

Project	IEEE 802.16 Broadband Wireless Access Working Group < http://ieee802.org/16 >	
Title	ASN.1 coding for AAI-DSx messages in IEEE 802.16.1a	
Date Submitted	2012-09-13	
Source(s)	Eunkyung Kim, Jaesun Cha, Anseok Lee, Wooram Shin, Kwangjae Lim ETRI	Voice: +82-42-860-5415 E-mail: ekkim@etri.re.kr
Re:	In response to Sponsor Ballot on P802.16.1a	
Abstract	ASN.1 coding for AAI-DSx messages in GRIDMAN Draft Standard	
Purpose	To discuss and adopt the proposed text in the draft amendment document on GRIDMAN	
Notice	<i>This document does not represent the agreed views of the IEEE 802.16 Working Group or any of its subgroups.</i> It represents only the views of the participants listed in the "Source(s)" field above. It is offered as a basis for discussion. It is not binding on the contributor(s), who reserve(s) the right to add, amend or withdraw material contained herein.	
Copyright Policy	The contributor is familiar with the IEEE-SA Copyright Policy < http://standards.ieee.org/IPR/copyrightpolicy.html >.	
Patent Policy and Procedures	The contributor is familiar with the IEEE-SA Patent Policy and Procedures: < http://standards.ieee.org/guides/bylaws/sect6-7.html#6 > and < http://standards.ieee.org/guides/opman/sect6.html#6.3 >. Further information is located at < http://standards.ieee.org/board/pat/pat-material.html > and < http://standards.ieee.org/board/pat >.	

ASN.1 coding for AAI-DSx messages in IEEE 802.16.1a

Eunkyung Kim, Jaesun Cha, Anseok Lee, Wooram Shin, Kwangjae Lim
ETRI

1. Introduction

This document provides ASN.1 encoding for AAI-DSA-REQ/RSP and AAI-DSC-REQ messages in P802.16.1a.

2. References

- [1] IEEE 802.16-12-0132-00, GRIDMAN System Requirement Document including SARM annex, January 2012.
- [2] IEEE P802.16nTM/D5, Air Interface for Broadband Wireless Access Systems - Draft Amendment: Higher Reliability Networks, June 2012.
- [3] IEEE P802.16.1aTM/D5, WirelessMAN-Advanced Air Interface for Broadband Access Systems - Draft Amendment: Higher Reliability Networks, June 2012.
- [4] IEEE P802.16TM-2012, IEEE Standard for Air Interface for Broadband Wireless Access Systems, August 2012.
- [5] IEEE P802.16.1TM-2012, IEEE Standard for WirelessMAN-Advanced Air Interface for Broadband Wireless Access Systems, September 2012.

3. Proposed Text on the IEEE 802.16.1a Amendment Draft Standard

[-----Start of Text Proposal-----]

