RT-WiFi: High-Speed Real-Time Communication Platform for Cyber-Physical Systems
RT-WiFi Design Goals

1. Real-Time Data Delivery and High Sampling Rate
   - Aim to provide at least 1 kHz sampling rate
     • minimum requirement for many mechanical control systems

2. Flexible Configuration
   - Configurable parameters: sampling rate, predictability of real-time data delivery, reliability, co-existence with regular WiFi networks

3. Transparent System Design
   - Use commercial-off-the-shelf network card
   - Transparent to upper layer protocols
Overview of a Control System using RT-WiFi Network

Network Manager

Control Application

RT-WiFi Access Point (AP)

RT-WiFi Station1
Actuator1 Sensor1

RT-WiFi Station2
Actuator2 Sensor2

RT-WiFi Station3
Actuator3 Sensor3
Architecture of RT-WiFi Protocol
Based on Timing Synchronization function (TSF) in IEEE802.11
- Synchronize all the stations in the RT-WiFi network.
- Generate timer interrupt
Architecture of RT-WiFi Protocol

- Coordinate channel access among the stations
- Link: Broadcast link, transmit link, receive link, shared link
- Superframe:

<table>
<thead>
<tr>
<th>Slot 0</th>
<th>Slot 1</th>
<th>Slot 2</th>
<th>Slot 3</th>
<th>Slot 4</th>
<th>Slot 5</th>
<th>Slot 6</th>
<th>Slot 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Shared</td>
<td>STA1</td>
<td>AP</td>
<td>STA2</td>
<td>AP</td>
<td>STA3</td>
<td>AP</td>
</tr>
<tr>
<td>Broadcast</td>
<td></td>
<td>AP</td>
<td>STA1</td>
<td>AP</td>
<td>STA2</td>
<td>AP</td>
<td>STA3</td>
</tr>
</tbody>
</table>

MAC

Message Handling Module

Link Scheduler

Superframe Table
Link Table
Device Profile

Queue

IEEE 802.11 Compatible Hardware

Physical
Architecture of RT-WiFi Protocol

Configuration parameters:
- Sampling rate
- Co-existence with regular Wi-Fi network
- Reliability (Retransmission)
Enabling Co-existence with Regular WiFi

1. Assume bounded length for regular WiFi data frames to ensure bounded latency
   - Limit maximum transmission unit to at most some upper bound
   - Limit lowest data rate to at least some lower bound

2. For RT-WiFi:
   - Enable carrier sense
   - Use shorter interframe space (IFS)
Increasing Reliability in RT-WiFi transmission

- Retransmission
  - In-slot retransmission
  - Out-of-slot retransmission

(a) Regular time slot

(b) Time slot with in-slot retransmission
RT-WiFi Testbed setup

Setup:
RT-WiFi baseline vs. regular WiFi
UDP socket program to emulate control applications

Metrics:
MAC layer to MAC layer latency
Packet loss ratio

Environment:
Interference free environment vs. office environment

<table>
<thead>
<tr>
<th>Slot 0</th>
<th>Slot 1</th>
<th>Slot 2</th>
<th>Slot 3</th>
<th>Slot 4</th>
<th>Slot 5</th>
<th>Slot 6</th>
<th>Slot 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Shared</td>
<td>STA1</td>
<td>AP</td>
<td>STA2</td>
<td>AP</td>
<td>STA3</td>
<td>AP</td>
</tr>
<tr>
<td>Broadcast</td>
<td></td>
<td>AP</td>
<td>STA1</td>
<td>AP</td>
<td>STA2</td>
<td>AP</td>
<td>STA3</td>
</tr>
</tbody>
</table>

Timeslot size: 500 μs
Results of RT-WiFi baseline (TDMA only) in Interference Free Environment

