# Central Coordinator

## Power-On Network Discovery Procedure

As with HomePlug AV, a GREEN PHY STA performs the Power-on Network Discovery Procedure to determine whether another HomePlug network is active and if a new AVLN can be instantiated. All STAs shall be capable of performing the Power-On Network Discovery Procedure.

Prior to initiating the Power-On Network Discovery Procedure, the GREEN PHY STA will select a BeaconBackoffTime (BBT). If the STA was the CCo of an AVLN before it was powered down, BBT should be chosen as a random value in the interval (MinCCoScanTime, MaxCCoScanTime). Otherwise, BBT shall be a random value in the interval (MinScanTime, MaxScantime). BBT is the maximum duration of time for which a STA will execute the Power-on Network Procedure.

During the Power-on Network Discovery Procedure, if one or more AVLNs are detected, the STAs shall attempt to transmit the CM\_UNASSOCIATED\_STA.IND MME using multi-network broadcast approximately once per Unassociated STA Advertisement Interval (USAI) to provide information to other stations that may be performing the same procedure. The transmission time of each CM\_UNASSOCIATED\_STA.IND MME (i.e., Unassociated Station Tranmission Time (USTT)) shall be randomly chosen (refer to Section 5.4.3.1). Further, the STA shall synchronize its PhyClk to one of the detected AVLNs, and shall use the SNID of that AVLN in the Multi-Network Broadcast transmissions. This provides a way for other STAs that also hear the same AVLN to apply the appropriate PhyClk correction for reception.

During the Power-On Network Discovery Procedure, while no AVLNs are detected, the STA shall search for Beacons and other unassociated STAs using Hybrid mode. After an AVLN is detected, the STA may adopt the HomePlug 1.0/1.1 coexistence mode of that AVLN. Optionally, the STA may continue to search in Hybrid mode (irrespective of the mode of the AVLN that it detected). This optional behavior will enable the STA to continue to detect Beacons of non-coordinating AVLNs (if any). It is important to note that proper execution of the CSMA/CA channel access mechanism requires the STA to perform virtual carrier sense in the mode of the Shared CSMA region (i.e., AV-only or Hybrid mode) of the AVLN(s) being tracked. Thus, if the Shared CSMA region is in AV-Only mode and the STA is searching in Hybrid mode, it has to transition to AV-only mode during the AVLNs Shared CSMA region as soon as it has a pending transmission. The STA should also transition back to Hybrid mode after the transmission is completed.

During the Power-On Network Discovery Procedure, detection of an AVLN with matching NID causes the unassociated STA to follow the procedure described in Section 7.3.5 for joining the AVLN . If the STA successfully joins the AVLNs, it shall terminate the power-on network procedure and become a STA in the AVLN (refer to Section 7.2.3). Failure to join successfully shall cause the STA to continue with the power-on network procedure.

Upon the expiration of the BBT, the STA shall process all the received Unassociated STA information (if any) as follows:

* If the STA detects other Unassociated STAs with a matching NID, it shall use the procedure described in Section 7.4.1 to determine whether it has to become the CCo and instantiate a new AVLN.
* If the STA detects no other AVLNs and there are no Unassociated STAs with matching NID, it shall become an Unassociated CCo (refer to Section 7.2.1),

In all other cases, the STA shall operate as an Unassociated STA (refer to Section 7.2.1).

A STA that failed to form or join an AVLN during the Power-on procedure shall operate in Unassociated STA Mode or Unassociated CCo Mode. Both these modes are generically called Unassociated STA Mode. Section 7.2.1 and Section 0 provide details on the Unassociated STA Mode and Unassociated CCo Mode respectively.

Figure 7‑1 shows the basic functional flow for Power-on Network Discovery Procedure.



Figure 7‑1: Power-on Network Discovery Procedure

## STA Behavior After Power-on

Once the power-on procedure is completed, a STA can be an Unassociated STA, Unassociated CCo, STA of an AVLN or CCo of an AVLN. This section describes the behavior of STA in each of these modes.

### Unassociated STA Behavior

An Unassociated STA that detects other AVLNs shall continue to send CM\_UNASSOCIATED\_STA.IND MME using multi-network broadcast approximately once per MaxDiscoverPeriod to provide information to other stations. The transmission time of each CM\_UNASSOCIATED\_STA.IND MME shall be randomly chosen. Further, the STA shall synchronize its PhyClk to one of the detected AVLNs, and shall use the SNID of that AVLN in the Multi-Network Broadcast transmissions. This provides a way for other STAs that also hear the same AVLN to apply the appropriate PhyClk correction for reception.

Detection of an AVLN with matching NID causes the unassociated STA to follow the procedure described in Section 7.3.5 for joining the AVLN. If the STA successfully joins the AVLN, it shall become a STA in the AVLN. Failure to join successfully shall cause the STA to continue operating as an Unassociated STA.

If an Unassociated STA determines that there are no AVLNs to track (i.e., AVLN(s) that it is tracking no longer exist), it shall become an Unassociated CCo.

Detection of a CM\_UNASSOCIATED\_STA.IND MME with matching NID causes the STA to follow the procedure described in Section 7.3.4.1.

Figure 7-2 shows the basic functional flow for Unassociated STA behavior.



Figure 7‑2: Unassociated STA Behavior

### Unassociated CCo Behavior

The mode of operation of an Uncoordinated CCo with respect to HomePlug 1.0/1.1 coexistence shall be based on the detection status of HomePlug 1.0/1.1 delimiters (refer to Section 9.3). The neighbor network mode of operation of an Unassociated CCo shall be either CSMA-Only mode or Uncoordinated mode, as described in Section 8.5. Unassociated CCos shall periodically send Discover Beacons as required by the Discover Process (refer to Section 7.6). An Unassociated CCo shall not transmit the CM\_UNASSOCIATED\_STA.IND MME.

Detection of an AVLN shall cause the Unassociated CCo to start operating as an Unassociated STA.

Reception of valid CC\_ASSOC.REQ shall cause the Unassociated CCo to become a CCo.

Figure 7‑3 shows the basic functional flow for Unassociated CCo behavior.



Figure 7‑3: Unassociated CCo Behavior

### Behavior as a STA in an AVLN

Upon joining the AVLN, if the STA is a user-appointed CCo (refer to Section 7.4.2) and the existing CCo of the AVLN is not a user-appointed CCo, the STA sends a CC\_CCO\_APPOINT.REQ to the existing CCo to hand over the CCo functionality. Successful handover of the CCo functionality will cause the STA to become a CCo.

A CCo-capable STA may also become a CCo as a result of CCo selection procedure execution (refer to Section 7.4) or subsequent to CCo failure, if the STA is a backup CCo (refer to Section 7.9.2).

If a STA in the AVLN fails to detect the Central or Proxy Beacons of the AVLN that it is part of for MaxNoBeacon and it is not the backup CCo, it should restart the Power-on procedure.

If a STA in the AVLN is requested to leave the AVLN, it shall become an Unassociated STA.

Figure 7‑4 shows the basic functional flow for the behavior as a STA in an AVLN.



Figure 7‑4: Behavior as a STA in an AVLN

#### Identifying HomePlug GREEN PHY Stations

A HomePlug GREEN PHY Station or a HomePlug AV station (with a version greater than 1.1) shall advertise its GREEN PHY Station type and capabilities (e.g., the station is HomePlug GREEN PHY ver 1.0 capable) to the CCo after the association and authentication process. This advertisement is done using CM\_STA\_IDENTIFY.IND MME. The format of the CM\_STA\_IDENTIFY.IND MME is described in Section 11.5.39. The CCo acknowledges the receipt of this MME using a CM\_STA\_IDENTIFY.RSP MME. A HomePlug GREEN PHY Station shall also advertise its HomePlug AV version number and/or GREEN PHY station type and capabilities to the other stations in the network using the discover beacons. A HomePlug GREEN PHY Station shall use the GREEN PHY Capability and HPAV version fields of Discovered Info BENTRY wherein it indicates its device type. Refer to Table 4-79 for more details. Alternately, any HomePlug GREEN PHY Station or a future version of HomePlug AV station may request the Station Identification information using a CM\_STA\_IDENTIFY.REQ MME. A HomePlug GREEN PHY Station or a future version of HomePlug AV station responds to this request using a CM\_STA\_IDENTIFY.CNF MME. The format of the CM\_STA\_IDENTIFY.CNF MME is as described in Section 11.5.38.

