AMI 1.1 High Level Design

, Onramp Wireless, Inc.

	Revision History	
Revision 0.1	12-17-2012	NPF
	Initial draft.	
	Revision History	
Revision 0.2	7-15-2012	AJ
	1.1 adjustments.	
	Revision History	
Revision 1.0	1-08-2014	AJ
	1.1 working HLD doc	
	Revision History	
Revision 1.1.4	1-24-2014	AJ
	1.1 working HLD doc	
	Revision History	
Revision 1.1.5	3-14-2014	AJ
	1.1 HLD doc	
	Revision History	
Revision 1.2.0	8-11-2014	AJ
	1.2 working HLD doc	

Abstract

The high-level design for Onramp's AMI 1.1 system.

Table of Contents

References

[1] Prithpal Khajuria. *AMI 1.1 PRD*. On-Ramp Wireless, Inc. June, 2013.

[2] Avinash Jain. *AMI 1.x Product Plan*. On-Ramp Wireless, Inc. June'30, 2013.

[3] Michael Hall, William Ashley. *OTR GW XML Interface*. https://omega.onramp.local/projects/OnRampTrac/browser/sw/commsys/trunk/ulp_data_inf/xml_schema/ulp_data_inf.xsd. On-Ramp Wireless, Inc. Updates with each CCS release.

[4] AMI Development Team. *AMI 1.1. OTA Specification*. On-Ramp Wireless, Inc. Ongoing, 2013.

[5] HES/OTV SW Team. *HES/OTV SW Architecture*. https://omega/projects/OnRampTrac/wiki/HESInfMon. On-Ramp Wireless, Inc. Ongoing, 2013.

[6] Nathan French. *OTA IDL DOC: AMI ami_ota_if.proto*. On-Ramp Wireless, Inc. 1 June 2013.

[7] Google Developers. *Google Protocol Buffers*. https://developers.google.com/protocol-buffers/. Google, Inc. June'30, 2013.

[8] IEC Standards Committee. *IEC Common Information Model (CIM) International Standard 61968-9*. IEC.

[9] Multispeak Initiative. *Multispeak Version 3.0 Specification*. http://www.multispeak.org. NRECA. Feb'01, 2007.

Glossary

Acronyms

AP	Access Point
CCS	Core Communication System: comprising of nodes, APs and GW
EMCM	Electric Meter Communication Module: Translates meter data for delivery over OTR
HES	Head End System: OTR network application server that interprets application data to/from various application platforms and translates for communication with third-party vendors (such as IEC-CIM/ Multispeak for communication with MDMS/CIS in AMI)
STn	Standard Table n: Refers to ANSI C12.19 organization of meter data. Standard Table has a well defined data structure with fields defined by C12.19. 'n' denotes table number.
SPn	Standard Procedure n: Refers to procedure number 'n' with functional impact to the meter well defined by ANSI C12.19. Standard procedure is a write to ST7 at right place with correct parameters.
MTn	Manufacturing Table n: In ANSI C12.19 organization of data, MTn refers to table n whose data structure and field interpretation is defined by the meter vendor. The data in this structure has no commonality between different meter types.
MPn	Manufacturing Procedure n: An ANSI C12.19 procedure whose functional impact to the meter is defined by the meter vendor.
MTP	Message Transfer Protocol: Also called uStream protocol is a transport layer protocol between HES and application platforms such as EMCM that provides end-to-end reliability, flow control, QoS and large file transfers for data transmitted over OTR network

OTA	Over-the-air: A generic term to represent state of transmission between modules and network components
OTR	On-Ramp Total Reach Network includes CCS, HES, OTV UI/web interface and various AMI and DA application platforms
OTV	On-Ramp Total View: A web portal to view data and events received from various applications and user communication with the remote devices

1. Introduction

This document contains high level design for the AMI 1.1 portion of Onramp's TotalReach (OTR) 1.1 system. The design is intended to address high-level requirements that are described in next subsection.

1.1. High-level Requirements

As a AMI system, OTR shall support a wide range of features that enables a utility to efficiently and effectively use capabilities offered by smart meters. It shall provide interface support for integration with 3rd party (O)MDMS. Further, OTV via REST/UI shall provide range of functionalities for monitoring and managing the AMI network that may not be enabled via 3rd party components. Requirements are divided into these three categories. Most of these requirements are derived from AMI 1.1 product requirement[1][2].

1.1.1. Meter-specific Features

1. Scheduled Billing Data Report where scheduling and reporting periods are configurable: AMI system shall provide ways to read billing data with two different scheduling and reporting periods; First schedule shall be capable of reading all data (summations/demands/coincients) from all tiers supported by the meter while second schedule (faster rate) shall be capable of reading and reporting summations from total tier (Note that second schedule is for meters deployed in energy only mode such as I210+ and kv2c/I210+c demand only mode)

   AMI 1.1.5 discrepancy: Only one billing data schedule is supported.

2. Scheduled Load Profile Reads if load profile is enabled: AMI system shall support all four LP sets (corresponding to ST64-ST67) if meter supports more than one LP set

   AMI 1.1.5 discrepancy: Only one LP set is supported.

3. On-demand Billing Data Read allowing reads of few billing quantities: Objective is to achieve a typical delay no greater than a minute for on-demand read of a single billing quantity irrespective of the meter model

4. On-demand Load Profile Read

   AMI 1.1.5 discrepancy: not supported.

5. Last GASP support: Power outage indication after power fail given supercap half life constraint of ~17J; Power outage detection shall discriminate between sustained and momentary outage reporting only sustained outages as last gasp; Detection algorithm should minimize probability of false alarm (momentary outage aka blinks confused with sustained outage) and probability of missed detection (real sustained power outages not reported)

6. Remote Connect/Disconnect (aka relay operation) with application timeouts

   a. Meter may support main relay and other auxiliary relays (for load control). There should be provision to switch the status of all relays

      AMI 1.1.5 discrepancy: only main relay is supported.

   b. Arm for connect shall be supported if meter has pushbutton for supporting arm to connect

      AMI 1.1.5 discrepancy: unsupported.

7. Reporting and managing all meter status indicators (alarms and notifications)

8. TOU/RTP support:

   a. Reporting of tier billing data in case where meter supports TOU tiers

   b. Supporting price signals as Season Change (in TOU mode) Or RTP/CPP on/off

      AMI 1.1.5 discrepancy: unsupported.

   c. Tier switch time changes via calander update (part of meter program upgrade: AMI 1.2)

      AMI 1.1.5 discrepancy: unsupported.

9. Demand Reset

   a. Support of user/interface triggered demand reset

   b. Automated demand reset after midnight read

      AMI 1.1.5 discrepancy: unsupported.

10. Remote EMCM configuration update that includes various schedules

11. Remote EMCM firmware upgrade

12. Power Quality reporting

13. Generic C1219 Operation Request Response

    AMI 1.1.5 discrepancy: 256 bytes max per queued table writes.

1.1.2. Interface requirements on HES for interfacing with MDMS

1. Support of following CIM queries on HTTP/SOAP

   a. Complete or partial read of billing quantities

   b. Load Profile request within a time interval

   c. Meter connect/disconnect/arm for close (aka connect with manual acknowledgment) request for all relays

d. Meter (power-switch) state request

e. Power quality request

f. Demand reset request

g. Season change request/ RTP change request

   AMI 1.1.5 discrepancy: unsupported.

h. Communication status (aka Ping) request

2. Asynchronous CIM reports

   a. All meter billing data that consists of summations, demands, coincidents, cumulative demands for tier and non-tier rates

   b. All load profile quantities and autonomous power quality reporting of sag and swells

   c. Different meter errors and cautions

1.1.3. Capacity/Performance Requirements

1. EMCM shall only report tier data corresponding to the tiers that are in use during the season

   AMI 1.1.5 discrepancy: all tiers always reported.

2. Newer data shall have priority over older/stale data that has not been sent

   AMI 1.1.5 discrepancy: LP scheme is still first in first out.

3. EMCM shall not queue periodic data for transmission at a rate faster than twice the normal configured rate: This is to ensure that backfilling by older data does not clog the OTA bandwidth.

4. Latency requirements:

   a. Checking meter connectivity should have a turn around time of less than 20 seconds: OTV shall provide UI hooks for OTR ping (infoNodeEchoRequest/Response as defined in [3])

   b. System shall support on-demand read that can have a typical latency of less than a minute irrespective of the meter and its configuration.

5. AMI 1.1 shall have capacity to support atleast 14000 meters/AP with following configuration (This ensures that a deployment with 7000 meters/AP has half the operational bandwidth left for non-regular data and other DA devices):

   AMI 1.1.5 warning: steady state capacity needs to be verified.

   a. Billing data with 10 or less channels (summations/demands/coincidents) and 2 tiers reported once per midnight

   b. Load Profile with 10 or less channels reported every 15 minutes OR (for I210+) a second billing data report of 5 or less channels (no tiers) reported every 15 minutes

1.1.4. Monitoring/AMI Management Requirements (via OTV)

1. Addition and removal of end-devices via marriage file import
2. Meter reading display and export of different reading quantities at a defined schedule
3. Configuration update of HES/OTV (local) and EMCM (note that meter configuration update is not supported in AMI 1.1 and will be available in future releases)
4. Group formation via flat/csv file
5. End device alarm management and notification reporting

2. System Architecture

2.1. Components

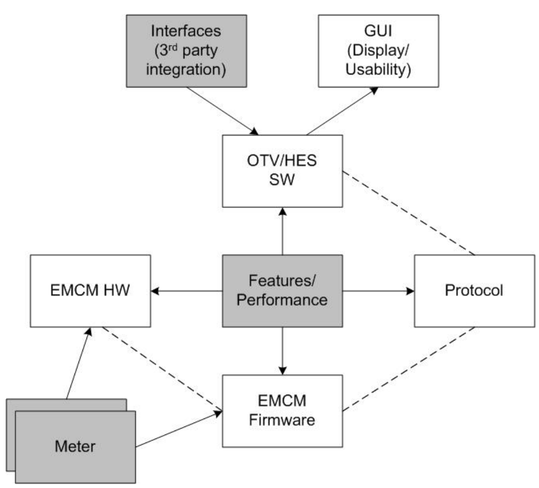
AMI system includes *core-communication system (CCS)* that is shared by different applications within OTR eco-system. CCS includes node modules that provides radio interface via network components such as Access Points (APs) and Gateway(GW). Element Management System (EMS) over SNMP provides network management functionality. CCS is not the focus of this documentation and is described in [1].

Node module is embedded in EMCM (Electric Meter Communication Module) that provide application interface to electric meter and translates meter data into OTA format. GW interfaces with Head End System (HES) which is OTR's application server. HES along with web portal for data display and web interface is also called OTV. HES and EMCM communicate using a propertiery OTA AMI protocol. In this document, we will focus on OTA AMI protocol and its relationship to the translation services provided by EMCM and HES/OTV.

In addition OTV will provide user interface for various monitoring and system management functionalities. Monitoring functions include alarms/notifications from end-devices and monitoring the general health of OTR system. Management functionalities include meter/EMCM/HES configuration management. Network management functionality rests with CCS and is provided by EMS.

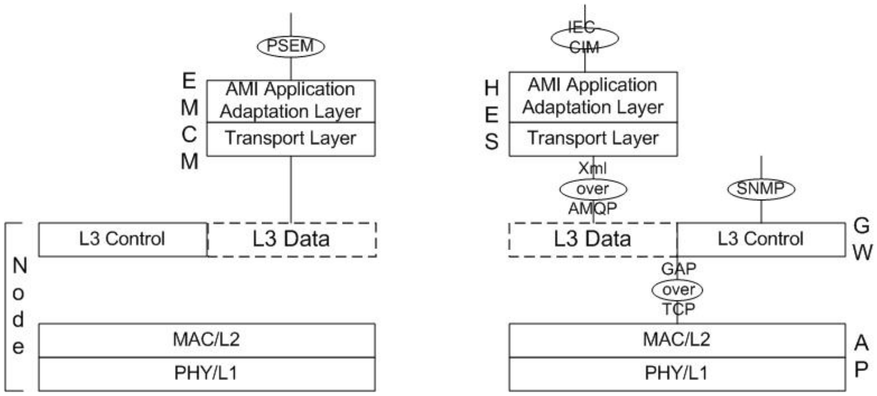
AMI protocol and translatation services provided are function of meters, features, and interfaces supported. Following figure illustrates logical relationship between various components of AMI system (excluding CCS)

Figure 1. AMI system logical relationship: Dotted lines indicate coupling



2.2. Protocol Stack

Figure 2. AMI Protocol Stack



Translation, adaptation and OTA transport are the various functionalities provided by both EMCM and HES/OTV. They correspond to different SW blocks that are described in detail in respective SW design documents ([4] for EMCM and [5] for HES/OTV). Figure above shows the logical protocol layers responsible for various functionalities.

MTP aka uStream is the transport protocol used by the transport layer. Adaptation layer encapsulates meter data or meter/EMCM requests/commands into OTA messages that are described in a .proto and uses a google protobuf encoder and decoder. On HES-side, translation to/from IEC-CIM (and Multipseak) is performed before interfacing with 3rd party MDMS over HTTP/SOAP. On EMCM side, translation to/from meter data in C12.19 tabular format is performed before communicating with meter over C12.18 PSEM.

2.3. Interfaces

2.3.1. Internal interfaces

Interface between different SW blocks in EMCM and HES/OTV are described in detail in respective SW documents. We will briefly discuss AMI adaptation layer interface to transport layer (aka uStream) to provide robustness of the data transfer, end-to-end reliability, QoS.

2.3.1.1. Using MTP: Interface between adaptation and transport layer

MTP aka uStream provides stream mapping which when coupled with memory management can provide appropriate QoS and reliability. Following different levels of priorities are desirable (in decreasing order) on uplink:

1. BootInd and responses to different requests (except large gap filling requests) will be rendered highest priority and will pre-empt any low priority autonomous transmission from EMCM.

2. Billing data, changes to meter cautions/errors and relay state (connect/disconnected) indication are considered next higher priority.

3. Load Profile Indication

4. Other autonomous indications such as Voltage events, EMCM events, EMCM diagnostics (such as panic block)

5. Large gap filling requests

On the downlink, control plane messages (messages generated by HES automatically and without an external trigger such as MeterConfigReportReq and SetDataOrderReportReq to keep in sync with EMCM) shall be mapped as higher priority over data plane requests such as OnDemandReads which will have higher priority over file downloads.

Exact stream mapping and memory handling is defined in [4].

2.3.1.1.1. Stream mapping on uplink

Table 1. UL streams

Stream ID	Resumable	Reliable	Usage	Min Memory size	Notes
3	no	yes	High priority responses to DL requests	1 kByte	**BootInd, MeterConfigReportRsp, BillingDataRsp, RelayOperationRsp, MeterFlagClearRsp**, MeterFlagRsp, DemandResetRsp, PriceSignalChangeRsp, SetConfigRsp, GetConfigValRsp, LineSideDiagnosticRsp
5	no	yes	Low frequency periodic reports	2 kBytes	**BillingDataInd, RelayStateInd, MeterFlagInd**
7	no	yes	High frequency periodic reports	0.5 kBytes	**LoadProfileInd, BillingDataInd (order2)**, LoadProfileRsp, PowerQualityEventInd, PowerQualityReportRsp
8	no	yes	Low priority autonomous reports	0.5 kByte	AppEventsInd, MeterCommErrInd, PanicBlockRsp, ReadMeterTableRsp, WriteMeterTableRsp
10	yes	yes	Large transfers: log retrieval, meter config changes	50 kByte	GenericC1219OpRsp,

2.3.1.1.2. Stream mapping on DL

Table 2. DL streams

Stream ID	Resumable	Reliable	Usage	Notes
3	no	yes	Control plane requests	MeterConfigReportReq, SetDataOrderReq
5	no	yes	Data plane requests	DemandResetReq, SeasonChangeReq, RelayOperationReq,

Stream ID	Resumable	Reliable	Usage	Notes
				BillingDataReq, LoadProfileReq, PowerQualityReportReq, FlagsReq, MeterFlagClearReq, MeterInfoReq, PanicBlockReq
10	yes	yes	Generic request and image transfers	GenericC1219OpReq, EMCM firmware, meter firmware, etc.

2.3.1.1.3. Enqueued Data Buffer Management on EMCM

Data between application and transport layers is stored in queues and exchanged via API call flows. MTPbuf is a wrapper on top of uStream to manage uplink (UL) data RAM between application and transport layer. MTPbuf minimum memory requirements are dictated by uStream (8*460 bytes for UL). EMCM application layer directly interfaces with uStream for messages queued on DL transport buffer. Following general rules are recommended:

1. On UL, MTPbuf shall have enough memory available to queue largest size message on highest priority streams 3/5 and largest size message on each of the other non-resumable streams 7 and 8.
2. UL streams 3 and 5 are highest priority streams and message on those streams can be queued anytime into USPbuf as long as there is enough total memory available.
3. *Backpressure*: For other non-resumable UL streams 7 and 8, EMCM should queue a message only if following two conditions are satisfied:
    a. MTPbuf has sufficient free memory available to queue a new largest size message on Stream 3 and 5 (most likely Billing data with all tiers).
    b. Query MTPbuf with GetWaitingMsgCountByStream(streamId x). Only if a there is no message waiting, i.e. count returned is equal to 0, should EMCM enque a message for that stream. WaitingMsgCount of non-zero value indicates backpressure to the application.
4. *Flow Control*: uStream DL transport buffer contains request messages (or fragments thereof) that were queued from HES. When uStream (via USP_RxMsgFrag()) indicates that a fragmentation of a message is available for processing, EMCM app shall query MTPbuf with GetWaitingMsgCountByStream(streamId 3).
    a. If the returned count is equal to 0, EMCM shall process the received DL fragment from uStream and indicate to uStream that the fragment was successfully recieved.
    b. If the returned count is equal to 1, EMCM should not retrieve the fragment and let uStream DL transport buffer hold the fragment. This is done to ensure that there is sufficient backpressure on DL requests and back-to-back DL requests are not dropped due to insufficient memory available on UL.

2.3.1.1.4. Handling uStream (enqueue) Failures

Even though uStream guarantees reliabile transmission, failures to deliver can happen due to various reasons such as unexected uStream sync and/or failure to enqueue a message due to queue size limitation.

Note: With multicast (on non-resumable streams), there is a possibility of failure to deliver a message for some nodes, which is in essence denotes failure to enqueue a message for those nodes.

2.3.1.1.4.1. At HES:

If HES adaptation layer is unable to queue (or uStream returns fail indication on) the following control plane messages, then it shall immediately resync the uStream:

1. MeterConfigReportReq
2. SetDataOrderReq

The two messages above are critical to synchronization between HES and EMCM. A failure shall result in resynchronization. HES shall also ensure that changes resulting from MeterConfigReportRsp are not dropped after it is acknowledged by uStream.

On the other, failure to deliver any other data plane requests shall immediately be returned as fail to the end-user. HES shall not make any automatic attempts to resend the data.

After a DL message is queued to uStream, uStream may respond with two different kinds of confirmations: a) *downlinkMessageComplete* will always be generated and will indicate whether EMCM app has received the message (transport layer confirmation) and b) *handleDownlinkMessageAcked* will indicate that all fragments of the message were successfully received by the node (MAC layer confirmation). Failure to deliver a message is determined from "downlinkMessageComplete". However for quick response HES shall generate a successful acknowledgment on receiving downlinkMessageDelivered after following requests:

1. RelayOperationReq

2.3.1.1.4.2. At EMCM:

Depending on the message class, different rules apply to EMCM on failure to enqueue (failure criteria described before):

1. If EMCM adaptation layer is unable to queue (or uStream returns fail indication) the following highest priority control plane response messages, then it shall immediately resync the uStream:
   a. BootInd
   b. MeterConfigReportRsp
2. There are some messages in which EMCM sends differential state information. EMCM polls the meter at regular intervals and if the state has changed, it sends an UL message. EMCM shall change its knowledge of the state information only when EMCM has successfully enqueued this "differential state indicator" to the MTPbuf, failing which the message can be dropped but state information remains unchanged. Following messages as currently defined fall under this category of differential state indicators:
   a. MeterFlagInd
   b. RelayStateInd
3. For all other messages mapped to non-resumable streams, failure to enqueue shall result in message being dropped. A count and time of drop can be maintained for diagnostic purposes.

