Supporting new TSN features in a decentralized and centralized controlled network

Franz-Josef Goetz, Siemens AG
Juergen Schmitt – Siemens AG
Thoughts about Centralized and Decentralized Systems

1. Centralized Organized Systems
   - Within industrial automation there are a lot of established centralized organized systems (e.g. EtherCat, PROFINet, VARAN, ...).

2. Decentralized Organized Systems
   - Ethernet and Internet are well known decentralized organized Systems.
   - AVB is also a decentralized organized system.

When talking about industrial automation we have to differentiate between administration, applications and communication.

With introducing TSN in industrial automation, vendors are also requesting for a decentralized organized communication system. One example is the ongoing discussion about OPC_UA over TSN.

ONE Solution for Centralized and Decentralized organized industrial networks

- TSN is talking about how to implement deterministic Ethernet
- Deterministic Ethernet can be implemented “all traffic is scheduled”
  BUT TSN has also specified other mechanism (e.g. traffic classes, reservation in combination with strict priority and pre-emption)
- The communication can be organized centralized and decentralized
- Diagnostic / double check is very important for industrial communication

- IEEE 802.1 has already standardized a lot of building blocks for a centralized or decentralized organized networks
- TSN has to reuse the in IEEE 802.1 standardized mechanism and building blocks
- TSN has to take care not to overload existing protocols
- If the existing building blocks have too much functionality we should specify a “profile” for an industrial TSN network to restrict the functionality
- If the existing building blocks do not cover the required functionality TSN has to fill the gap

=> The task of TSN within IEEE 802.1 is to standardize ONE and only ONE solution handling centralized or decentralized organized TSN networks.
=> This is the only way TSN becomes successful for the industrial market!
IEEE 802.1 has standardized a lot of mechanism and building blocks
(e.g. .1Qai, .1Qak, .1Qal, .1AS, .1Qat, .1Qav, .1Qbu, .1Qbh, .1Qca, .1CB, ...)

The following slides shows how these building blocks can be used for
TSN to support centralized or decentralized organized TSN networks!

The following slides shows
• **gaps**, which must be filled and
• **interfaces** for which the TSN group has to specify **data objects**
Current Discussion!

The current .1Qcc draft shows three concepts for network configuration:

1. Fully Distributed Model
2. Centralized Network (based on system protocols) / **Distributed User Model**
3. Fully Centralized Model (based on system protocols) + supporting “Scheduled Traffic”

This presentation shows an alternative based on mechanism and building blocks already introduced in IEEE802.1:

1. Fully Distributed Model (not supporting “Scheduled Traffic”)
2. Centralized Network (based on .1Qca) / **Distributed User Model** + supporting “Scheduled Traffic”
3. Fully Centralized Model (based on .1Qca) + supporting “Scheduled Traffic”

AVB: Decentralized controlled Network with Registration & Reservation based on RSTP

1. Fully Distributed Model (specified with AVB )

ASSUMPTION:
- *Loop free topology based on STP (spanning tree protocol)*
TSN has introduced **new features** like (seamless) redundancy based on path computing.

To support the new features like “**Seamless Redundancy**” in a **decentralized controlled network** additional data objects and protocols are necessary:

- ISIS-PCR (specified in .1Qca) for topology discovery and path computing
  (also path computing algorithm like Dijkstra, SP – shortest path or MRT – Multiple-Redundant-Tree)
  =&gt; **BLCE’s** – Bridge-Local-Computing-Elements (specified in .1Qca)

- **NEW MRRP Multiple-Relation-Registration Protocol** to nail down the path for the registration of network attributes

- MVRP is used to establish the data planes (VLAN’s / VID’s)

- MMRP (optional) to configure the forwarding behavior for unregistered MAC addresses

- MSRP to register the Stream Attributes (e.g. SR-DA, Tspec, availability, ..)