### Behavior as a CCo in an AVLN

Once a STA becomes a CCo, if it determines that there are no STAs that have successfully joined the AVLN, it shall start a Join Wait Timer. Join Wait Timer is the duration of time for which the STA will act like a CCo if no other STA has successfully joined the AVLN. It is recommended that Join Wait Timer be set to at least MaxDiscoverPeriod to provide sufficient time for STAs to join the AVLN. If a STA joins the AVLN before the expiration of Join Wait Timer, the timer shall be cleared. Expiration of a Join Wait Timer shall cause the CCo to operate as an Unassociated CCo if there are no other AVLNs present; otherwise, it shall start operating as an Unassociated STA.

Handing over of the CCo functionality to another STA in the AVLN shall cause the CCo to become a STA in the AVLN.

If all the STAs associated with the AVLN leave the AVLN and there are no other AVLNs detected, the CCo shall operate as an Unassociated CCo. If all the STAs associated with the AVLN leave the AVLN and there is at least one other AVLN detected, the CCo shall become an Unassociated STA.

Figure 7‑5 shows the basic functional flow for the behavior as a CCo in an AVLN.



Figure 7‑5: Behavior as a CCo in an AVLN

### Deciding AV-Only or Hybrid Mode

If a STA joins an AVLN, it shall adopt the mode of that AVLN (although it shall inform the CCo of the traffic it has detected, which may change the mode later – refer to Section 9.3.1 and Section 9.3.2). When forming an AVLN (either as an Unassociated CCo or as the CCo of an AVLN with other STAs), the STA that becomes the CCo must determine the mode of the AVLN (refer to Section 9.3.2).

All STAs shall search for HomePlug 1.0.1 and HomePlug 1.1 transmissions during the power-on procedure. This information is used to determine the coexistence mode of the AVLN when the STA becomes a CCo and instantiates a new AVLN. During Power-on, detection of HP1\_FC\_Thresh valid HomePlug 1.0.1 or HomePlug 1.1 transmissions over an interval of HP1\_FC\_Thresh\_Interval or less shall constitute detection of HomePlug 1.0.1 or HomePlug 1.1 activity, respectively.

## Forming or Joining an AVLN

### AVLN Overview

An AVLN is formed by STAs that possess a common NID and CCo. STAs in an AVLN will typically possess a common NMK and Security Level (SL), but a CCo may divide an AVLN into sub-AVLNs, each with its own NMK. All NMKs that are associated with the same NID shall have the same SL, which is provided in the **CM\_SET\_KEY.REQ** MME or in the **APCM\_SET\_KEY.REQ** primitive. When no AVLN exists and a STA discovers one or more other stations with the same NMK and SL, the STAs shall form an AVLN if at least one is CCo-capable. If a STA discovers an existing AVLN with the NMK and SL it possesses, it shall join the existing AVLN. An AVLN is formed or joined using Association and Authentication.

To join (participate in) an AVLN, a STA must have:

* A valid NMK and Security Level (NMK-SL, obtained by Authorization — a.k.a. NMK Provisioning, as described in Section 7.10.3).
* A unique TEI (obtained by Association, as described in Section 7.3.2).
* The current NEK (obtained by Authentication, as described in Section 7.3.3).

Usually, the NMK-SL and NID are stored in non-volatile memory.

#### Network Identification

Each AVLN has a Network Identifier (NID) that is provided with or generated from the NMK-SL and is used to help a STA identify another STA or AVLN with the same NMK-SL. Section 4.4.3.1 describes how the NID is generated.

It is possible, though highly unlikely, for different NMKs to hash to the same NID value. Consequently, the NID is not guaranteed to be unique and is not guaranteed to identify a STA or AVLN with the same NMK. Neighboring AVLNs could have different NMKs, but the same NID. STAs shall attempt to join each STA or AVLNs that possess the NID that is associated with or generated from the NMK the STA possess. Networks with the same NID are uniquely identified by the SNID and NID pair.

It is also permitted for the CCo to use the same NID for multiple NMKs, forming a sub-AVLN with each NMK. As the NIDs are the same, a STA with an (NID,NMK) pair whose NID matches the NID of an AVLN shall attempt to join that AVLN. The CCo must disambiguate the NMK depending upon its knowledge of the STA’s MAC address.

#### Human-Friendly Station and AVLN Names

STA manufacturers shall provide default “Human Friendly” Identifiers (HFIDs) to the STAs. An HFID shall be a string of up to 64 ASCII characters chosen from the range ASCII[32] to ASCII[127]. The HFIDs shall be stored in non-volatile memory. The HFID shall be null-terminated (ASCII[00]). If the HFID is a full 64 characters in length, an implicit null character shall be interpreted in the “65th character position.”

User Interface (UI) software should provide functionality to enable the user to enter/modify user-entered “human friendly” names for each STA actively joined to the AVLN and for the AVLN itself. These user-entered HFIDs are distinct from the manufacturer-set HFID for the STA, which is permanent. The UI may provide a way for the user to cause the HFID to be sent or replaced with the null string when it is sent in the clear. Authenticated stations may refuse to provide the HFID to STAs outside the AVLN (i.e., always encrypt MMEs containing the HFID with the NEK).

The HFID of an AVLN is obtained by sending a CC\_WHO\_RU.REQ MME and receiving the associated CC\_WHO\_RU.CNF reply. These may be sent and received unencrypted. The HFID of a STA is obtained by sending a CM\_ HFID.REQ MME and receiving the associated CM\_ HFID.REQ reply. These are ordinarily sent and received encrypted with the NEK.

#### Get Full AVLN Information

The NID is contained in each Beacon transmitted by the CCo. Ordinarily, this will be sufficient, but there will be circumstances when the STA needs additional information about the AVLN (e.g., to identify the AVLN to the user it will need to get the HFID of the AVLN). The Message Sequence Chart (MSC) shown in Figure 7‑6 allows the new STA to get full AVLN information prior to obtaining a TEI. If the STA already has a TEI, that value shall be used within the messages.

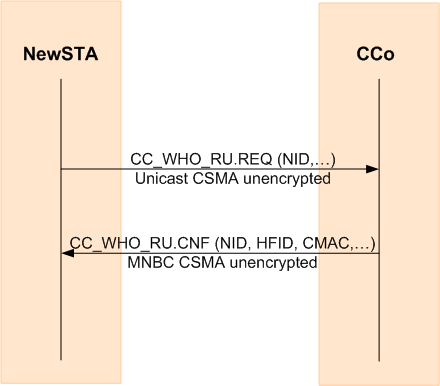


Figure 7‑6. Getting Full AVLN Information

#### Get Full STA Information

The SNID and TEI are contained in each MPDU transmitted by a STA. Ordinarily, this will be sufficient, but there will be circumstances when a user needs additional information about the STA (e.g., the HFID of the STA is useful to identify the STA to the user).

The CM\_GET\_HFID.REQ MME may be used to get the HFIDs of the STA (manufacturer-set HFID or user-set HFID) or the HFID of the Network. A STA that belongs to an AVLN may elect not to provide the actual HFIDs if the request is not encrypted with the NEK, but to replace them with the null string. A STA whose NMK has never been used by it to join or to form an AVLN should respond with its actual HFID in the clear when requested to do so.

The CM\_STA\_CAP.REQ MME may be used to get detailed information about the STA including optional features the STA supports, the HomePlug AV version, the OUI and the product manufacturer’s version number.

### Association

Association is a process by which a STA obtains a valid TEI from the CCo of the AVLN with which it wants to associate. Disassociation is a process by which a STA should stop using a valid TEI for an AVLN with which it was once associated. Upon disassociation, the TEI becomes invalid until reassigned.

There is a single method of Association, but there are several methods of Disassociation.

When a STA initially wants to communicate with an AVLN, it shall perform Association as shown in Figure 7‑7.

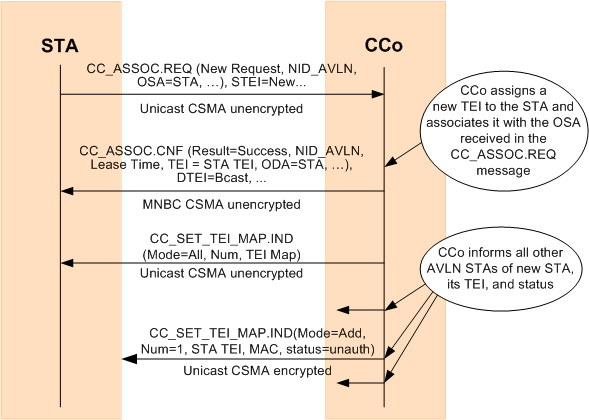


Figure 7‑7. STA Association

When two Unassociated STAs try to form an AVLN, a STA that determines that another STA is better suited to become the AVLN’s CCo and that the two STAs should form an AVLN (by matching NIDs or SC-Join states) shall wait for the other STA to establish the AVLN and start transmission of Central Beacons. A STA that decides it should become the CCo based on CCo capabilities, matching NIDs, or SC-Join/Add states shall start transmitting Central Beacons and wait for a CC\_ASSOC.REQ MME from the other STA, as described in Section 7.1.