2.3.2. External Interfaces

2.3.2.1. EMCM to Node

The EMCM uses the Host Common library to interact with the Node. MTP is an add-on functionality provided by the library and the interface to that is discussed in Section 2.3.1.1, “Using MTP: Interface between adaptation and transport layer". For power outage indication aka Last Gasp, EMCM bypasses MTP and uses the messaging provided directly by host common library. Last Gasp design is disccussed later.

2.3.2.2. EMCM to Meter

The EMCM communicates with a Meter via ANSI C12.19 table/procedure interactions over an ANSI C12.18 (PSEM) port. Various schedules and triggers that lead EMCM to interact with meter are discussed later. In addition, EMCM may monitor other lines to determine power outage as discussed later. Meter-centric details are described in EMCM SW design document.

2.3.2.3. HES to 3rd party MDM via CIM/Multispeak

HES architecture is described in [5]. Interaction to the 3rd party MDMS/CIS/OMS/EA are provided over this interface using either IEC-CIM [8] or Multispeak [9] XML format. Queries on this interface that shall be supported by HES are specified by the high-level requirements described above.

2.3.2.3.1. Supporting CIM requests/asynchronous reports

Detailed schema for meter readings are described in clause H of CIM spec [8]. In this document, we list, how the various payloads are populated. Following payloads are supported:

1. MeterReadings: Detailed schema is in Clause H.6 of [8]. Timestamp, ReadingType and Value elements are required in responses/asynchronous indications. Different asynchronous indications from EMCM containing meter readings will be mapped to valid ReadingType CIM codes using "iec-cim-http.properties" txt file. CIM based GetMeterReading request is supported as below:

    a. **TimeSchedule element included**: Shall be interpreted as load profile read.

    If ReadingType is "0.0.4.0.0.0.2.n.0.0.0 " where n is a number between 1 and 4: This will be interpreted as MDMS is requesting measurement quantities recorded by the meter in load profile set 'n'. *Note that an unknown ReadingType shall result in CIM reply with ReplyCode of 2.6 with request rejected.*

    If ReadingType is not included, then default value of '1' will be assumed for load profile set. In AMI 1.1, there is no plan to support reading of specific load profile quantities which means any ReadingTypes if included can be ignored. OTA this request will translate to GetLoadProfileReq.

    b. **TimeSchedule element not included, ReadingType is "12.0.0.0.2.19.0.0.0.0.0"**: This GET request indicates request for instantaneous/average power quality quantities and will be lead to "PowerQualityReportReq" OTA. Read request for specific quantities is not supported.

    c. **TimeSchedule element not included, (one or more) ReadingTypes may be included**: If no specific ReadingType element is present, it shall be interpreted as request for all measurement quantities recorded in bulk data table ST23. If specific ReadingType(s) is included, those shall be translated to appropriate billing UOMs by the HES. OTA this request will lead to BillingDataReq.

    Note that an unknown ReadingType shall result in CIM reply with ReplyCode of 2.6 with request rejected.

2. EndDeviceControls: Detailed schema is specified in Clause H.4 of [8]. Type element is required. Different Types supported are:

    a. **Connect/Disconnect/Armed for Closure Request**: Connect is supported as type "3.31.6.42.x"; Disconnect is supported as type "3.31.6.68.x" and Armed for Closure (aka Connect with manual acknowledgment is supported as type "3.31.6.11.x". It is mapped OTA to RelayOperationReq with appropriate switch state. 'x' here is a number that may or may not be present. If x=0 or x is not present in the request, then operation on main relay=0 is being requested. Otherwise relay 'x' (auxiliary relays) is being operated on.

Disconnect command in addition to type, may contain endTime tag which will be mapped to corresponding TTL(time-to-live) OTA. The TTL tag will indicate to EMCM that it shouldn't initiate disconnect after the time on the message.

b. **Demand Reset**: Demand Reset command is supported as type "3.8.6.61" and is mapped OTA to DemandResetReq.

   Demand Reset command in addition to type, may contain startTime tag which will be mapped to corresponding startTime OTA. The startTime will indicate to EMCM that it shouldn't initiate demand reset procedure prior to that time.

c. **Season Change**: Season Change command is supported as type "2.36.6.228.x", where x denotes the season. It is mapped OTA to SeasonChangeReq.

   Season Change command in addition to type, may contain startTime tag which will be mapped to corresponding startTime OTA. The startTime will indicate to EMCM that it shouldn't initiate Season change procedure prior to that time.

3. <u>EndDeviceEvents</u>: EndDeviceEvents are reported as a result of three different triggers: a)autonomous events reported by EMCM; b)events resulting as a result of EndDeviceControl commands; and c)events resulting as a result of EndDeviceStatus requests. Detailed schema is in Clause H.5 of [8]. Timestamp and Category elements are required. Category elements are populated as specified below:

a. *Autonomous Events*:

   i. **Meters Flags**: Mapping for various quantities reported in MeterFlagInd to CIM codes is provided in iec-cim-http.properties file.

   ii. **Sustained Power Outage**: On received sustained meter power outage indication, HES shall generated a power outage event (category code: 6.20.1.185) and on determining power outage restored, HES shall generate clearing of the event (category code: 6.20.9.185).

   iii. **Momentary Power Outage**: EMCM may determine that power fail observed from the meter is not sustained outage but instead a momentary (also called blink). It may report number of blinks and timestamp and duration of each blink if it can correctly determine in BlinkReport (TBD) message. Timestamp of the first blink will determine occurance of momentary outage. Following will be populated for upstream consumption: Momentary Outage (6.26.13.165) and Number of Momentary Outages (6.26.13.166.n) where n indicates the number reported by EMCM. (Question: should we just specify them in properties file using .proto tags?).

   iv. **Power Quality Events**: Generated from powerQualityEventInd (aka GEVoltageEventInd). Message includes eventType which will indicate the event and relevant information pertaining to the event. iec-cim-http.properties filewill contain the category code of these events (such as powerQualityEventInd.sagInfo:6.38.17.223 and VoltageEventInd.swellInfo:6.38.17.248 etc).Note that voltage and current values reported in the information element will lead to them being reported as MeterReadings. However, with these readings HES **should** QualityCode 1.2.0 to indicate the receiving entity to "look for related meter events".

b. *Response to control commands*:

   i. **Connect/Disconnect/Armed for closure response**: OTA message RelayStateOperationRsp will be mapped to appropriate category codes as defined in iec-cim-http.properties file.

ii. **Demand Reset response**: A successful demand reset will be mapped to event with category code ""

iii. **Season Change response**: On receiving a SeasonChangeRsp message OTA, a successsful delivery shall result into event with category code "2.36.16.228.x" where x denotes season.

c. *EndDevice(node+MCM+Meter) status requests*: These kind of requests are supported under the umbrella of GetMeterReadings. However no ReadingType is included but instead an EndDeviceEvents are included with categories defined below.

i. **Communication Status (aka Ping) Request**: Ping is supported as type "1.23.5.18" (Check Network Interface Communication Status. It is mapped OTA to echoRequest over GW interface.A successful echoRsp over GW interface will be mapped to event with category code "1.23.18.244" and an unsuccessful response (or on timeout) will be mapped to an event with category code "1.23.18.85".

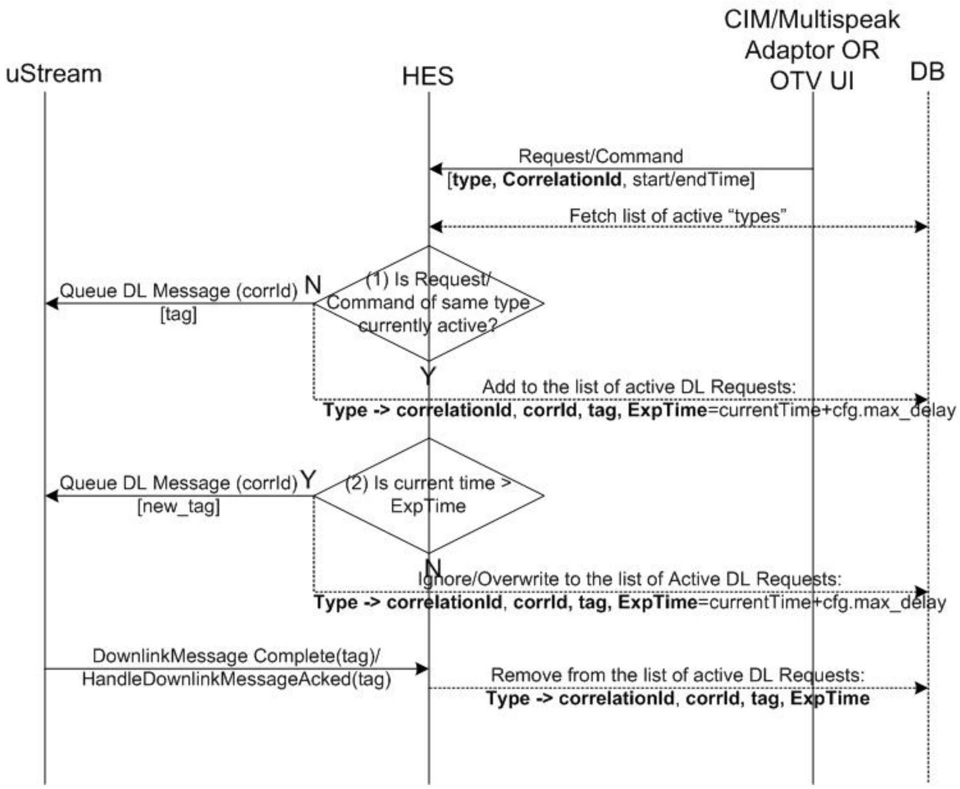
ii. **Power Switch State Request**: Request is supported with cateogry code "3.31.6.11". (*Note that in AMI 1.0 implementation, this is implemented as GetMeterReadings with ReadingType included. We can keep it that way as well as long as we can distinguish it from billing data read of a specific quantity.*)

2.3.2.3.2. Managing data requests

Data requests (as described in previous section) on OTR system are delivered with highest reliability even on highly unrelaible communication links. There are no OTA timeouts enforced on individual messages. Since OTV interfaces with other 3rd party servers and user interfaces with that may trigger multiple requests within small interval, HES shall manage the requests to ensure that multiple requests are not queued OTA to make efficient use of system bandwidth.

Call flow shown in Figure 3, “” describes how HES can throttle multiple requests of the same "type". Type here refers to different requests/commands associated with following OTA messages: "BillingDataReq", "GetLoadProfileReq", "PowerQualityReportReq", "RelayOperationReq", "DemandReset", "SeasonChange" and "PowerSwitchStateReq". For eachEMCM, HES can maintain a list of types that are currently active. cfg.max_delay is a HES configuration parameter (default value 1 hour?) which determines the duration for which HES will throttle a new request of the same "type" if uStream has not acknowleged delivery of the request.

Figure 3.



3. Features

High-level requirements outline all the meter-specific features that shall be supported in AMI 1.1. Next few subsections will give a brief sketch of how these features are supported with call flows and messages that are exchanged OTA. Exact description of these messages is in the IDL[6]. An OTA message may contain many optional variables. The boldfaced variables in illustrative calls flows shall be included in the exchanged messages as they are critical to the support of that feature. Rhombus boxes are used to represent decision-making process either EMCM or HES has to go through which makes call flows partly represent a flowchart.

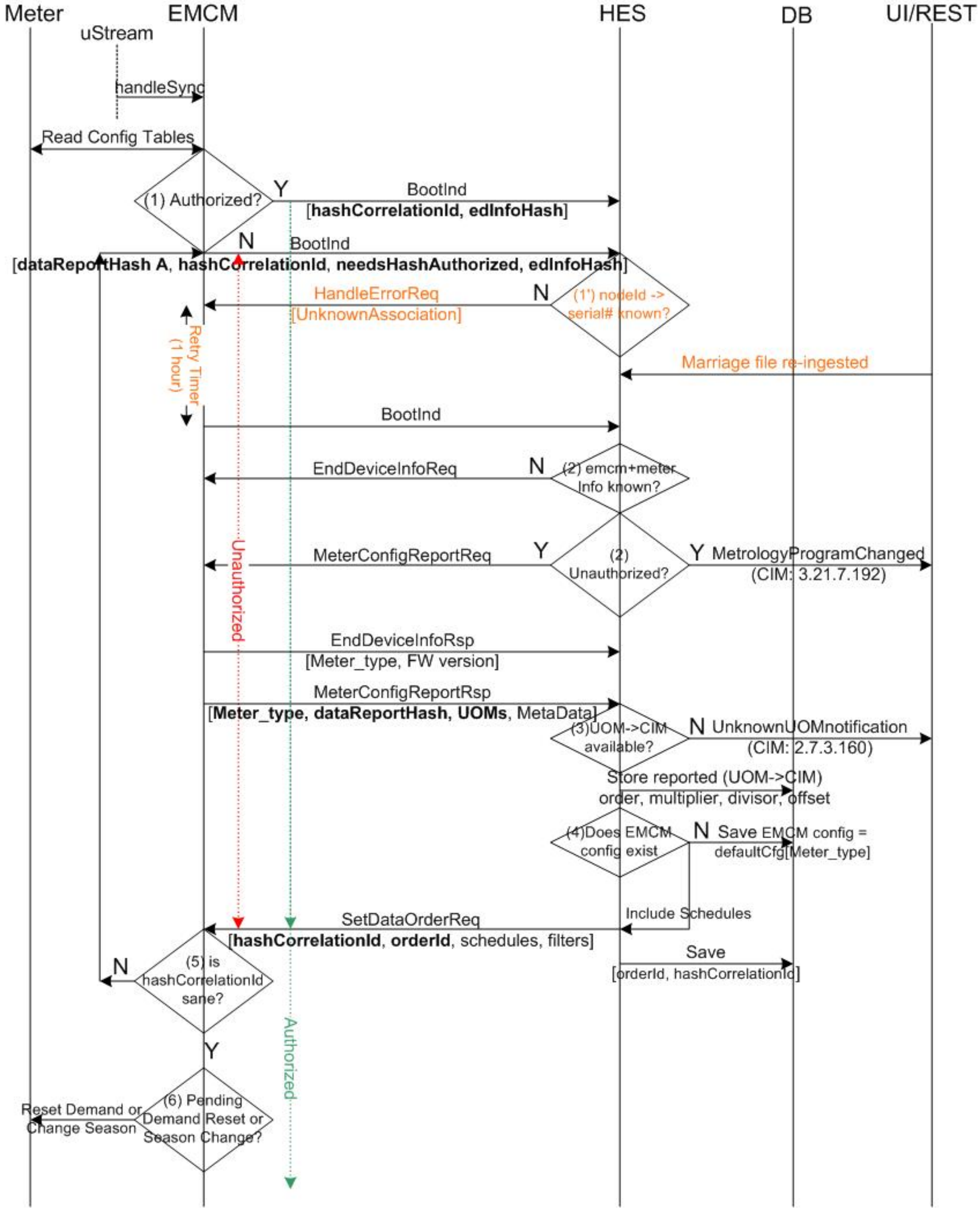
For detailed behavior of the OTA protocol, see AMI OTA Specification[4].

For definitions of the AMI OTA messages, see the IDL ami_ota_if.proto [6].

3.1. Device Boot

BootInd is an OTA message that shall be sent by EMCM whenever there is uStream synchornization. uStream synchronization can happen post EMCM boot (hence the name), triggered by AMI application (at HES or EMCM) due to some internal timeouts or in some rare scenarios may happen due to some unexpected event. Irrespective of the cause of the sync, uStream will generate a callback "handleSync()" to the EMCM app at the time sync is initiated. On receiving handleSync indicating start of resync, EMCM application shall queue BootInd message OTA and bootstapping call flow is executed as shown in Figure 4, “Boot sequence” below.

Figure 4. Boot sequence



Call flow above shows OTA messages that are exchanged with boldfaced variables critical to the bootstrapping sequence. Decision-making process at HES/EMCM is described in detail below:

1. Authorized and Association known?: "Authorized" check on EMCM side is required to ensure that meter program is valid and HES/OTV have the right program information to correctly parse the meterology related information. For this purpose, EMCM shall maintain a 128-bit hash called

"meterDataReportHash" over meterology configuration in its persistent memory. Further to generate "meterDataReportHash"

a. Meterology configuration is contained in ST0,ST12,ST14,ST15,ST16,ST21,ST22,ST61 and ST62 and meterDataReportHash is computed over the contents of these tables.

b. If the newly computed hash does not match the hash stored in its persistent memory, meter program shall be considered unauthorized and EMCM shall send a BootInd message to the HES with meterDataReportHash (and "hashCorrelationId" which is different from the previous ones, for example: new hashCorrelationId = (prev hashCorrelationId +1)%256) if hashCorrelationId is supported to reduce BootInd overhead). Note that this shall also be sent if EMCM doesn't have valid hash. After this EMCM will remain unauthorized and shall not queue any meterology data till it receives SetDataOrderReq with same hashCorrelationId.

c. On the other hand if newly computed hash matches the one stored in persistent memory, EMCM must send either hashCorrelationId or meterDataReportHash. hashCorrelationId is a small overhead tag that EMCM increments everytime hash changes. In this scenario, large (128-bit) meterDataReportHash is not required and replacing it decreases the OTA overhead of BootInd message in majority of the cases. EMCM will consider itself authorized and move on to step 5.

In addition to meterDataReportHash, EMCM shall also compute a small 32-bit hash called endDeviceInforHash over meter and EMCM information such as FW and HW version numbers that are reported in EndDeviceInfoRsp. EMCM shall also include endDeviceInfoHash in the BootInd.

"Association" check is done on HES side to ensure that HES can associate an EMCM/meter to valid meter serial number and it knows the meter model and meter type to appropriately service that meter. This association on HES/OTV side is part of marriage file ingestion. Unknown association will lead HES to generate handleErrorReq message sent on DL to EMCM.

To generate "meterDataReportHash", EMCM should be able to read the meter at the time BootInd is being generated. If EMCM has been unsuccessful to generate a valid BootInd message OR it receives from HES handleErrorReq message, EMCM shall generate a retry timer (default value 1 hour) and retry sending BootInd after the end of the timer.

If EMCM is unauthorized and has successfully enqueued BootInd for transmission, EMCM may start a long timeout (> 6 hours) to receive either MeterConfigReportRsp or SetDataOrderReq authorizing the EMCM.

2. Unauthorized or hashCorrelationId unknown or end device info unknown?: This check is on the HES side.

   a. If either "needsHashAuthorized" flag was received or "hashCorrelationId"/"meterDataReportHash" does not match the ones in HES database, then HES shall consider EMCM unauthorized and request a new meter configuration from EMCM and notify the OTV of the configuration change for display over user interface. Note that in first deployment, either of the two conditions should always be true.

   b. If meterDataReportHash does not match what the HES has stored in its database or HES doesn't have the end device information in its database, then HES shall send EndDeviceInfoReq to the EMCM.

   c. If EMCM is authorized, then there is nothing to be done and boot sequence terminates.

3. UOM->CIM map: MeterCfgReportRsp received from EMCM will contain various summation, demand, load profile UOMs (depending on meter mode and functionalities supported by the meter). UOMs in entirety contain 32-bit ANSI UoM and vendor specific NFS information as well as multiplier, divisor and offset.

a. ANSI UoM and NFS will uniquely identify measurement quantity and shall be used by HES to determine corresponding CIM code from iec-cim.http properties file. This map shall be saved by the HES persistently.

b. Multipler, divisor and offset is the information that shall be applied to raw register values that EMCM later reports in periodic messages and hence shall be stored persistently.