[Remedy: Replace text in line#33-35, page 237, P802.16.1a/D5 by following]

```
-- *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*--
-- Connection management Messages *
-- *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*--
ServiceClassName ::=          IA5String (SIZE (2..128))
DataRate ::=                  INTEGER (0..4294967295)
Interval ::=                   INTEGER (0..65535)
GlobalServiceClassName ::=    SEQUENCE {
    directionIndicator          DirIndicator                OPTIONAL,
    maxSustainedRate            INTEGER (0..63)              OPTIONAL,
    maxTrafficBurst             INTEGER (0..63)              OPTIONAL,
    minReservedTrafficRate      INTEGER (0..63)              OPTIONAL,
    maxLatency                  INTEGER (0..63)              OPTIONAL,
    pagingPreference            PagingPreference            OPTIONAL,
    reqTransmissionPolicy       ReqTransmissionPolicy       OPTIONAL,
    ulGrantScheduling           UlGrantScheduling           OPTIONAL,
    toleratedJitter             INTEGER (0..63)              OPTIONAL,
    trafficPriority              INTEGER (0..7)              OPTIONAL,
    unsolicitedGrantInterval    INTEGER (0..63)              OPTIONAL,
```

```

1      unsolicitedPollingInterval    INTEGER (0..63)                OPTIONAL
2  }
3  VendorID ::=                     BIT STRING (SIZE (24))
4  QosParameterSetType ::=          BIT STRING {
5      provisionedSet                (0),
6      admittedSet                  (1),
7      activeSet                      (2)
8  } (SIZE (3))
9
10 DirIndicator ::=                 ENUMERATED {
11     uplink,
12     downlink
13 }
14 ConfirmationCode ::=             ENUMERATED {
15     successful,
16     failure
17 }
18 -- The mapping of predefined BR index used in quick access message to
19 -- BR size and BR actions
20
21 PredefinedBrIndex ::=            SEQUENCE {
22     brIndex                         INTEGER (0..15) OPTIONAL,
23     brAction                        INTEGER {
24         ertPS                        (0),
25         aGP                          (1),
26         br                            (2)
27     } (0..3)                        OPTIONAL,
28     brSize                          INTEGER (0..2047)                OPTIONAL
29 } -- bytes
30
31 UplinkQosInfo ::=                SEQUENCE {
32     ulGrantScheduling                ULGrantScheduling                OPTIONAL,
33     ulSchedulingType                 CHOICE {
34         ertPs                        SEQUENCE {
35             unsolicitedGrantInterval Interval
36         }, -- ms
37         ugs                          SEQUENCE {
38             unsolicitedGrantInterval Interval
39         }, -- ms
40         rtPs                          SEQUENCE {
41             unsolicitedPollingInterval Interval
42         }, -- ms
43         agPs                          SEQUENCE {
44             priGrantPollingInterval  Interval,                -- ms
45             secGrantPollingInterval  Interval                OPTIONAL, -- ms
46             priGrantSize              GrantSize,                -- bytes
47             secGrantSize              GrantSize                OPTIONAL, -- bytes
48             adaptationMethod          ENUMERATED {
49                 absInitiated,
50                 amsInitiated
51             }
52     }
53 }
54
55 },
56 accessClass                        INTEGER (0..3)                OPTIONAL,
57 differentiatedBrTimer              INTEGER (1..64)                OPTIONAL, -- frames
58 predefinedBrIndexList              SEQUENCE (SIZE (0..15)) OF PredefinedBrIndex OPTIONAL,
59 initialBackoffWindowSize           INTEGER (0..15)                OPTIONAL,
60 maxBackoffWindowSize               INTEGER (0..15)                OPTIONAL,
61 backoffScalingFactor               INTEGER (0..15)                OPTIONAL
62 }
63
64 QosParameter ::=                  SEQUENCE {
65

```

```

1
2     trafficPriority                INTEGER (0..7)                OPTIONAL,
3     maxSustainedRate              DataRate                OPTIONAL, -- bps
4     maxTrafficBurst               DataRate                OPTIONAL, -- bytes
5     minReservedTrafficRate       DataRate                OPTIONAL, -- bps
6     maxLaytency                  DataRate                OPTIONAL, -- ms
7     pagingPreference              PagingPreference        OPTIONAL,
8     reqTransmissionPolicy         ReqTransmissionPolicy  OPTIONAL,
9     toleratedJitter               DataRate                OPTIONAL, -- ms
10    directionalQosInfo            CHOICE {
11        downlinkInfo              NULL,
12        uplinkInfo                UplinkQosInfo
13    },
14    macInOrderDelivery            ENUMERATED {
15        notPreserved,
16        preserved
17    }                                OPTIONAL,
18    orderPreservation             CHOICE {
19        notPreservation            NULL,
20        preservation              SEQUENCE {
21            nonArqReorderTimeout   INTEGER (1..32)
22        }
23    }, -- 5ms
24    vendorId                      VendorID                OPTIONAL
25 }
26
27
28
29 GrantSize ::=                    INTEGER (0..65535)
30
31 UlGrantScheduling ::=            ENUMERATED {
32     undefined,
33     bestEffort,
34     nrtPs,
35     rtPs,
36     ertPs,
37     ugs
38 }
39
40 ReqTransmissionPolicy ::=       BIT STRING {
41     broadcastBr                   (0),
42     multicastBr                   (1),
43     piggyback                     (2),
44     noFragmentation               (3),
45     noPhs                         (4),
46     noPcking                      (5),
47     noRohc                       (6)
48 } (SIZE (7))
49
50 PagingPreference ::=            ENUMERATED {
51     noPagingGeneration,
52     pagingGeneration
53 }
54
55 DataDeliveryServices ::=        INTEGER {
56     ugs                           (0),
57     rtVrs                         (1),
58     nrtVrs                        (2),
59     be                             (3),
60     ertVrs                        (4),
61     aGPs                          (5)
62 } (0..255)
63
64 CsSpecification ::=             INTEGER {
65     packetIpv4                    (1),