- SRP to do stream reservation (**min. latency, max latency, ..**)
1. Fully Distributed Model (for TSN to support redundancy)

NEW:

- Each bridge has to support BLCE functionality *(specified in 1Qca, distributed path computation)*
- ISIS-PCR is just for topology discovery and path computing (“Next Hop”)
- New MRP application MRRP – Multiple Relation-Registration protocol to nail down the path for Stream registration & reservation *(is a replacement of RSTP)*
BUT in TSN we need mechanisms that allow Stream Reservation class (SR class) parameters to be configured because TSN has introduced new shaper, pre-emption, CT, … (in comparison to AVB we have predefined traffic classes)

- Managed Objects are required to configure traffic classes for a time sensitive network (observation interval, priority, VID, shaper, redundancy, max. MTU size, …)

- Managed Objects are required to configure max. available bandwidth for each traffic class (traffic classes for stream and traffic classes for best effort traffic)

- …
1. Fully Distributed Model (with Network Manager)

NEW:
- **Network manager to distribute managed objects** (supporting new managed objects for new TSN features)
BUT within TSN we still have the requirement to (parts of the .1QCC PAR)

- **Support for more streams.** The current worst case limit is less than 500 streams; there are use cases that require two orders of magnitude greater than this.

- **Inclusion of additional parameters and mechanisms in the stream reservation protocol that support additional applications, such as higher reliability, latency requirements, and latency changes due to network reconfiguration.**

- **Support for higher layer streaming sessions, such as Real-Time Protocol (RTP)-based sessions.**

- **Deterministic stream reservation convergence.**

With MSRP/SRP we have already overloaded MRP AND with MRRP and additional parameters to describe streams (supporting high reliability). TSN is continuing overloading MRP *(more data objects, more MRP PDU’s, more applications, more …)*

**Proposal:**

Splitting Registration and Reservation into MRP++ for registration and MSP for reservation *(more details in the end of this presentation)*
Decentralized controlled Network with Path Computation, Registration & Reservation

1. Fully Distributed Model (distinguishing registration & reservation)

NEW:
- **MRP++** for registration to support more data objects, streams, ... with one PDU (ISIS like on link)
- **MSP** for reservation to support better performance and to support more reservation applications (which a necessary for converged networks like rate constrained best effort traffic)
BUT
we have to be compatible to the current version of MRP (MRRP, MVRP, MMRP, MSRP/SRP)

AND
we should expand the current version of MRP to support the new TSN features.
Decentralized controlled Network with Path Computation, Registration & Reservation

1. Fully Distributed Model (using existing MSRP/SRP as outbound interface)

**ToDo:**

- **Using existing MRP (including MRRP, MVRP, MMRP, MSRP/SRP) for registration & reservation between end station and edge bridge** (guess UNI-Interface)
- **Adding to existing MRP data objects for control (TLV’s) to support new TSN features like redundancy**
Centralized Network / Distributed User Model with Path Computation, Registration & Reservation

ISIS-PCR, specified in .1Qca, supports also a centralized controlled network by introducing PCEs (Path-Computing-Element specified in IETF).

New TSN features like (seamless) redundancy, time aware shaper (TAS) are based on path computing.

Proposal 1 - Using ISIS-PCR also for registration:

- Using PCE for centralized path computing

- ISIS-PCR is used for topology discovery
  - and to distribute Stream specification (currently not in .1Qca)
  - and to distribute the “Explicit Trees” for streams

- Using MSP for Stream reservation (E2E signaling)
Centralized Network / Distributed User Model with Path Computation, Registration & Reservation

2. Centralized Network / Distributed User Model (Using ISIS-PCR also for registration)

Proposal 1:
- **PCE for centralized path computing and registration**
- **ISIS-PCR is used for topology discover and distributing registration for data objects for network control** (e.g. stream specification)
- **MSP is used for stream reservation and also E2E signaling**
Centralized Network / *Distributed User Model* with Path Computation, Registration & Reservation

ISIS-PCR, specified in .1Qca supports also a centralized controlled network by introducing PCE’s (Path-Computing-Element specified in IETF) supporting the new TSN features like (seamless) redundancy based on path computing.

