Present a series of contributions to help closing the gaps between current Qdj and the industry requirements.
why (and how) User/Network Interface must be modified
Agenda

• Industrial automation requirements
  • System level
  • End station level
  • Network level (out of scope of this contribution)

• Stream request and response requirements

• Proposed UNI parameters
  • Timeliness aspects
  • Other aspects in future contributions
System level: Plug & Produce

Dynamics and converged network
Plug & produce

Machines (network parties)
- added and/or removed without additional network engineering steps

- Incremental network configuration needed
  - Applications request network resources
  - If request is granted, new applications communicate over existing network
    - New streams established
    - Old streams still work
  - Stream removal possible
    - Releasing no longer needed resources

Incremental network configuration and resource allocation
Converged network supporting Plug & produce

Application engineering
• Functionality
• Timeliness requirements
• Including communication

Independent of network knowledge
• Shaper agnostic
• MAC addresses unknown
• …
End Station (ES) level: ES Model
End station by IEEE/IEC 60802

- Communication requirements from IA application
- Data objects (application and communication relations)
- ES middleware (CUC part of middleware)
- Translates application layer requirements into stream requirements
- Sends Stream requests to CNC via UNI
- CNC responses to middleware via UNI
- Setup comm based on stream responses
- CNC establishes stream using remote management
• End station capabilities discovered by CNC via remote management
ES Model: Application Layer
IOC and IOD have the same understanding of task cycle.
Wide range required by applications

<table>
<thead>
<tr>
<th>Cycle length</th>
<th>25µs to 4s (10^{-9} to 10^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame size</td>
<td>64 to 2000 (octets)</td>
</tr>
<tr>
<td>Cycle length</td>
<td>512</td>
</tr>
<tr>
<td>Talkers *1</td>
<td>512</td>
</tr>
<tr>
<td>Listeners *1</td>
<td>1000</td>
</tr>
<tr>
<td>Devices</td>
<td>&gt; 9k</td>
</tr>
<tr>
<td>Streams *2</td>
<td>large amount</td>
</tr>
</tbody>
</table>

*1 per IOC
*2 C2D and C2C, with 8 PLCs
ES Model: Middleware Shim
Tasks of middleware (relevant for UNI)

Manage stream requests
• Translates application layer requirements into stream requirements

Setup end station communication
ES Model: Ethernet Interface
Ethernet Interface as per IEEE/IEC 60802

**Time Aware Offset Control**
- Stream(stream group)-based queue
- Talker added by application layer

**Gate Control (Qbv)**
- Priority-based queue
- Triggered by gating control list
Frame injection: reduction ratio concept

Based on reduction ratio concept

- Common base-cycle
  - $Q_{bv}$ gating cycle
- Stream interval expressed by reduction ratio
  - power of 2 multiple of gating cycles
Frame injection: reduction ratio concept

Based on reduction ratio concept
- Common base-cycle
  - Qbv gating cycle
- Stream interval expressed by reduction ratio
  - power of 2 multiple of gating cycles

Facilitates implementation
End station centric representation
- Common gating cycle behavior
  - Defined once
- Individual gating cycle behavior defined for each phase
  - Constrained scope (per phase)

Phase: identifies out of a set of gating cycles (w.r.t reduction ratio) the one in which the transmission of a stream starts.
For a stream with RR=n, max number of phases=n
Frame injection: reduction ratio concept

Based on reduction ratio concept
- Common base-cycle
  - $Q_{bv}$ gating cycle
- Stream interval expressed by reduction ratio
  - Power of 2 multiple of gating cycles

Facilitates implementation
- End station centric representation
  - Common gating cycle behavior
    - Defined once
  - Individual gating cycle behavior defined for each phase
    - Constrained scope (per phase)
Stream A: RR=4, Phase=4
Stream B: RR=4, Phase=4
Stream C: RR=4, Phase=3
Stream D: RR=2, Phase=2

Phase 1  Phase 2  Phase 3  Phase 4  Phase 1  Phase 2  Phase 3  Phase 4

Gate Control (Qbv) configures behavior of gating cycles (once for all gating cycles)

“wire view” after gate control

Time Aware Offset Control assigns streams to gating cycles by means of phases (per stream response)
Transmission of frames in **same gating cycle (phase)**
- As burst ordered by **priority** (gate control)
- Then ordered by **reduction ratio**
- Then by **sequence**
  - For instance after:

**Time Aware Offset Control**
This solution favors

- **Dynamic stream requests**
- **Predictability**
  - Defined transmission order of streams per gating cycle
- **Lower dispatching complexity**

over

- **Optimal network utilization**

<table>
<thead>
<tr>
<th>25µs to 4s</th>
<th>Cycle length</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-9} to 10^{0}</td>
<td>Talkers '1</td>
</tr>
</tbody>
</table>

512
Stream Request & Response
Stream traffic types in IEEE/IEC 60802

- Isochronous
  - Transmission in sync with network and task

- Cyclic synchronous
  - Transmission in sync with network

- Cyclic asynchronous
  - Transmission in sync with network is optional

Examples in this contribution
- Isochronous

UNI parameters presented in this contribution
- covers all three types
Stream timeliness: One IO-Controller, one IO-Device, isochronous tasks
Stream timeliness: One IO-Controller, one IO-Device, isochronous tasks
Stream request & response parameters

**Stream request parameters**

1. MinUpdateInterval
2. Communication deadline
3. Frame size
4. ...

**Further stream properties**

- Traffic class
  - ISO, CYC-S, CYC-A, ...
- Redundancy
- Source, Sinks
- ...

