Interference-Aware Real-Time Flow Scheduling for Wireless Sensor Networks

Octav Chipara, Chengjie Wu, Chenyang Lu, William Griswold

University of California, San Diego
Washington University in St. Louis
Motivation

• Challenges in real-time wireless sensor networks
  • **predictable:** real-time + reliability
  • **dynamic environments:** interference, unreliable links
  • **scalability:** support large networks

• **RFS: real-time flow scheduling**
RFS vs. WirelessHART

• Wireless interference
  • prevent interference [WirelessHART]
  • interference aware [RFS]

• Wireless channels
  • single channel [RFS]
  • multiple channels [WirelessHART]

• Scalability
  • centralized [WirelessHART]
  • two-level [RFS]
Real-time flows

- Established between arbitrary end-points
- Flow instances are released periodically
- Timing properties (period, deadline, phase)
- Precedence constraints due to hop-by-hop forwarding
- Links are unreliable => retransmissions
Our approach

• Planner: construction of plan when route is established
  • all instances of a flow are executed according to the same plan
  • accounts for **precedence constraints** + **unreliable links**

• Scheduler: run-time execution of concurrent flows
  • dynamically schedules multiple concurrent flows
  • handles **interference** + **prioritization**

• **This setup allows for a clean separation of concerns!**
Plans

- **Link estimator** provides retransmission needed for reliability
- **Plans**: constructed when routes are established
  - precedence constraints enforced
  - links transmit sufficient times to deliver packets with high likelihood

**Plan**: (AB) (BC) (BC) (CD) (DE) (EF)
Generic interference model

- Different approaches to modeling and estimating interference.
  - graph
  - Signal to Noise plus Interference Ratio (SNIR)

- Decouple real-time scheduling and specific interference models
  - scheduler works with interference estimator via a generic interface

- Generic model
  - conflict between two transmissions => cannot occur simultaneously
  - mapping from graph and SNIR threshold models
  - new interference models and estimators may be easily incorporated
Scheduler

• consider flows in decreasing order of priority
• execute the next transmission of a flow if it does not conflict with the transmissions of higher priority flows scheduled in the current slot

Spatial reuse! Prioritization!

<table>
<thead>
<tr>
<th>Slot</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>F₁</td>
<td></td>
<td></td>
<td>AB</td>
<td>BC</td>
<td>BC</td>
<td>CD</td>
<td>DE</td>
<td>EF</td>
<td></td>
</tr>
<tr>
<td>F₂</td>
<td>FE</td>
<td>ED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DG</td>
<td>GH</td>
<td></td>
</tr>
</tbody>
</table>
Schedulability analysis

- Response-time analysis for flows with arbitrary interference
  - static priorities
  - deadlines ≤ periods

- Response time of flow \( l \) is:

\[
R_l = L_l + \sum_{h \in h_p(l)} \left\lfloor \frac{R_l}{P_h} \right\rfloor \cdot I(l, h)
\]

Length of plan

Inter-flow interference

Key challenge: assess the interference between pairs of flows
Conflict matrices

- Captures the **sequential execution** of plans \(\rightarrow\) ordering of rows/columns
- Captures **links unreliability** \(\rightarrow\) duplicate rows/columns
- Captures **conflicts** \(\rightarrow\) “x”-es in the matrix

![Diagram](image)

<table>
<thead>
<tr>
<th></th>
<th>AB</th>
<th>BC</th>
<th>BC</th>
<th>CD</th>
<th>DE</th>
<th>EF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ED</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DG</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GH</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Mapping executions onto conflict matrices

### F₁ exec, F₂ exec

\[ I(2,1) = I(2,1) + 0 \]

### F₁ exec, F₂ suspended

\[ I(2,1) = I(2,1) + 1 \]

### worst-case interference

worst-case path in matrix solved through dynamic programming
Two-level scheduler

- **Divide the network into neighborhoods**
  - nodes need to synchronize their schedules within the neighborhood
  - limits the storage and communication overheads

- **Two-level scheduling:**
  - RFS within neighborhood
  - graph coloring among neighborhoods: distributed TDMA

- **Release guard** synchronizes flows across neighborhoods
  - dynamic schedulers allows for efficient implementation of phase shifts
  - real-time analysis for release guard still applies
Simulations results

- Simulator based on real traces collected from WU testbed
- Interference estimated using RID protocol
- Four flows with rates $F_0:F_1:F_2:F_3$ with ratios 1:1.5:2.2:4.3
  - connect the corners of the topology
- Baselines:
  - graph coloring (GC)
  - NCR-NF: node fair
  - NCR-WF: workload fair
Results

- Significant improvements in capacity and real-time capacity
  - account for the precedence constraints of hop-by-hop forwarding
  - provides prioritization among flows
Theoretical bound vs. Simulations

- Bounds are safe for tested configurations
- Pessimism: 23% lower real-time capacity
  - Still 63% higher than the network capacity under NCR-WF
Conclusions

• **RFS: real-time flow scheduling for wireless sensor networks**
  - provides prioritization among flows with static priorities
  - achieves scalability via two-level scheduling

• **Novel schedulability analysis for real-time flows**
  - efficiently calculates inter-flow interference based on conflict matrices

• **Significant performance improvements over TDMA solutions**
  - 2x improvements in network capacity
  - 6x improvements in real-time capacity