When a STA joins an AVLN, the CCo shall provide the complete TEI Map to that STA and update all other STAs in the AVLN with the new TEI Map information using the CC\_SET\_TEI\_MAP.IND MME.

#### TEI Assignment and Renewal

The CCo assigns a Terminal Equipment Identifier (TEI) to each STA when it successfully associates with the AVLN. The TEI shall be 8 bits long and shall be unique within the AVLN.

Table 7‑1 shows the possible values of the 8-bit TEI. Note the TEI value of 0x00, which may be used by a new STA or by a CCo that wants to communicate with another CCo. The new STA shall recognize that the CC\_ASSOC.CNF MME (sent with the broadcast TEI) is for it by verifying that the ODA in the MME is its MAC address.

Table 7‑1: TEI Values

|  |  |
| --- | --- |
| TEI Value | Interpretation |
| 0x00 | TEI not yet assigned. This value may be used by a CCo to communicate with another CCo or by a new STA that has not yet associated with the AVLN. |
| 0x01 - 0xFE | Identifies a STA within the AVLN. |
| 0xFF | Broadcast TEI (All STAs shall treat message as addressed to them). This value shall never be used as an STEI. |

The CCo maintains the list of assigned TEIs and the corresponding mapping of MAC addresses. The CCo shall send the CC\_SET\_TEI\_MAP.IND MME (unencrypted) to the new STA to provide it with the current (TEI, MAC address) mappings of all existing STAs in the AVLN.

The CCo shall send the CC\_SET\_TEI\_MAP.IND MME (encrypted) to update all authenticated STAs with any changes in the mappings (e.g., a new STA has joined the AVLN).

After a STA is disassociated (or has been expelled), its TEI will be reclaimed by the CCo. The reclaimed TEI shall not be reused for a period of at least 5 minutes and the CCo shall send the CC\_SET\_TEI\_MAP.IND MME to update all authenticated STAS with the disassociation(s).

An authenticated STA may use the CC\_SET\_TEI\_MAP.REQ MME to query the CCo for a full copy of the TEI Map. The CCo shall respond to the STA with the CC\_SET\_TEI\_MAP.IND to provide the TEI map.

##### Disambiguated TEIs

Although the CCo ensures that the TEIs it assigns are unique within the AVLN, the same TEI value may be assigned to a different STA in a neighbor AVLN, so it is important to disambiguate the TEI by associating it with the network where it is generated. The NID or the SNID can be used for this purpose.

##### TEI Leases and Renewals

When the CCo assigns or renews a TEI it shall specify a lease time. This is the length of time for which the STA may use the TEI. If the lease time expires before the TEI is renewed, the STA shall stop using the TEI and apply for another. Permitted lease values are shown in Table 7‑2.

Table 7‑2: Lease Values

|  |  |
| --- | --- |
| Lease Value | Interpretation |
| 0x0000 | Reserved |
| 0x0001 - 0xFFFF | Lease time in minutes (maximum value exceeds 45 days)  0x000F = 15 minutes is default lease time for STAs that are associated but not authenticated.  0x0B40 = 48 hours is default lease time for an authenticated STA. |

Lease time is measured from the time the CCo generated the corresponding CC\_ASSOC.CNF. Variable amount of delay can be incurred from the time the CCo generated the CC\_ASSOC.CNF and the STA receives and processes the message. Implementations should consider this when determining the expiry time of the TEI lease and the time at which the STA starts renewing its TEI lease. Before the lease time has expired, a STA shall go through the association process again to renew its TEI. It is recommended that the STA starts its TEI renewal process at least 5 seconds before the expiration of its lease time. The ReqType field in the CC\_ASSOC.REQ message shall be set to indicate that it is a renewal request. If the CCo accepts the renewal request, the same TEI shall be assigned to the STA. If the STA fails to renew its TEI before it expires, the CCo shall s the STA from the AVLN.

A renewal CC\_ASSOC.REQ message may also be sent by an existing STA to a PSTA to indicate that the existing STA can no longer decode the (Central) Beacons reliably. The PSTA shall forward the CC\_ASSOC.REQ message to the CCo. The CCo shall accept the renewal request, create a Proxy Network, and assign the same TEI to the existing STA. Refer to Section 7.7.

##### When to Stop Using a TEI

A STA shall stop using a TEI if any one of the following events occurs:

* The TEI’s lease time expires before the STA has successfully renewed the lease.
* The STA disassociates from the AVLN (refer to Section 7.3.5.1).
* The AVLN’s CCo asks the STA to leave the AVLN (refer to Section 7.3.6).

##### Updating STAs with the TEI MAP

When there is a change in the TEI Map, such as when a STA associates, authenticates, or leaves the AVLN, the CCo shall send a CC\_SET\_TEI\_MAP.IND message to all STAs in the AVLN.

A STA can request the current TEI Map from the CCo using the CC\_SET\_TEI\_MAP.REQ message.

### Method for Authentication

Once a STA has associated and has a valid NMK, it shall use this NMK to join the AVLN. If the CCo verifies the STA’s NMK, it will give the STA an NEK. Once a STA is authenticated successfully, the CCo shall maintain the STA’s authentication status as long as the STA’s association status is maintained. Thus, a STA is not required to re-authenticate subsequent to TEI renewal.

There is one method for Authentication; it is used by all STAs. The joining STA shall send a CM\_GET\_KEY.REQ containing KeyType = NEK, the STA’s TEI and MAC address. It shall also contain a freshly generated nonce. The message shall be placed in an Encrypted Payload of a CM\_ENCRYPTED\_PAYLOAD.IND MME encrypted by the NMK, with PID=0x00 and PMN=0x01.

If the CCo confirms that the correct NMK was used to encrypt the message, it shall send a CM\_GET\_KEY.CNF message (in a CM\_ENCRYPTED\_PAYLOAD.IND with the payload encrypted with the NMK) to the STA. This message shall contain the NEK and EKS along with the nonce sent in the request and shall be placed in an Encrypted Payload encrypted using the NMK. The procedure is shown in Figure 7‑8.

If the CCo cannot decrypt the request encrypted by that NMK (indicated by CM\_ENCRYPTED\_PAYLOAD.RSP with Result = Failure), the new STA shall flag the NMK as invalid on this AVLN (NID and SNID) and either restart the process of obtaining a (valid) NMK on this AVLN or try to join a different AVLN (same NID but different SNID).

The new STA can begin using the NEK as soon as it successfully receives it from the CCo.

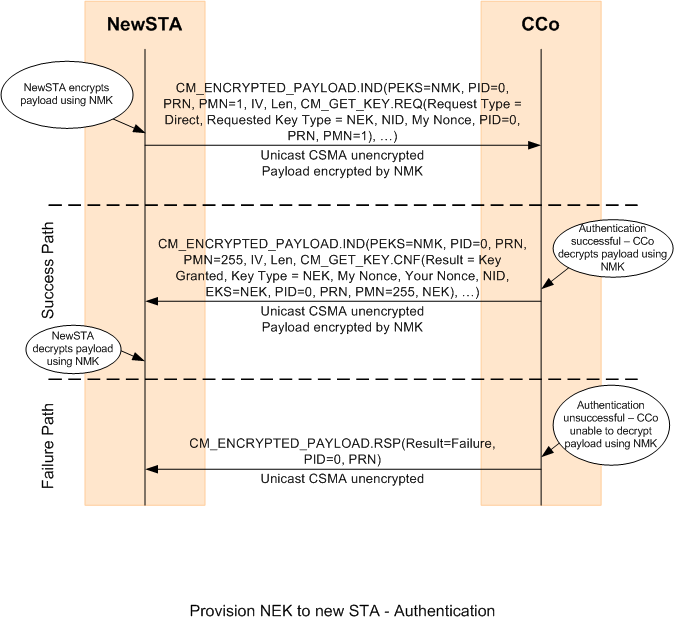


Figure 7‑8: Provision NEK for a new STA (Authentication)

### Forming a New AVLN

Two Unassociated STAs can form a new AVLN when one of four conditions is met:

* They have the same (NID,NMK) pair, and one or both receives the other’s CM\_UNASSOCIATED\_STA.IND MME (refer to Section 7.3.4.1).
* One STA sends the other STA its NMK encrypted with the other STA’s DAK (refer to Section 7.3.4.2).
* They both have NMK-SCs, and one’s HLE indicates that it should enter the SC-Join state and the other’s HLE indicates that it should enter the SC-Add state (refer to Section 7.3.4.3).
* They both have NMK-SCs and the HLE of each indicates they should enter the SC-Join state (refer to Section 7.3.4.4).