4. EMCM config: Creating an EMCM configuration is an OTV UI feature via which new configurations could be defined. If user at any time has selected a certain configuration to be applied, then that configuration shall be chosen for any future authorizations. Otherwise HES shall chose default configuration corresponding to Meter Type. MeterType information is contained in MeterCfgReportRsp (that could also be overwritten using the ingest file).

   HES shall send SetDataOrderReq which contains filters and schedules for different quantities reported in periodic reports. (In AMI 1.1, we intend to keep it simple and not filter quantities that meter can read except for GE I210+ fast periodicity billing data reports.) SetDataOrderReq will be identified by orderId which later will be provided by EMCM in all periodic reports to identify filters applied by EMCM in its report. Hence saving orderId and various filters in persistent memory is critical to HES functionality.

5. IsHashCorrelationId Sane?: SetDataOrderId message sent by HES must includes hashCorrelationId last received from the eMCM otherwise HES is indicating boot time mismatch. If hashCorrelationId is not included or it is not same as what is expected by eMCM, then eMCM shall consider itself unauthroized and restart the synchronization process by sending BootInd.

   If SetDataOrderId is received with matching hashCorrelationId, then eMCM shall consider itself authorized and save the data orderId.

6. Pending demand reset or season change?: If unauthorized, then after receiving SetDataOrderReq from HES, EMCM will store hashCorrelationId and orderID persistently and consider itself authorized. Otherwise it considers itself authorized after queuing BootInd.

   Before EMCM gets back to its periodic routine, EMCM shall check if any user-triggered demand resets or season/RTP changes are pending. This can happen either because of user-triggered demand reset or season/RTP change commands received OTA where "start time" passed the time EMCM was not powered/functional. For pending demand reset and season/RTP change, EMCM shall ensure that they are either executed or timers to execute them are restarted as described in Section 3.5.2, “Backend-triggered Demand Reset” and Section 3.5.3, “Back-end triggered "Critical-Peak/Real-Time Pricing (CPP/RTP)" or "Season" Change”.

3.2. Scheduled Metrology Reads

Scheduled metrology reads correspond to meter metrology table reads that EMCM performs on a schedule. This includes ST23 (or ST26) read for billing data and ST64-67 read for load profile data for ANSI compliant meters. Schedules and order filters are provided by HES in SetDataOrderReq. Two schedules for billing data read are supported and one schedule for each of the load profile set supported by the meter.

Billing data and demand resets can be supported in following different ways (for ANSI compliant meters other than GE I210+):

1. *Meter does not perform automatic demand reset*: In this case, EMCM shall read the data from current register table ST23 and OTV EMCM configuration provides a way to enable demand reset either daily or monthly executed as a procedure by EMCM.

   Meter may however perform automatic self read. Self-reads provide a way for previous data to be manually read over the optical port when EMCM communication to the backend over OTR is broken.

2. *Meter is configured to do automatic demand reset and self read on demand reset*(Support has been postponed to AMI 1.2): Note that this option is not available in "demand-only" mode due to lack of time support on the meter. However when meter supports time, this option is available and proper configuration as described will ensure self-read table ST26 is used appropriately:

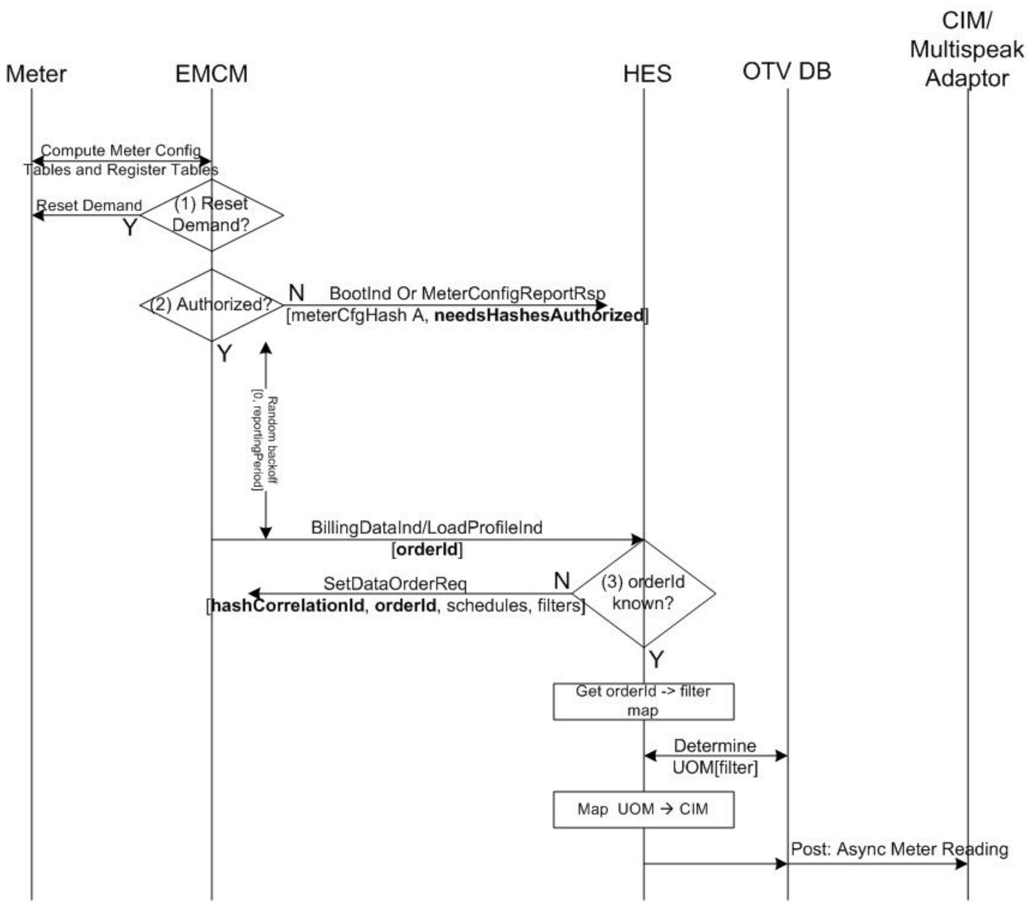
    a. Calendar table ST54 is programmed to support automatic demand reset on the meter. Further self-read on demand reset shall be enabled.

    b. EMCM shall not be configured for automatic demand reset. OTV UI will provide a way to disable it.

    c. EMCM shall be configured to read self read table ST26. OTV UI shall provide a way to enable EMCM configuration parameter "useSelfReadData". When useSelfReadData is set, EMCM shall read self-read billing data from ST26. Instead if useSelfReadData is not set, EMCM shall read data billing data from ST23.(*Note that current register table ST23 is not recommended due to race condition in which automatic demand reset by the meter may happen prior to EMCM read making demand quantity read from ST23 meaningless. Therefore it is important that useSelfReadData is enabled if meter is configured to do automatic demand resets and self reads*). Further EMCM shall avoid reading ST26 at midnight but should few minutes of delay before ST 26 is parsed to allow meter the time to store the data in self-read table.

    AMI 1.1.5 discrepancy: one one billing order and only ST23 reads supported.

Figure 5.



Call flow above shows triggers, checks and translations that ensure billing/load profile data read from meter is reported for consumption. At the pre-defined schedule, EMCM shall read all configuration tables necessary for computation of meterCfgHash and appropriate register tables for billing/load profile report. Decision-making process at HES/EMCM is described in detail below:

1. Reset Demand?:If scheduled read corresponds to billing data order 1 (NOT APPLICABLE to order 2), then billingDataOrder may include either the flag "doDailyDemandReset" or "doMonthlyDemandReset" to indicate if EMCM is expected to automatically reset demand at first read of every day or every month (if offset is 0, then it corresponds to midnight read). If EMCM is expected to do automatic demand reset and if local time corresponds to time of the automatic demand reset, EMCM shall reset demand in the meter.

   AMI 1.1.5 discrepancy: scheduled demand resets unsupported.

   Note that when doMonthlyDemandReset received in data orders is set, "monthlyDemandResetOffsetInDays" received in data orders will indicate number of days from the start of month at which demand reset shall be executed. If monthlyDemandResetOffsetInDays is not received, then first day of the start of month shall be used for demand reset.

   To ensure that demand was reset, EMCM should read ST8 to retrieve RESULT_CODE corresponding to SP9 and return "demandResetResultCode" in BillingDataInd.

2. Authorized?:Before queuing billing data read for transmission, EMCM shall verify meterCfgHash over the meter tables specified in Section 3.1, “Device Boot”. If configuration has changed, then EMCM will consider itself unauthorized, throw away the reading and enter Boot process as shown in Figure 4, “Boot sequence”. If however EMCM is authorized, EMCM shall send either BillinDataInd or LoadProfileInd message containing the contents of the register data. orderId (and loadprofileOrderIdx in case of load profile) will uniquely specify quantities being reported

3. Known orderId?: If orderId is unknown to HES, then HES shall throw away the data and send schedules and filters under current orderId in OTA message setDataOrderReq.

   If orderId is known, HES shall determine from the context and the filter(orderId), applicable UOM and CIM and constants (multipliers, divisors and scalars) to each of the raw register values. Processed data is then computed as multipler*raw_data/divisor + offset and posted as MeterReadings with MeterReadingType given by the CIM code. HES shall store both the processed and raw data for audit trail purposes (90 days?).

   If demandResetResultCode is included, then HES shall generate CIM events indicates either successful or unsuccessful demand reset execution.

3.2.1. TOU/RTP support

A meter in TOU mode (dependent on the configuration) has ability to support data in multiple tiers based on tier switches programmed into the meter. Some GE meters(in demand/demandLP mode) do not have explicit TOU support (no tier switches) but provide an ability to store data in a separate tier based on explicit command that enables storage in this tier. GE meters call this as "RTP" tier, however AMI system will call this with generic term "tier A".

3.2.1.1. Tier Data Report

Tier switches programmed into calendar table ST54 determines the tiers in which billing data is accumulated. For capacity reasons, it is critical that only the tier data that has been updated is reported OTA. EMCM determines which of the tiers have been accumulated in either of the following two ways:

AMI 1.1.5 discrepancy: all tiers always sent.

1. Option 1: Meter Agnostic way:

   a. EMCM stores in its persistent memory tier data or hash(tier data) for every tier reported by meter at midnight read (Note that even with single flash page, this allows approximately 30 year flash life with 10k flash write cycles).

   b. At every billing data, hash(tier x data) for tier x is compared against hash(tier x) stored in flash. If the tier data has changed, then it shall be queued for transmission in BillingDataInd message. New data (or hash of the new data) is stored in EMCM's persistent memory only after BillingDataInd message has been successfully delivered by the uStream.

2. Option 2: Relying on CURR_TIER in ST55:

   a. For ANSI meters that support TOU and those that have ST55, EMCM can poll for CURR_TIER field every 10-15 minutes. All tiers that are polled since last successful BillingDataInd are then stored in EMCM flash undera "used tiers" structure. To avoid multiple flash writes if tier hasn't changed and same tier is returned polled multiple times, EMCM should avoid a new write.

   b. Whenever BillingDataInd corresponding to orderIndex 1 is succesfully delivered (successful delivery confirmation from uStream), then all tiers polled previously should be erased.

   Slight disadvantage of Option 2 is that it can potentiall miss back-to-back tier updates that may happen in between EMCM polls. However, it is highly unlikely for tier durations to be smaller than 15 minutes.

3.2.1.2. Real time price support based on price signals

RTP or real time pricing (all called CPP aka critical peak pricing) is an AMI feature that can be supported whether the meter is in TOU mode or GE-specific demand/demandLP mode. In TOU mode, different price signals can be mapped to different "seasons" and therefore price signal change corresponds to season change. Calendar table should be programmed so that tier switches within a specific season accumulate data in the desired tier. In GE demand/demandLP mode, price signal will be mapped to enabling/disabling of RTP, i.e. price signal will determine whether the data is accumulated in tier A (aka GE RTP tier) or not.

There are two ways to support season change:

AMI 1.1.5 discrepancy: unsupported.

1. *On-demand season change*: startTime is specified to indicate when the season change is to be executed by EMCM. EMCM shall ensure that if scheduledRead and SeasonChange are triggered at the same time, then scheduledRead is executed prior to SeasonChange.After season change procedure is executed, EMCM shall generate SeasonChangeRsp message with value of the current season indicated in ST55. This is discussed in more detail in Section 3.5.3, “Back-end triggered "Critical-Peak/Real-Time Pricing (CPP/RTP)" or "Season" Change”.

2. *Season Change programmed into calader Table ST54*: Support uncertain/TBD in AMI 1.1. Biggest issue is determining when season triggers happen. This requires EMCM to keep walk through ST54. GE meters do not auto-correct for current season when date is changed. Further a season change pushed OTA doesn't lead to season data being available in previous season table(ST24) making it less useful. This will require that season change is only supported via calendar table ST54 changing which is cumbersome (requires management of old calendar table at HES and pushing large data set OTA to EMCM.

Option 2 decreases usability and is far more complex. Advantage of Option 2 over Option 1 are corner scenarios in which EMCM is unable to set the correct seasons on meter due to connectivity issues (something that can be addressed in EMCM by added complexity).

3.2.2. Virtual Metering

GE I210+ is an energy only meter. Other meters even though support load profile could be configured to be in energy-only mode, *i.e.* report billing data with no support for load profile - a configuration that is called demand-only mode in GE meters.

Energy only metering may require frequent (15 min-1 hour) reporting of billing quantities to enable creation of interval data (aka load profile) as well as tier data at the backend (hence called virtual metering).

Reporting billing data recorded in ST23 frequently can be a problem. This is because the recorded data could be large. This causes buffering issues in eMCM due to RAM limitations (since reporting time is different from read time) and use of flash for storage will cause wear-leveling issues. Further transmissing large amount of data frequently will create OTA capacity bottlenecks. However, large data size comes from demands, coincidents and multiple tiers of data. Reporting of all tiers and demands and coincidents frequently is an overkill and unnecessary as virtual metering probably doesn't even require it. Hence in AMI 1.1:

1. Billing Order 1 will be mapped to read that takes place every 24 hours (at midnight): This is configurable but expected use case is approximately once (or twice) a day. All data will be reported (including tier data that has changed since the previous day as described in next section.

2. Billing Order 2 will be mapped for higher frequency read. Only summations of total tier will be read and reported in Billing Order 2. This will shrink down the buffering/transmission requirements to maximum of 8 summation quantities (8 in SGM3x, 5 in kV2c, 2 in I210+c are the max).

3.2.3. Load Profile Quality Reads and Optimizations

In ANSI C12.19 compliant meters, load profile interval may be qualified by 4-bit "Common Interval Status" flags applicable to all channels in the interval and "Channel Status" indicator specifically applicable to that channel. These statuses qualify the reading and potentially useful to an MDM.

EMCM shall use the following rules when creating an indication or a response message contained load profile data:

1. commonIndStatus is included only if one or more of the flags are set i.e. it is non-zero.

   AMI 1.1.5 discrepancy: unsupported.

2. channelStatus is included if for any of the channel the value is anything other than 0.

   AMI 1.1.5 discrepancy: unsupported.

3. (optimization for capacity reasons) EMCM may drop and not include channelValues if channelStatus for all channels is 4 (skipped interval). (*Note: I fail to see why it will be 4 for one interval not 4 or others, so checking a single channelStatus bitfield should be enough.*)

HES (IEC-CIM adaptor) shall use the following rules when parsing a load profile report:

1. "commonIndStatus" if received with load profile will map to following quality codes for each of the channels reported in that interval:

   - Bit0 set (DST_ FLAG) -> 1.4.16

   - Bit 1 set (POWER_FAIL) -> 1.2.32
   - Bit 2 set (CLOCK_SET_FORWARD) -> 1.4.64
   - Bit 3 set (CLOCK_SET_BACKWARD) -> 1.4.128

2. If channelStatus corresponding to a channelValue is not received, then HES shall assume channelStatus is 0. If the channelStatus is 0 and commonIndStatus is not included, then reading quality is 1.0.0. Otherwise, HES shall attach following quality codes based on received channelStatus value:
   - 0 (no commonIndStatus included) -> 1.0.0
   - 1 (Overflow) -> 1.4.1
   - 2 (Partial Interval) -> 1.4.2
   - 3 (Long Interval) -> 1.4.3
   - 4 (Skipped Interval) -> 1.4.4
   - 5 (Test Data) -> 1.4.5
   - 6 (Configuration Changed) -> 1.4.6
   - 7 (LP recording stopped) -> 1.4.7
3. If no channelValues are recieved but channelStatus indicates value of 4, then HES shall assume channelValus are 0 and generate the corresponds interval reads.

3.2.4. Load Profile Auto-gap Filling

AMI 1.1.5 discrepancy: unsupported.

Most meters store load profile data in their persistent memory so that the data can be fetched from the meter at times connection is available. However as high-level requirements state, it is desirable to have the newest data transmitted first so as to avoid scenarios where a meter is always sending stale data. Stale data is not suitable for any real time analytics which is one of the use cases of load profile. Further backfilling should be slow so as to not overstress the available capacity of the system. Following outlines an algorithm that meets the desired objectives:

1. For each load profile set, EMCM maintains timestamp of last load profile interval data read from the meter. Along with it, EMCM shall maintain a 96 bit bimap that indicates state of past 96 load profile reports. If they are queued for transmission to uStream, they are marked as sent.
2. At every scheduled load profile read (loadprofileReadInterval), EMCM shall read timestamp of the latest data and shift the bitmap marking all since last read as not sent.
3. It shall then queue latest unsent data to uStream (governed by rules described in Section 2.3.1.1.3, “Enqueued Data Buffer Management on EMCM”). No more than 2 x loadProfileReadInterval/MAX_INT_TIME_SETk interval data shall be queued every loadProfileReadInterval and bitmap corresponding to interval data queued shall be marked as sent.

Note that data can still be lost. This happens if there are power outages or if backlog is bigger than 96 reads (~ 1 day for 15 min interval data). A backend gap filling process is expected to take care of data more than a day old if desired.

3.2.5. I210+ Gap filling

AMI 1.1.5 discrepancy: unsupported.

I210+ gap filling of frequent billing data is similar to load profile gap filling for ANSI C12.19 compliant meter. However, this requires EMCM to store the data in the flash. A bit can be associated with each datablock that indicates if data was previously queued for transmission.

3.3. Power Quality Support

AMI 1.1.5 discrepancy: Only supported for GE kV2c and i210+ meters.

Power Quality is a generic term that means host of different things recorded by a meter to quantify how much line voltage and current have deviated from the norm. In many cases, the quantities of interest could be programmed to be reported periodically in load profile and they are by default therefore considered supported. However, they are interval values (min/max/average over an interval) and therefore may not be sufficient.

Given the widely open interpretation of power quality, quantities relevant to power quality are supported by different meters in an adhoc fashion and therefore supporting this meter-specific feature in a unified fashion is extremely challenging. We will broadly classify power quality support (outside of load profile) into three different categories: A)Metrics; B)Events; and C)Diagnostics.

A. <u>Momentary/average power quality metrics</u>: Following is an extended list with the data structure of the recorded quantities (if available).

   1. LtoNVoltageMagnitude[]: (phase, harmonicFlag, value)

      - There are 3 possible entities for phase: 0->PhaseA-N; 1->PhaseB-N, 2->PhaseC-N. If no phase is included, it will be interpreted (in HES via properties file) as PhaseA-N.
      - If known whether harmonics are included or not, the harmonicFlag is set, otherwise this field is not included in quantity description and will be considered unknown.
      - Meter-specific details:
        - I210+c(MT113) supports with harmonicFlag unknown
        - kV2c(MT110) supports all
        - SGM3xxx(MT113) supports with harmonicFlag unknown

   2. LtoLVoltageMagnitude[]: (phase, harmonicFlag, value)

      - There are 3 possible entities for phase: 0->PhaseA-B; 1->PhaseB-C; 2->PhaseC-A.Phase must be included.
      - If known whether harmonics are included or not, the harmonicFlag is set, otherwise this field is not included in quantity description and will be considered unknown.
      - Meter-specific details:
        - I210+c(MT113): No known support
        - kV2c(MT110) supports all

- SGM3xxx(MT113): No known support

3. currentMagnitude[]: (phase, harmonicFlag, value)

- There are 3 possible entities for phase: 0->PhaseA-N; 1->PhaseB-N, 2->PhaseC-N and EMCM shall map it appropriately. If phase is not included, it will be interpreted (in HES via properties file) as PhaseA-N.