<table>
<thead>
<tr>
<th>Link</th>
<th>Max Latency(μs)</th>
<th>Mean Latency(μs)</th>
<th>Latency stdev. (μs)</th>
<th>Loss Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT-WiFi</td>
<td>Wi-Fi</td>
<td>RT-WiFi</td>
<td>Wi-Fi</td>
</tr>
<tr>
<td>STA1-&gt;AP</td>
<td>535</td>
<td>16448</td>
<td>173</td>
<td>191</td>
</tr>
<tr>
<td>STA2-&gt;AP</td>
<td>529</td>
<td>13387</td>
<td>172</td>
<td>181</td>
</tr>
<tr>
<td>STA3-&gt;AP</td>
<td>525</td>
<td>13589</td>
<td>174</td>
<td>202</td>
</tr>
<tr>
<td>AP-&gt;STA1</td>
<td>827</td>
<td>16472</td>
<td>184</td>
<td>250</td>
</tr>
<tr>
<td>AP-&gt;STA2</td>
<td>544</td>
<td>17465</td>
<td>187</td>
<td>298</td>
</tr>
<tr>
<td>AP-&gt;STA3</td>
<td>1055</td>
<td>17049</td>
<td>188</td>
<td>248</td>
</tr>
</tbody>
</table>

16-32x       47-90x
Results of RT-WiFi baseline (TDMA only) in Office Environment

<table>
<thead>
<tr>
<th>Link</th>
<th>Max Latency(μs)</th>
<th>Mean Latency(μs)</th>
<th>Latency stdev. (μs)</th>
<th>Loss Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT-WiFi</td>
<td>Wi-Fi</td>
<td>RT-WiFi</td>
<td>Wi-Fi</td>
</tr>
<tr>
<td>STA1-&gt;AP</td>
<td>3865</td>
<td>100078</td>
<td>176</td>
<td>401</td>
</tr>
<tr>
<td>STA2-&gt;AP</td>
<td>4193</td>
<td>81499</td>
<td>171</td>
<td>348</td>
</tr>
<tr>
<td>STA3-&gt;AP</td>
<td>3861</td>
<td>75298</td>
<td>174</td>
<td>429</td>
</tr>
<tr>
<td>AP-&gt;STA1</td>
<td>1197</td>
<td>78089</td>
<td>184</td>
<td>788</td>
</tr>
<tr>
<td>AP-&gt;STA2</td>
<td>1342</td>
<td>78923</td>
<td>189</td>
<td>790</td>
</tr>
<tr>
<td>AP-&gt;STA3</td>
<td>2186</td>
<td>77860</td>
<td>189</td>
<td>799</td>
</tr>
</tbody>
</table>

2.0-4.3x
36-184x
Flexible Channel Access Controller

Experiment Setup

• **Network A:**
  – Regular WiFi network
  – 10 Mbps UDP traffic generated by iperf

• **Network B:**
  – UDP program to emulate control applications
  – Compare:
    • Regular WiFi
    • RT-WiFi baseline
    • RT-WiFi with co-existence enabled
    • RT-WiFi with co-existence enabled and one in-slot retransmission enabled

• **Metrics:**
  MAC to MAC layer latency
  Packet loss ratio
## Flexible Channel Access Controller

### Experiment Results

<table>
<thead>
<tr>
<th></th>
<th>Max Latency (μs)</th>
<th>Mean Latency (μs)</th>
<th>Latency Stdev. (μs)</th>
<th>Loss Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular WiFi</td>
<td>62629</td>
<td>580</td>
<td>1679.04</td>
<td>0%</td>
</tr>
<tr>
<td>RT-WiFi baseline</td>
<td>953</td>
<td>183</td>
<td>27.75</td>
<td>50.21%</td>
</tr>
<tr>
<td>RT-WiFi co-ex</td>
<td>2507</td>
<td>220</td>
<td>149.27</td>
<td>10.92%</td>
</tr>
<tr>
<td>RT-WiFi co-ex + retry</td>
<td>2831</td>
<td>254</td>
<td>166.37</td>
<td>4.96%</td>
</tr>
</tbody>
</table>
Data Link Layer Schedule Assignment for Minimizing Communication Jitter

• TDMA communication scheduler design for wireless CPS applications.
  – Meet diverse application and system requirements
    • Application performance requirements
    • Hardware capability
  – Minimize communication jitters
    • Easier controller design
    • Better application performance
RT-WiFi Communication Task Model

• Model communication tasks in an RT-WiFi network as a periodic task set $T = \{T_i\}_{i=1}^n$

