Real-Time Systems

Lecture Topic - Rate Monotonic Theory Advantages and Pitfalls

Dr. Sam Siewert

Electrical, Computer and Energy Engineering

Embedded Systems Engineering Program

University of Colorado Boulder

Real-Time Systems Practitioner

- By completion of RT-Systems course, you should be a beginning RT practitioner
- IEEE RTSS, RTAS, RTCSA, IEEE Computer Society TCRTS
- Linux Foundation - RT Linux, Zephyr,
- Standards - ARINC 653, RTCA DO-178 B & C

A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems (Electronic Materials: Science & Technology) 1993rd Edition

by Mark Klein (Author), Thomas Ralya (Author), Bill Pollak (Author), Ray Obenza (Author), Michael González Harbour (Author)

1 customer review

Look inside

Kindle	Hardcover	Paperback	Digital	Other Sellers
$73.22 - $257.33	$24.99 - $273.30	$378.90	$45.40	See all 5 versions

Buy used — prime $24.99

Buy new — prime $273.30

FREE Delivery by Friday
If you order within 10 hrs 3 mins or
Get it Tomorrow if you order within 3 hrs 33 mins and choose paid One-Day Shipping at checkout Details
Only 1 left in stock (more on the way).
Ships from and sold by Amazon.com. Gift-wrap available.
prime

List Price: ~~$399.99~~
Save: $126.69 (32%)
14 New from $66.80
Deliver to Sam - Prescott Valley 86315
Qty: 1

Add to Cart

Buy Now

Turn on 1-Click ordering

RMA is the state of Practice

1. Technical Society and Committee Support
2. Industry
3. Standards
4. Linux Foundation
5. Software Engineering Institute
6. Publications, Textbook, and online education and training

Major Issue Summary – Assumptions and Constraints

1. Inexact RM LUB – Solved with Worst Case Analysis
2. T=D Limitation – Solved by DM and by RM Practitioner's Guide
3. Fails in cases where Dynamic Priorities succeed – Trade-off for simpler failure mode and scheduler implementation
4. Does not include Secondary, Tertiary, … etc. resource issues – Mostly solved by priority inheritance or Priority ceiling (Course #2)
5. Does not encode importance of services – solved to a degree by period transform
6. Supported best by Cyclic Executive or RTOS rather than general OS – POSIX real-time extensions
7. Lack of broad understanding – courses like this, textbooks, Practitioner's guide, Course #2 full derivation of RM LUB
8. WCET variability – ARM R-Series Microprocessors
9. Accuracy, precision and deterministic timing limitations – Use a co-processor (FPGA, ASIC, GP-GPU)
10. Does not scale well – AMP with multi-core SoCs, and with SMP and VMs in research

RM Necessary and Sufficient Feasibility

- RM LUB is Only Sufficient (Order N)
- DM is Only Sufficient (Order N^2)
- N&S Feasibility Tests (Order N^3)
  - Lehoczky, Shah, Ding Theorem
  - Worst-Case Analysis
    - If S_i Can be Schedule over Longest T_i => Feasible
    - Best To Look over LCM of All T_i (Full Story)
  - N&S Test: Iterate Over Subsets of S_i for Each Longest T_i for All S_i
    - Encodes Process of Hand Drawing Timing
    - With Preemption According to RM Policy
    - Scheduling Point (N&S) – Automates WC Analysis (hand)
    - Completion Test (N&S) – Automates WC Analysis
- RM LUB Sufficient Test Will Never Incorrectly Pass an Infeasible Service Set
- But Will Fail Some Feasible Ones
- An N&S Test is Exact – Will Not Pass Infeasible Sets Nor Will it Fail a Feasible Set

RM LUB is Sufficient
- if scenario passes, it is feasible,
- but if scenario fails LUB, it might still be feasible

RM Worst-Case Analysis is EXACT (Necessary and Sufficient)