- If known whether harmonics are included or not, the harmonicFlag is set, otherwise this field is not includin quantity description and will be considered unknown.

- Meter-specific details:

  - I210+c doesn't support momentary currents

  - kV2c(MT110) supports all

  - SGM3xxx(MT113) supports RMS_CURRENT_FUND_PLUS[] with harmonicFlag unknown

4. momentaryPF (Momentary Power Factor): (phase, value)

- There are 3 possible entities for phase: 0->PhaseA-N; 1->PhaseB-N, 2->PhaseC-N and EMCM shall map it appropriately. If phase is not included, phase will be considered unknown (in CIM code generated).

- Meter-specific details:

  - I210+c(MT113) with no phase

  - kV2c(MT110) supports with no phase information

  - SGM3xxx(MT113) supports all 3 phases

5. averagePF (Average Power Factor): (phase, value)

- There are 3 possible entities for phase: 0->PhaseA-N; 1->PhaseB-N, 2->PhaseC-N and EMCM shall map it appropriately. If phase is not included, phase will be considered unknown (in CIM code generated).

- Meter-specific details:

  - I210+c(MT81) mentions existence of MT81, but meter and programming guide doesn't show one!

  - kV2c(MT81) supports with no phase information; question is whether prev or current value is returned?

  - SGM: No known support

6. frequency (Momentary Frequency): value

- Meter-specific details:

  - I210+c: no known support

  - kV2c(MT110) supports

    - SGM3xxx: No known support

7. TDD(Total Demand Distortion): (phase, value)

- There are 3 possible entities for phase: 0->PhaseA-N; 1->PhaseB-N, 2->PhaseC-N and EMCM shall map it appropriately.
- Meter-specific details:
    - I210+c: no known support
    - kV2c(MT110) supports
    - SGM3xxx: No known support

8. ITHD(Current Total Harmonic Distortion): (phase, value)

- There are 3 possible entities for phase: 0->PhaseA-N; 1->PhaseB-N, 2->PhaseC-N and EMCM shall map it appropriately.
- Meter-specific details:
    - I210+c: no known support
    - kV2c(MT110) supports
    - SGM3xxx: No known support

9. VTHD(Voltage Total Harmonic Distortion): (phase, value)

- There are 3 possible entities for phase: 0->PhaseA-N; 1->PhaseB-N, 2->PhaseC-N and EMCM shall map it appropriately.
- Meter-specific details:
    - I210+c: no known support
    - kV2c(MT110) supports
    - SGM3xxx: No known support

10. Distortion PF: (phase, value)

- There are 4(?) possible entities for phase.
- Meter-specific details:
    - I210+c: no known support
    - kV2c(MT110) supports
    - SGM3xxx: No known support

11. Temperature: value

- Meter-specific details:

- I210+c(MT113): Supported
- kV2c: No known support
- SGM3xxx(MT113): Supported

12.Momentaries:

B. Power Quality Cautions/Events: Events may be generated and logged by a meter when a power quality metric exceeds a certain configured threshold. In addition to the event, logs may also contain information about duration of such an event, actual value of power quality metrics etc. Here is a list of different power quality events and optional information qualifying the event:

1. sagInfo/swellInfo: Sag or Swell information data block which may include

   a. nbrEvents: that will indicate number of sag or swell events that may have passed since last report

   b. record[]: An array of sag or swell records that may include

      i. eventTime

      ii. duration

      iii. LtoNVoltageMagnitude[]: (phase, value)

      iv. currentMagnitude[]: (phase, value)

   - Meter-specific details:
     - I210+(Table 85): Meter reports sagCtr and swellCtr. EMCM shall keep a copy, diff it and send an event only if different. There is no associated record. If EMCM doesn't keep it in flash (no requirement to store persistently across boots), EMCM should read it during boot and store a local copy (but don't generate a power quality event).
     - I210+c(MT112): supported but nbrEvents should be computed from the log table
     - kV2c(MT112): Supported but nbrEvents should be computed
     - SGM3xxx(MT114): Supported as EVENT_CODE=74 for sag and EVENT_CODE=76 for swell (argument description in Figure 9 of RnP guide)

2. voltageImbalance: This information element may include

   a. nbrEvents: that will indicate number of voltage imbalance events that may have been recorded since last report

   b. record[]: An array of voltage imbalance records that may include

      i. eventTime

      ii. minVoltage

      iii. maxVoltage

   - Meter-specific details:
     - I210+: No known support

- I210+c: No clean support except for event logs table ST76 (no support in AMI 1.1)
- kV2c: No clean support except for event logs table ST76 (no support in AMI 1.1)
- SGM3xxx(MT114): Supported as EVENT_CODE=79(argument description in Figure 9 of RnP guide)

3. underFrequency/overFrequency: These two information elements may include

   a. nbrEvents: that will indicate number of under/over frequency events that may have been recorded since last report

   b. record[]: An array of under/over frequency records that may include

      i. eventTime

      ii. frequency

   - Meter-specific details:
     - I210+: No known support
     - I210+c: No clean support except for event logs table ST76 (no support in AMI 1.1)
     - kV2c: No clean support except for event logs table ST76 (no support in AMI 1.1)
     - SGM3xxx(MT114): Supported as EVENT_CODE=75 for under frequency and 77 for over frequency (argument description in Figure 9 of RnP guide)

C. <u>Line-side diagnostics(LSD)</u>: This a GE kV2c feature that has potential to bring benefits for 3-phase CNI meters (though it is supported partly on GE I210+c). Essentially meter can be programmed to flag some abnormal conditions as diagnostics and provide complete information about voltage and currents that can be used to create phasor diagrams. Following information is contained in LSD:

1. Diagnostic bitfield: A bit indicates diagnostic condition that is active. Following 8-bit diagnostics are defined in kV2c:

   i. *Diag1(D1)*: Polarity, Cross Phase, Rev. Alert

   ii. *Diag2(D2)*: Voltage Imbalance Alert

   iii. *Diag3(D3)*: Inactive Current Alert

   iv. *Diag4(D4)*: Current Phase Angle Alert

   v. *Diag5(D5)*: Distortion Alert

   vi. *Diag6(D6)*: Under Voltage Alert

   vii. *Diag7(D7)*: Over Voltage Alert

   viii. *Diag8(D8)*: High Neutral Current Alert

2. LtoNVoltageMagnitude[]: (phase, value)

3. CurrentMagnitute[]: (phase, value)

4. VoltageAngles[]: (phase, value)
5. CurrentAngle[]: (phase, value)
6. DistortionPF

3.3.1. Power Quality Metrics

3.3.1.1. On demand power quality read

Instantaneous/average power quality metrics (Type A) generated by the meter can be requested OTA by the message "PowerQualityReportReq". HES shall generate and queue PowerQualityReportReq message either due to an explicit request from OTV UI or via CIM interface as described in Section 2.3.2.3.1, “Supporting CIM requests/asynchronous reports”.

On receiving this message, EMCM shall query the meter (queries are meter-specific) and based on supported quantities generate "PowerQualityReportRsp" message. Quantities reported in PowerQualityReportRsp message shall be mapped to readingType CIM code based on mapping in properties file and publish it accordingly.

3.3.1.2. Scheduled power quality read

Schedule for power quality read is determined by SetDataOrderReq. In addition a filter can be provided to reduce the number of quantities reported when power quality report is expected at a fast rate.

3.3.2. (Autonomous) Power Quality Events Read

Note for IDL change: Change the name from GEVoltageInd to powerQualityEventsInd

Power quality events (Type B) as described before are logged by the meter. EMCM shall poll the meter for new logged events at periodic interval and report newly logged events. EMCM should generate powerQualityEventsInd message only if a new event is logged.

Configurable parameters of interest (should be configurable OTA as PQEventOrder):

1. powerQualityEventOrder.schedule.reportingPeriod (default=15 mins): EMCM poll rate for power quality events. This will also determine maximum rate at which a new powerQualityEventsInd message is generated by the EMCM.
2. powerQualityEventOrder.filter.maxRecordsPerReport (default=5): Note that EMCM shall read through all events recorded by the meter that were logged since the last read and include the relevant information element (Type B) with "nbrEvents" corresponding to that information element. However, EMCM shall limit the total number of records (across all events) to maxRecordsPerReport. EMCM shall only report the oldest "maxRecordsPerReport" unread since last read if events logged exceed this quantity.

As described in Section 2.3.2.3.1, “Supporting CIM requests/asynchronous reports”, on receiving powerQualityEventsInd message from EMCM, HES shall map them to relevant "EndDeviceEvents" category code and associated "MeterReadings" to appropriate readingType **with quality code 1.2.0** and publish them for OTV and MDMS consumption.

Being a brand new feature, next few subsections give an outline of meter-specific algorithm used to support this feature. Intention is to show that feature can be supported without any state management done by EMCM or doing a large read that requires large storage (storage requirements are dependent on maximum number of records that are reported). Post implementation this could potentially move to EMCM SW spec.

3.3.2.1. Reading power quality events from GE I210+c/kV2c (Informative)

Support: Since voltage recording in MT112 is enabled only if relevant soft-switch is installed (bit 12 of upgrade_bfld in MT0 is set) and monitoring enabled, EMCM should verify whether voltage event monitoring will be done by reading MT111 and checking "voltage event monitoring enabled" flag.

Relevant Procedures: MP79 to reset list pointers (not certain if needed); MP80(with Entries Read) reduces NBR_UNREAD_ENTRIES in MT112 by "Entries Read"; and MP 84 which is a snapshot procedure that should be executed prior to MT112 read.

Algorithm: Both I210+c/kV2C store the data in circular buffer. Each record is 23 bytes and there are 200 records stored at offset 15.

- Variables that determine which records should be read:
  1. LAST_ENTRY_ELEMENT (2 bytes @ offset 3 of MT112);
  2. NBR_UNREAD_ENTRIES (2 bytes @ offset 9 of MT112)
- If recording is enabled, then every reportingPeriod, EMCM should read NBR_UNREAD_ENTRIES of 23 bytes records in MT112 as follows:
  - Oldest: 15+23*((LAST_ENTRY_ELEMENT-NBR_UNREAD_ENTRIES)%200)
  - ith from the oldest: 15+23*((LAST_ENTRY_ELEMENT- NBR_UNREAD_ENTRIES +i-1)%200)
- Based on EVENT_TYPE, EMCM should determine nbrEvents for sagInfo and swellInfo from all unread records.
- EMCM shall include maximum of maxRecordsPerReport records irrespective of whether they correspond to sagInfo or swellInfo. Data from rest of the record should be thrown.
- After reading MT112, execute MP80 with Entries Read = NBR_UNREAD_ENTRIES. This will ensure that during next read, logged records read during this read will not be doubly counted.

3.3.2.2. Reading power quality events from GE SGM3xxx (Informative)

3.3.3. (On-demand) Line-side diagnostics/power quality (GE kV2c)

AMI 1.1.5 discrepancy: Supported but as part of PowerQualityReportReq

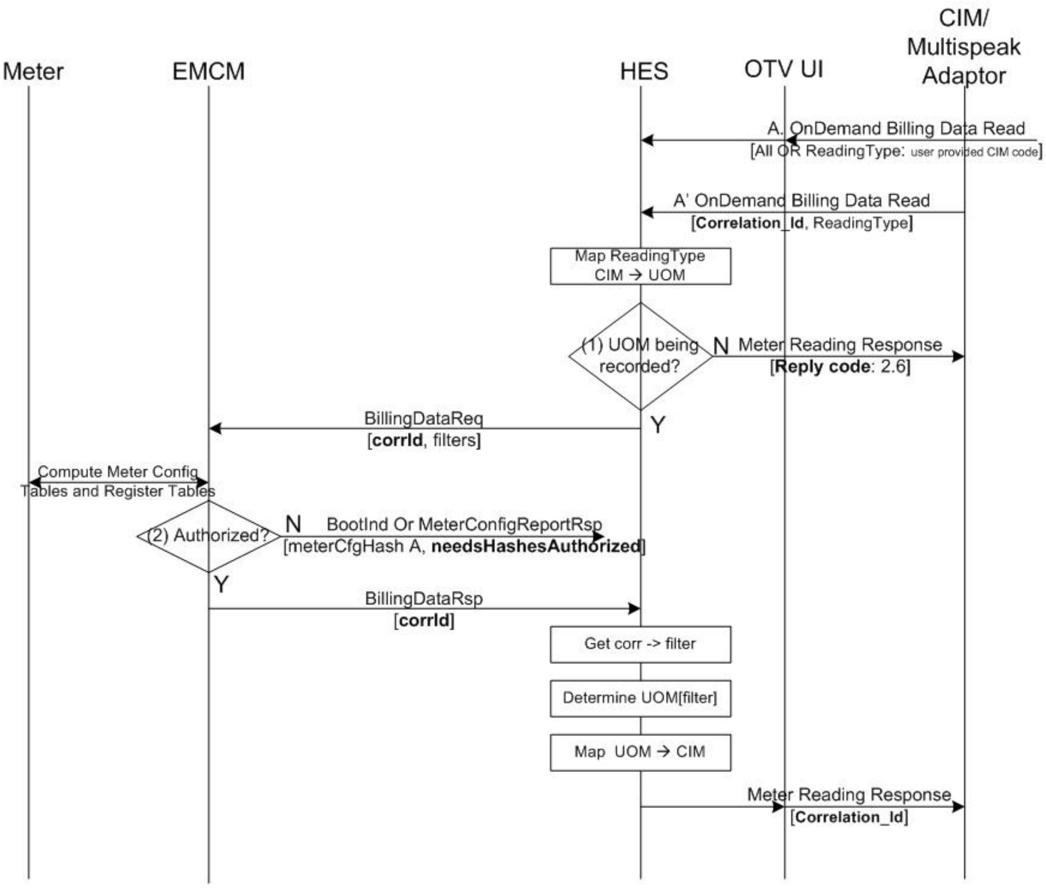
Note than diagnostic bit-field is part of meter flags (Section 3.6, “Meter Events/Flags and Relay State”) and will be mapped to events generated by HES/OTV. On observing an event, a user may wish to diagnose the cause of the event. GE kV2c (as well as I210+c, but less useful in I210+c) provide comprete descriptions of voltages and currents that using phasor diagramns can be used for diagnose the installation issues. Given post-installation use case, line-side diagnostics at this point is envisioned to be solely an OTV feature with no CIM support. A user may request LSD information which will result in HES queuing diagnosticPowerQualityReq to EMCM. EMCM shall read the data present in MT73 (kV2c (I210+c maybe), other meters still TBD) and generate diagnosticPowerQualityRsp. HES is not required to map this information to CIM but store it in OTV database for visual display over OTV UI.

3.4. On Demand Reads

Load profile, billing data (that includes summations, demands and coincidents when supported), and instantaneous power quality are few of the on-demand requests that are supported. Over the CIM interface, this request is made as described in Section 2.3.2.3.1, “Supporting CIM requests/asynchronous reports”. Following subsections outline how each of these requests are supported by OTR.

3.4.1. On Demand Billing Read

Figure 6.



1. UOMs being recorded?:If billing data read request is with specific CIM codes, HES shall map them to UOM and thereby determine the billing data filters for the response. However if that UOM is not supported by meter being queried, then it shall reject the request with Reply code of "2.6" indicating "Invalid ReadingTypeId".

2. Authorized?:Before queuing billing data read for transmission, EMCM shall verify meterCfgHash over the meter tables specified in Section 3.1, “Device Boot”. If configuration has changed, then EMCM will consider itself unauthorized, throw away the reading and enter Boot process as shown in Figure 4, “Boot sequence”. If however EMCM is authorized, EMCM shall send BillinDataRsp either with orderId or with specific quantities requested.

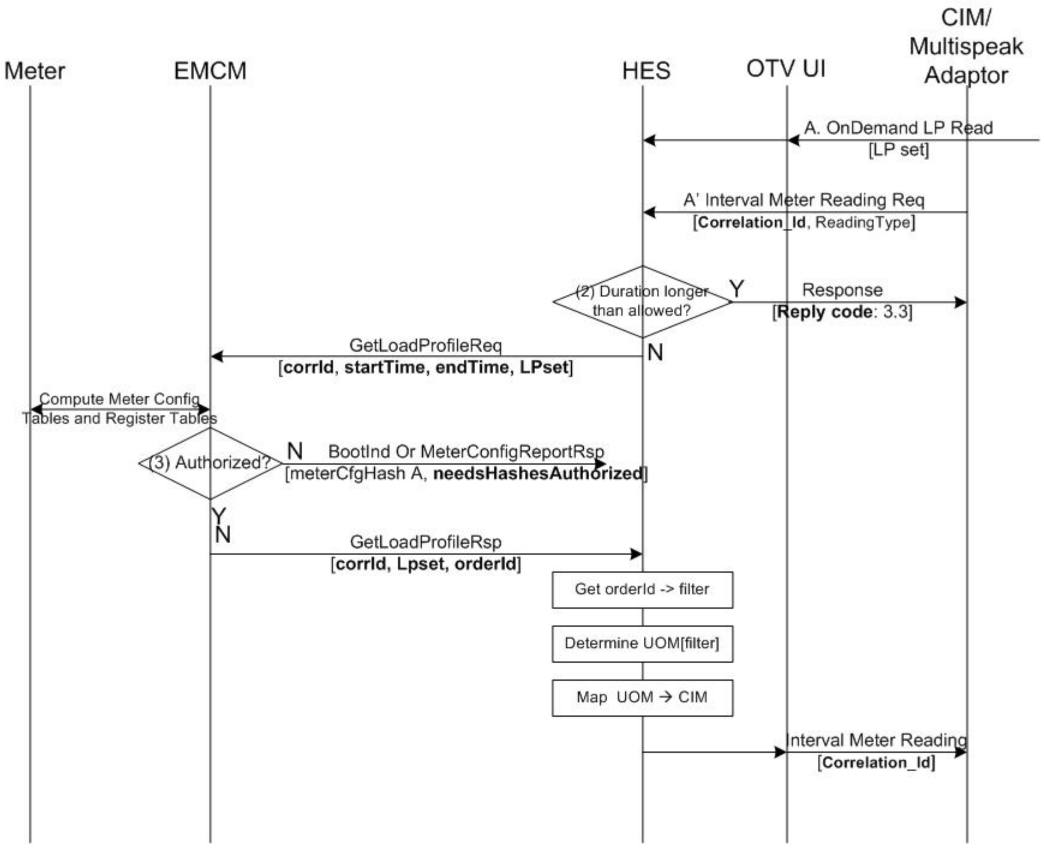
3.4.2. On Demand Load Profile Read

⚠

AMI 1.1.5 discrepancy: unsupported.

Unlike billing data, on demand read of selected load profile quantities are not supported in AMI 1.1. All quantities recorded by the meter within a certain load profile set will be returned.

Figure 7.



1. Duration longer than allowed?: Load profile (LP) read request contains duration (with start and endTime) for which LP data is being requested. **HES shall maintain configuration that limits the number of intervals being requested (default value of 16 intervals)** . If duration requested corresponds to more than maximum number of allowed intervals, HES shall reject the request with reply code of "3.3" indicating "too much requested data".

   Note that number of intervals is typically related to EMCM storage capacity and so is not really changeable till a new EMCM upgrade which changes this storage capacity.

2. Authorized?: As described previously.

3.5. Meter Control

3.5.1. Backend-triggered Connect/Disconnect/Arm-to-Connect and autonomous Relay State Ind

EMCM can be programmed to periodically monitor relay state of all the relays and report any changes to the state of any relay. In ANSI C12.19 compliant meter, NBR_OF_CONTROL_POINTS in ST111 denotes number of supported relays. For each of the supported relays, relay state as per ANSI C12.19 is a 3-variable state: "sensed" level, "current" or "output" level, and "pending" or "requested" level.

