A Hot-Patching Protocol for Repairing Time-Triggered Network Schedules

Francisco Pozo
April 11, 2018, RTAS 2018 Porto, Portugal

Mälardalen University
Time-Triggered Paradigm

- Offline Schedule
- Nodes follow the schedule
- Shared notion of time
Time-Triggered Paradigm Example

- Frame 1 and 2:
  - From N1 to N6
  - Path: {1-7-11}
  - Period: 8

- Frame 3 and 4:
  - From N2 to N6
  - Path: {3-9-11}
  - Period: 8
Time-Triggered Communication Disadvantages

- **Not flexible:**
  - If something changes during run-time, we must change the whole schedule
  - Vulnerable to failures
We want to enhance the reliability with small cost increase repairing the schedule during run-time
Hot-Patching Protocol

- **React** to Link failures
- **Patch** a small part of the schedule
- **As fast as** possible
- **Limit the number** of lost frames
Hot-Patching Protocol

4 Steps:

• Notification
• Membership
• Solving
• Update

Network

Time-Triggered Schedule
Hot-Patching Protocol

4 Steps:

- Notification
- Membership
- Solving
- Update

Network

Time-Triggered Schedule
Schedule Reparability

How do we create schedules that are easy to repair?
Schedule Reparability

- Choose a high reparability initial schedule
- Low cost to transform to other valid schedules
Schedule Reparability

- **Maximize frame distances:**
  1. Between the same frame
  2. Between frames in links

Minimize Make-span

Maximize frame distances
Hot-Patching Protocol

- Minimize the cost transformation
- Allows for fast patching

Network

Time-Triggered Schedule
A Hot-Patching Protocol for Repairing Time-Triggered Network Schedules

Francisco Pozo
April 11, 2018, RTAS 2018 Porto, Portugal

Mälardalen University