Truth table

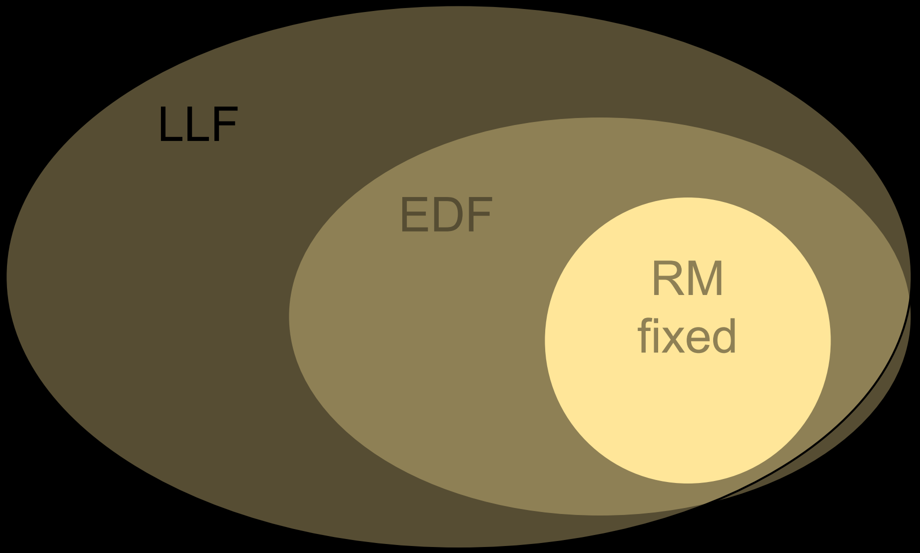
S	N	$S \Rightarrow N$	$S \Leftarrow N$	$S \Leftrightarrow N$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

The assertion that a statement is a "necessary *and* sufficient" condition of another means that the former statement is true if and only if the latter is true. That is, the two statements must be either simultaneously true or simultaneously false.[2][3][4] -Wikipedia

RM compared to Dynamic Priority 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)
- Both Encode Urgency, so they have Distinct Advantage
- Failure Mode on Over-runs harder to DEBUG (less deterministic)
- Can Schedule More Cases (especially EDF compared to RM)
- Harmonic Cases, U=100% for all 3
- RM and EDF Most Common
- LLF – Hard to Implement
- LLF and EDF often Same



Deadline Driven Scheduling (Liu & Layland) is EDF

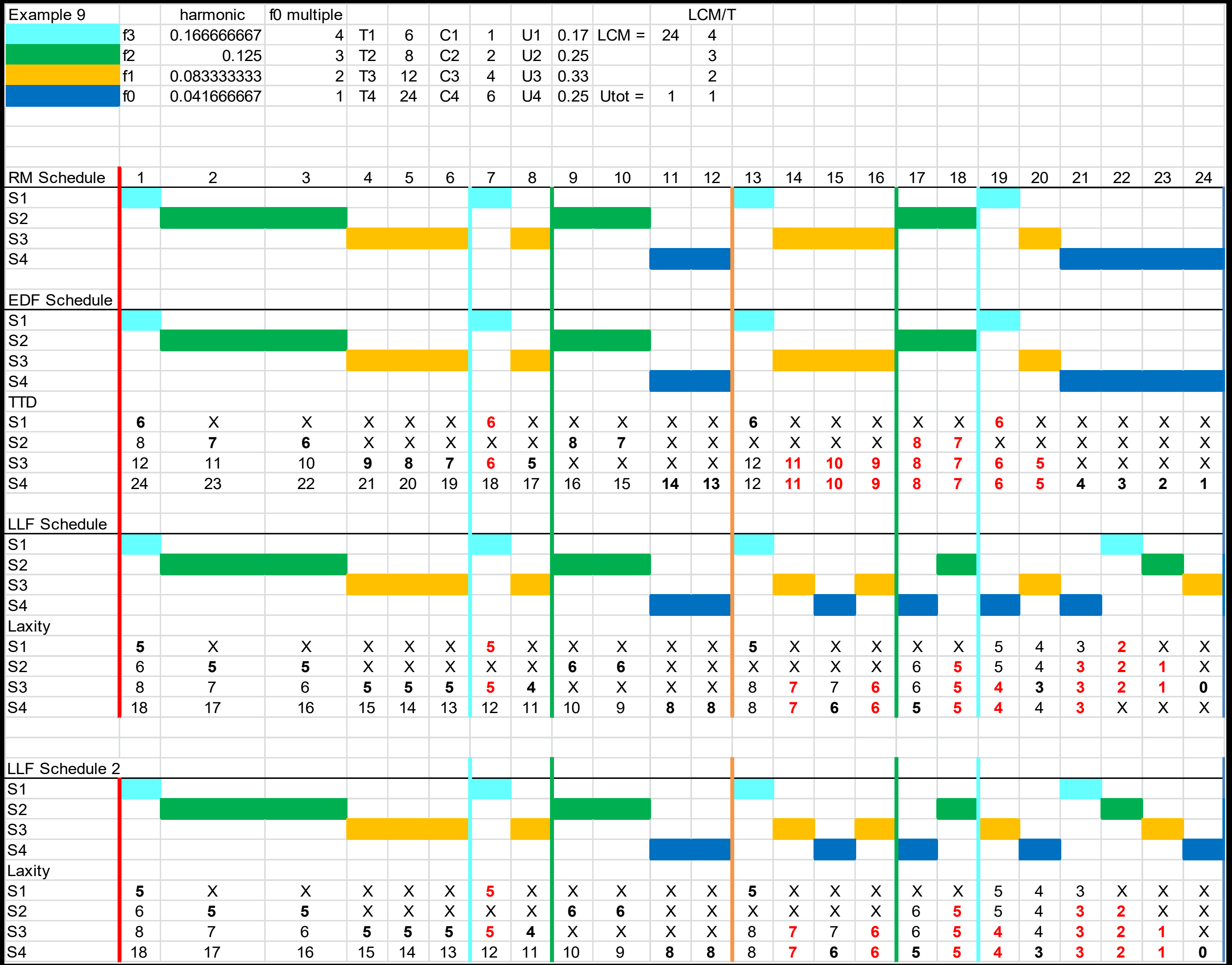
1) Course #2 in RTES series covers EDF and LLF
2) RMA and Fixed Priority are a standard of practice
3) Open area of debate and research for HRT
4) Accepted for SRT