Meter specific information:

1. GE kV2c as an CNI meter has no relays.

2. GE I210+ can have maximum of one relay.

3. GE I210+c can have maximum of one relay but NBR_OF_CONTROL_POINTS is 6. Index 0 corresponds to main relay but others are mapped to enabling and disabling of ECP(1), DLP(2), PPM(3), LOCKOUT(4) and Manual acknowledgement(5).

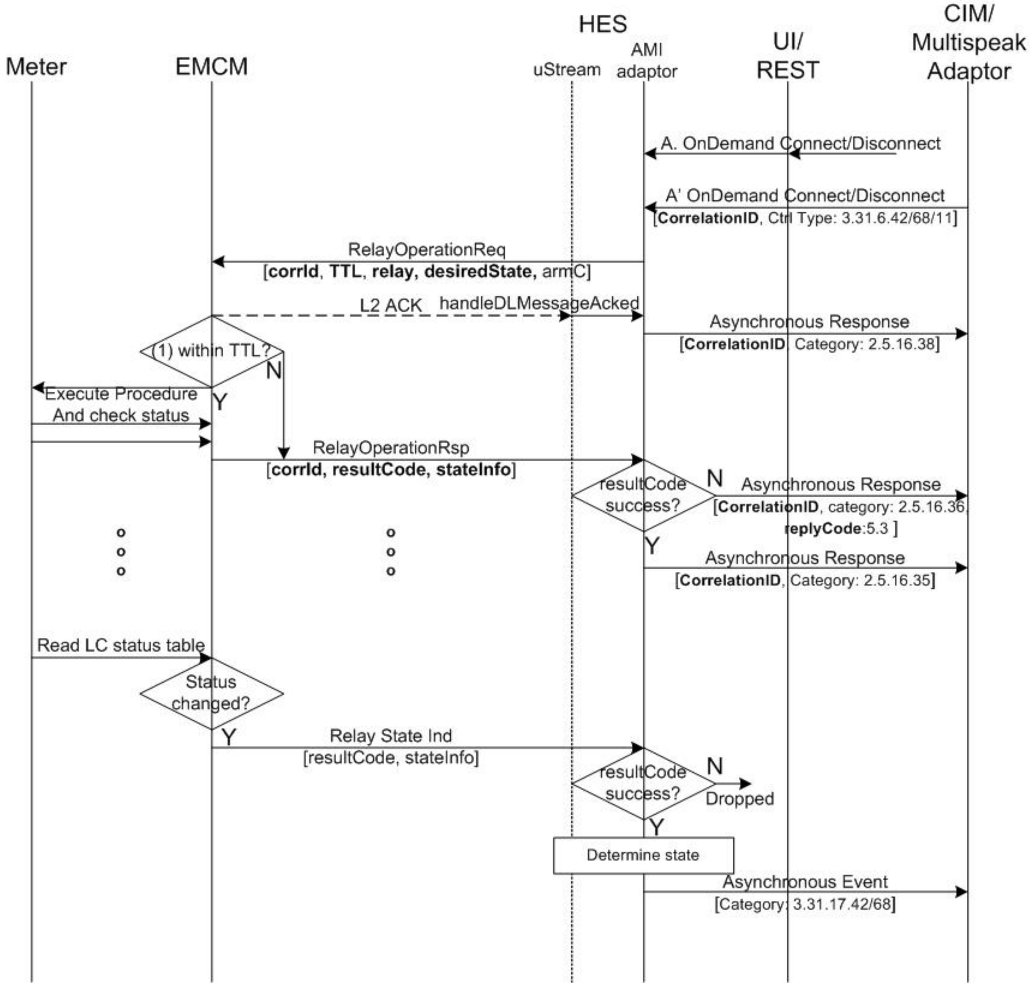
   ORW AMI system won't treat these logical mappings as relays (i.e. periodically monitored changes to relay state will only correspond to relay state of control point index 0) though enabling and disabling of ECP/DLP/PPM/Lockout can be achieved by sending a RelayOperationRsp (or by generic OTA meter configuration update process). State information changes corresponding to ECP/DLP/PPM will be retrieved and reported as part of meter flags.

4. GE SGM3xxx can have maximum of 3 relays which corresponds to NBR_OF_CONTROL_POINTS. First one corresponds to main relay (also called service disconnect relay) and other two will be auxiliary relays.

   ORW AMI system shall poll for and report the changes to the state of any of the three relays when supported.

   Just like I210+, state transition of main relay can be affected by internal SGM modes such as PPM/ESC/UFLS/SC/PU. However enabling/disabling of these features is not explicitly provided. Enabling/disabling of these modes can only be achieved by OTA meter configuration update process.

Figure 8.



A connect/disconnect/arm for closure command (for any of the multiple relays) is mapped to RelayStateOperationReq message. If commanded to arm for closure, i.e. connect with manual acknowledgment (applicable only to meters with pushbutton installed), armC variable will be included with armC = 1. This variable will be used by eMCM to use appropriate parameters while execute connect command.

TTL is determined in one of the two ways:

1. If endTime is included in the command received from the backend, then TTL corresponds to the endTime

2. If endTime is NOT included in the command received from the backend, then HES shall determine TTL from configurable parameter "cfgMaxDeliveryDuration". TTL = current_time + cfgMaxDeliveryDuration

3.5.2. Backend-triggered Demand Reset

Demand reset can be auto programmed into EMCM configuration (or meter) as was described in Section 3.2, “Scheduled Metrology Reads”. Demand reset can also be triggered either via CIM/ Multispeak interface (Section 2.3.2.3.1, “Supporting CIM requests/asynchronous reports”) or UI. Following characterize the demand reset request:

1. Demand reset from backend may contain "start time" and/or "end time" and implies that the demand reset has to be executed at the start time but no later than the end time. Start time allows user to schedule the demand reset in advance. End time allows the user to time out the demand reset in cases where OTA delivery delays are past start time or EMCM is not active at the time of the demand reset (booted or unauthorized). Both are optional. If start time is not included, then EMCM executes demand reset at the time of the request. If end time is not included, then demand reset request never expires.

   AMI 1.1.5 discrepancy: start time and end time unsupported.

2. On receiving a user-triggered demand reset request, if start time is greater than the current time, EMCM shall store the demand reset request in its persistent memory and set up a timer to trigger demand reset at the start time.

3. After EMCM reboots, it is authorized and has acquired time, EMCM should check if there is any pending demand reset. If there is a pending request, then EMCM shall

   - execute immediately if the current time is in between start time and end time;
   - throw away the demand reset request if current time has passed;
   - start a timer for demand reset to be executed at start time if start time is in the future.

4. If a new demand reset request is received prior to the execution of old DemandResetReq, then previous request shall be overwritten, *i.e.* EMCM needs to provide storage for only a single demand reset command.

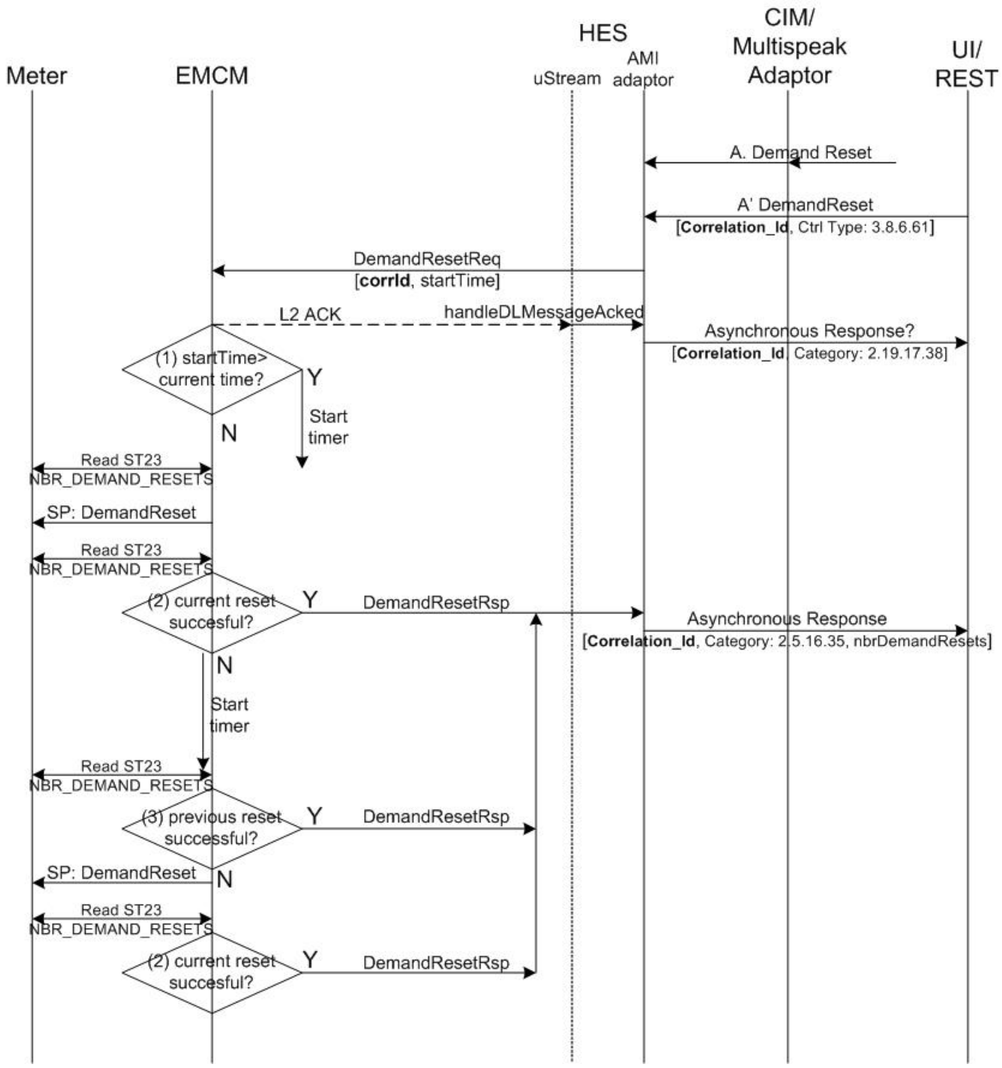
5. Demand reset request can contain optional parameters to request demand data and coincident data prior to demand reset execution.

   HES-side consideration: These optional parameters are set and included based on HES properties file or can always be set by default. Demand and coincident data can be large and so by providing this flexibility, we can turn off reporting of data that is not required.

   AMI 1.1.5 discrepancy: Ability to request demand and coincident data not supported.

What is described below (Figure 9) is for future consideration (FFS)

Figure 9. Retries on user-triggered demand reset: FFS



1. EMCM shall have an ability to store one pending demand reset request in its persistent memory. When a demandResetReq command is received, it can overwrite previous pending demand reset command. If startTime is not included in the message, then time of the receipt will be used as start time. *In addition to start time, there should be a field for NBR_DEMAND_RESETS which can be considered "unknown" at the time request was queued and not yet attempted*. EMCM shall start a timer to trigger a demand reset at the start time. If EMCM boots prior to start time, EMCM shall re-start timer to trigger demand reset at appropriate time as illustrated in step 5 of Figure 4, “Boot sequence”.

2. At start time or at some periodic interval, EMCM should first query for NBR_DEMAND_RESETS (from ST23) before executing demand reset and then query it again. If NBR_DEMAND_RESETS has changed, then it indicates successful demand reset. On the other hand, if NBR_DEMAND_RESETS variable has not changed or meter communication failed in the process, then it indicates either pending or failed demand reset. EMCM shall update NBR_DEMAND_RESETS (if available) to the pending request and wait for a periodic interval to retrigger the request.

3. At start time of some periodic interval, EMCM should first query for NBR_DEMAND_RESETS (from ST23). If NBR_DEMAND_RESETs polled has changed (from a known value), then it indicates previous reset attempt was successful else we are back to step 2.

3.5.3. Back-end triggered "Critical-Peak/Real-Time Pricing (CPP/RTP)" or "Season" Change

AMI 1.1.5 discrepancy: CPP/RTP and/or Season change request not supported.

3.5.3.1. Season Change or CPP/RTP change?

Season change is applicable when meter is in time-of-use (TOU) mode. Tiers that are accumulated during a particular season are governed by tier times programmed in calendar table (ST54) stored in the meter. Therefore a season change can in principal affect price change if different tiers are accumulated during different seasons.

CPP or RTP is meter vendor-specific feature. They basically mean the same thing but it is called CPP in GE SGM3xxx and RTP in GE I210+c/kV2c. When CPP/RTP is enabled, meter accumulates a specific tier associated with critical pricing. The differences in meter-specific support are outlined below:

- In GE I210+c/kV2c, RTP is available ONLY when meter is not in TOU mode (aka what GE calls Demand or Demand/LP mode). When RTP is activated, meter starts accumulating consumption in tier A and when deactivated, stops accumulating consumption in tier A. Meter configuration in MT76 may further qualify this accumulation but meter configuration is out-of-band and specifics of it are irrelevant to this description and out-of-scope of this document.
- In GE SGM3xxx, CPP is available even when meter is in TOU mode. When CPP is activated, meter starts accumulating consumption in a pre-programmed tier corresponding to CPP and when deactivated, it goes back to normal TOU mode operation where accumulation is governed by tier switches programmed in Calendar Table. Meter configuration in MT76 may further qualify the accumulation when CPP is enabled but meter configuration is out-of-band and specifics of it are irrelevant to this description and out-of-scope of this document.

What all this means to ORW AMI system?:Two downlink requests are supported: i) SeasonChangeReq; ii) RtpChangeReq.

1. SeasonChangeReq:
   a. It is only applicable when meter is in TOU mode. If meter is not in TOU mode (applicable only to GE I210+/I210+c/kV2c), and a SeasonChangeReq is received, then EMCM should generate an error response indicating not supported.
   b. Season change request can take a parameter with four values indicating the season.
   c. It can be mapped over IEC-CIM xml interface in following two ways:
      i. As a season change request by mapping in iec-cim-http.properties to season change CIM code: EndDeviceControl.SeasonChangeReq=3.36.6.228.x where x indicates season number;
      ii. As a price signal change request by mapping in iec-cim-http.properties to price signal change CIM code: EndDeviceControl.SeasonChangeReq=3.34.6.201.x where x indicates season number;
2. RtpChangeReq:
   a. In case of GE I210+c/kV2c meters, it is only applicable when meter is in non-TOU mode. If meter is in TOU mode, and a RTPEnableReq is received, then EMCM should generate an error response indicating not supported.
   b. RTP change request can take a parameter with two values: 0 and 1 where '0' indicates disabling RTP/CPP while '1' indicates enable RTP/CPP.

c. It is mapped over IEC-CIM xml interface in following way:

   i. As a price signal change request by mapping in iec-cim-http.properties to price signal change CIM code: EndDeviceControl.RtpChangeReq=3.34.6.201.x where x=0 indicates RTP/CPP off and x=1 indicates RTP/CPP "on";

Exact process of backend triggered season/RTP change very much mirrors that for demand reset. Following characterizes the season/RTP change request:

1. Season change or RTP change request from backend may contain "start time" and/or "end time" and implies that the command has to be executed at the start time but no later than the end time. Start time allows user to schedule the command in advance. End time allows the user to time out the command in cases where OTA delivery delays are past start time or EMCM is not active at the time of the demand reset (booted or unauthorized). In addition, end time faciliates an additional purpose: it allows the user to bring the season/RTP state to same state as before the startTime. Both are optional. If start time is not included, then EMCM executes the command at the time of the request. If end time is not included, then request never expires and meter season/RTP state is alterned till an explicit command is received to change it back.

2. On receiving a user-triggered season/RTP change request, if start time or end time is greater than the current time, EMCM shall store the request in its persistent memory and set up a timer to trigger demand reset at the start time if start time is greater than current time.

3. If end time is included, then before executing season change or RTP change, EMCM shall store the current state of season/RTP change in its persistent memory and set up a timer to trigger season/RTP change to its original state at the end time.

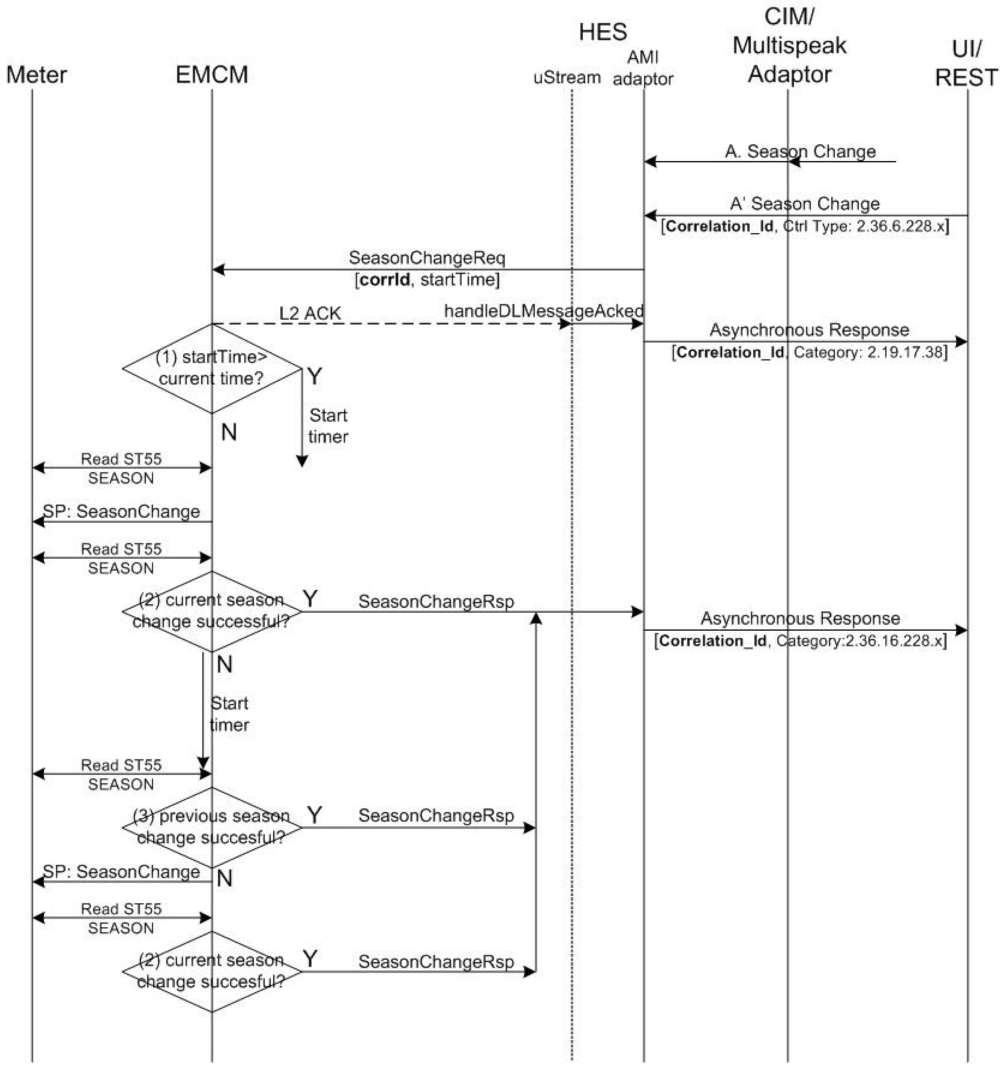
4. After EMCM reboots, it is authorized and has acquired time, EMCM should check if there is any pending season/RTP change request. If there is a pending request, then EMCM shall

   - execute immediately if the current time is in between start time and end time;
   - throw away never executed request if current time has passed the end time;
   - start a timer for request to be executed at start time if start time is in the future;
   - if the request was executed and end time exists, start a timer to reexecute the change and revert it to original state at the end time.