Not addressed in this contribution
Stream timeliness: One IO-Controller, one IO-Device, isochronous tasks

Stream request parameters
1. Min, MaxUpdateInterval
2. (backwards) Communication deadline
3. Frame size
4. ...

Stream response parameters
1. Computed update interval
2. Phase
3. Sequence identifier
4. ...

- Max - Min update interval
- (backwards) Comm deadline
- Frame size
User/Network Interface: Proposed parameters
Stream Request over UNI: timeliness related parameters

**Stream request** parameters

1. Min-,MaxUpdateInterval
2. (backwards) Communication deadline
3. Frame size
4. ...

[Diagram showing stream parameters with labels for new and existing parameters]

- New parameter
- Existing parameter
Stream Request over UNI: timeliness related parameters

Stream request parameters
1. Min-,MaxUpdateInterval
2. (backwards) Communication deadline
3. Frame size
4. ...

New: Minimum and Maximum Update Interval

specify the minimum and maximum update interval of time in which the traffic specification cannot be exceeded. The traffic specification is specified by MaxFramesPerInterval=1 and MaxFrameSize.

Note that definition of interval is very similar to the one in Qcc: TrafficSpecification Interval (46.2.3.5.1)
Stream Request over UNI: timeliness related parameters

Stream request parameters

1. Min-, MaxUpdateInterval
2. (backwards) Communication deadline
3. Frame size
4. ...

New: Backwards Communication Deadline

Latest arrival time at the listener(s) of this stream.

This value should be specified ‘backwards’, i.e.:

- the ‘first point’ is the end of the update interval.
- the ‘second point’ is at the reference plane at the Listener marking the boundary between the network media and PHY.
Stream Request over UNI: timeliness related parameters

Stream request parameters

1. Min-, MaxUpdateInterval
2. (backwards) Communication deadline
3. Frame size ✅
4. ...

Existing: TrafficSpecification.MaxFrameSize
Stream Response over UNI: timeliness related parameters

Stream response parameters
1. Calculated update interval
2. Transmission phase offset
3. Sequence identifier
4. …
Stream Response over UNI: timeliness related parameters

**New: Calculated update interval**

Value between provided min und max update interval
Is a result from CNC, not a requirement from CUC

Note that definition of interval is very similar to the one in Qcc: TrafficSpecification Interval (46.2.3.5.1)
Stream Response over UNI: timeliness related parameters

New: Transmission phase offset

Per burst: offset to indicate the start time of the gating cycle in which the stream is transmitted. Reference point is the start of the planned update interval.

Per frame: offset to indicate the start time when the frame is transmitted. Reference point is the start of the planned update interval.

Stream response parameters
1. Calculated update interval
2. Transmission phase offset
3. Sequence identifier
4. …
Stream Response over UNI: timeliness related parameters

New: Sequence identifier

Sequence identifier defines a tie-break rule to decide the transmission order of streams planned to be sent as burst in:

- Same gating cycle
- With same priority
- With same reduction ratio

To allow for simple incremental scheduling implementation, this parameter represents a pointer to the previously sent stream.

For instance:

- after: ☐ ☐

Stream response parameters

1. Calculated update interval
2. Transmission phase offset
3. Sequence identifier ☐
4. …
Summary
Summary

Goal: allow for Qdj to fulfill the industry requirements

This contribution shows that currently Qdj does not cover all industry requirements

This contribution focused at UNI and presented
- main timeliness issues
- first modification proposals

Next contributions
- address comments from this contribution
- present further industry related issues at UNI and propose respective Qdj modifications, e.g.
  - Stream ID management by CNC
  - Network agnostic representation of
    - Source and sink
    - Stream traffic type
  - Multiple CUCs
- present textual contributions to Qdj
Questions ?
Contact

Published by Siemens AG

Dr. Rodrigo Ferreira Coelho
System Architect
DI FA CTR ICO ARC
Siemenspromenade 1
91058 Erlangen
Deutschland

Phone: +49 9131 17-45546
E-mail: rodrigo.ferreira_coelho@siemens.com
End station and its interactions with CNC by IEEE/IEC 60802

Application Layer

Middleware Shim

Ethernet Interface

Requirements

Stream Requests

Capabilities

App Requirements

Ethernet IF Capabilities

End station

CNC

UNI (Stream Request/Response)

Configure

Discovery

Network

Capabilities

Configure

Discovery
## Stream traffic types and requirements

<table>
<thead>
<tr>
<th>Periodic</th>
<th>Isochronous</th>
<th>Yes</th>
<th>Cyclic synchronous</th>
<th>Yes</th>
<th>Cyclic Asynchronous</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission in sync with network</td>
<td>Yes</td>
<td>Yes</td>
<td>Optional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmission in sync with task (application layer)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tolerance to frame loss due to congestion</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timeliness</td>
<td>Deadline (w.r.t. task cycle, i.e. the start of gating cycle in which update interval starts)</td>
<td>Max latency (w.r.t. start of gating cycle in which the stream is sent)</td>
<td>Max latency (w.r.t. start of gating cycle in which the stream is sent)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>