In each of these cases, two Unassociated STAs exchange MMEs that includes the CCo capability of each STA, and from these MMEs each STA recognizes that a new AVLN needs to be formed. One of the STAs will become the CCo (as described in Section 7.4.1) and the other will associate with the new CCo, and possibly perform the NMK key exchange and ultimately authenticate.

CM\_SET\_KEY.REQ/CNF, CM\_UNASSOCIATED\_STA.IND, and CM\_SC\_JOIN.REQ/CNF MMEs contain CCo capability information; this information allows the recipient to determine which STA should become the CCo of the new AVLN. Whichever STA has the greater CCo capability (as defined in Section 7.4.1 shall become the new CCo. Once the AVLN is formed, the CCo may be changed due to other factors.

A STA that determines that another STA should become the CCo (refer to Section 7.4.1) shall wait to try to associate with it until it starts receiving the Central Beacon from that STA or possibly another STA. Such STAs may only repeat the MMEs that the other STA must receive to prompt it to form an AVLN. The other STA should reach the same conclusion and become a CCo and start issuing Central Beacons, perhaps in Coordinated Mode if a neighboring network is present. A STA that becomes a CCo shall begin transmitting the Central Beacon and shall continue to do so as long as at least one STA has successfully associated with it or while it hears MMEs from other STAs that should associate with it.

When the STA has or receives an NMK associated with an NID matching that of the new CCo (possibly as a result of one of the processes described above), it shall try to authenticate using the protocol described in Section 7.10.4. If authentication fails, the NMKs are not the same and the protocol aborts. In this case, if the CCo has no associated STAs, it shall cease AVLN operation and returns to being an Unassociated STA (unless it cannot detect any other AVLN). If authentication failed for reason other than different NMKs, the STAs should try again to form a new AVLN. If authentication succeeds, the initial AVLN formation is complete and the STAs may go on to add more STAs to the AVLN, select a new CCo, etc.

#### Two Unassociated STAs with Matching NIDs

Two STAs with identical NMKs and identical Security Levels will also have identical default NIDs. Identical NIDs do not guarantee that the NMKs are identical, but the probability against this is very small. The NID is advertised in the CM\_UNASSOCIATED\_STA.INDMME, so when one STA receives the other STA’s CM\_UNASSOCIATED\_STA.INDMME, it will observe the matching NID.

At this point, one of the STAs (or a third STA) must become a CCo as defined in Section 7.4.1. The CM\_UNASSOCIATED\_STA.INDMME contains the CCo capability and the MAC address of the sender (as part of the generic MME format). The receiver can then decide whether it or another STA should become the CCo.

If the STA determines that it should become the CCo, it shall form a new AVLN and start sending Central Beacons, then wait for the other STA(s) to associate.

If the STA determines that another STA should become the CCo, it shall wait until it detects a Central Beacon with matching NID. In the meanwhile, the STA shall continue to send CM\_UNASSOCIATED\_STA.IND MMEs periodically in case the other STA did not correctly receive its earlier advertisements and hence does not know to become the CCo of a new AVLN. After the non-CCo STA has detected the new CCo’s Beacon with matching NID, it shall send the CCo a CC\_ASSOC.REQMME asking it for a TEI.

If more than one AVLN with the same NMK is formed, the STA may attempt to join any AVLN with matching NID that it detects. These AVLNs may merge using the mechanism described in Section 8.6.

Once the STA has associated with the CCo, it shall try to authenticate as described in Section 7.10.4.

This entire process is shown in Figure 7‑9.

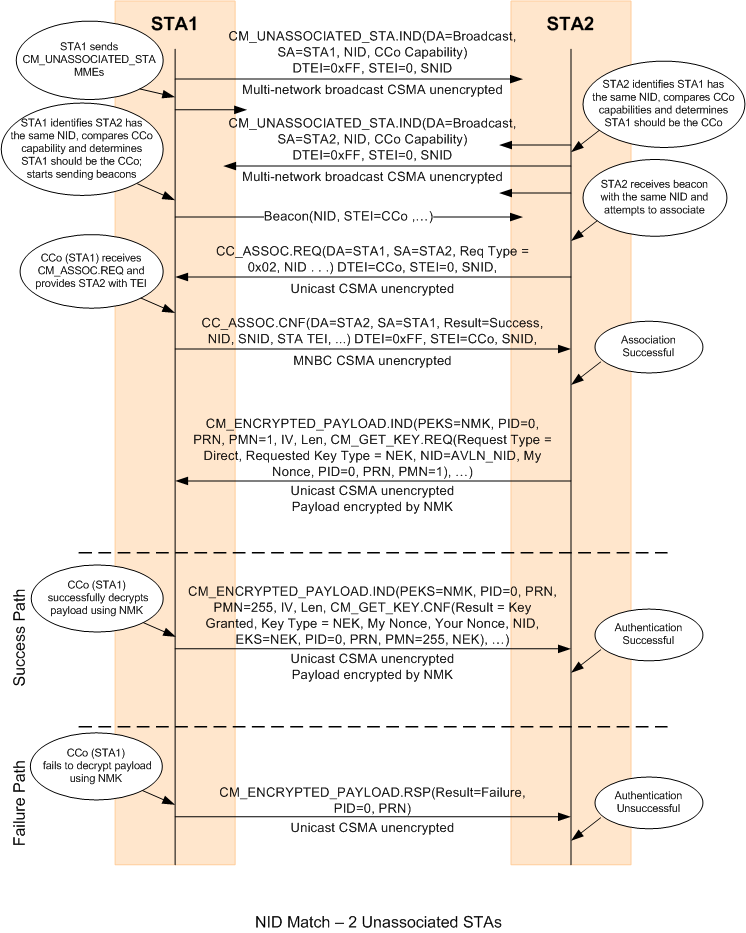


Figure 7‑9: AVLN Formation by Two Unassociated STAs with Matching NIDs

#### Two Unassociated STAs Form an AVLN Using a DAK-encrypted NMK

When the HLE provides a STA with the DAK of another station and tells it to send the other STA an NMK and NID, the STA shall broadcast a CM\_SET\_KEY.REQ MME containing a TEK and encrypted with the DAK as the payload of a CM\_ENCRYPTED\_PAYLOAD.IND MME, sent unencrypted using multi-network broadcast. All STAs that receive the MME shall try to decrypt it; if one succeeds, that successful STA shall respond with a CM\_SET\_KEY.CNF MME encrypted with the TEK as the payload of a CM\_ENCRYPTED\_PAYLOAD.IND MME sent unencrypted using unicast.

Note: A STA that fails to decrypt a CM\_ENCRYPTED\_PAYLOAD.IND MME that uses a DAK for payload encryption shall not respond to that MME with a CM\_ ENCRYPTED\_PAYLOAD.RSP MME, but shall silently drop the message.

Each STA compares the CCo capability fields (present in the first two messages) and determines which STA should become the CCo, as defined in Section 7.4.1. Once one STA becomes the CCo and the other STA has associated with it, the STAs complete the protocol as described in Section 7.10.3.4. When the protocol is completed successfully, the STA that is not the CCo shall authenticate with the CCo to obtain the NEK. The new CCo shall use in its Central Beacon the NID that was sent with the NMK; the other STA must also use the NID associated with the NMK and wait for a Central Beacon with the matching NID before it authenticates with the CCo.

The STA sending the DAK-encrypted payload shall continue to transmit these periodically until it either receives a response or until it times out.

If the STA that was given the DAK by its HLE later joins with some other STA to form an AVLN, it may restart this protocol by sending the DAK-encrypted payload as an AVLN STA.

An associated (and even authenticated) STA that receives a new NMK via DAK-encrypted payload shall leave its current AVLN and form an AVLN with the STA that initiated the DAK-based protocol, even if that STA is not initially part of any AVLN.

The entire process (omitting failure paths) is shown in Figure 7‑10. See also Section 7.10.3.4.