5. If a new season/RTP change request is received prior to the execution of old request, then previous request shall be overwritten, *i.e.* EMCM needs to provide separate storage for a single season change request command and a single RTP change request.

What is described below (Figure 10) is for future consideration (FFS)

Figure 10. Retries on user-triggered season change: FFS



1. EMCM shall have an ability to store one pending season change request in its persistent memory. When a SeasonChangeReq is received, it can overwrite previous pending season change command. If startTime is not included in the message, then time of the receipt will be used as start time. EMCM shall start a timer to trigger a season change at the start time. If EMCM boots prior to start time, EMCM shall re-start timer to trigger season change at appropriate time as illustrated in step 5 of Figure 4, “Boot sequence”.

2. At start time or at some periodic interval, EMCM should execute procedure to change the season to desired value. It shall then query SEASON (from ST55) to determine if season has changed to the desired value. If it has, then it indicates successful season change resulting in SeasonChangeRsp message. On the other hand, if SEASON variable is not the desired value, then it indicates either pending or failed season change. EMCM shall wait for a periodic interval to retrigger the request.

3. At start time of some periodic interval, EMCM should first query for SEASON (from ST55). If SEASON polled has changed to the desired value, then it indicates previous attempt was successful else we are back to step 2.

3.6. Meter Events/Flags and Relay State

Meter events include status indicator bitfields reported by ANSI C12.19 compliant meters in meter status table ST3. These are also commonly referred to as meter flags and grouped into two categories: "standard status indicators" and "manufacturer status indicators". All C12.19 compliant meters set/clear standard status indicators (if enabled via meter program) and result in common set of events. Manufacturer status indicators are vendor and model specific. Relay state is the basic state of relays (main relay which controls power from the main line and various load control relays also called auxiliary relays). Basic state of a relay in ANSI C12.19 compliant meters is recordedin "Load Control Status Table (ST112)". Basic state of relay may be further qualified by manufacturer-defined status indicators.

There are a large category of statuses that meters record but that are not selected in meters to be reported in ST3. Use case is line-side diagnostic indicators in GE kV2c and and load control status indicators in residential meters such as GE I210+c and SGM3xxx. These will be dubbed as "extra manufacturer" state indicators as they are vendor and model dependent and are discussed in detail below.

3.6.1. Extra Manfuacturer Flags

State indicators that shall be read and reported as "extra manufacturer" indicators are as described below:

A. *GE kV2c*:

- First 8-bits of extraMfgFlags shall be mapped to 8-bit diagnostics bitfield (DIAG_BFLD) in MT72. These when set, in increasing order of numeration, indicate: b0) Cross Phase; b1) Voltage imbalance; b2) Inactive current; b3) Current phase angle exceeds limits; b4) Distortion; b5) Under voltage; b6) Over voltage; b7) High neutral current.
- For future consideration.

B. *GE I210+c*:

- Line-side diagnostic indicators: First 8 bits (b0-b7) are same as that in kV2c and correspond to 8 bit diagnostic bitfield (DIAG_BFLD) in MT72. Note that even though diagnostic bitfield is 8 bits, I210+c only sets the following: b5) Under voltage and b6) Over voltage.
- Load control status indicators: Next 16 bits (b8-b23) shall correspond to LC_STATUS_FLAGS (2 bytes @offset 0) recorded in Load Control Status Table (MT115). Of these, the following when set will indicate: b8) Communication error with switch control board; b9) Switch state inconsistent; b10) Switch failed to close; b11) Alternate source present; b13) Switch Bypassed; b14) Switch failed to open; b15) Negative prepayment credit; b16) Manual Arm timer expired.
- RC_STATE (with first 3 bits zeroed out): Next 8 bits (b24-b31) shall correspond to RC_STATE(1 byte @offset 4). Of these first 3 bits should be zeroed out (either not supported or unnecessaryduplicates of relay state: ACTUAL_SWITCH_STATE, DESIRED_SWITCH_STATE and OPEN_HOLD_FOR_COMMAND) and only relevant one is b30) Lockout in effect.
- Load control state indicators: Next 8 bits (b32-b39) shall correspond to LC_STATE(1 byte @offset 5) recorded in Load Control Status Table (MT115). Of these, the following when set will indicate: b33) Energy conservation period active; b34) Demand limiting period active; b35) Prepayment mode is currently active; b37) Service disconnected due to ECP; b38) Service disconnected due to DLP; b39) Service disconnected due to PPM.
- FFS

C. *GE SGM3xxx*:

- (Main) Relay status indicators: First 16 bits (b0-b15) shall correspond to RELAY_STATUS_BFLD (2 bytes @offset 0) recorded in "Service Disconnect and Load Control Relay Status Table" (MT124). Of these, following when set shall indicate: b0) Switch failed to close; b1) Switch failed to open; b2) Switch bypassed; b3) Switch opertion error; b4) Alternate source; b5) Phase Error; b8) Relay command rejected; b9) Switch armed for closure; b10) Manual arm timeout expired; b11) RTC Alert; b12) Negative prepayment credit; b13) Prepayment overdraft limit exceeded.

- Main relay state indicators: Next 16 bits (b16-b31) shall correspond to SDR_RELAY_STATE_RCD (2 bytes @offset 2) recorded in MT124. Of these, following when set shall indicate: b16) Switch override active; b17) Safety alerty; b18) Prepayment mode active; b19) ESC active; b20) Under frequency load shedding active; b21) SC active; b22) PU active; b23) AD after remote connection active.

- Relay state additional information: Next 8 bits (b32-b39) shall correspond to STATE_ADDITIONAL_INFO (1 byte @offset 4) recorded in MT124. Of these b32-b34 indicate different reasons for auto-service disconnect and b35-b39 indicates which active auto-disconnect mode is in progress.

- For future consideration.

3.6.2. Autonomous reporting

- EMCM-side support: EMCM shall read stdStatus and mfgStatus bitfields, non-ST3 flags, and relay state for each of the supported relays after the boot and report it. On successive reads (governed by periodic schedule configured into EMCM), if there are changes to the bitfields, EMCM shall generate meter indication message with status indicators and/or relay state included. EMCM shall ensure that it has been successfully able to enqueue any updates to uStream before changing state in its internal memory.

  Note that a uStream failure to send a reliable message on uplink shall lead to uStream resynchronization which as previously defined shall lead to boot process between EMCM and HES. Therefore status indicators shall be included in bootInd, a message that is sent post boot to ensure HES and EMCM state wrt flags is in sync.

- HES-side support: On receiving meter status indicators, HES shall map them to appropriate EndDeviceEvents with category codes as defined in properties file.

  OTV side support for managing alarms/notifications based on reported events is described later.

3.6.3. Clearing of status indications (not applicable to relay state)

Any clearing of status indication on the meter will be user-triggered and will lead to HES sending MeterFlagClearReq to EMCM with status indicators bits that needs to be cleared included. EMCM shall execute the procedure to clear these fields.

Clearing of status indication bit will trigger an updated MeterFlagInd message (as described in last section) that will lead to clearing of the meter event on OTV.

3.7. Meter Program Update and Generic Reads (via ANSI C12.19 Operations)

AMI 1.1.5 discrepancy: AMI 1.0-style table reads and writes are supported.

For ANSI compliant meters, EMCM users ANSI C12.18/C12.21 PSEM protocol to communicate with the meter assuming meter data is stored and operated upon as per the structure imposed by ANSI C12.19. In C12.19 compliant meters, meter data (register data as well as configuration data) is organized as meter tables. C12.19 via C12.18 read/write exchanges allows EMCM to read data from meter tables, write to meter tables and perform procedures to update the meter data in a coherent and consistent fashion.

Meter program update (aka meter configuration update) is achieved via a meter-specific sequence of writes and procedure executes. In addition, meter-specific constraints may require delay between operations, provisions for setting up time, and validatation based on some meter data. Note that meter program is not just a set of configuration parameters to be updated but also has a complex set of dependencies whose consistency is important for proper functioning of the meter. It is the responsibility of the meter vendor tool to generate a sequence of write and executes that is permitted within the space of allowed configurations and it is the responsibility of AMI system to faithfully execute this sequence.

A generic read is a way to read raw data from the meter (presented in various meter tables) that with further processing at the backend can be used to generate relevant reports. Note that a generic read may require procedure executed (and even some table writes) prior/post read. For example reading register data and event logs may require snapshot procedure prior to the read for coherent data gathering and may also require some pointers to be updated post read.

Intent of this feature is to provide a generic C12.19 Operation request and response with provisions for validation, delay between operations and setting time. It can be used for full/partial meter configuration update or for 3rd party to develop features with meter-specific data not provided in standard AMI product (*i.e.* features not available over IEC-CIM interface). Following characterize a generic meter operation request:

1. **Request start and end time**: Start time indicates the time at which the C12.19 operation request should be executed. EMCM must start a timer to execute the request at start time. If EMCM reboots before the start time, it shall restart the timer. If start time was in the past and request has not been executed, EMCM shall execute the request immediately provided the end time has not elapsed. Request should be discarded if the end time has elapsed and a C1219OperationRsp should indicate failure due to TTL expiry.

   Note that both start and end times are optional. Absence of start time indicates that request should be executed immediately while absence of end time means no expiry time.

2. **Number of retries**: If a sub-operation results in error or meter communication fails during the middle of execution, number of retries will indicate number of times C12.19 operation request should be re-executed before EMCM declares failure. Absence of this parameter implies that C12.19 operation request should not be retried.

   - Note that backoff between retries is determined based on EMCM configuration parameter (default ~30 seconds)

   - If failure happens during compatibility check, compatibility check is retried. Otherwise if compatibility check has passed, then only programming/read and audit operations are retried. Results of operations performed in previous retry will be overwritten.

3. **Request type**: A generic C12.19 operation request consists of sub-operations which fall into one of the following three categories. These categories are labeled by requestType:

   A. *CompatibilityTest*: CompatibilityTest consists of one or more read sub-operations where each operation is limited to a single table read (either partial or full) and comparing it either against a binary string or verifying a range. CompatibilityTest is usually done to validate meter model, meter type, meter FW version number etc.

- CompatibilityTest comprises of sub-operations that should be specified before sub-operations corresponding to other requestTypes. Its presence is optional. If the process is interrupted during CompatibilityTest, CompatibilityTest is retried as dictated by number of retries indicated in the request.
- If CompatibilityTest at any stage fails, **i.e.** read data does not match the validation string or falls in the specified range, then the entire upgrade process is suspended and response is sent with result code indicating "CompatibiltiyTest failure" and results of read operations performed during CompatibilityTest.

B. *ProgrammAndReadOps*: A sequential execution of different kinds of operations described below. This is the core execution step that marks either program upgrade or table reads for backend processing.

- If the process is interrupted during this stage, then execution returns to beginning and entire request is retried while satisfying maximum number of retries constraint. However, any operation involved in CompatibilityTest is skipped and not retried.

C. *AuditTest*: An AuditTest like CompatibilityTest consists of one or more read operations where each individual operation is limited to a single table read (either partial or full) and comparing it against a binary string or verifying a range. Purpose of AuditTest is to ensure that operations performed during ProgramAndReadOps have correctly taken affect.

- AuditTest generally follows ProgramAndReadOps though two may be interlaced. Its presence is optional.
- If the process is interrupted during AuditTest (eg. meter communication error) or AuditTest fails at any stage fails, then execution returns to the beginning and entire request is retried while satisfying maximum number of retries constraint. However, any operation involved in CompatibilityTest is skipped and is not retried.

4. **Operations**: A generic request is concatenation of following different kinds of operations. Each operation results in a response which is determined on the request.

- Meter table read: Either full or partial read

  Response is generic success or failure indication along with either the read data or in case of failure C12.18 result code.

- Meter table write: Either full table write or partial write

  Response is generic success or failure indication along with C12.18 result code.

- Meter procedure execute

  Response is generic success or failure indication along with result code read from ST8 read corresponding to procedure execute.

- Delay: Delay is the time for which EMCM shall wait before executing the next operation in the list

  No result code is required.

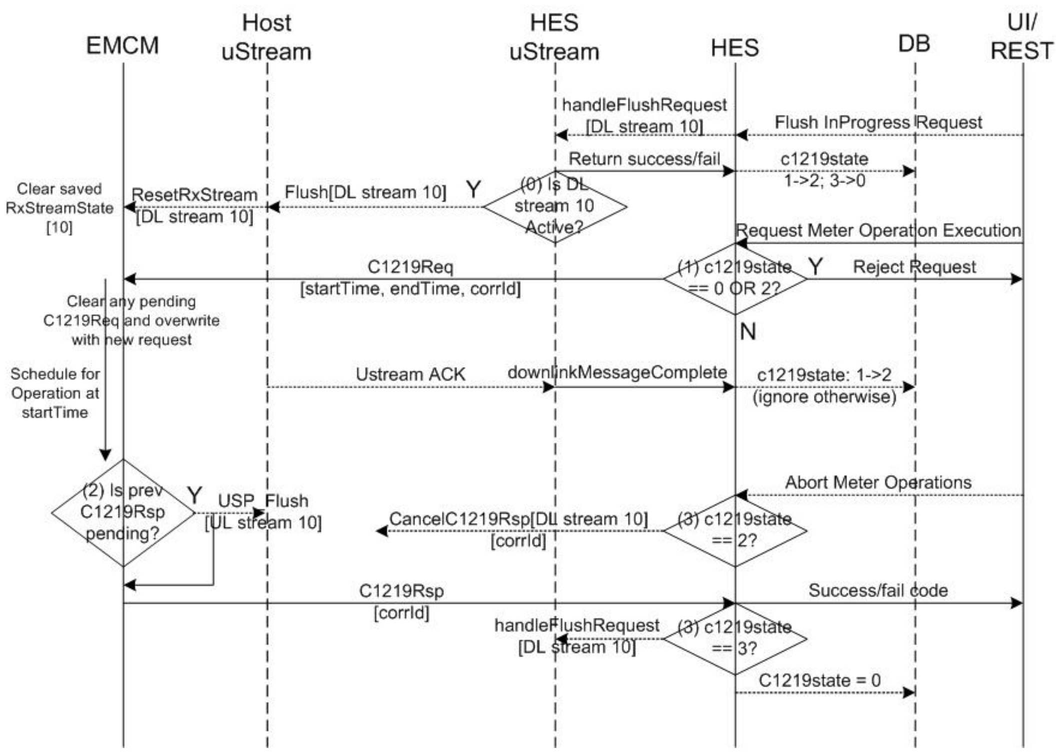
- Set time: Setting of time can be based on time zone offset and DST flag provided by either the script or something that EMCM shall successfully read from the meter before starting the operation.

  Response includes generic success or failure.

5. **Validation**: Each read operation can optionally contain validation context. This is used only when the requestType is either CompatibilityTest or AuditTest. Following validation types are supported:
   - Number Compare
   - Hex String Compare
   - Range Compare

3.7.1. The protocol

Figure 11. Generic C1219 Operations Request Response Call Flow



Following salient features outline the protocol:

1. Both the C1219Operations Request and Response are on resumable stream 10 (on DL and UL respectivvely).
   - Why?: StreamId 10 is higher than streamId used for non-resumable data. A higher stream Id imples that it has lower priority as compared to non-resumable requests and responses (which have stream Id 3 on both UL and DL) and will be pre-empted when higher priority uplinks and downlinks are in progress.
   - Resumability implies that request and response shall be delivered OTA irrespective of device reboots or power outages. This allows partially downloaded large program upgrades or partially uploaded large log reads do not get thrown away due to unexpected power outages.
   - Stream Id for program upgrade should be lower than stream Id for meter firmware upgrade.
2. To support this feature, following three requests shall be supported:

A. "Meter Operations Request" with operations as defined before: An xml request received by HES/OTV is translated to an OTA message for execution at the EMCM.

B. "Flush Inprogress Transmission" is a request transmitted to the EMCM to cancel and flush any in-progress C1219OperationRsp: A user-triggered REST/UI command that will lead to an OTA request on DL stream 10.

C. "Abort Meter Operations" that results in flushing of any in-progress DL request: A cuser-triggered REST/UI command that will lead HES to request uStream to flush any inprogress C1219OperationRequest message OTA or CancelC1219OperationsRsp

3. OTV UI shall allow only one C1219Operation request till it is fully delivered or explicitly canceled by the user.

   Note that in unicast case, uStream will automatically reject a new request if previous request has not been successfully delivered. However in multicast case, active management of delivery state to all members in the group should be done by the OTV. Avoiding a new DL multicast to a group, one or few members of which are still processing the old request provides a sane solution.

4. If a second C1219Operation request is enqueued while C1219OperationRsp to a previous request has not been delivered by EMCM is permitted. Following describe the behavior on EMCM-side:

   - If fragments corresponding to new C1219OperationReq is received, then the previous C1219Operations is overwritten. In that event, EMCM shall cancel any timers for processing of previous request.

   - During the time a new C1219OperationReq is processed, EMCM shall check if C1219OperationRsp from a previous request is still under transmission. If a previous request is under transmission, it shall be cancelled and a uStream flush on UL stream 10 be generated.

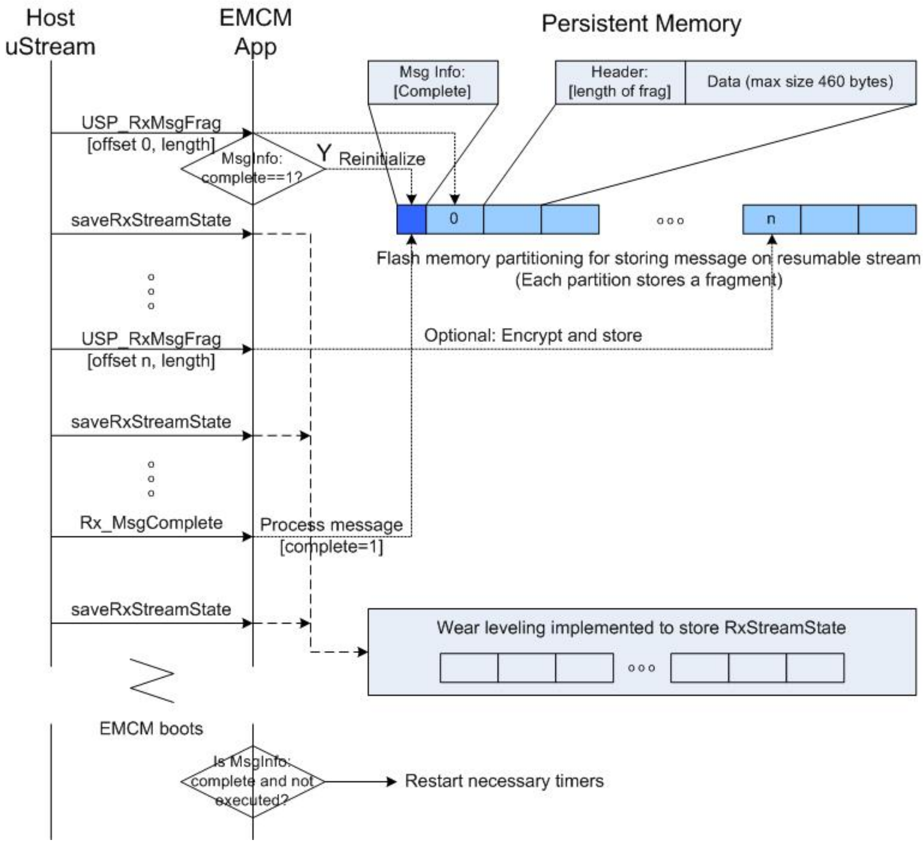
3.7.2. Handling C1219OperationRequest at EMCM

Following characterize the main issues to be addressed at EMCM with suggested solutions:

1. **Persistent storage of large size request**: C1219OperationReq is received on DL resumable stream 10 and requires that EMCM application shall be able to store the fragments (can be received out-of-order) in persistent memory along with RxStreamState (uStream state information about the state of Rx resumable streams required to persist across device reboots and uStream syncs).