Buttazzo, Giorgio C. "Rate monotonic vs. EDF: judgment day." *Real-Time Systems* 29.1 (2005): 5-26.

Brun, Adrien, Chunhui Guo, and Shangping Ren. "A Note on the EDF Preemption Behavior in “Rate Monotonic Versus EDF: Judgment Day”." *IEEE Embedded Systems Letters* 7.3 (2015): 89-91.

Timing Diagrams - sched-example-9

- EDF and LLF have Priority TIES
- For EDF/LLF, More than ONE valid schedule (Course #2)
- RM most often used for HRT
- EDF for SRT
- RM failure mode is simpler
  - Highest frequency succeeds
  - Lowest frequency fails
  - Services higher priority than over-run unaffected



Example 9		harmonic	f0 multiple									LCM/T
	f3	0.166666667	4	T1	6	C1	1	U1	0.17	LCM =	24	4
	f2	0.125	3	T2	8	C2	2	U2	0.25			3
	f1	0.083333333	2	T3	12	C3	4	U3	0.33			2
	f0	0.041666667	1	T4	24	C4	6	U4	0.25	Utot =	1	1

RM Schedule / EDF Schedule

TTD	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	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

Laxity	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	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

Laxity	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	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

Application of RM Theory

Policy to Assign Priorities in RTOS or SCHED_FIFO for Service Set

Feasibility Test (On-Line, Off-Line, By Mode)

- Off-Line for Fixed Set of Services
  - Transition from Off-Line Boot to On-Line RT Services
  - Service Set Can't Change – No Dynamic Admission
- On-Line Admission of Services
  - Feasibility Test Must be Run On-Line to Re-Test Feasibility of New Services Before They Go On-Line
  - Feasibility Test Itself is a Service that Must Be Admitted by Off-Line Test

Most RT Systems Have Static Services Sets

Or, Modes with Different Service Sets

- E.g. Space Telescope in Observing Mode, Ready Mode, Safe Mode
- Can Apply RM Feasibility Testing to Each Mode

RM Theory Pessimism

Pessimistic Assumptions Require Resource Margin for Safety

C = WCET

- Longest Path for Code
- Path is Executed Least Efficiently
- WCET Is Worse than 3-Sigma Deviation (Rarely Happens)

T = Worst Case Inter-arrival Rate (Period Transform Required)

Importance Encoding Requires Period Transform

Critical Instant - Assumes that Service Requests May All Come At Once

RM LUB not EXACT

- RM LUB is Based Upon Service Periods That Are Not Necessarily Harmonic, So Partial Interference can be Over-Accounted for
- Feasible Service Sets May Be Rejected, Although Unsafe Will Never Be Admitted
- Can Use Alternative Scheduling Point or Completion Test RM N&S Feasibility Tests

Pitfalls Applying RM Theory

Audsley, Neil, and Alan Burns. "Real-time system scheduling." (1990).

Klein, Mark, et al. A practitioner's handbook for real-time analysis: guide to rate monotonic analysis for real-time systems. Springer Science & Business Media, 2012.