Proposal 2 - Introducing PCEP for path computing request / response and using MRP++ for “Explicit Tree” registration:

- Using ISIS-PCR **just** for topology discovery
- Using PCE for centralized path computing
- Introducing PCEP (Path-Computing-Element-Protocol original specified in IETF) to
  - request / response for path-computing (communication relation)
- Using MRP++ to distribute
  - “Explicit Tree” for streams (gained by PCEP response)
  - stream specification
- Using MSP for Stream reservation (E2E signaling)
Centralized Network / Distributed User Model with Path Computation, Registration & Reservation

2. Centralized Network / Distributed User Model (Introducing PCEP for path computing request / response and using MRP++ for “Explicit Tree” registration)

Proposal 2:
- **PCE for centralized path computing**
- **ISIS-PCR is just used for topology discover**
- **PCEP is used to request / response path computing (“Explicit Tree”) for streams / relations**
- **MRP++ to register data objects for network control, stream specification, distributing also the data object for “Explicit Tree” and all the others**
- **MSP is used for stream reservation and also E2E signaling**
Centralized Network / Distributed User Model with Path Computation, Registration & Reservation

**BUT** to support SCHEDULING (TAS- time-aware-shaper) introducing new SCHEDULING-Function into PCEs is necessary. The current functionality of PCEP and also MRP must be extended.

**Proposal 3 – Supporting “SCHEDULING”:**

- Using ISIS-PCR die topology discovery
- Using PCE for centralized path computing and scheduling for TAS (time aware shaper)
- Using PCEP+ to
  - request / response for path-computing and for scheduling for specified streams
  - distributing the window size for each scheduled traffic class and also distributing the information like which streams are scheduled
- Using MRP++ to distribute
  - “Explicit Tree” for streams (gained by PCEP response)
  - stream specification
- Using MSP for Stream reservation (looking that the Stream is correctly scheduled)
Centralized Network / Distributed User Model with Path Computation, Registration & Reservation

2. Centralized Network / Distributed User Model (supporting “SCHEDULING”)

Proposal 3:
- **PCE for centralized path computing**
- **ISIS-PCR is just used for topology discover**
- **PCEP+ is used to request / response path computing and scheduling** + distributing data objects to each bridge along the path like window size
- **MRP++ to register data objects for network control, stream specification, distributing also the data object for “Explicit Tree” and all the others**
- **MSP is used for stream reservation and also E2E signaling**
Centralized Network / Distributed User Model
with Path Computation, Registration & Reservation

BUT to support Path-Computing and Scheduling with a Network-Controller

Proposal 4:

- Using ISIS-PCR die topology discovery
- Using PCE in a Network-Controller for centralized path computing and scheduling for TAS
- Using PCA (Path-Computing-Agent) to communicate with a Network-Controller (includes a PCE)
- Using PCEP+ to
  - request / response for path-computing and for scheduling for streams
  - distributing the window size for each scheduled traffic class and also distributing the information like which streams are scheduled
- Using MRP++ to distribute
  - “Explicit Tree” for streams (gained by PCEP response)
  - stream specification
- Using MSP for Stream reservation (looking that the Stream is correctly scheduled)
2. Centralized Network / Distributed User Model (supporting “SCHEDULING”)

Proposal 4:

- **PCE** for centralized path computing
- **ISIS-PCR** is just used for topology discover
- **PCEP+** is used to request / response path computing and scheduling + distributing data objects to each bridge along the path like window size
- **PCAP** is used for communication between a PAC and PCE
- **MRP++** to register data objects for network control, stream specification, distributing also the data object for “Explicit Tree” and all the others
- **MSP** is used for stream reservation and also E2E signaling
To support the “Fully Centralized Model”

**Proposal 5:**

- Using ISIS-PCR for topology discovery

- Using PCE in a Network-Controller for centralized path computing and scheduling for TAS
  - The Network-Controller gets the communication relations and stream specification from a “Centralized-User-Configuration”

- Using PCA (Path-Computing-Agent) to communicate with a Network-Controller (includes a PCE)

- Using PCEP+ to
  - **distributing result** of path-computing and scheduling
  - distributing the window size for each scheduled traffic class and also distributing the information like which streams are scheduled

- Using MRP++ to distribute
  - “Explicit Tree” for streams (gained by PCEP response)
  - stream specification

- Using MSP for Stream reservation (looking that the Stream is correctly scheduled)
Centralized Network / Distributed User Model with Path Computation, Registration & Reservation