```

```

1          packetIpv6                (2),
2          packetEthernet             (3),
3          packetIpv4OrIpv6          (14),
4          multiProtocol              (15),
5          talkAroundDC               (18)
6
7          } (0..255)
8 PacketErrorRate ::= SEQUENCE {
9     perMeasurement                 ENUMERATED {
10         postArqHarq,
11         beforeArqHarq
12     },
13     perValue                       CHOICE {
14         percentage                 INTEGER (0..63), -- 0 to 63%
15         negExponentialOf10         INTEGER (0..63)
16     } -- PER = 10EXP -N/10
17 }
18 ArqAttribute ::= SEQUENCE {
19     arqEnable                       ENUMERATED {
20         arqNotRequested,
21         arqRequested
22     }
23     OPTIONAL,
24     arqWindowSize                   INTEGER (1..65535)
25     OPTIONAL, --
26     arqBlockLifeTime                INTEGER (0..65535)
27     OPTIONAL, -- 100us
28     arqSyncLossTimeout              INTEGER (0..65535)
29     OPTIONAL, -- 100us
30     arqPurgeTimeout                 INTEGER (0..65535)
31     OPTIONAL, -- 100us
32     -- sub-block size = 2^(P+3), 0<=P<=7
33     arqSubBlockSize                 INTEGER (0..7)
34     OPTIONAL, -- Bytes
35     arqErrorDetectionTimeout        INTEGER (0..65535)
36     OPTIONAL, -- 100s
37     arqFeedbackPollRetryTimeout    INTEGER (0..65535)
38     OPTIONAL -- 100s
39 }
40 IpAddrAndMask ::= CHOICE {
41     ipv4                            SEQUENCE {
42         ipAddr                      BIT STRING (SIZE (32)),
43         ipMask                      BIT STRING (SIZE (32))
44     },
45     ipv6                            SEQUENCE {
46         ipAddr                      BIT STRING (SIZE (128)),
47         ipMask                      BIT STRING (SIZE (128))
48     }
49 }
50 ProtocolPortRange ::= SEQUENCE {
51     lowPortNumber                   IPPortNumber,
52     highPortNumber                  IPPortNumber
53 }
54 ClassificationRule ::= SEQUENCE {
55     priority                         INTEGER (0..255)
56     OPTIONAL,
57     protocolField                    INTEGER (0..255)
58     OPTIONAL,
59     ipMaskAndSrcAddr                 IpAddrAndMask
60     OPTIONAL,
61     ipMaskAndDestAddr                IpAddrAndMask
62     OPTIONAL,
63     protocolSrcPortRange             ProtocolPortRange
64     OPTIONAL,
65     protocolDestPortRange            ProtocolPortRange
66     OPTIONAL,
67     associatedPhsiIndex              INTEGER (0..255)
68     OPTIONAL,
69     classificationRuleIndex          INTEGER (0..65535)
70     OPTIONAL,
71     vendorID                         VendorID
72     OPTIONAL,
73     ipv6FlowLabel                    BIT STRING (SIZE (24))
74     OPTIONAL,
75     classificationAction              ENUMERATED {
76         none,
77         discard
78     }
79     OPTIONAL,

```

```

1      dscpValue                INTEGER (0..63)                OPTIONAL,
2      phsDscAction             PHSDSCAction             OPTIONAL,
3      phsRuleAdded             PhsRuleActions           OPTIONAL,
4      ethernetAttributes       EthernetAttributes       OPTIONAL,
5      innerIpProtocolInfo      InnerIpProtocolInfo   OPTIONAL
6  }
7  InnerIpProtocolInfo ::=      SEQUENCE {
8      innerSrcIpAddr           BIT STRING (SIZE (32))        OPTIONAL,
9      innerDestIpAddr         BIT STRING (SIZE (32))        OPTIONAL,
10     innerProtocolField       BIT STRING (SIZE (8))         OPTIONAL,
11     innerTos                 BIT STRING (SIZE (8))         OPTIONAL,
12     innerIpv6FlowLable       BIT STRING (SIZE (8))         OPTIONAL
13 }
14 }
15 PHSDSCAction ::=            ENUMERATED {
16     addPHSRule,
17     setPHSRule,
18     deletePHSRule,
19     deleteAllPHSRules
20 }
21 PhsRuleActions ::=          CHOICE {
22     addPhsRule                PhsRule,
23     setPhsRule                PhsRule,
24     deletePhsRule             INTEGER (0..255),
25     deleteAllPhsRules         NULL
26 }
27 }
28 PhsRule ::=                  SEQUENCE {
29     phsiField                 INTEGER (0..255)                OPTIONAL,
30     phsfField                 OCTET STRING (SIZE (0..255))        OPTIONAL,
31     -- phsmField[x] = 0 don't surpress x's byte,
32     -- phsmField[x] = 1 subpress x's byte
33     phsmField                 BIT STRING (SIZE (0..255))        OPTIONAL,
34     phsvField                 ENUMERATED {
35         verify,
36         doNotVerify
37     }
38     vendorSpecificPHSParameters OCTET STRING (SIZE (0..999))    OPTIONAL
39 }
40 }
41 EthMacAddrAndMask ::=        SEQUENCE {
42     ethMacAddress             MACAddress,
43     ethMaskAddress            MACAddress
44 }
45 }
46 EtherType ::=                ENUMERATED {
47     noLayer3Matching,
48     dixSnap,
49     dsap
50 }
51 EtherType16Bits ::=          SEQUENCE {
52     eprot1                    BIT STRING (SIZE (16)),
53     eprot2                    BIT STRING (SIZE (16))
54 }
55 }
56 EtherType8Bits ::=           SEQUENCE {
57     eprot1                    BIT STRING (SIZE (8)),
58     eprot2                    BIT STRING (SIZE (8))
59 }
60 Ieee802Dot2Sap ::=           SEQUENCE {
61     ethType                   EtherType,
62     layer3ProtocolId          CHOICE {
63         noL3Matching          NULL,
64         dixSnapEncp           EtherType16Bits,
65

