Embedded reconfiguration of TSN: Dual reconfiguration with dropping and reclaiming

Álex Gracia, Alitzel G. Torres-Macías, *Juan Segarra*, José L. Briz, Antonio Ramírez-Treviño, Héctor Blanco-Alcaine

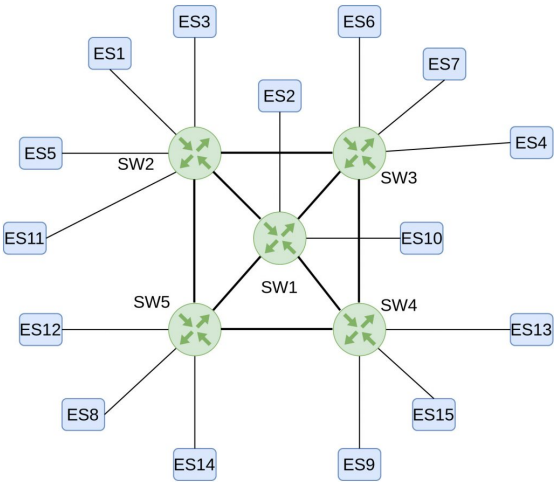
Universidad Zaragoza 1542 · Cinvestav · intel

37th Euromicro Conference on Real-Time Systems
Brussels, July 8-11, 2025

GOBIERNO DE ESPAÑA · MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES · AGENCIA ESTATAL DE INVESTIGACIÓN

Outline

1 Introduction



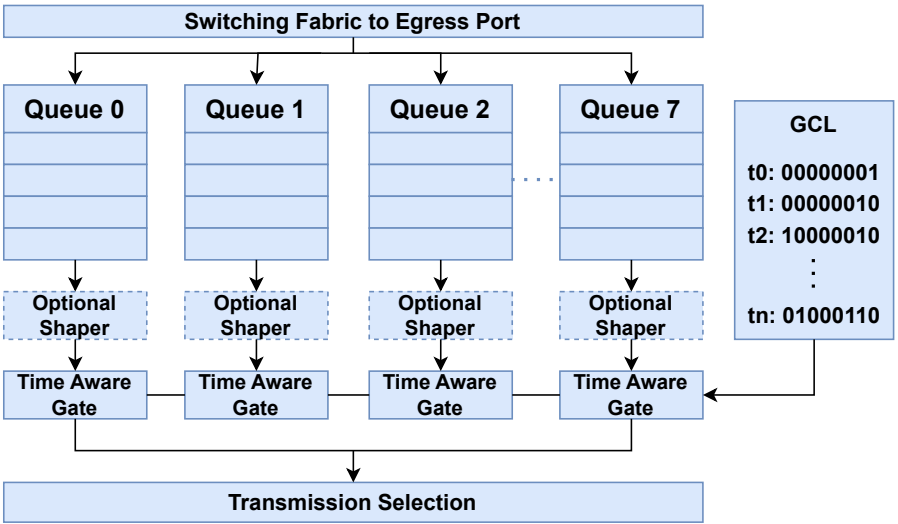
Challenge:

- 241 mixed criticality streams
- 8 potentially failing links
- Reroute and reschedule upon link failures
  - NP-complete problem

Tested solution:

- Mixed Integer Linear Programming models for scheduling
- Incremental + global reconfiguration
- Time Aware Shaper for all streams

1.1 Time Aware Shaper



2 Proposed procedure

1. Detect faulty link

2 Proposed procedure

1. Detect faulty link
2. Get ordered sublist of disposable (affected) schedules and purge them
   - ➤ DROP policy until reschedule

2 Proposed procedure

1. Detect faulty link
2. Get ordered sublist of disposable (affected) schedules and purge them
   - DROP policy until reschedule
3. Incremental stage: schedule and deploy stream-by-stream
   - Schedule streams one-by-one on an existing schedule
   - Faster, one-by-one optimal (no heuristics)

[1] A. Galilea Torres-Macías et al. Optimal and Fast IEEE802.1Qbv Incremental Scheduling. Submitted to Computer Networks.