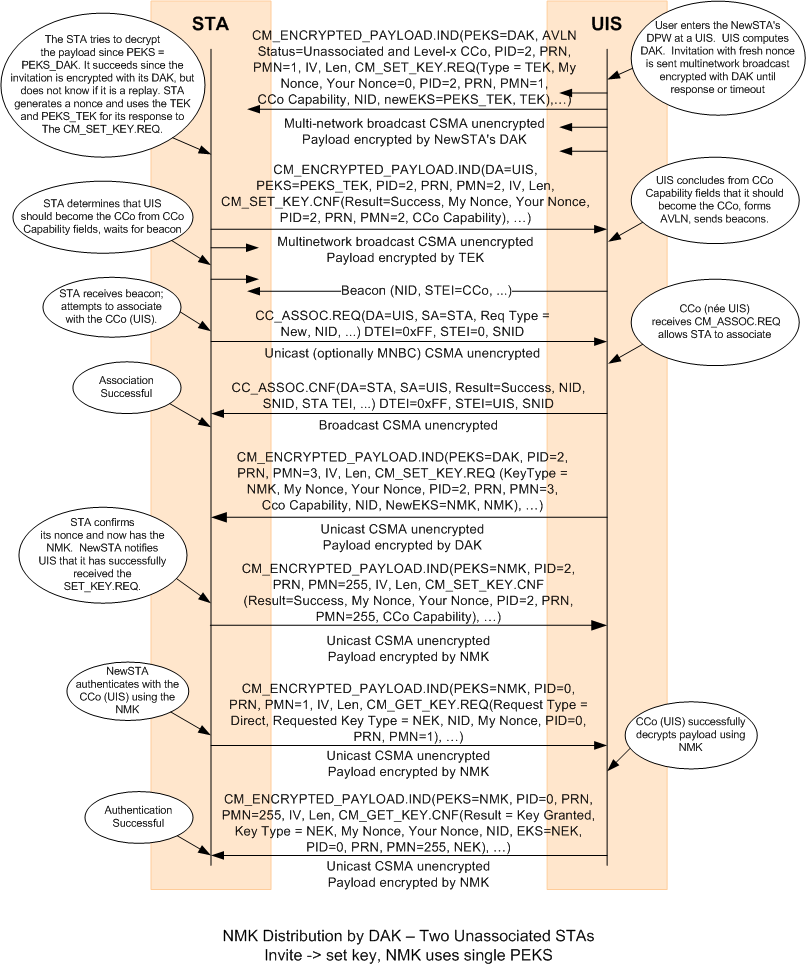


Figure 7‑10: AVLN Formation Using a DAK-Encrypted NMK

#### Two Unassociated STAs: One in SC-Add and One in SC-Join

When the HLE places a STA into the SC-Join state, it transmits CM\_SC\_JOIN.REQ MMEs using multi-network broadcast periodically until it either joins an AVLN or times out. If the HLE places the STA into the SC-Add state, however, it does not advertise this, but waits to hear another STA transmitting CM\_SC\_JOIN.REQ MMEs until it either adds a new STA or times out. Optionally, a STA in the Simple Connect SL may cache recently received CM\_SC\_JOIN.REQ MMEs in anticipation of its HLE placing it into the SC-Add state.

When a STA in the SC-Add state detects a CM\_SC\_JOIN.REQ MME, it responds to it with a CM\_SC\_JOIN.CNF MME. The STA that was in the SC-Add state then becomes the CCo of a new AVLN and starts issuing Beacons. When the STA in the SC-Join state detects the Beacons, it associates with the CCo, regardless of relative CCo capabilities. This case is distinguished from the case below in which two Unassociated STAs are in SC-Join by the STA in SC-Add setting the CCo Status field to 0b1. If the joining STA has greater CCo Capability, it will later become the CCo through autoselection of the CCo and the CCo Handover process (refer to Section 7.4.3 and Section 7.5).

After the new STA has associated with the new CCo, the two STAs optionally perform Channel Adaptation (refer to Section 5.2.6) to have channel adapted tone maps. Finally, the CCo shall start the UKE protocol. This first establishes a shared TEK, which is used by the CCo to provide the new STA with its NMK (refer to Section 7.10.3.5). When the new STA has the NMK, it shall authenticate and join the AVLN.

Note: The STA in SC-Add state will pass the NMK-SC it posses to the STA in SC-Join state. A random NMK-SC is not generated. A user may place an Unassociated STA in the SC-Add state if that STA had been part of an AVLN (in the SC security level) that is not currently present to add a new STA to that AVLN (i.e., to pass the NMK-SC of that AVLN to the new STA in SC-Join state). See also Section 7.10.3.1.2.

This entire process is shown in Figure 7‑11.

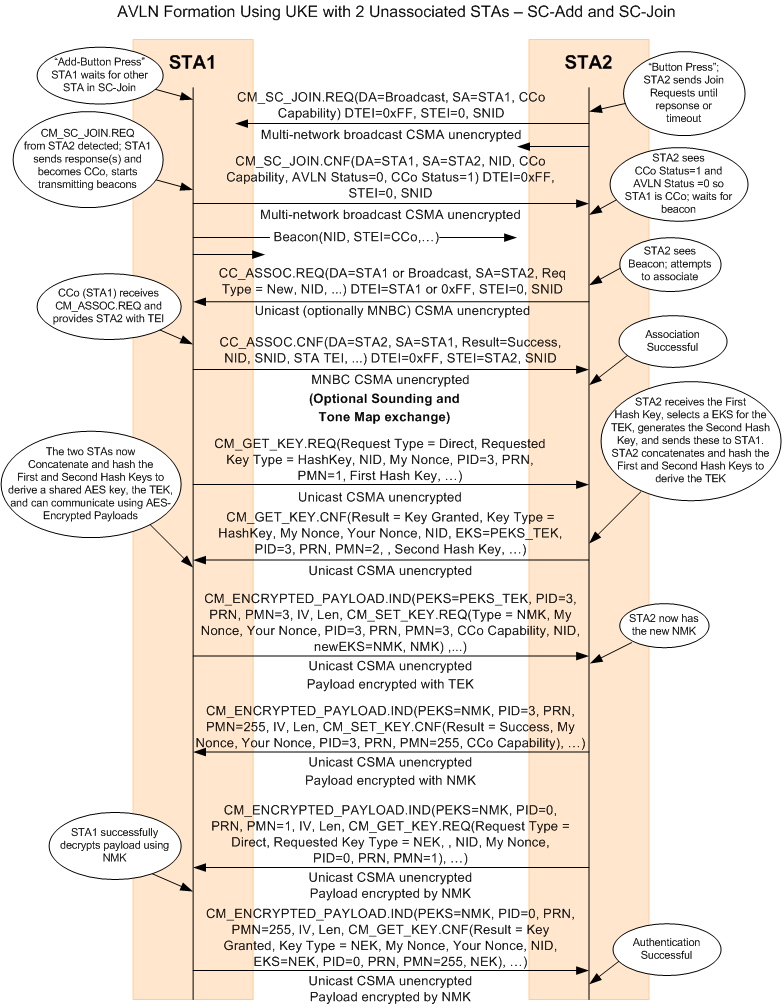


Figure 7‑11: AVLN Formation Using UKE by One STA in SC-Add and One STA in SC-Join

#### Two Unassociated STAs: Both in SC-Join

It is possible that both STA’s HLEs can be placed in the SC-Join state. In this case, both will begin to transmit CM\_SC\_JOIN.REQ MMEs with their CCo capability using multinetwork broadcast. When a STA in SC-Join mode receives another STA’s CM\_SC\_JOIN.REQ MME, it shall determine which STA should become the CCo as defined in Section 7.4.1. If it is the one to become the CCo, it shall change its state to SC-Add, generate a new random NMK-SC, send a CM\_SC\_JOIN.CNF MME to the other STA with its NID, establish itself as a CCo, and start issuing Central Beacons, forming a Neighbor Network if necessary. When the other STA receives the CM\_SC\_JOIN.CNF MME and detects the Beacon with the same NID, it shall associate with the new CCo. The two STAs shall optionally perform channel adaptation prior to commencing the UKE protocol as above (refer to Section 7.3.4.3).

The STA that determines it should not become the CCo must wait for the other STA to send the CM\_SC\_JOIN.CNF MME and for that STA to begin issuing Central Beacon. In the meanwhile, the STA shall continue to send CM\_SC\_JOIN.REQ MMEs periodically in case the other STA did not correctly receive its earlier transmissions and hence does not know to send the CM\_SC\_JOIN.CNF MME and become the CCo of a new AVLN.

This entire process is shown in Figure 7‑12.