```

```

1      dsapEncp                      EtherType8Bits
2      }
3  }
4  UserPriority ::= SEQUENCE {
5      priorityLow                    INTEGER (0..7),
6      priorityHigh                   INTEGER (0..7)
7  }
8
9  EthernetAttributes ::= SEQUENCE {
10     ethMaskAndDestMacAddr          EthMacAddrAndMask          OPTIONAL,
11     ethMaskAndSrcMacAddr           EthMacAddrAndMask          OPTIONAL,
12     ieee802Dot2Sap                 Ieee802Dot2Sap           OPTIONAL,
13     userPriority                    UserPriority              OPTIONAL,
14     vlanId                          BIT STRING (SIZE (12))      OPTIONAL
15 }
16 RohcAttributes ::= SEQUENCE {
17     maxContextId                    INTEGER (0..65535)        OPTIONAL,
18     largeContextId                  INTEGER {
19         smallCid                    (0),
20         largeCid                    (1)
21     } (0..255)            OPTIONAL,
22     -- 0: no segmentation,
23     -- 1..65535 Maximum reconstructed reception unit in Byte
24     rohcMrru                        INTEGER (0..65535)        OPTIONAL
25 }
26
27 EMBSService ::= BIT STRING {
28     embsInServingABOnly             (0),
29     embsInZoneWithMacroDiversity    (1),
30     embsInZoneWithoutMacroDiversity (2)
31 } (SIZE (3))
32
33 ClassifierDSCAction ::= ENUMERATED {
34     dscAddClassifier,
35     dscReplaceClassifier,
36     dscDeleteClassifier
37 }
38
39 GroupID ::= INTEGER (0..4095)
40 GroupResourceAllocInfo ::= SEQUENCE {
41     groupID                          GroupID,
42     longTTIIndicator                  ENUMERATED {
43         oneAAISubframe,
44         fourOrAllAAISubframes
45     },
46     periodicity                       ENUMERATED {
47         oneFrame,
48         twoFrames,
49         fourFrames,
50         eightFrames
51     },
52     mimoModeSet                       ENUMERATED {
53         mode0,
54         mode1And2,
55         mode2,
56         mode2And4
57     },
58     userBitmapSize                    ENUMERATED {
59         size4,
60         size8,
61         size16,
62         size32
63     },
64 }
65

