TIES
- Therefore, More than ONE valid schedule
- RM Used often for HRT
- EDF for SRT



f3	0.17	4	T1	6	C1	1	U1	0.17	CM =	24	4	
f2	0.13	3	T2	8	C2	2	U2	0.25			3	
f1	0.08	2	T3	12	C3	4	U3	0.33			2	
f0	0.04	1	T4	24	C4	6	U4	0.25	Jtot =	1	1	

RM Schedule	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
S1																								
S2																								
S3																								
S4																								
EDF Schedule																								
S1																								
S2																								
S3																								
S4																								
TTD																								
S1	6	X	X	X	X	X	6	X	X	X	X	X	6	X	X	X	X	X	6	X	X	X	X	X
S2	8	7	6	X	X	X	X	X	8	7	X	X	X	X	X	X	8	7	X	X	X	X	X	X
S3	12	11	10	9	8	7	6	5	X	X	X	X	12	11	10	9	8	7	6	5	X	X	X	X
S4	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
LLF Schedule																								
S1																								
S2																								
S3																								
S4																								
Laxity																								
S1	5	X	X	X	X	X	5	X	X	X	X	X	5	X	X	X	X	X	5	4	3	2	X	X
S2	6	5	5	X	X	X	X	X	6	6	X	X	X	X	X	X	6	5	5	4	3	2	1	X
S3	8	7	6	5	5	5	5	4	X	X	X	X	8	7	7	6	6	5	4	3	3	2	1	0
S4	18	17	16	15	14	13	12	11	10	9	8	8	8	7	6	6	5	5	4	4	3	X	X	X
LLF Schedule 2																								
S1																								
S2																								
S3																								
S4																								
Laxity																								
S1	5	X	X	X	X	X	5	X	X	X	X	X	5	X	X	X	X	X	5	4	3	X	X	X
S2	6	5	5	X	X	X	X	X	6	6	X	X	X	X	X	X	6	5	5	4	3	2	X	X
S3	8	7	6	5	5	5	5	4	X	X	X	X	8	7	7	6	6	5	4	4	3	2	1	X
S4	18	17	16	15	14	13	12	11	10	9	8	8	8	7	6	6	5	5	4	3	3	2	1	0

RM vs EDF

- Read Rate Monotonic vs EDF: Judgement Day by Giorgio Buttazzo
  - Implementation Complexity
  - Runtime Overhead
  - Feasibility Analysis
  - Permanent and Transient Overload Response
  - Response Jitter and Latency
  - RT Resource Sharing Issues
  - Aperiodic Task Handling