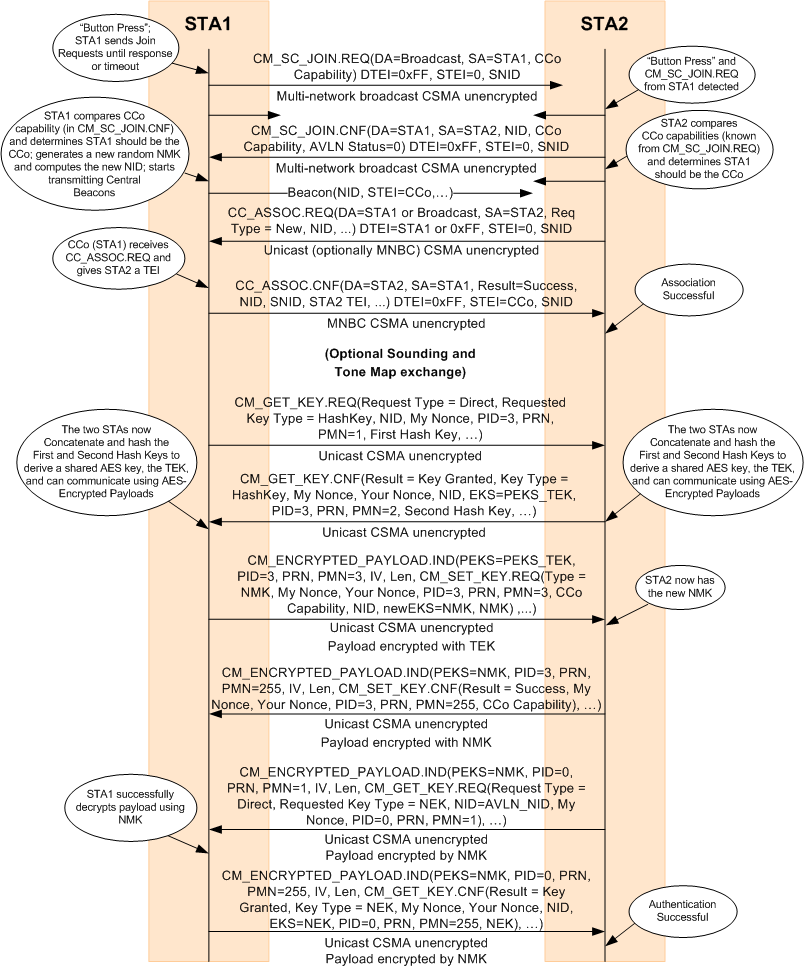


Figure 7‑12: AVLN Formation Using UKE by Two STAs in SC-Join

### Joining an Existing AVLN

An Unassociated STA may join an existing AVLN when one of the following three conditions is met:

1. It has the same NMK and Security Level and detects the AVLN’s Central Beacon or the Discover Beacon of one of the STAs in the AVLN (refer to Section 7.3.5.1).
2. One of the AVLN STAs sends the Unassociated STA its NMK encrypted with the Unassociated STA’s DAK (refer to Section 7.3.5.2).
3. The AVLN has an NMK-SC, the Unassociated STA’s HLE indicates that it should enter the SC-Join state, and the HLE of a STA in the AVLN indicates that it should enter the SC-Add state (refer to Section 7.3.5.3).

In each of these cases, based on the initial information that is received or exchanged, the Unassociated STA will recognize that it needs to associate with an existing AVLN. After association, the STA proceeds with the protocol to receive the NMK for the AVLN if it does not already posses it. Upon successful reception of the NMK for the AVLN, the STA then authenticates using the NEK distribution protocol described in Section 7.3.3.

#### Matching NIDs

Two STAs with identical NMKs and identical Security Levels will also have identical default NIDs. Identical NIDs do not guarantee that the NMKs are identical, but the probability against this is very small. The NMK held by a STA may be associated with a non-default NID; in this case, the non-default NID associated with the NMK shall be used for matching purposes. The NID is advertised in the Central Beacon, Proxy Beacon, and Discover Beacons, so when an Unassociated STA receives one of these, it shall observe the matching NID.

The Unassociated STA must take the first step; the AVLN STAs (including the CCo) must wait for it to initiate the process. The Unassociated STA shall send the CCo a CC\_ASSOC.REQ MME asking for an initial TEI within the AVLN. The CCo shall reply with a CC\_ASSOC.CNF MME with STATUS=Success and a TEI assignment, unless it is out of TEIs or is in the process of transferring CCo status to another STA. In the latter cases, the CCo shall reply with STATUS=Defer, and the new STA will have to retry later.

Once the STA has associated with the CCo, it shall try to authenticate as described in Section 7.10.4.

This entire process is shown in Figure 7‑13.

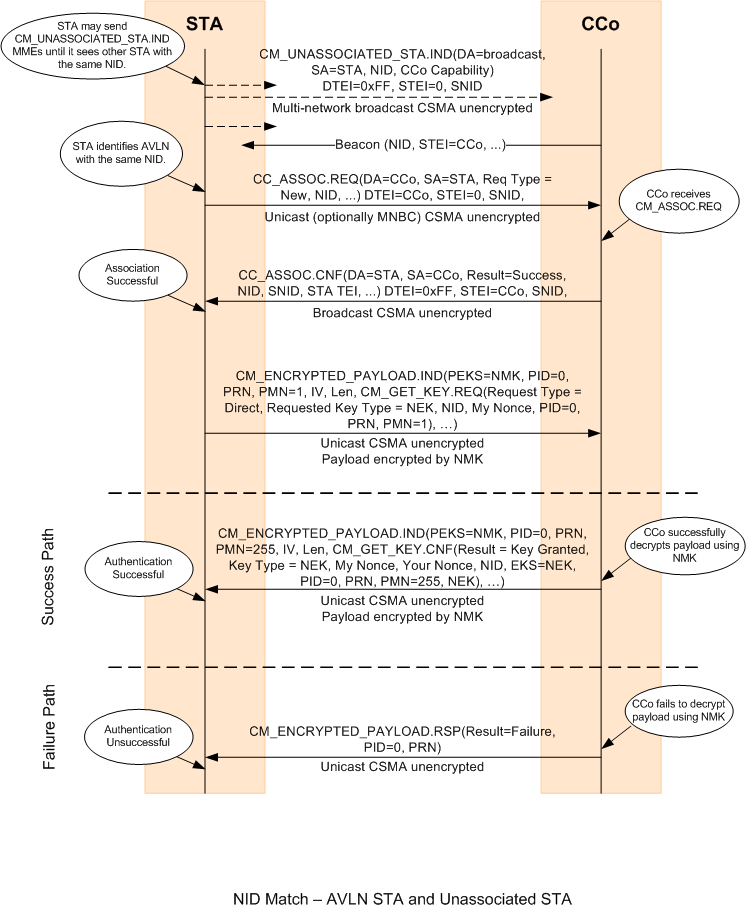


Figure 7‑13: New STA Joins Existing AVLN with Matching NID

#### DAK-encrypted NMK

When a STA with a suitable User Interface (the UIS) already on an AVLN is provided with the DAK of another STA and told by the HLE to send the other STA its current NMK, the UIS shall transmit a CM\_SET\_KEY.REQ MME containing a TEK and encrypted with the DAK as the payload of a CM\_ENCRYPTED\_PAYLOAD.IND MME, sent unencrypted using multi-network broadcast. The STA need not be the CCo to do this; it is sufficient that it is in an AVLN.

All STAs that receive the MME shall try to decrypt it; if one succeeds, that successful STA shall respond with a CM\_SET\_KEY.CNF MME encrypted with the TEK as the payload of a CM\_ENCRYPTED\_PAYLOAD.IND MME, sent unencrypted, then it shall associate with the AVLN’s CCo and complete the protocol as described in Section 7.10.3.4 A STA that fails to successfully decrypt a payload encrypted with a DAK shall ignore the message

When the DAK-encrypted NMK provisioning protocol is completed successfully, the new STA shall accept the NMK and SL, then shall try to authenticate with the CCo as described in Section 7.10.4 and join the AVLN.

This entire process is shown in Figure 7‑14.