• Each task $T_i$ is defined as $T_i = (P_i^{\text{min}}, P_i^{\text{max}}, C_i)$
  
  – $P_i^{\text{min}}$: minimum sampling period supported by the task: determined by hardware and software capability
  
  – $P_i^{\text{max}}$: maximum sampling period required by the task: determined by controller requirement
  
  – $C_i$: number of time slots for data transmission

  – $P_i^{\text{min}}, P_i^{\text{max}}$ and $C_i$ are integers, representing the number of time slots
  
  – Scheduler allows preemption only at time slot boundaries.
Static Link Schedule Assignment

• Each task $T_i$ consists of an infinite sequence of jobs. Each job is a fixed number ($C_i$) of data fragments to be transmitted, with each fragment taking one time slot to transmit.

• **Jitter** is defined as the variation of jobs’ inter-completion times.

• Given a communication task set $T = \{T_i\}_{i=1}^n$ with $T_i = (p_i^{min}, p_i^{max}, C_i)$, the **Jitter-Free Scheduling (JFS) Problem** is to determine the period $P_i$ and phasing $\left\{\phi_{i,1}, ..., \phi_{i,C_i}\right\}$ for each task $T_i$, so that:
  1) $p_i^{min} \leq P_i \leq p_i^{max}$
  2) The $j$-th fragment of task $T_i$ is scheduled at $(\phi_{i,j} + P_i \cdot k)$-th time slot, where $k = 0, 1, 2, ...$
  3) only one fragment is scheduled at a time slot
  4) The network utilization $U = \sum_{i=1}^n \frac{C_i}{P_i}$ is minimized

• JFS problem is NP-hard
Harmonic Chain Based Jitter-Free (HCJF) Scheduler

- Find a sufficient condition that yields a pseudo-polynomial time solution to the JFS problem.

- **Stage 1: Period Selection**
  - Select period \( P_i \in [P_i^{min}, P_i^{max}] \) for each task, so that all the \( \{P_i\}_{i=1}^{n} \) form a harmonic chain and the network utilization \( U = \sum_{i=1}^{n} \frac{C_i}{P_i} \) is minimized.

  [A set \( S \) of positive integers forms a **harmonic chain** if and only if \( \forall x, y \in S, (x | y) \lor (y | x) \).]

- **Stage 2: Phasing Assignment**
  - Assign phasing using a greedy algorithm
Period Selection Algorithm

- Without loss of generality, assume: for $i > j$:
  \[ P_i^{max} > P_j^{max}, \text{ OR } P_i^{max} = P_j^{max} \text{ and } P_i^{min} \geq P_j^{min} \]

- Under this assumption, we have proved that the periods assigned to the task are non-decreasing.
  i.e. $\forall i \in [1, n - 1], \ P_i \mid P_{i+1}$

$p_{i,j}$: $j$-th admissible period of task $T_i$

$p_{1,1} = P_1^{min}$
$p_{1,1-1} = P_1^{max} - 1$
$p_{1,2} = P_1^{min} + 1$
$p_{1,1} = P_1^{min}$

$T_1$, $T_2$, $T_3$, $T_{n-1}$, $T_n$
Period Selection Algorithm

If \( p_{i,j} \) can form a harmonic chain with some admissible periods from \( T_1 \) to \( T_{i-1} \), then:

- \( u_{i,j} \) stores the minimum utilization of these harmonic chain
- \( \text{prev}_{i,j} \) is the backpointer to the previous node in the harmonic chain with minimum utilization

\[
\begin{align*}
    u_{i,j} &= \frac{c_1}{p_{1,h_1}} + \ldots + \frac{c_{i-1}}{p_{i-1,h_{i-1}}} + \frac{c_i}{p_{i,j}} \\
    &= u_{i-1,h_{i-1}} + \frac{c_i}{p_{i,j}} \quad \text{prev}_{i,j}
\end{align*}
\]
Period Selection Algorithm