```

```

1      userBitmapIndex          INTEGER (0..31),
2      initialACID              INTEGER (0..15),
3      numberOfACIDs            INTEGER (0..7),
4      resourceSizeInclusionBitmap BIT STRING (SIZE (16))
5  }
6  SleepRspCode ::=
7      ENUMERATED {
8          unsolicitedRequest,
9          slpReqApproved,
10         slpReqRejected
11     }
12  FastFbkChannelOp ::=
13      ENUMERATED {
14          fastFbkChKept,
15          fastFbkChDeallocAtFrame,
16          autoFastFbkChDealloc
17     }
18  SleepCycleSetting ::=
19      CHOICE {
20         request          SleepReqAction,
21         response         SleepRspAction
22     }
23  SleepReqAction ::=
24      SEQUENCE {
25         operation        CHOICE {
26             switchSleepCycle SEQUENCE {
27                 sleepCycleID          SCID,
28                 startFrameNumber      INTEGER (0..63)
29             },
30             changeSleepCycle SEQUENCE {
31                 sleepCycleID          SCID,
32                 startFrameNumber      INTEGER (0..63),
33                 trafficIndicationFlag TrafficIndicationFlag,
34                 listeningWindowExtFlag ListeningWindowExtFlag,
35                 nextSleepCycleIndicator NextSleepCycleIndicator,
36                 initialSleepCycle     INTEGER (0..15),
37                 finalSleepCycle       INTEGER (0..1023),
38                 listeningWindow       INTEGER (0..63),
39                 newInitialSleepCycle  INTEGER (0..31)          OPTIONAL,
40                 tAMS                  INTEGER (0..31)          OPTIONAL
41             }
42         }
43     }
44  SleepRspAction ::=
45      SEQUENCE {
46         responseCode      SleepRspCode,
47         operation        CHOICE {
48             switchSleepCycle SEQUENCE {
49                 sleepCycleID          SCID          OPTIONAL,
50                 startFrameNumber      INTEGER (0..63) OPTIONAL,
51                 ffbchOperation        FastFbkChannelOp OPTIONAL
52             },
53             changeSleepCycle SEQUENCE {
54                 sleepCycleID          SCID          OPTIONAL,
55                 startFrameNumber      INTEGER (0..63) OPTIONAL,
56                 ffbchOperation        FastFbkChannelOp OPTIONAL,
57                 trafficIndicationFlag TrafficIndicationFlag OPTIONAL,
58                 listeningWindowExtFlag ListeningWindowExtFlag OPTIONAL,
59                 nextSleepCycleIndicator NextSleepCycleIndicator OPTIONAL,
60                 initialSleepCycle     INTEGER (0..15) OPTIONAL,
61                 finalSleepCycle       INTEGER (0..1023) OPTIONAL,
62                 listeningWindow       INTEGER (0..63) OPTIONAL,
63                 slpId                 SLPID          OPTIONAL,
64                 newInitialSleepCycle  INTEGER (0..31) OPTIONAL,
65

```



```

1          tAMS                                INTEGER (0..31)                OPTIONAL
2      }
3  } OPTIONAL,
4  reqDuration                                INTEGER (0..255)                OPTIONAL
5  }
6
7  HRMulticastService ::= BIT STRING {
8      hrMulticastInServingABSONly            (0),
9      hrMulticastInMultiBSZone                (1)
10     } (SIZE (2))
11
12
13  HRMulticastFlowAddition ::= SEQUENCE (SIZE (1..16)) OF SEQUENCE {
14      hrMulticastGroupID                      HRMulticastGroupID,
15      fID                                       FID,
16      feedbackReqInd                           CHOICE {
17          feedbackReqInd                       BOOLEAN,
18          feedbackResInd                       ENUMERATED {
19              ackOnly,
20              nackOnly,
21              noFeedback
22          }
23      } OPTIONAL,
24      -- 0: ACK only, 1: NACK only
25      logicalChannlInd                          INTEGER (0..4095)                OPTIONAL,
26      probInd                                    INTEGER (0..1023)                OPTIONAL
27  }
28
29  HRMulticastFlowChange ::= SEQUENCE (SIZE (1..16)) OF SEQUENCE {
30      currentHRMulticastGroupID                HRMulticastGroupID,
31      currentFID                               FID,
32      newHRMulticastGroupID                    HRMulticastGroupID,
33      newFID                                    FID
34  }
35
36  HRMulticastFlowDelete ::= SEQUENCE (SIZE (1..16)) OF SEQUENCE {
37      hrMulticastGroupID                      HRMulticastGroupID,
38      fID                                       FID
39  }
40
41  HRMulticastSupportedAdd ::= SEQUENCE {
42      hrMulticastGroupZoneId                  HRMulticastGroupZoneID,
43      hrMulticastIndicationCycle              HRMulticastIndicationCycle    OPTIONAL,
44      hrMulticastInfo                          HRMulticastFlowAddition      OPTIONAL
45  }
46
47  HRMulticastSupportedChange ::= SEQUENCE {
48      hrMulticastGroupZoneId                  HRMulticastGroupZoneID      OPTIONAL,
49      hrMulticastIndicationCycle              HRMulticastIndicationCycle    OPTIONAL,
50      hrMulticastInfo                          HRMulticastFlowChange        OPTIONAL
51  }
52
53  HRMulticastSupportedDelete ::= SEQUENCE {
54      hrMulticastInfo                          HRMulticastFlowDelete        OPTIONAL
55  }
56
57  -- +-----+
58  -- DSA Request
59  -- +-----+
60  AbsInitDsaInfo ::= SEQUENCE {
61      sfid                                     SFID,
62      fid                                       FID
63  }
64  UnicastAvailIntervalBitmap ::= CHOICE {
65