   Figure below illustrates an example of how the request received on resumable stream should be handled.

Figure 12. Resumable Stream Storage and Handling



Wear-leveling is required for both RxStreamState as well as data contained in C1219OperationReq.

Recommendation: Targetted maximum size of C1219OperationReq is 50 kbytes (tentative). Since this generic request can be used to enable 3rd party features (in addition to program upgrade/configuration change), wear leveling should allow for around 10 requests/day. Further wear leveling should allow for RxStreamState to be updated approximately every 20 minutes (~72 writes per day).

2. **Persistent storage of large size response**: C1219OperationReq can lead to a large response generated (*esp.* if log tables are being retrieved). We will size the maximum response to 50 kbytes. EMCM shall be able to store it in persistent memory and provide wear leveling that allows for a maximum of 10 responses per day (even though maximum sizing is 50 kbytes, a typical use case may be for short reads). Since EMCM uses resumable stream with UL streamId 10 to send this response, TxStreamState (uStream state information about the state of Tx resumable streams required to persist across device reboots and uStream syncs) should be stored persistently and wear leveling shall account for TxStreamState updated approximately every 20 minutes (~72 writes per day).

3. **Determining execution time**: Once C1219OperationReq has been successfully received, EMCM shall read the message to start a timer for execution of the request based on startTime. EMCM shall ensure that an unexecuted request is executed post EMCM reboot as long as the time post boot falls within the range [startTime, endTime].

4. **C1219OperationReq Execution**:

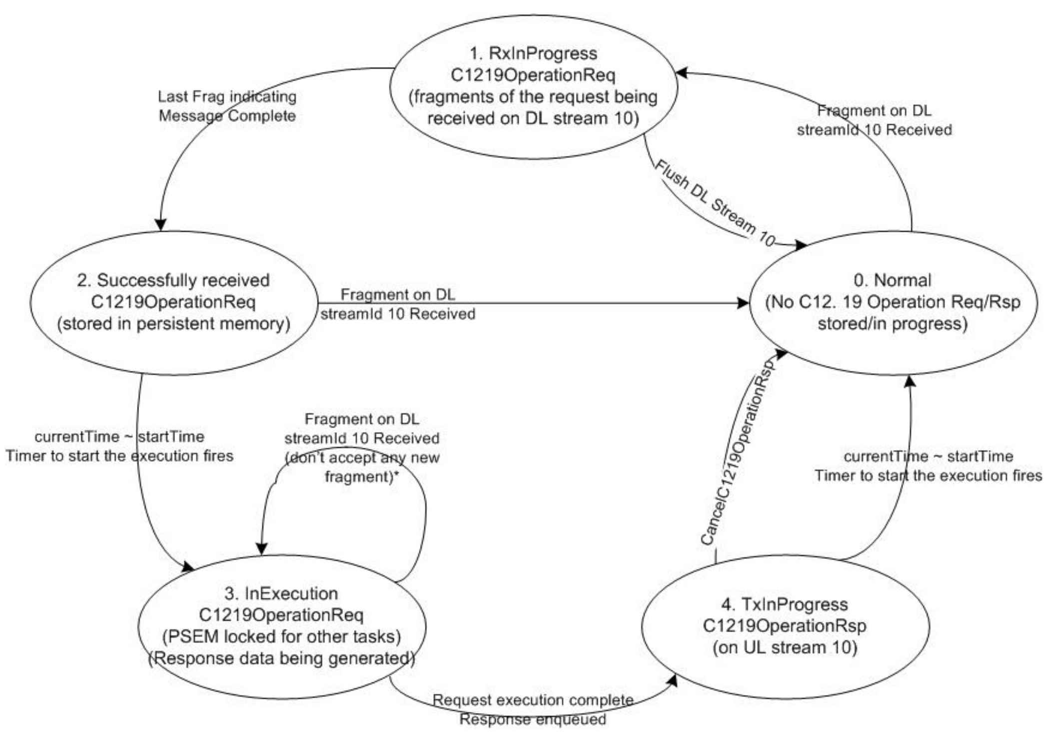
- Execution of the request on C12.18 layer shall happens as a single EMCM operation. This implies that for the duration of the time C1219OperationReq is executed (even when there is delay between suboperations), PSEM is locked and not available to other scheduled and requested events which shall wait for their turn and may fall behind their scheduled time.

- C1219OperationReq when being executed shall not be interrupted or cancelled due to application request. EMCM behavior to cancellation is described later. If EMCM boots during the request exection state but before the request is successfully executed (or before it has past number of retries), then EMCM shall re-try the execution.
- Since the request can be a long message that can not be stored or decoded in RAM, EMCM shall be able to sequentially read, decode and execute partial request data stored in persistent memory.
- Table read operations can be larger than maximum data that can be exchanged during a single C12.18 handshake. In that scnenario, EMCM is responsible for reading fragments of table data and reassembling them as a single table read. Any scratch space in persistent memory used to process the request shall be were-levelled at the same rate as C1219OperationReq/Rsp.
- Table write operations can be larger than maximum data that can be exchanged during a single C12.18 read request or RAM available to EMCM. In that scnenario, EMCM is responsible for reading fragments of table write data from persistent memory and writing them as partial writes.
- Since the response can be long consisting of sequence of executed operations with large data contents (in case of large reads), EMCM shall be able to sequentially build and store the response data in persistent memory.

3.7.3. Clearing a C12.19 Operation Request at EMCM

As described above, processing of a C12.19 operation request or transmission of C12.19 operation response can be cancelled due to an explicit request or a new C12.19 operation request. EMCM behavior is determined by the state EMCM is in as shown below.

Figure 13. Clearing C12.19 Operation processing dependent on EMCM states



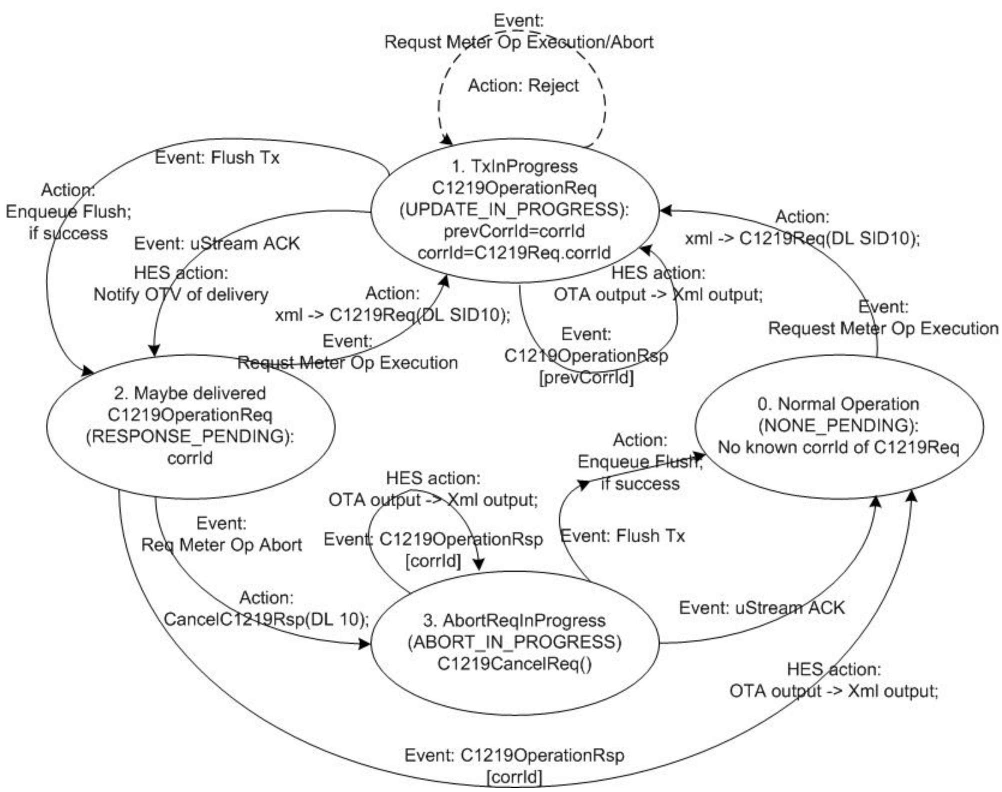
Note that when C1219OperationReq execution is in progress, EMCM application would not be able to accept fragment from ustream corresponding to a new request on DL stream 10. There are two ways to handle this:

A. EMCM provides storage for new and old C1219OperationReq. This way while new one is in progress, old one is not overwritten. Old request is overwritten only when new request has been completely received and old request is no longer in execution state 3.

B. If there is only a single memory location for C1219OperationReq, EMCM shall place the uStream in "Rx Pause" mode to stop it from delivering "rx fragments" before it starts marshalling through C1219Req for execution. Once the execution has completed, it shall re-enable delivery of rx fragments and on receipt of a single fragment, the old request is overwritten.

3.7.4. HES/OTV state transitions due to C1219Req

Figure 14. HES state transitions during C12.19 Request and cancellation



3.7.5. Multicast group handling of C12.19 Operation Request

Following is recommended on OTV to handle a C12.19 Operation Request to a group of meters:

1. For each EMCM, OTV database shall maintain a variable indicating whether C1219OperationReq is in progress (i.e. request has started and response has not been received). This variable is cleared only when either the response is received or an instruction to explicitly clear the request has been received.

   Note that OTV should NOT clear the inProgress variable in OTV DB on an instruction to clear the C12.19 operation request. It should instead be cleared by HES only after uStream indicates that in-progress transmission over DL stream 10 has either finished or flushed.

When a new C12.19 operation request is enqueued to the group, then OTV shall check that no EMCM in the group has a C12.19OperationReq in progress.

2. Once a C12.19 operation request is sent to a multicast group, removal of the EMCMs from the multicast group should not be allowed till the C12.19 operation response from all nodes in the group has been received or the request has been explictly cleared by the user.

3. Progress for each request shall be viewable. This is achievable with HES updating the OTV database with following state updates to transmission on DL stream 10:

    a. Successful enqueing of the request on DL stream 10

    b. When transmitting over multicast, uStream pushes out "handleNodeMessageProgress" that indicates request receive progress at the EMCM

    c. Successful delivery of request on DL stream 10

3.8. Power Outage Reporting

Power outage observed on a line can be sustained or momentary power outage. Sustained outage reporting is often also called "Last Gasp". Detection between the two is responsibility of the end device (aka Meter +EMCM). A detection algorithm is gauged by following two yardsticks:

A. **Probability of false alarm (Pfa)**: corresponds to falsely detecting and reporting momentary power outage as sustained power outage.

B. **Probability of missed detection (Pmiss)**: corresponds to failure to detect sustained power outage.

Our desired goal for Pfa and Pmiss are both to be less than 1%. Note that Pmiss above doesn't include probability of successfully reporting sustained outage OTA which is a function of supercap charge, geographical spread of outage and link conditions. Note that a higher Pfa is far more undesirable than higher Pmiss.

Though many meters attempt to provide this information, definition of momentary interruption differs from one utility to another. Further meter is not available to query during an outage (momentary or sustained). Since it is required that in events of sustained outage, EMCM should immediately report the outage, EMCM can not typically rely on meter to make the distinction between outages (momentary or sustained). However, in cases where EMCM may have determed momentary outage, EMCM could potentially use meter recorded "momentary interruptions" for reporting instead of EMCM determined after the power has been restored.

Following configurable parameters will characterize a momentary outage:

1 *maxMomentaryOutageDuration*: After a power fail is detected by EMCM/Meter, this is the maximum duration for which if the power is not restored, it will be labeled as sustained outage by AMI system. Desired goal of AMI 1.1 is to support a configurable value in the range of 0-120 seconds.

2 *maxMomentaryInteruptionDuration*(default=15sec): A momentary outage may consist of many back-to-back interruptions. If any single interruption is of duration longer than "maxMomentaryInteruptionDuration", then it will be considered as sustained momentary outage. Error in determining duration of momentary interruption is the source for false/missed detection.

3 *minPowerOnDuration*(default=3000msec): A momentary outage may consist of many back-to-back interruptions. Minimum duration for which power is on (aka PF line from meter is deasserted) before it is considered restored is determined by this parameter.

3.8.1. Power Outage Detection: An outline

Across differernt meters (may change as more meters are added), following three pin inputs are of interest:

1. **PWRFAIL/PF_AMI (simply PF)**: Most meters (all GE meters) provide a signal which is asserted when power to the meter is interrupted and meter doesn't want EMCM drawing current from the meter. This marks the start of outage detection in a meter.

2. **Vin**: Direct line (stepped down, DC converted) voltage from the meter which in some way is measure of input line voltage to the meter.

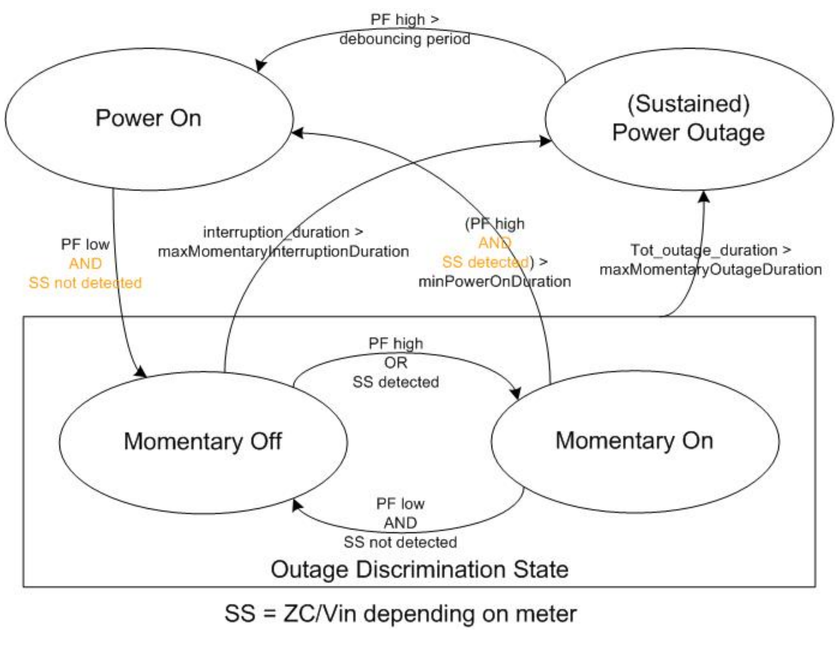
   Note that Vin above a threshold while accounting for some hysterisis can be considered as successful Vin detection.

3. **Zero-crossing (ZC)**: This may or may not be available. This is mostly a processed signal available to EMCM for every zero crossing of AC line voltage.

   Note that two consecutive rising edges within 30ms can be considered as successful ZC detection (works for both 50 Hz and 60 Hz with enough frequency error margin). This criterion could be extended to include multiple rising edges (meter dependent) if spurious edges are of concern but should be avoided as it decreases the efficacy of momentary detection when circuits are closed only for a short duration (potentially shorter than 100 ms).

We will call PF as the primary signal from EMCM indicating power fail and ZC/Vin as the secondary signal (SS).

Figure 15. State transitions on power outage detection (momentary or sustained)



In general PF (changing from high to low) is sufficient to indicate that either momentary/sustained event has happened. Note that however there are exceptions to this rule such as GE SGM3xxx where it is clearly stated "This signal also goes active (low) during internal resets of the meter." So is the case with L+G S4X which has ZC but PF by itself may simply indicate meter boot. Therefore to detect an outage (either momentary or sustained) from a non-outage, EMCM may rely on both PF AND SS to determine power outage as shown in Figure above.

Once PF is low AND SS is not detected (Implementation alert: EMCM may not be monitoring secondary signal all the time. So it can start monitoring SS after PF goes low), EMCM shall enter "Outage Discrimation" state. On entering Outage Discrimination state, EMCM should start two timers: "tot_outage_duration" and "interruption_duration" and a counter: "nbr_interruptions" (all initialized to zero). During Outage Discrimination state, following gives an outline on how the three variable are updated:

AMI 1.1.5 warning: Implementation follows spirit of above, but does not literally implement those counters as described.

- PF asserted AND SS not detected => outage exists, momentary power off. EMCM should be accumulating both tot_outage_duration and interruption_duration and increment nbr_interruptions on entering "Momentary Off" state as shown in the figure above.

- PF deasserted OR SS detected but still in "Outage Discrimation" period => outage exists, momentary power on. EMCM should increment nbr_interruptions on first detecting this condition and clear out interruption duration. EMCM shall continue accumulating tot_outage_duration.

Following criteria will be used to detect the end of Outage Discrimation period with the result being either momentary outage or sustained outage detected:

- If PF gets not asserted AND secondary signal is detected for a period greater than debouncind period +minPowerOnDuration, EMCM shall clear out both tot_outage_duration and interruption_duration and return to power on state declaring the outage event as momentatry outage event.

  AMI 1.1.5 discrepancy: implmenation uses only PF to clear outage; ZC is currently ignored for this purpose.

- Any time during Outage Discrimination state, if either tot_outage_duration > *maxMomentaryOutageDuration* or if interruption_duration > *maxMomentaryInteruptionDuration*, EMCM should declare the outage event as sustained outage and return to power fail state.

Note that ZC is prefered as secondary signal as this will greatly improve detection performance. Following information is meter-specific:

A. PF and Vin available, no ZC: GE SGM3xxx fall in this category.

AMI 1.1.5 discrepancy: use of Vin signal in this algorithm is not implemented.

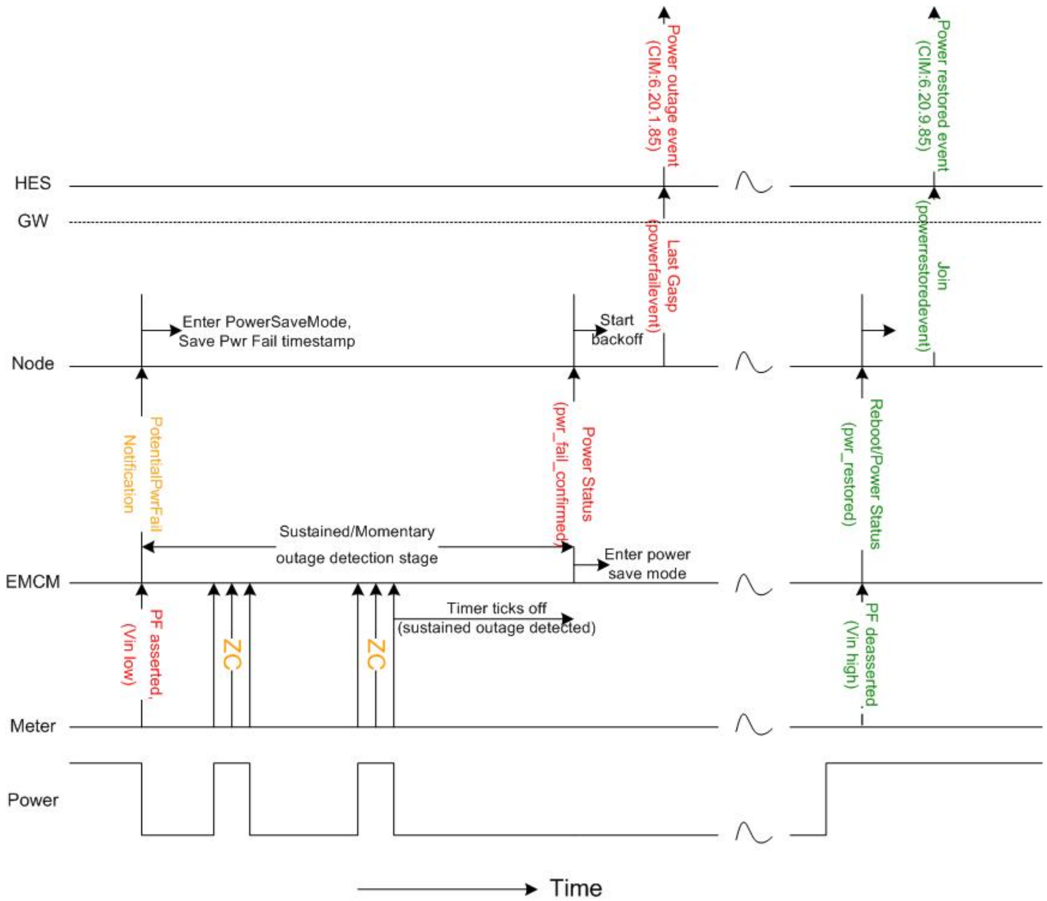
B. PF and ZC available: GE I210+c and GE kV2c fall under this category.