3. Fully Centralized Network (supporting “SCHEDULING”)
3. Fully Centralized Network (supporting “SCHEDULING”)

Proposal 5:
- **PCE for centralized path computing**
- **ISIS-PCR is just used for topology discover**
- **PCEP+ is used to distribute path computing and scheduling + distributing data objects to each bridge along the path like window size**
- **PCAP is used for communication between a PAC and PCE**
- **MRP++ to register data objects for network control, stream specification, distributing also the data object for “Explicit Tree” and all the others**
- **MSP is used for stream reservation and also E2E signaling**
MRP++ and MSP
Reasons for splitting Registration and Reservation

At the moment within the TSN group there is a discussion

- How to “support for more streams. The current worst case limit is less than 500 streams; there are use cases that require two orders of magnitude greater than this.”
- How to get “deterministic stream reservation convergence.”
- ...

(excerpt from the .1Qcc PAR)

The following slides

- explain registration
- explain reservation
- and show the difference also in the architecture between both
Registration:

Properties:
- Attributes get synchronized between links (*ISIS-like on link*)
- Synchronized data is constant (no modification within a Bridge)
- No creation of new Attributes
- Has to scale to larger amount of data (Fragmentation of PDU is necessary)
- Performance of attribute propagation is not the main focus

Main focus:
- Reliable synchronization of network attributes within an active topology given by a context. (In contrast to ISIS where Attributes are flooded all over the network to everybody)
MRP++ Architecture

synchronizing Network Attributes (e.g. VLAN, MAC Address, Stream description, ...)

Page 28  15. Mai 2015
One Registration Application (out of others) is MSRP (Multiple Stream Registration Protocol)

Used to propagate the static properties of a stream along the path.

Such properties are for example:
- VID
- Max. frame size
- Frame priority
- Rank
- Stream-ID
- Tree-ID (for path)
- Stream destination MAC
- …
Signaling

Properties:
- Directed from source to sink
- Attributes can get modified at every Hop
- Changes along the Path between source and sink has to be signaled very fast
- Attribute disappears if source withdraws or times-out
- Beside cyclic Link-To-Link synchronization, event based PDUs are necessary

Main focus:
- End-to-End signaling
- Monitoring the route between source and sink
- Fast signaling of changes along the road to source and sink
- Signal the source and the sink what they get if they go along the route
Basic MSP Architecture

Participant

Signalling application

MAD

Reservation synchronisation

LLC

Transmission

Reception

Attribute Sink

15. Mai 2015

Page 31

Attribute Source

Basic MSP Architecture

Signalling application

Bridge-Global-Signalling application

MAP CONTEXT

MAD

Reservation synchronisation

LLC

Transmission

Reception

MAC Relay Entity

Transmission

Reception

Attribute Sink

15. Mai 2015

Page 31

Attribute Source
One Signaling Application (out of others) is MSSP (Multiple Stream Signaling Protocol)

Used to:

- propagate the dynamic properties of a stream along the path (Upstream AND Downstream).

  E.g.:
  - Accumulated Latency (Downstream)
  - Required Latency (Upstream)
  - Min. receive Interval (Upstream)
  - Effective receive Interval (Downstream)
  - Stream send state (Ready/Failed) (Downstream)
  - Stream receive state (Ready/Failed/ReadyFailed) (Upstream)

- Use the event based messages to by-pass the slow cyclic Link-To-Link synchronization to signal disruptive events on the path (e.g. Link-Down due to wire break)
MSSP Architecture
(Multiple Stream Signaling based on the MSP-Architecture)
Conclusion for decentralized and centralized Approaches

General
- **Ongoing task in .1Qcc**
  - Adding to existing MRP data objects for control (TLV’s) to support new TSN features like redundancy
  - Specifying new Managed Objects which required to configure traffic classes
- **New work item:**
  - Splitting Registration and Reservation into MRP++ for registration and MSP for reservation

For the “Centralized Network / Distributed User Model” and also for the Fully Centralized Model” there are 4 proposals:
- **Proposal 1** - Using ISIS-PCR also for registration -> will overload ISIS-PCR (scaling issue)!
- **Proposal 2** - Introducing PCEP for path computing request / response and using MRP++ for “Explicit Tree” registration
- **Proposal 3** - Supporting “SCHEDULING”
- **Proposal 4** - Network-Controller with PCE functionality
- **Proposal 5** - Implementation proposal for the “Fully Centralized Model”

**New Work item for Proposal 2,3,4,5:**
- Standardizing PCEP and its data objects for Ethernet (supporting also optional “Scheduled Traffic”) within IEEE 802.1
- Splitting MRP and its applications in registration (MRP++) and reservation (MSP)

=> Discussion: How to proceed?
Thank you for your attention!