```

```

1      nmsi4                BIT STRING (SIZE (4)),
2      nmsi8                BIT STRING (SIZE (8)),
3      nmsi16               BIT STRING (SIZE (16)),
4      nmsi32               BIT STRING (SIZE (32))
5  }
6  AdditionalSfInfo ::=    SEQUENCE {
7      qosParameterSet      QosParameterSetType      OPTIONAL,
8      serviceName          ServiceClassName          OPTIONAL,
9      globalServiceClass   GlobalServiceClassName    OPTIONAL,
10     csSpecificationType   CsSpecification          OPTIONAL,
11     dataDeliveryServices  DataDeliveryServices    OPTIONAL,
12     arqAttributes         ArqAttribute              OPTIONAL,
13     classificationRules   ClassificationRule        OPTIONAL,
14     rohcAttributes        RohcAttributes            OPTIONAL,
15     sduInterArrival       Interval                  OPTIONAL, -- 0.5ms
16     timeBase              Interval                  OPTIONAL, -- ms
17     sduSize               INTEGER (0..255)          DEFAULT 49,
18     targetSaid            INTEGER (0..255)          OPTIONAL,
19     packetErrorRate       PacketErrorRate          OPTIONAL,
20     macHeaderType         ENUMERATED {
21         agmh,
22         spmh
23     }
24     } OPTIONAL
25 }
26
27 GroupParaCreateChange ::= SEQUENCE {
28     commonQosParameters   SEQUENCE {
29         qosParameter      QosParameter              OPTIONAL,
30         additionalSfInfo  AdditionalSfInfo            OPTIONAL
31     },
32     qtySfid               INTEGER (1..32)              OPTIONAL,
33     groupFidList          SEQUENCE (SIZE (1..32)) OF SEQUENCE {
34         fid                FID
35     },
36     groupFidParameterArray SEQUENCE (SIZE (1..32)) OF SEQUENCE {
37         fid                FID,
38         nonCommonQosParameters SEQUENCE {
39             qosParameter      QosParameter              OPTIONAL,
40             additionalSfInfo  AdditionalSfInfo            OPTIONAL
41         }
42     } OPTIONAL
43 }
44
45 CoupledGroupCreateChange ::= SEQUENCE {
46     commonQosParameters   SEQUENCE {
47         qosParameter      QosParameter              OPTIONAL,
48         additionalSfInfo  AdditionalSfInfo            OPTIONAL
49     },
50     qtyCoupledSfid        INTEGER (0..32)              OPTIONAL,
51     coupledFidArray       SEQUENCE (SIZE (1..16)) OF SEQUENCE {
52         fid                FID
53     },
54     coupledNonCommonFidArray SEQUENCE (SIZE (1..16)) OF SEQUENCE {
55         fid                FID,
56         nonCommonQosParameters SEQUENCE {
57             qosParameter      QosParameter              OPTIONAL,
58             additionalSfInfo  AdditionalSfInfo            OPTIONAL
59         }
60     } OPTIONAL
61 }
62
63 AAI-DSA-REQ ::=          SEQUENCE {
64
65