Problem: Period is not Constant

- Service Releases have Period Jitter, are Sporadic, or A-periodic
- Possible Solution: Period Transform
  - Assume Service Period Based on Worst Case Inter-arrival Rate (Highest Known Frequency of Requests)
  - When Frequency is Lower than Worst Case, Margin Available for Slack Stealers
  - May not Handle Truly A-periodic Services (Worst Case Frequency Unknown)

Problem: C is not Deterministic

- Service Execution has Jitter due to Processor Architecture and Data Driven Algorithms
- Possible Solution: WCET
  - Assume Worst Case Execution Time
  - Longest Path Possible for Algorithms
  - Worst Efficiency Possible Executing that Path (e.g. all cache accesses miss)

Pitfalls Applying RM Theory

Problem: RM Policy Does Not Encode Importance

- The Longest Period Service May not Be the Service We Want to Fail in an Overload Situation
- ***Possible Solution***: Period Transform
  - Assume High Importance Service with Long Period has a Shorter Period than it Actually Does to Artificially Increase Priority
    - Increases System CPU Resource Margin
    - Overload Failures Result in Less Important Services Missing Deadlines

Problem: Deadline Not Equal Period (Longer – overlapping duplicates, Shorter – output time)

- Deadline may be Less than Period when Response I/O Latency is Accounted for (e.g. if Service Response is Transported on a Network)
- Deadline may be Greater than Period if Service Release Processing is Allowed to Overlap (e.g. 2 or more service requests within T)
- ***Possible Solution***: Deadline Monotonic Policy and Feasibility Test

Deadline Monotonic Theory

DM Policy: Higher Priority Given to Services with Shorter Deadlines

DM Sufficient Feasibility:

$$\forall i : 1 \leq i \leq n : \frac{C_i}{D_i} + \frac{I_i}{D_i} \leq 1.0$$

For All Services from 1 to n, if the Deadline Interval is Long Enough To Contain The Service Execution Time Interval Plus All Interfering Execution Time Intervals, Then the Service is Feasible - If all Services are Feasible, Then the System is Feasible.

$$I_i = \sum_{j=1}^{i-1} \left\lceil \frac{D_i}{T_j} \right\rceil C_j$$

$$\left\lceil \frac{D_i}{T_j} \right\rceil$$

Worst Case Number of Releases of S_j Over Deadline Interval for S_i

Interference to Service S_i is due to Preemption By All Higher Priority Services S_1 to S_{i-1} and the total Interference Time is the Number of Releases of S_j Over The Deadline Interval D_i for S_i Multiplied by S_j Execution Time C_j Summed for All S_j which Have Greater Priority Than S_i.

DM Improved Feasibility Test – Still Not EXACT

DM Sufficient Feasibility Pessimistic

- Assumes that All Releases of S_j Over D_i Fully Interfere with S_i
- Last Release of S_j Over D_i May Only Partially Interfere with S_i

Improved Interference Estimation:

$$I_i = \sum_{j=1}^{i-1}\left[\left[\left\lfloor \frac{D_i - D_j}{T_j}\right\rfloor + 1\right] C_j + \left[\left\lceil \frac{D_i}{T_j}\right\rceil - \left[\left\lfloor \frac{D_i - D_j}{T_j}\right\rfloor + 1\right]\right] \times Min\left[C_j, D_i - \left\lfloor \frac{D_i}{T_j}\right\rfloor T_j\right]\right]$$

Full Interference Time

=0 if No Partial Interference
OR =1 if Partial Interference

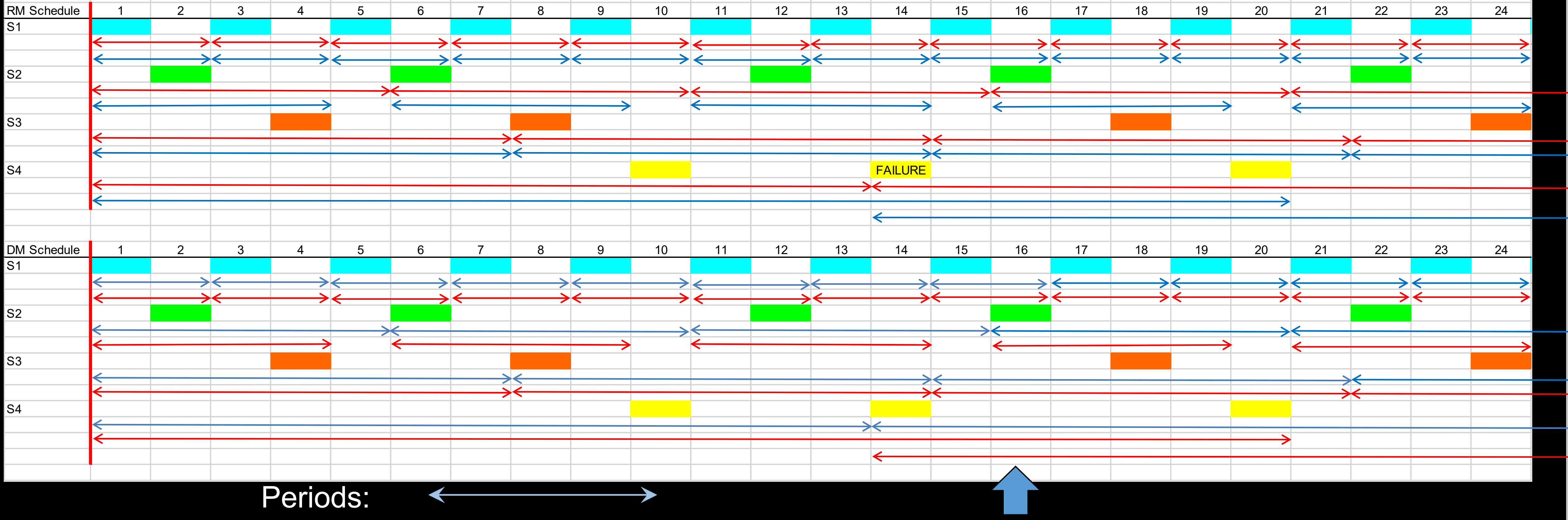
Minimum of S_j Execution Time
OR S_j Partial Execution Time