If sustained outage is detected, momentary interruption profile should not be reported OTA. It may be kept for debugging purposes.

3.8.2. Timeline of events in power outage reporting

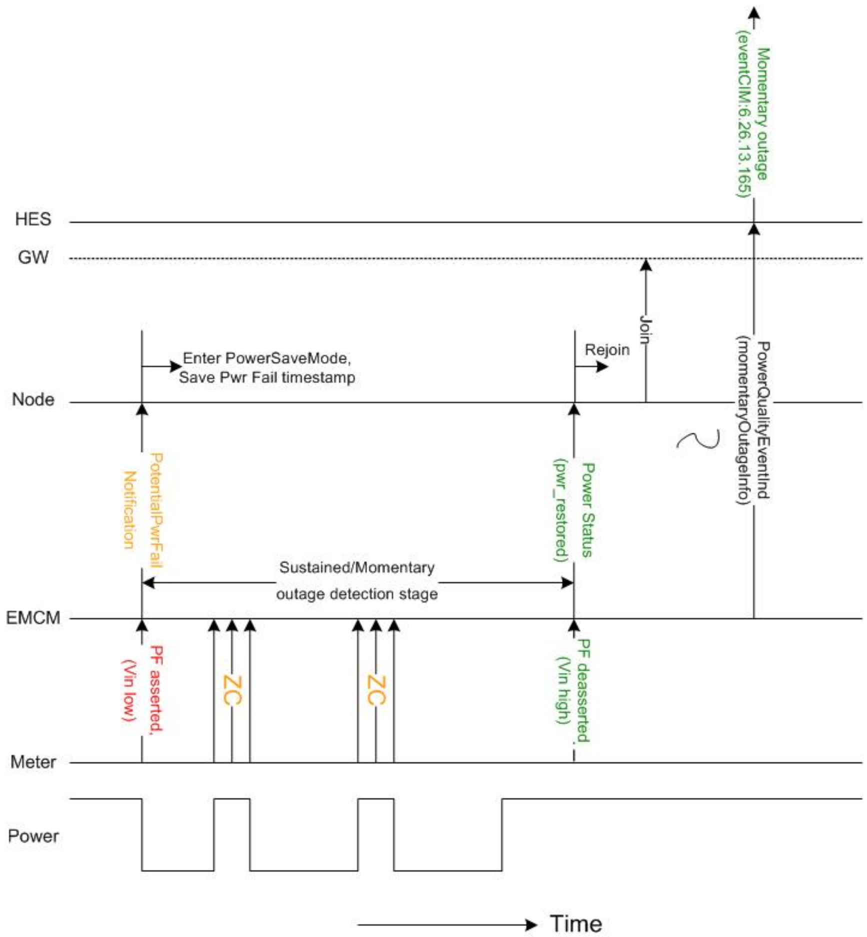
Figures below illustrate timeline of events whether momentary or sustained outage is detected. Interface messages (powerfailevent and powerrestoreevent) from GW lead HES to generate sustained power outage and power restore events (CIM codes as described in properties file: expected CIM code shown in the figure). In case momentary outage is detected, no Last Gasp is sent, and EMCM sends a momentaryOutageInfo in BootInd message that shall lead to corresponding momentary outage event generated by HES for MDMS/OTV consumption.

Figure 16. Timeline: Sustained Outage Detected



AMI 1.1.5 discrepancy: power save mode is allowed once PF is detected.
AMI 1.1.5 discrepancy: we do not send Power Restore to the node upon reboot.

Figure 17. Timeline: Momentary Outage Detected



AMI 1.1.5 bonus: power save mode is allowed once PF is detected.

EMCM sleep state: In scenarios where EMCM detects **sustained power outage**, EMCM sends a power_fail_confirmed signal to the node and goes to a *low power state* aka sleep state. In this low power state EMCM shuts of all operations except monitoring power line to be woken by a high power line signal and line from the node to process transmit status of LastGasp. In case power is restored, EMCM should query the node to check if node is still joined. If node is joined (Last Gasp didn't go through), EMCM should start a maxLastGaspTxDelay (default~5 mins) timer for Last Gasp to go through. After the timeout or after node is not joined, EMCM should sent Power Status with "power_restore" flag to indicate node to join the network.

AMI 1.1.5 discrepancy: sleep state/low power state is allowed once PF is detected.
AMI 1.1.5 discrepancy: once committed to Last Gasp, EMCM waits for 2 hours for PF_SDU_STATUS; after PF_SDU_STATUS or this timeout, EMCM reboots.

Supercap charge-discharge cycles: Repeated charging and dischraging effects supercap's life. To reduce the effect of frequent failures and interruptions on supercap's life, everytime meter toggles its PF signal, EMCM should disable supercap from charging for minChargingDisabledPeriod (default=5 mins?). To ensure that EMCM doesn't miss recharging of supercap due to boots/asserts, minChargingDisabledPeriod after boot, EMCM should ensure that supercap charging is enabled.

3.8.3. Momentary Outage Reporting

Momentary outage event information will be reported as part of power quality event in next PowerQualityEventInd message (based on PowerQualityEventInd schedule). Due to this, there could be delay from the time a momentary outage clears to when it is reported (like all other power quality events).

Following information shall be in included in a MomentaryOutageInfo:

- nbrEvents: nbrEvents indicate number of momentary outage events that have happened since last reported PowerQualityEventInd was scheduled for transmission.

- cumNumEvents: cumNumEvents indicate number of momentary outage events that have happened since persistent memory on eMCM was last initialized. cumNumEvents should roll after it reaches maximum value (max 16-bit or 32-bit?).

  Note that difference between two subsequent cumNumEvents should be equal to nbrEvents if PowerQualityEventInd was not lost due to session loss (uStream resync).

- nbrInterruptionsFirstEvent: nbrInterruptions refer to number of blinks (or momentary interruptions) that happened within in a single momentary outage event. FirstEvent refers to momentary outage event that was the first one since last PowerQualityEventInd was scheduled for transmission.

- firstEventTimestamp: Timestamp corresponding to the beginning of the momentary outage event indicated above.

Note that given the above approach, if more than one momentary outage happens within power quality event reporting schedule, then detailed information (time of event and number of interruptions) about only the first event will be available and detailed information about other events will be lost. This ensures that flood of momentary outages do not overwhelm OTR network while at the same time informing the operator about number of momentary outages that have happened.

3.8.3.1. Algorithm (Informative)

MomentaryOutageInfo data structure in persistent memory:

- nbrEvents

- cumNumEvents

- nbrInterruptionsFirstEvent

- firstEventTimestamp

Algorithm then consists of following two steps based on different triggers:

1. On every positive momentary outage detection (note that it is not applicable if outage is classified as sustained outage even if it is followed by blinks), EMCM shall

   a. If nbrEvents==0, update nbrInterruptionsFirstEvent and firstEventTimestamp based on findings of the detection algorithm outlined above;

   b. Increment nbrEvents and cumNumEvents by 1 modulo 2^16^ (or keep it 2^32^?).

2. When PowerQualityEventInd is scheduled for transmission, EMCM shall:

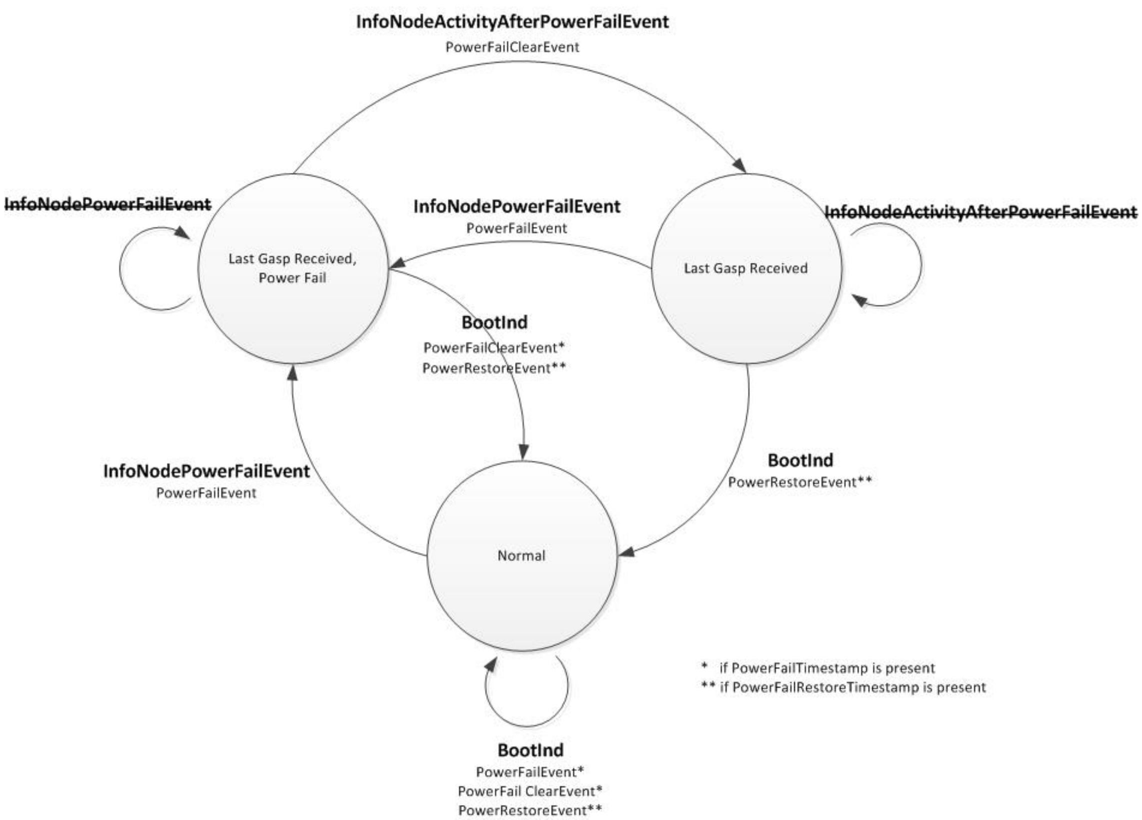
a. If nbrEvents==0, not include MomentaryOutageInfo in PowerQualityEventInd.

b. If nbrEvents>0, populate MomentaryOutageInfo with the information in the data structure and clear nbrEvents to 0.

3.8.4. Power Outage Processing at HES

Momentary Outage info is reported in PowerQualityEventInd as described in power quality section. Following behavior outlines how HES/OTV shall handle sustained outage event indications:

Figure 18. EMCM power fail state transition in HES



1. Both indications received from GW:

   a. HES receives infoNodePowerFailEvent, a non-SDU xml message over AMQP from GW for a node in "normal" operation. HES places the node/eMCM into "last gasp receied, power fail" state and generates a power fail event (CIM code defined in iec-cim properties file) which results in power fail alarm.

   b. HES receives infoNodeActivityPowerFail, a non-SDU xml over AMQP. If the eMCM is in power fail state, HES shall take it, generate power fail clear event based on which OTV shall reset power fail alarm. On generating the event, HES shall change the node state to "last gasp received" state.

   c. HES receives BootInd. Since node state in HES is "last gasp received", HES shall not generate any clear event but shall change the node state to "normal".

2. infoNodeActivityPowerFail is lost:

   a. HES receives infoNodePowerFailEvent over AMQP from GW for a node in "normal" operation. HES places the eMCM into "last gasp received, power fail" state and generates a power fail event (CIM code defined in iec-cim properties file) which results in power fail alarm.

   b. HES doesn't receive infoNodeActivityPowerFail (potentially lost over AMQP). However, after some time HES receives BootInd when the eMCM is in power fail state. HES shall generate power fail clear event and clear "last gasp received, power fail" state and change node state to "normal".

3. infoNodePowerFailEvent is lost:

   a. If HES receives "infoNodeActivityPowerFail" over AMQP from GW when the /eMCM is in "normal" operation (since infoNodePowerFailEvent was lost), HES shall ignore the message.

   b. If HES receives BootInd from the node/EMCM indicating power fail when the node is in "normal" operation, then HES shall generate power fail and power fail clear event with timestamps of power fail and power fail clear included in the BootInd. Node state stays unchanged to "normal".

In addition to the above, whenever the BootInd indicates power fail as boot reason and includes power fail clear timestamp, HES shall generate another event called 'power restore' with restoration timestamp included in bootInd. Purpose of this event is to indicate the exact time of power restore as compared to 'power fail clear' timestamp which in many cases will correspond to the time at which CCS network collector first heard from the node after power fail.

4. AMI system management via OTV

CIM/Multispeak interface provides a way for 3rd party components to be consumer of various quantities recorded by a metering system and act upon the information provided. However none of these external components (MDMS/CIS/OMS/EA etc) have functional responsibility to manage the various components of an AMI system. While EMS provides functionality for network management of CCS components, OTV is ultimately responsible for managing application-specific components. Management functionality can be broadly classified under following different categories:

1. Addition and removal of new end devices (aka meters+EMCM) to AMI system: This is managed via ingestion of a marriage file. Marriage file will determine associations and allow HES to provide services to the user of an end device. Given that this is customer-facing interfacing issue that will require modifications to adapt to specific customer requirements, it is not discussed here.

2. Configuration update of HES/OTV (local), EMCM and meters.

3. SW/FW upgrade: HES/OTV SW upgrade is local. At present EMCM firmware upgrade is supported via EMS (similar to AMI 1.0). Meter firmware upgrade is currently postponed to AMI 1.2

4. EMCM/Meters alarm management and notification reporting

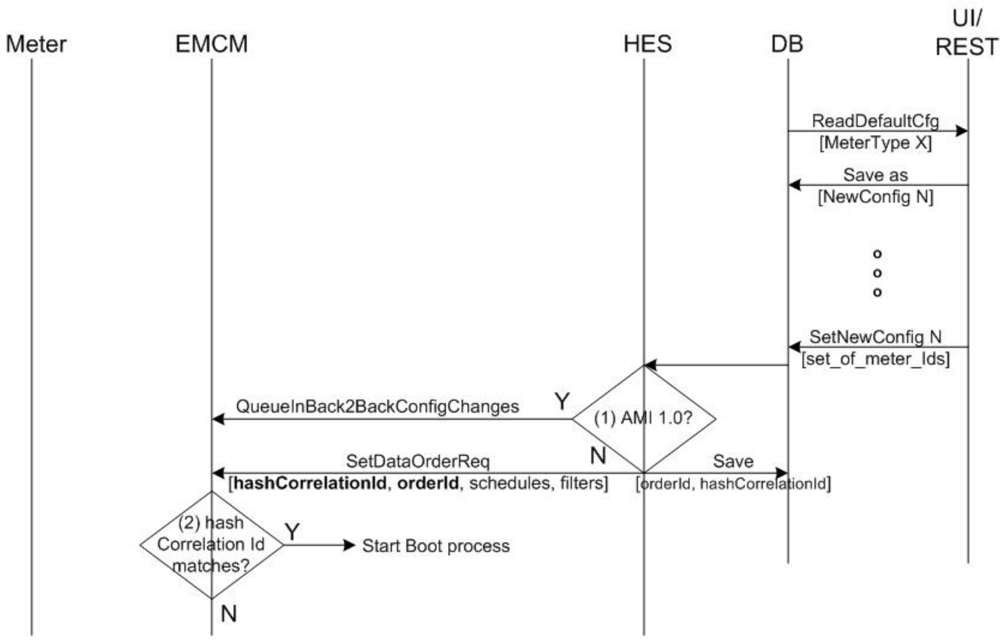
5. Diagnostics/trouble shooting

While the above list broadly classifies functions that shall be managed via OTV, it is imperative that OTV shall provide means to effectively and easily manage large number of devices of different kinds. This requires a good interface for group formation and management.

4.1. EMCM Configuration Update

EMCM configuration change is an OTV feature via which operator could specify schedules and filters for different measurement reports. Note that in AMI 1.1, to keep it simple, filter specification will be used only for quantities reported in GE I210+ fast periodicity billing data reports. Exact OTV UI details are described here: https://omega/projects/OnRampTrac/wiki/OTV/device_configuration.

Figure 19.



As shown above, an operator could modify a configuration and save it as a new file. HES shall send a SetDataOrderReq to EMCMs with AMI 1.1 firmware on receiving a command to apply new EMCM configuration. This will be associated with a new orderId. *Note that while new configuration is being transmitted OTA on DL, some periodic reports may be in air on UL with an olderorderId. To avoid the data in previous reports from losing, it is recommended that HES shall save the older configuration with older orderId.*

In case configuration change is for EMCMs on AMI 1.0, HES shall parse and send each configuration change back-to-back. Configuration that does not apply shall not be transmitted OTA.

If EMCM is authorized and SetDataOrderReq is received, EMCM shall verify hashCorrelationId matches the one it was previously assigned by HES. Mismatch shall retrigger the boot process as shown in Figure 4, “Boot sequence”.

4.2. Reported meter reads

From a usability perspective, readings corresponding to the same readingType should be displayed as separate time sequence with the most recent one showing on the front panel (#6201).

4.3. Managing AMI events

AMI events are generated due to different kinds of triggers. Some conditions are stateful which when triggered will generate an **alarm** or a **notification**. Others are non-stateful. Only a best effort notification shall be generated whenever an indication is received.

Stateful events are those for which protocol ensures that HES is in sync with latest state on the end device. Statefulness requires that state changes are reliably and without fail communicated to the backend. All best effort indications on the other hand are by default non-stateful as it is assumed that they can be lost in situations like full queues and link failures.

User does not control whether an event is stateful or not. It is a result of intentional design choices as described in this document. However user shall have control on determining whether stateful events are displayed on alarm window. An alarm window on OTV UI is a window that shows certain events such as power fail, high temperature etc (specified by user) are still active. A notification on OTV UI is not persistent and is displayed on the message window. An alarm on OTV UI shall be persistently displayed till it is cleared. This flexibility is provided to the end-user in the following way:

1. Electricami file will provide a list of "amialarms", each line specifying the corresponding set and clear CIM code.

   Note that it is expected from a trained user that non-stateful events are not configured to be displayed on the alarm window otherwise failure to indicate will lead to incorrect state of the event.

2. Notifications: Every event will be displayed as a notification on the message window.

Next we describe different categories of events observed in the AMI system as stateful or not.

1. App (aka EMCM) events: These are indications that EMCM generates such as time sycn error detected, meter communication error etc. Power outage report is special kind of app event that will be considered stateful.

2. Meter Flags (aka ST3 status indicators): All of these are stateful. Everytime HES receives meterFlagInd, it shall compare received copy with the one in its persistent memory. Any change in the bit is an event. However it may happen that clearing of a bit does not have a corresponding CIM code defined in IEC-CIM properties file. In this case, a CIM notification is not triggered.

3. Other Meter Events such as Power Quality Events: All of these (sag events/swell events) are not stateful, i.e. only notifications with E3 not equal to "1" or "9".

4. Events generated internally by HES manager: These can be stateful or not.

4.4. Group formation and maintainance

OTV shall provide a way to create group of devices based on flat/csv file. Ingestion process shall request a group name, whether it is a multicast group and if it is a subgroup of a bigger group, name of the bigger group. Following features are recommended

1. OTV should restrict the group formation to devices that belong to the same application.

2. OTV should reject the group if device_id included in the flat file is unknown (or does not belong to the group, this subgroup is part of).

3. Multicast group?: If indicated as a multicast group, OTV shall create a multicast group with the devices within the group. Devices should correspond to the application that are powered nodes on CCS 2.x.

4.5. Diagnostics and trouble shooting