```

```

1      fidChangeCount          FidChangeCount,
2      absInitDsaInfo          AbsInitDsaInfo          OPTIONAL,
3      directionIndicator      DirIndicator,
4      qosParameters           QosParameter           OPTIONAL,
5      additionalSfInfo        AdditionalSfInfo        OPTIONAL,
6      emergencyIndication     BOOLEAN           OPTIONAL,
7      embsService             EMBSService           OPTIONAL,
8      fullEMBSIDArray         SEQUENCE (SIZE (1..8)) OF SEQUENCE {
9          embsZoneID          EMBSZoneID,
10         carrierIndex        PhyCarrierIndex,
11         embsidFIDMappingList SEQUENCE (SIZE (1..15)) OF SEQUENCE {
12             embsid          EMBSID,
13             fid             FID
14         }
15     } OPTIONAL,
16     unicastAvailIntervalBitmap UnicastAvailIntervalBitmap    OPTIONAL,
17     groupParameterCreateChange GroupParaCreateChange        OPTIONAL,
18     coupledGroupCreateChange  CoupledGroupCreateChange    OPTIONAL,
19     multicastGroup            SEQUENCE (SIZE (1..16)) OF SEQUENCE {
20         multicastGroupId      MulticastGroupID,
21         fid                   FID
22     } OPTIONAL,
23     -- for HR-Network
24     hrMulticastService        HRMulticastService        OPTIONAL,
25     hrMulticastSupportedAdd   HRMulticastSupportedAdd   OPTIONAL,
26     sleepCycleSetting         SleepCycleSetting         OPTIONAL,
27     harqChannelsList          SEQUENCE (SIZE (1..16)) OF INTEGER (0..15) OPTIONAL,
28     -- for HR-Network
29     twdcAddressforTX          INTEGER (0..4095)          OPTIONAL,
30     twdcAddressforRX          INTEGER (0..4095)          OPTIONAL,
31     fbisConInd                BOOLEAN                      OPTIONAL,
32     ...
33 }
34 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
35 -- DSA Response
36 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
37
38 AAI-DSA-RSP ::= SEQUENCE {
39     fidChangeCount          FidChangeCount,
40     confirmationCode        ConfirmationCode,
41     fid                     FID          OPTIONAL,
42     groupParameterCreateChange SEQUENCE {
43         fidArray            SEQUENCE (SIZE (1..16)) OF SEQUENCE {
44             fid             FID          OPTIONAL
45         }
46     } OPTIONAL,
47     embsService             EMBSService           OPTIONAL,
48     fullEMBSIDArray         SEQUENCE (SIZE (1..8)) OF SEQUENCE {
49         embsZoneID          EMBSZoneID,
50         carrierIndex        PhyCarrierIndex        OPTIONAL,
51         embsidFIDMappingArray SEQUENCE (SIZE (1..15)) OF SEQUENCE {
52             embsid          EMBSID,
53             fid             FID
54         }
55     } OPTIONAL,
56     carrierSwitching        CHOICE {
57         unicastAvailIntervalBitmap UnicastAvailIntervalBitmap,

```

```

1         aaiEmbsRepMsg                NULL
2     } OPTIONAL,
3     multicastGroup                    SEQUENCE (SIZE (1..16)) OF SEQUENCE {
4         multicastGroupId              MulticastGroupID
5     } OPTIONAL,
6
7
8     -- for HR-Network
9     hrMulticastService                HRMulticastService                OPTIONAL,
10    hrMulticastSupportedAdd          HRMulticastSupportedAdd          OPTIONAL,
11
12    sleepCycleSetting                  SleepCycleSetting                OPTIONAL,
13
14    fbisConInd                        BOOLEAN                          OPTIONAL,
15    ...
16 }
17 -- +-----+
18 -- DSA Acknowledge
19 -- +-----+
20 AAI-DSA-ACK ::=                      SEQUENCE {
21     fidChangeCount                    FidChangeCount,
22     confirmationCode                  ConfirmationCode,
23     ...
24 }
25 -- +-----+
26 -- DSC Request
27 -- +-----+
28
29
30 AbsInitDscInfo ::=                  SEQUENCE {
31     sfid                              SFID
32 }
33
34 AAI-DSC-REQ ::=                     SEQUENCE {
35     fidChangeCount                    FidChangeCount,
36     absInitDscInfo                    AbsInitDscInfo                OPTIONAL,
37     directionIndicator                 DirIndicator,
38     serviceName                       ServiceClassName                OPTIONAL,
39     globalServiceClass                 GlobalServiceClassName          OPTIONAL,
40     qosParameterSet                    QosParameterSetType            OPTIONAL,
41     qosParameters                      QosParameter                    OPTIONAL,
42     sduInterArrival                    Interval                        OPTIONAL, -- 0.5ms
43     timeBase                            Interval                        OPTIONAL, -- ms
44     classifierDSCAction                 ClassifierDSCAction              OPTIONAL,
45     classificationRules                 ClassificationRule                OPTIONAL,
46     rohcAttributes                     RohcAttributes                  OPTIONAL,
47     packetErrorRate                    PacketErrorRate                  OPTIONAL,
48     emergencyIndication                 BOOLEAN                          OPTIONAL,
49     embsService                         EMBSservice                      OPTIONAL,
50     fullEMBSIDArray                    SEQUENCE (SIZE (1..8)) OF SEQUENCE {
51         embsZoneID                     EMBSZoneID,
52         newEmbsZoneID                   EMBSZoneID,
53         carrierIndex                    PhyCarrierIndex                OPTIONAL,
54         serviceFlowUpdateType           CHOICE {
55             bitmapAndNew                SEQUENCE {
56                 serviceFlowUpdateBitmap BIT STRING (SIZE (16)),
57                 embsidFIDMappingArray  SEQUENCE (SIZE (1..16)) OF SEQUENCE {
58                     newEMBSID          EMBSID,
59                     newFID              FID
60                 }
61             }
62     },
63     currentAndNew                       SEQUENCE (SIZE (1..16)) OF SEQUENCE {