Still Not N&S Feasibility Test

- Over-Accounts for Partial Interferences (They Don't All Just Sum)
- Some Partial Interferences Over-lap in Time and Preempt Each Other

Example #6 - DM Ok, RM Fails, but DM is RM

Example 6	Service	Period		Freq f	f0 multiple	WCET		RM Utility								DM Utility			
	S1	T1	2	0.5	6.5	C1	1	U1	50.00%	**LCM =**	**70**	For DM	D1	2		U1	50.00%		
	S2	T2	5	0.2	2.6	C2	1	U2	20.00%				D2	4	EARLIER	U2	25.00%		
	S3	T3	7	0.142857	1.857143	C3	1	U3	14.29%	**LUB =**	**75.68%**		D3	7		U3	14.29%		
	S4	T4	13	0.076923	1	C4	2	U4	15.38%	**Utot =**	**99.67%**		D4	20	LATER	U4	10.00%	**Utot =**	**99.29%**



Periods:

Deadlines:

Note: overlap of Deadlines for S4

Rate Monotonic or EDF?

Liu & Layland Rate Monotonic - Fixed Priority Policy

- Feasibility Test - Sufficient RM LUB
- Most Hard RT Systems Have Static Services Sets

$$U = \sum_{i=1}^{m} (Ci / Ti) \le m(2^{\frac{1}{m}} - 1), U \lim m \to \infty = \ln(2) \approx 0.69$$

EDF (Deadline Driven) – Dynamic Priority Policy

- Feasibility Test Based on Hyper-Period (Product of All Periods)

$$\forall tasks \in 1...m, T_{hyperperiod} = 0...(T_1 T_2 ... T_{m-1} T_m)$$

$$(T_1 T_2 ... T_{m-1} T_m) = \frac{(T_1 T_2 ... T_{m-1} T_m)}{T_1} C_1 + ... + \frac{(T_1 T_2 ... T_{m-1} T_m)}{T_m} C_m$$

$$\therefore \sum_{i=1}^{m} (Ci / Ti) \le 1$$

RMA LUB for quick analysis

RM Worst Case analysis for EXACT feasibility

EDF has feasibility tests, but can be inefficient at runtime

Sáez, Sergio, Joan Vila, and Alfons Crespo. "Using exact feasibility tests for allocating real-time tasks in multiprocessor systems." *Proceeding. 10th EUROMICRO Workshop on Real-Time Systems (Cat. No. 98EX168)*. IEEE, 1998.

Albers, Karsten, and Frank Slomka. "Efficient feasibility analysis for real-time systems with EDF scheduling." *Design, Automation and Test in Europe*. IEEE, 2005.

Singh, Jagbeer. "An algorithm to reduce the time complexity of earliest deadline first scheduling algorithm in real-time system." *arXiv preprint arXiv:1101.0056* (2010).

Notes on Dynamic Admission

- Off-Line for Fixed Set of Services
  - Transition from Off-Line to On-Line RT Services
  - Service Set Can't Change – No Dynamic Admission

- On-Line Admission of Services
  - Feasibility Test Must be Run On-Line to Re-Test Feasibility of New Services Before They Go On-Line
  - Feasibility Test Itself is a Service that Must Be Admitted by Off-Line Test or Run on Dedicated Hardware

SMP Load-balancing – Migration of services essentially requires dynamic admission