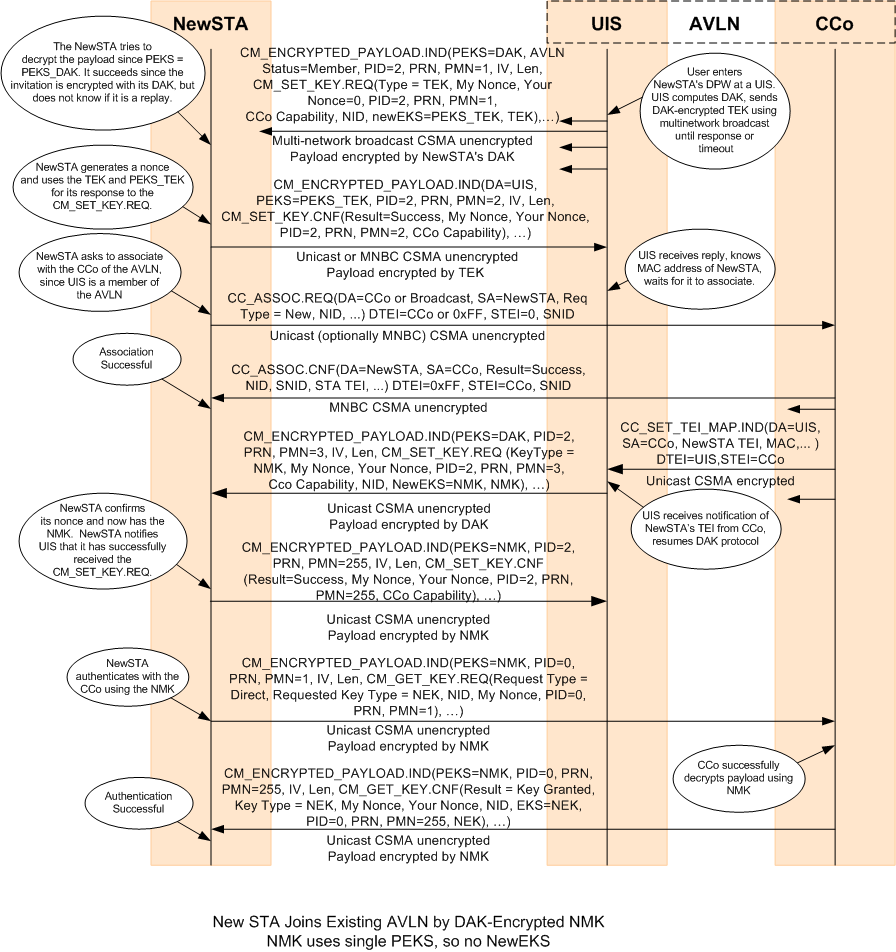


Figure 7‑14: New STA Joins AVLN by DAK-Encrypted NMK

#### SC-Join and SC-Add

When the HLE places a STA into the SC-Join state, the STA shall transmit CM\_SC\_JOIN.REQ MMEs using multi-network broadcast periodically until it either joins an AVLN or times out. If the HLE places the STA into the SC-Add state, however, the STA shall not advertise this, but shall wait to hear another STA transmitting CM\_SC\_JOIN.REQ MMEs until it either adds a new STA or times out. Optionally, a STA in the Simple Connect SL may cache recently received CM\_SC\_JOIN.REQ MMEs in anticipation of its HLE placing it into the SC-Add state.

When the AVLN STA in the SC-Add state detects a CM\_SC\_JOIN.REQ MME, it shall respond with a CM\_SC\_JOIN.CNF MME. The STA need not be the CCo to do this; it is sufficient that it is in an AVLN. The new STA in the SC-Join state associates with the AVLN. At this point, the two STAs optionally perform Channel Adaptation (refer to Section 5.2.6.2) to have channel adapted tone maps.

Once the new STA has associated with the AVLN (and optionally established channel adapted tone maps), the AVLN STA shall start the UKE protocol. The AVLN STA knows the new STA has associated due to the updated TEI Map received from the CCo. The UKE protocol first establishes a shared TEK, which is used by the AVLN STA to provide the new STA with its NMK-SC (refer to Section 7.10.3.5). When the new STA has the NMK-SC, it shall try to authenticate with the CCo as described in Section 7.10.2.5 and join the AVLN.

This entire process is shown in Figure 7‑15.

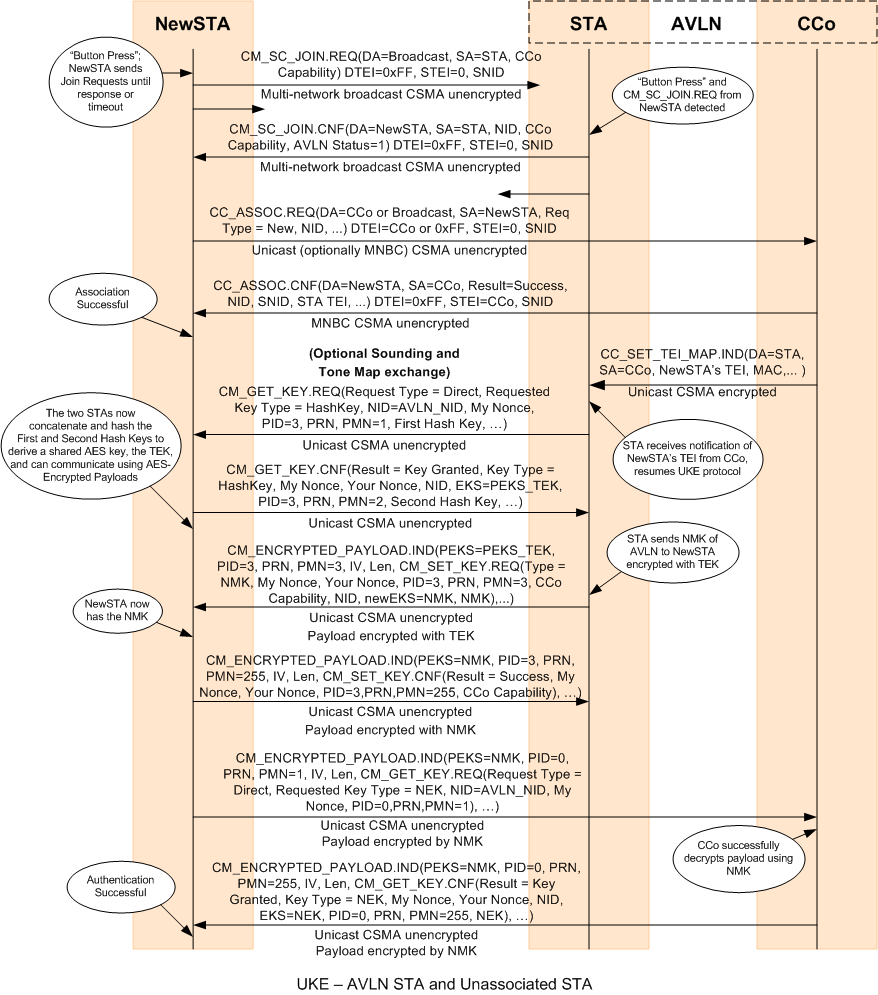


Figure 7‑15: New STA Joins Existing AVLN Using UKE

##### SC-Join and SC-Add for GREEN PHY

When a STA is in the SC-Add state detects a CM\_SC\_JOIN.REQ MME, it will inform the HLE using an APCM\_SC\_JOIN.REQ (refer to Section 12.2.2.52) and wait for the HLE to reply with an APCM\_SC\_JOIN.CNF. The STA with then reply to the new STA using CM\_SC\_JOIN.CNF MME containing the information from the APCM\_SC\_JOIN.CNF.

### Leaving an AVLN

If the STA is powered down or instructed to leave the AVLN by the user, it shall notify to CCo of its departure as shown in Figure 7‑16.

The STA shall wait until it receives an acknowledge response from the CCo before actually leaving. If it does not receive an ACK within 3 Beacon Periods, it will try to send the message a second time, after which it shall not use the TEI it had been assigned for any further communications with the AVLN (the CCo or any member STA).

If the user has overtly requested disassociation (note that power down is an implicit request, not an overt request), the user may also want to tell the STA to not try to re-associate with the AVLN in future. In this case, the STA must also change the NMK.

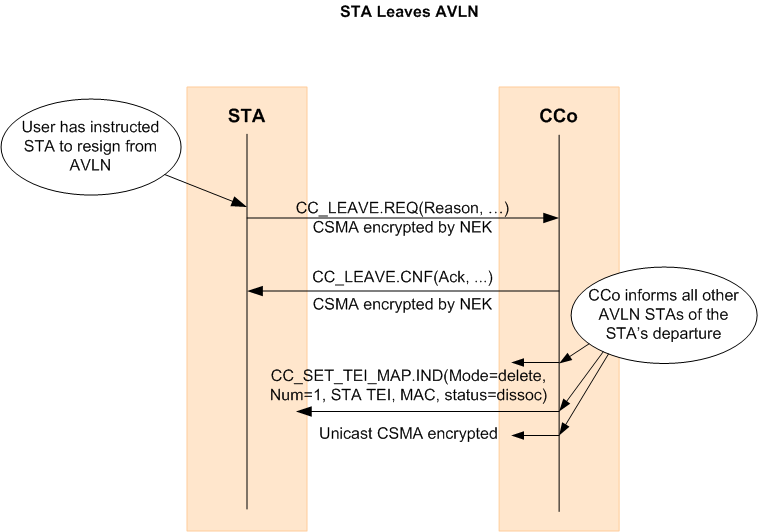


Figure 7‑16: Disassociation - STA Leaves AVLN

When a STA leaves an AVLN (including TEI lease expiration), the CCo shall update all other STAs in the AVLN with the new TEI Map using the CC\_SET\_TEI\_MAP.IND MME.