```

```

1          currentEMBSID           EMBSID,
2          currentFID             FID,
3          newEMBSID             EMBSID,
4          newFID               FID
5      }
6  }
7  } OPTIONAL,
8  unicastAvailIntervalBitmap   UnicastAvailIntervalBitmap   OPTIONAL,
9  groupParameterCreateChange   GroupParaCreateChange       OPTIONAL,
10 coupledGroupCreateChange      CoupledGroupCreateChange    OPTIONAL,
11 multicastGroupAddition         SEQUENCE (SIZE (1..16)) OF SEQUENCE {
12      multicastGroupAddedId     MulticastGroupID
13  } OPTIONAL, -- multicast group id to be added
14  hrMulticastGroupZoneIdChange   HRMulticastGroupZoneID   OPTIONAL,
15  hrMulticastIndicationCycleChange  HRMulticastIndicationCycle OPTIONAL,
16  hrMulticastSupportedAdd         HRMulticastSupportedAdd     OPTIONAL,
17  hrMulticastSupportedChange      HRMulticastSupportedChange OPTIONAL,
18
19  multicastGroupDeletion        SEQUENCE (SIZE (1..16)) OF SEQUENCE {
20      multicastGroupDeletedId    MulticastGroupID
21  } OPTIONAL, -- multicast group id to be deleted
22  hrMulticastSupportedDelete      HRMulticastSupportedDelete   OPTIONAL,
23  sleepCycleSetting            SleepCycleSetting           OPTIONAL,
24  initialBackoffWindowSize     INTEGER (0..15)             OPTIONAL,
25  maxBackoffWindowSize         INTEGER (0..15)             OPTIONAL,
26  backoffScalingFactor         INTEGER (0..15)             OPTIONAL,
27  harqChannelsList            SEQUENCE (SIZE (1..16)) OF INTEGER (0..15) OPTIONAL,
28  ...
29  }
30 }
31 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
32 -- DSC Response
33 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
34
35
36 AAI-DSC-RSP ::= SEQUENCE {
37     fidChangeCount            FidChangeCount,
38     sfid                      SFID                OPTIONAL,
39     confirmationCode          ConfirmationCode,
40     groupParameterCreateChange SEQUENCE {
41         fidArray              SEQUENCE (SIZE (1..16)) OF SEQUENCE {
42             fid                FID                OPTIONAL
43         }
44     } OPTIONAL,
45     sleepCycleSetting         SleepCycleSetting    OPTIONAL,
46     ...
47 }
48 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
49 -- DSC Acknowledge
50 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
51
52 AAI-DSC-ACK ::= SEQUENCE {
53     fidChangeCount            FidChangeCount,
54     confirmationCode          ConfirmationCode,
55     ...
56 }
57 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
58 -- DSD Request
59 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
60
61 DsdEmbsInfo ::= SEQUENCE {
62     embsZoneID                EMBSZoneID,
63     embsidFIDMappingArray     SEQUENCE (SIZE (1..15)) OF SEQUENCE {
64         embsid                EMBSID,
65
```

```

1          fid          FID
2      }
3  }
4
5  AAI-DSD-REQ ::=          SEQUENCE {
6      fidChangeCount      FidChangeCount,
7      fid                  FID OPTIONAL,
8      dsdEmbsInfo         DsdEmbsInfo          OPTIONAL,
9      sleepCycleSetting   SleepCycleSetting    OPTIONAL,
10     ...
11 }
12
13 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
14 -- DSD Response
15 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
16 AAI-DSD-RSP ::=          SEQUENCE {
17     fidChangeCount      FidChangeCount,
18     fid                  FID
19     confirmationCode    ConfirmationCode,          OPTIONAL,
20     sleepCycleSetting   SleepCycleSetting          OPTIONAL,
21     ...
22 }
23
24 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
25 -- Group Configuration
26 -- +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
27
28 [-----End of Text Proposal-----]
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```