Initialization:

\[ u_{1,j} = \frac{c_i}{p_{1,j}} \quad 1 \leq j \leq P_1^{max} - P_1^{min} + 1 \]

\[ \text{prev}_{1,j} = 1 \]
Recursion: Let set $S = \{ x \mid (p_{i-1,x} | p_{i,j}) \land (prev_{i-1,x} \neq \text{null}) \}$

If $S \neq \emptyset$:

$$u_{i,j} = \frac{c_i}{p_{i,j}} + \min_{y \in S} u_{i-1,y}$$

$$prev_{i,j} = \arg \min_{y \in S} u_{i-1,y}$$

If $S = \emptyset$:

$$u_{i,j} = +\infty$$

$$prev_{i,j} = \text{null}$$
Recursion: Let set $S = \{x \mid (p_{i-1,x} | p_{i,j}) \land (prev_{i-1,x} \neq \text{null})\}$

If $S \neq \emptyset$:

$$u_{i,j} = \frac{c_i}{p_{i,j}} + \min_{y \in S} u_{i-1,y}$$

$$prev_{i,j} = \text{argmin}_{y \in S} u_{i-1,y}$$

If $S = \emptyset$:

$$u_{i,j} = +\infty$$

$$prev_{i,j} = \text{null}$$
**Period Selection Algorithm**

Recursion: Let set $S = \{x \mid (p_{i-1,x} \mid p_{i,j}) \land (prev_{i-1,x} \neq \text{null})\}$

If $S \neq \emptyset$:

- $u_{i,j} = \frac{c_i}{p_{i,j}} + \min_{y \in S} u_{i-1,y}$
- $prev_{i,j} = \arg \min_{y \in S} u_{i-1,y}$

If $S = \emptyset$:

- $u_{i,j} = +\infty$
- $prev_{i,j} = \text{null}$
Termination:

\[ U = \min_{j} u_{n,j} \quad 1 \leq j \leq m_n \quad q = \arg\min_{j} u_{n,j} \]

If \( U = +\infty \): Harmonic chain not found

If \( U \neq +\infty \): The optimal network utilization is \( U, P_n = p_{n,q} \), then follow the backpointers \( \text{prev} \) to construct the full period sequence \( P_{n-1}, P_{n-2}, \ldots, P_1 \)
Phasing Assignment in Static Network

- Create a superframe with size $P_n$
- Assign phasing from $T_1$ to $T_n$ in sequence
- Assign the first unallocated phasing in the schedule to a fragment of a task
- The task set is schedulable if the network utilization $U$ is less than or equal to 1.

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$\phi_1$</th>
<th>$\phi_2$</th>
<th>$\phi_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>2</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-3</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4-5</td>
<td></td>
<td></td>
<td>12</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>6-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$T_1: P_1 = 2, C_1 = 1 \quad \phi_1 = 0$

$T_2: P_2 = 6, C_2 = 1 \quad \phi_2 = 1$

$T_3: P_3 = 12, C_3 = 1 \quad \phi_3 = 3$
Dynamic Network Scheduling

- Devices may join and leave the network frequently
- No prior knowledge on future requests
- Existing schedule may need to be adjusted to admit new tasks

Example:

At $t_1$, $T_1$: $P_1 = 4$, $C_1 = 1$ joins.
- $\varphi_1 = 0$

At $t_2$, $T_2$: $P_2 = 4$, $C_2 = 1$ joins.
- $\varphi_2 = 1$
- $\varphi_2 = 2$

At $t_3$, $T_3$: $P_3 = 2$, $C_3 = 1$ joins.
- $\varphi_3 = 1$
Dynamic Network Scheduling

- Devices may join and leave the network frequently
- No prior knowledge on future requests
- Existing schedule may need to be adjusted to admit new tasks

- Adjusting an existing schedule will cause:
  - network configuration overhead
  - control overhead to adjust sensor/actuator

- An efficient on-line scheduling algorithm to minimize schedule adjustment overhead is needed.
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule
Dynamic Phasing Assignment

• S-tree: a data structure to represent the schedule
Dynamic Phasing Assignment

• **S-tree**: a data structure to represent the schedule

![Diagram of S-tree]

\[
\phi = 0
\]
Dynamic Phasing Assignment

- **S-tree**: a data structure to represent the schedule

![Diagram of S-tree](Image)