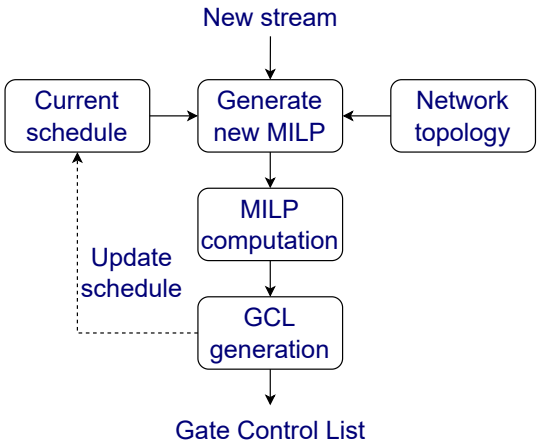
2 Proposed procedure

1. Detect faulty link
2. Get ordered sublist of disposable (affected) schedules and purge them
   - DROP policy until reschedule
3. Incremental stage: schedule and deploy stream-by-stream
   - Schedule streams one-by-one on an existing schedule
   - Faster, one-by-one optimal (no heuristics)

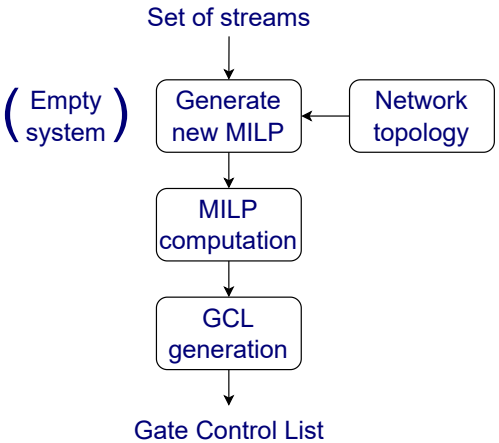
   [1] A. Galilea Torres-Macías et al. Optimal and Fast IEEE802.1Qbv Incremental Scheduling. Submitted to Computer Networks.
4. Global stage, if needed (not needed for the use case)
   - Schedule several streams together on an empty system
   - Slower, better schedulability, globaly optimized, with heuristics

   [2] A. Galilea Torres-Macías et al. Fast IEEE802.1Qbv gate scheduling through integer linear programming. IEEE Access, 12:111239–111250, 2024.

2.1 Scheduling by MILP models



(a) Incremental scheduling [1]



(b) Global scheduling [2]

2.2 Incremental stage

Process streams one by one:

1: **for all** $stream$ in disposableStreamList **do**
2: $pathList \leftarrow$ Get ordered list of possible paths to reschedule $stream$
3: **for all** $path$ in $pathList$, shortest first **do** // Single iteration suffices!
4: Generate MILP model [1] to schedule $stream$ on $path$
5: Solve model
6: **if** schedulable **then**
7: Deploy schedule for $stream$
8: Exit loop on pathList

2.3 Global stage

Process set of streams together; maximize its size with binary search:

1: **while** not satisfied with the schedule **do**
2: Generate MILP model [2] to schedule a $set\ of\ streams$ (shortest paths)
3: Solve model
4: **if** schedulable **then**
5: Purge system and Deploy
6: **if** scheduled $set\ of\ streams$ contains all streams **then**
7: The system is optimized; exit
8: **else**
9: Add streams to the $set\ of\ streams$
10: **else**
11: Remove streams from the $set\ of\ streams$
12: **if** the $set\ of\ streams$ has already been tested **then**
13: The system cannot be further improved; exit

3 Experiments

- Intel Xeon Gold 5120 CPU: 28 cores (56 total threads), released on 2017
- Frame isolation supported but not addressed (TC7 streams isolated)
- 1 ms frame processing time; 0.5 ms clock skew