Franz-Josef Goetz
PD TI ATS TM 4 2
Gleiwitzer-Str. 555
90475 Nürnberg
Phone: +49 (911) 895-3455
E-Mail: franz-josef.goetz@siemens.com

Jürgen Schmitt
PD TI ATS TM 4 2
Gleiwitzer-Str. 555
90475 Nürnberg
Phone: +49 (911) 895-5338
E-Mail: juergen.jues.schmitt@siemens.com
Backup

The following slides contain further details!
## Motivation for V2 MRP (Multiple Registration Protocol) and V1 MSP (Multiple Signaling Protocol)

<table>
<thead>
<tr>
<th>Pro (also Supported by new Version)</th>
<th>Cons</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution of network attributes over context</td>
<td>No fragmentation - limits the number of attributes. This problem is partly solved by spending one separate frame for each application or application instance. The disadvantage of the current solution that high computing power is required for serialization and deserialization.</td>
<td>+ Supported Fragmentation</td>
</tr>
<tr>
<td>One basic mechanism for different applications (MVRP, MMRP,...)</td>
<td>Very complex and transparent state machines -&gt; difficult to synchronize implementations from different vendors</td>
<td>+ Supported Fragmentation</td>
</tr>
<tr>
<td>Common architecture (application--&gt;instance--&gt;attribute)</td>
<td>MRP combines registration and reservation, the attribute size (advertise) is very large and extended the MAP mechanism and introduced four packed events exclusive for MSRP</td>
<td>+ Supported Fragmentation</td>
</tr>
<tr>
<td></td>
<td>The pack mechanism form MRP is not practical (only for special use cases)</td>
<td>+ By introducing fragmentation the packed mechanism is no longer necessary</td>
</tr>
<tr>
<td></td>
<td>MSRP combines registration and reservation, the attribute size (advertise) is very large and extended the MAP mechanism and introduced four packed events exclusive for MSRP</td>
<td>+ Extending existing applications (MVRP, MMRP, MSRP) to support redundancy and seamless redundant on precalculated trees</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ If necessary add a new application like MRRP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ Optional support for higher layers like IP (e.g. transport higher layer addresses, QoS specifier, ..) by e.g. using TLV's</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ Managed Objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ TLV's are used to specify the MRP attributes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ The mechanism to synchronize the attribute list on a link is comparable to the synchronization mechanism used by ISIS (ISIS-like)</td>
</tr>
<tr>
<td></td>
<td>MSRP combines registration and reservation, the attribute size (advertise) is very large and extended the MAP mechanism and introduced four packed events exclusive for MSRP</td>
<td>MSP (“RSVP like”)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(“MSP is a separate transport-protocol” for e.g. stream reservation)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ MSPP (Multiple Stream Signaling Protocol) is an application for MSP which is used for stream reservation, e2e signalling and diagnostic. The context, which is required for forwarding the signal / reservation, is either built by MRP or ISIS-PCR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ Optional support for higher layers like IP (e.g. transport higher layer addresses, QoS specifier, ..) by e.g. using TLV's</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+ Managed Objects</td>
</tr>
</tbody>
</table>

**MRP v2** "transport-protocol" for applications like MVRP, MMRP, MSRP, ...

**MSP** ("RSVP like")

**Support for more streams. The current worst case limit is less than 500 streams; there are use cases that require two orders of magnitude greater than this.**

**Support for higher layer streaming sessions, such as Real-Time Protocol (RTP)-based sessions.**

**Deterministic stream reservation convergence -> request for performance.**
Data model for splitting the existing MSRP to MSRP on MRP++ and MSSP on MSP