- Harmonic Chain
  - P = 1
  - P = 2
  - P = 4
  - P = 8

0 1 2 3 4 5 6 7
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule

![Diagram of S-tree with phasing assignments]
S-tree Node Status

Free Node  Occupied Node  Semi-occupied Node

Root

\[
\phi = 0 \quad \phi = 1 \\
\phi = 0 \quad \phi = 1 \quad \phi = 3 \\
\phi = 0 \quad \phi = 1 \quad \phi = 5 \\
\phi = 0 \quad \phi = 1 \quad \phi = 7
\]

P = 1  P = 2  P = 4  P = 8

<table>
<thead>
<tr>
<th>B</th>
<th></th>
<th></th>
<th>B</th>
<th>A</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
S-tree Node Attribute:
supported-periods & num-of-tasks

- Free Node
- Occupied Node
- Semi-occupied Node

**Supported-periods**

**Num-of-tasks**

```
Root  4,8  2

P = 1

φ = 0  φ = 1
4  8
1  1

P = 2

φ = 0  φ = 1
B   4
φ = 2  φ = 3
8  4
φ = 6  φ = 7
8  8

P = 4

φ = 0  φ = 1
8  8
φ = 4  φ = 5
A
φ = 6  φ = 7
8  8

P = 8
```

**Node Attributes**

- supported-periods
- num-of-tasks

```
0  1  2  3  4  5  6  7

B  B  A
```
Add-task in S-tree

Assignment Policy:
To add a task with period $P_i$: start from the root, follow a path with supported-period $\leq P_i$ until we find a free node with period $P_i$.
When more than one node can support the new task, select the node whose supported-period is closest to $P_i$. 

New task C: $P = 8$, $C = 1$
Add-task in S-tree

Replacement Policy:
If only semi-occupied nodes are available for new task, the semi-occupied node with least number of tasks will be replaced.
Add-task in S-tree

Replacement Policy:
If only semi-occupied nodes are available for new task, the semi-occupied node with least number of tasks will be replaced.

Replacement Policy:
If only semi-occupied nodes are available for new task, the semi-occupied node with least number of tasks will be replaced.

New task E: P = 2, C = 1

Adjusted Tasks List:
- Task C is removed from S-tree temporarily
Add-task in S-tree

Replacement Policy:
If only semi-occupied nodes are available for new task, the semi-occupied node with least number of tasks will be replaced.

Adjusted Tasks List:
- New task E: P = 2, C = 1
- Root 4,8
- P = 1
- φ = 0 φ = 1 φ = 2 φ = 3
- Adjusted
- Tasks List :
Add-task in S-tree

Adjustment Policy:
When adjusting the schedule of a task, we assign a node that has the closest phasing to its original phasing.

New task E: P = 2, C = 1

Adjusted Tasks List:
Adjustment Policy:
When adjusting the schedule of a task, we assign a node that has the closest phasing to its original phasing.

New task E: P = 2, C = 1

Adjusted Tasks List:

Add-task in S-tree
HCJF Scheduler Performance Evaluation
- Network Performance

– Compare with:
  • Earliest Deadline First scheduler (EDF)
  • Rate Monotonic scheduler (RM)
  • Contention-Free periodic message scheduler (CF)

– Simulation Setup
  • Time slot size: 200 μs
  • Randomly generate 500 join/leave requests
  • \( P_{\text{min}} \): [2, 20] uniformly distributed (simulating 250Hz to 2500Hz max. supported sampling rate)
  • \( P_{\text{max}} \): [10, 600] uniformly distributed (simulating 8.33Hz to 500Hz min. required sampling rate)
  • Number of fragments: [1,3] Poisson distribution with \( \lambda = 1 \)
  • 100 runs for each simulation
HCJF Scheduler Performance Evaluation
- Network Performance (Cont.)

**Overall Jitter**

![Graph showing overall jitter vs. number of nodes for different scheduling algorithms.]

**Number of Adjustments**

![Graph showing number of adjustments vs. number of nodes for different scheduling algorithms.]

- RM
- EDF
- CF
- HCJF