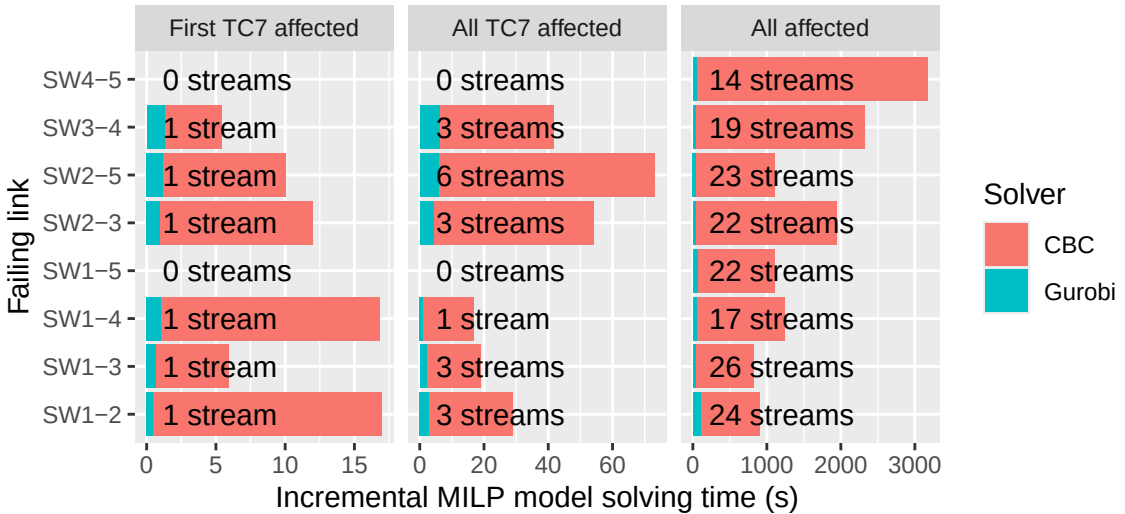
Experiment 1. Network dimmensioning (consecutive link failures)

- ✓ Adequate: all reschedulable with 3 failing links

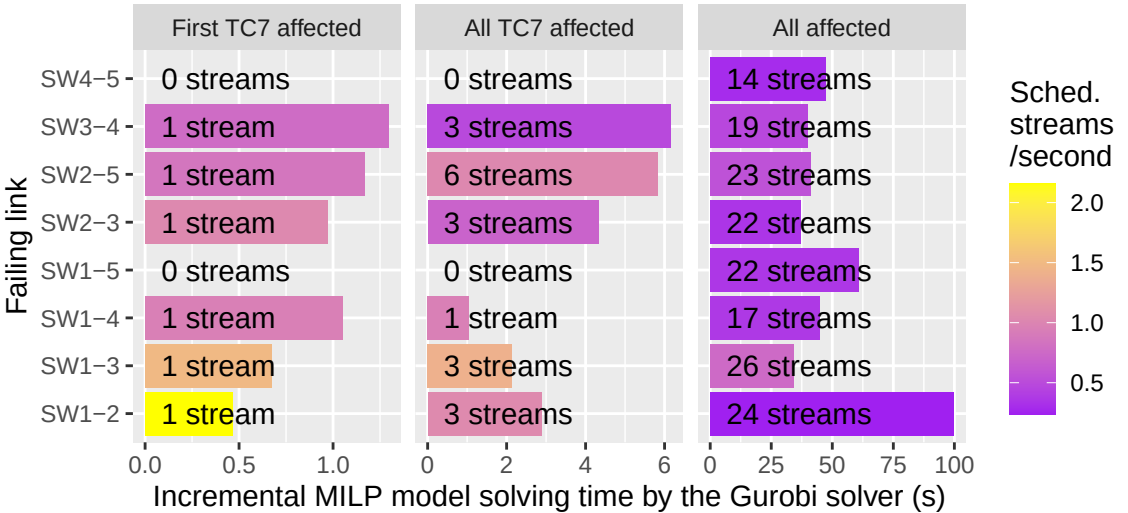
Experiment 2. Rescheduling times (single link failure)

- ✗ Remove streams affected by link failure and lower priority working ones
  - Rescheduling working schedules is not worth
  - ✓ Reschedule streams affected by link failure only
- ✗ Get valid schedules (not necessarily optimal)
  - Faster for the first streams, but globaly slower
  - ✓ Get optimal schedules always

3.1 Rescheduling times (single link failure)



3.1 Rescheduling times (single link failure) (II)



4 Conclusions and future work

For the considered use case:

- Recovery time (Gurobi): from 0.5 s (TC7) to 1-2 min. (all streams)
- Incremental (one-by-one) rescheduling [1] suffices
- Hints for rescheduling:
  - Rescheduling working low-criticality streams is globally slower
  - Computing optimal schedules is globaly faster than just valid ones

Work in progress:

- Speed-up solving times through heuristics
- Fault tolerance by CNC-managed frame replication
- Alternative implementations of the Gate Control List of TAS

Embedded reconfiguration of TSN: Dual reconfiguration with dropping and reclaiming

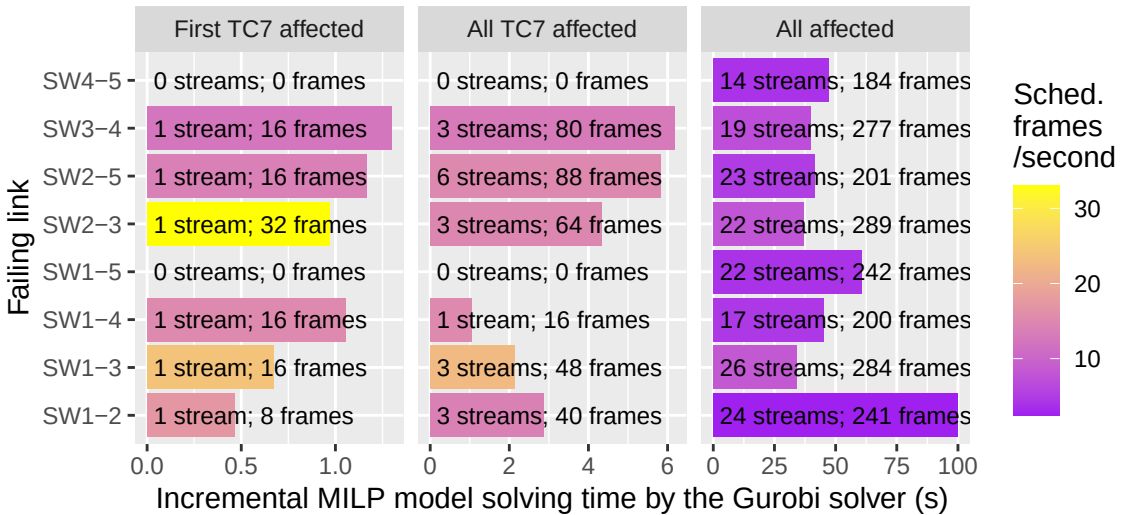
Álex Gracia, Alitzel G. Torres-Macías, *Juan Segarra*, José L. Briz, Antonio Ramírez-Treviño, Héctor Blanco-Alcaine

Universidad Zaragoza 1542 · Cinvestav · intel

37th Euromicro Conference on Real-Time Systems
Brussels, July 8-11, 2025

GOBIERNO DE ESPAÑA · MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES · AGENCIA ESTATAL DE INVESTIGACIÓN

4 Rescheduling times (single link failure)



4 Mixed Integer Linear Programming

- Set of linear (in)equations: $ax + by + \ldots \leq cz$
- Only linear operations ($+/-$) between variables
- Constants may multiply variables
- Integer/binary variables (hard) mixed with real variables
- Objective function to minimize: min: $x + y$
- E.g. $sendInstantBridge1 + delay = receiveInstantBridge2$

5 CBC and script times (paper)

Faulty link	Scheduled streams (number)		Optimal solution	
			MILP	Scripts
SW1-SW2	First TC7 affected	(1)	16.5	9.8
SW1-SW2	All TC7 affected	(3)	26.1	50.3
SW1-SW2	All affected	(24)	808.8	668.3
SW2-SW3	First TC7 affected	(1)	11.0	164.6
SW2-SW3	All TC7 affected	(3)	49.9	429.4
SW2-SW3	All affected	(22)	1914.3	1188.3
SW2-SW5	First TC7 affected	(1)	8.9	59.2
SW2-SW5	All TC7 affected	(6)	67.3	307.6
SW2-SW5	All affected	(23)	1058.4	971.2
SW3-SW4	First TC7 affected	(1)	4.1	20.8
SW3-SW4	All TC7 affected	(3)	35.6	864.4
SW3-SW4	All affected	(19)	2283.3	1951.3

5 Frame isolation

- Both our papers support frame isolation by changing the traffic class of each stream after scheduling the system
- In this use case, it is not possible to isolate all streams
- Our tests indicate that all TC7 streams are isolated
- Isolating a subset of streams in this way is straightforward, but not addressed