<table>
<thead>
<tr>
<th>New</th>
<th>Static Information</th>
<th>Dynamic Information</th>
</tr>
</thead>
</table>

### Table 1: Talker Advertise vs Talker Failed vs Listener vs Domain

<table>
<thead>
<tr>
<th>Domain</th>
<th>StreamID</th>
<th>StreamClassID</th>
<th>StreamClassPriority</th>
<th>StreamClassVid</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRPP</td>
<td>StreamID</td>
<td>StreamID</td>
<td>StreamClassID</td>
<td>StreamClassPriority</td>
</tr>
<tr>
<td></td>
<td>Talker Sys-ID</td>
<td>Unique-ID</td>
<td>Unique-ID</td>
<td>Unique-ID</td>
</tr>
<tr>
<td></td>
<td>Dest-Address</td>
<td>VID</td>
<td>VID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MaxFrameSize</td>
<td>MaxInterval</td>
<td>MaxInterval</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MaxInterval</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DataFramePriority</td>
<td>Rank</td>
<td>Rank</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DataFramePriority</td>
<td>Rank</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AccumulatedLatency</td>
<td>portTxMaxLatency</td>
<td>AccumulatedLatency</td>
<td>portTxMaxLatency</td>
</tr>
<tr>
<td></td>
<td>FailureInformation</td>
<td>BridgeID</td>
<td>FailureCode</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: MSRPv2 on MRP++ vs MSSP on MSP

<table>
<thead>
<tr>
<th>Domain</th>
<th>StreamID</th>
<th>StreamClassID</th>
<th>StreamClassPriority</th>
<th>StreamClassVid</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRPP</td>
<td>StreamID</td>
<td>StreamID</td>
<td>StreamClassID</td>
<td>StreamClassPriority</td>
</tr>
<tr>
<td></td>
<td>Talker Sys-ID</td>
<td>Unique-ID</td>
<td>Unique-ID</td>
<td>Unique-ID</td>
</tr>
<tr>
<td></td>
<td>Dest-Address</td>
<td>VID</td>
<td>VID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MaxFrameSize</td>
<td>MaxInterval</td>
<td>MaxInterval</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MaxInterval</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DataFramePriority</td>
<td>Rank</td>
<td>Rank</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DataFramePriority</td>
<td>Rank</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AccumulatedLatency</td>
<td>portTxMinLatency</td>
<td>AccumulatedLatency</td>
<td>portTxMaxLatency</td>
</tr>
<tr>
<td></td>
<td>RequiredLatency</td>
<td>portRxMinLatency</td>
<td>portRxMaxLatency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>List&lt;FailureInformation&gt;</td>
<td>BridgeID</td>
<td>State</td>
<td>AccumulatedRspec</td>
</tr>
<tr>
<td></td>
<td>FailureCode</td>
<td>BridgeID</td>
<td>FailureCode</td>
<td>List&lt;FailureInformation&gt;</td>
</tr>
<tr>
<td></td>
<td>FailureCode</td>
<td>BridgeID</td>
<td>FailureCode</td>
<td>List&lt;FailureInformation&gt;</td>
</tr>
</tbody>
</table>
MRP++ Frame Format

Frame:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>Expected Length (in Bytes)</td>
<td></td>
</tr>
<tr>
<td>Application List</td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td></td>
</tr>
<tr>
<td>Application-ID</td>
<td></td>
</tr>
<tr>
<td>Length (in Bytes)</td>
<td></td>
</tr>
<tr>
<td>ApplicationInstance</td>
<td></td>
</tr>
<tr>
<td>Instance-ID</td>
<td></td>
</tr>
<tr>
<td>Length (in Bytes)</td>
<td></td>
</tr>
<tr>
<td>SortedAttributeList</td>
<td></td>
</tr>
<tr>
<td>List count (Number of Elements in the List)</td>
<td></td>
</tr>
<tr>
<td>AttributeSize (in Byte) + Status-Size (in Byte)</td>
<td></td>
</tr>
<tr>
<td>Checksum over Attribute Values</td>
<td></td>
</tr>
</tbody>
</table>

Fragment:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected Length in Bytes (= Rest)</td>
<td>REST OF FRAME</td>
</tr>
</tbody>
</table>