### Removing a Station from an AVLN

The only secure way to remove a STA from an AVLN is to change the NMK (refer to Section 7.10.3.7. The DAK of the removed STA should be discarded.

A CCo may send a CC\_LEAVE.INDMME to a STA to remove the STA from the AVLN. The CCo may also change the NEK and not provide the new NEK to the STA it wants to remove.

## Selection of CCo

The first STA to instantiate a network becomes the CCo. As the network evolves with more STAs joining or leaving the AVLN, another STA may be more suitable to fulfill the role of CCo. The current CCo shall apply the CCo selection procedure on an ongoing basis to identify the best STA within the AVLN to perform the function. If a more suitable STA is identified by the selection process, the current CCo process must hand over the function to the STA selected by the Auto-Select function.

The CCo can be automatically selected and does not require the user to have any knowledge of the CCo function or its operation. This function is called Auto-Selection of the CCo. All STAs must support the Auto-Selection function.

Alternately, the STA operating as the CCo may have been appointed by the user through a UIS provided on the AVLN. The ability to provide a user interface for enabling the user to appoint the STA or another STA as a CCo is optional for the STA. All CCo’s shall be capable of handing over CCo functionality when requested by another authenticated STA in the AVLN.

### CCo Selection for a New AVLN

When an Unassociated STA determines that a new AVLN needs to be formed based on the MMEs it received, the STA shall determine whether it should become the CCo for the new AVLN based on the CCo Capability field and the OSA contained in those MMEs. Refer to Section 7.3.4 for more information about how a STA determines when a new AVLN needs to be formed.

If the CCo Capability of the Unassociated STA is greater than that of the other STAs detected, or if the STA’s MAC address is greater than the other STAs’ when the CCo Capability is equal to the greatest capability detected, the STA shall become the CCo, possibly in Coordinated Mode if neighboring networks are detected, and begin transmitting the Central Beacon. For comparing MAC Addresses, the Individual/Group (I/G) bit of the 48-bit MAC address shall be treated as the least-significant bit in the least-significant octet.

An exception is an Unassociated STA that is in SC-Add state will always become the CCo (refer to Section 7.3.4.3).

#### Determination of SNID by a STA Acting as CCo in a Newly Formed AVLN (GREEN PHY)

When a new AVLN is formed by an unassociated STA and the unassociated STA has become the CCo of the AVLN, the STA (as the new CCo) shall randomly choose a SNID value which is not being used by any other AVLN heard during Power-On Network Discovery, and set the RSF and TEI Range according to the number of TEIs required. If all the SNID values are being used, the STA (as the CCo of the new AVLN) shall randomly select a SNID among those with RSF equal to one and set its own RSF to one. The TEI values of the STAs associated to this network shall be in the range of [0x01-0x3F], or [0x40-0x7F] or [0x80-0xBF] or [0xC0-0xFE] in order not to collide with the other AVLN with matching SNID.

### User-Appointed CCo

The following procedure describes the user-appointed CCo process. Figure 7‑17 shows this function.

1. The user enters the MAC address of the STA that should be assigned the role of CCo. The user enters this MAC address into a UI made available by a STA that is already associated and authenticated with the network.
2. The UI STA shall communicate with the existing CCo via a CC\_CCO\_APPOINT.REQ message, with the Request Type indicating a request to appoint a STA with the MAC address contained in the CC\_CCO\_APPOINT.REQ message as a user-appointed CCo (i.e., ReqType = 0x00).
3. If the current CCo is a user-appointed CCo or the STA that needs to be appointed as a CCo is not part of the AVLN, the current CCo shall send CC\_CCO\_APPOINT.CNF indicating a failure. Otherwise, the current CCo responds by querying the appointed STA with a CC\_HANDOVER.REQ message, requesting the STA to assume the role of the CCo. The message shall indicate that the handover is due to user appointment.
4. The STA responds with a CC\_HANDOVER.CNF message, where the STA either accepts the transfer of the CCo function or declines to do so.
5. The current CCo passes on this response to the UI STA through the CC\_CCO\_APPOINT.CNF message. If the user-appointed STA is the same as the UI STA that sent the CC\_CCO\_APPOINT.REQ message, the existing CCo shall send a CC\_CCO\_APPOINT.CNF message to the UI STA with a successful code and initiate the handover function.

* If the response from the appointed STA is positive, the existing CCo must initiate a CCo handover using the Handover function described in Section 7.5.
* If the response is negative, no further action is required from the existing CCo.

1. The current CCo shall carry out the remaining steps of the handover function (refer to Section 7.5).

A user-appointed CCo shall not perform the Auto-Selection of the CCo function. The CCo role is only transferred when the current user-appointed CCo disassociates (or is powered down) from the network. All CCo-capable STAs shall store in non-volatile memory information about whether they are a user-appointed CCo for an AVLN. If a STA that is a user-appointed CCo disassociates from the AVLN and re-associates with it at some later time, it shall first determine whether the existing CCo of the AVLN is user appointed. If the existing CCo is not a user-appointed CCo, it shall become the CCo by following the procedure described above.

A user-appointed CCo can be un-appointed as a user-appointed CCo by transmitting a CC\_CCO\_APPOINT.REQ message with a Request Type indicating un-appointment of the existing CCo as a user-appointed CCo (i.e., ReqType = 0x01). Upon un-appointment as a user-appointed CCo, the CCo shall continue to act as a CCo and start performing Auto-Selection of the CCo function. The CC\_CCO\_APPOINT.REQ message with ReqType = 0x01 can also be sent to any STA in the AVLN that is not acting as a CCo, but is configured to act like a user-appointed CCo. Such scenarios can occur when the user inadvertently appoints more than one STA in the AVLN as a user-appointed CCo. If a STA in the AVLN that is not acting as a CCo receives the CC\_CCO\_APPOINT.REQ message with ReqType = 0x01 , it shall cause the STA to un-appoint itself as a user-appointed CCo and respond with a CC\_CCO\_APPOINT.CNF message.

A user-appointed CCo can be un-appointed as a user-appointed CCo and a new STA in the AVLN can be appointed as a user-appointed CCo by transmitting a CC\_CCO\_APPOINT.REQ message with Request Type indicating a simultaneous un-appointment of the existing CCo as a user-appointed CCo and transfer the CCo functionality to a new user-appointed CCo (i.e., ReqType = 0x02).

Un-appointment of a user-appointed CCo shall only be performed by the user (or Host). Thus, CC\_CCO\_APPOINT.REQ MMEs with Request Type 0x01 and 0x02 shall only be generated by the Host.

The user-appointed CCo status field in the Discover Info BENTRY shall be used by STAs to notify their user-appointed CCo status to all other STAs in the AVLN. This information can be used by STAs to determine whether the existing CCo is a user-appointed CCo. This information can also be used to detect the presence of multiple user-appointed CCos in the AVLN.

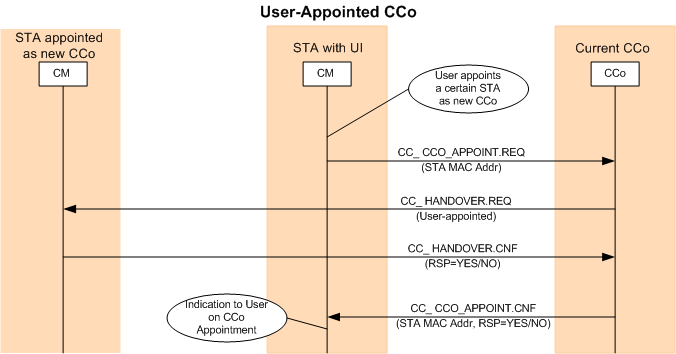


Figure 7‑17: User-Appointed CCo

|  |
| --- |
| Informative Text  The user-appointed CCo selection method may result in undesirable consequences, such as disruption of ongoing Connections, reduction in coverage within the home, no connects, and so on. Since this option requires the user to understand the CCo function, it is recommended that the option be provided only to advanced users. |

### Auto-Selection of CCo

The current CCo, unless user appointed, must analyze the Topology Table for the AVLN at least once every MaxDiscoverPeriod duration. The rules of precedence described below apply to all STAs in the AVLN to rank their suitability in assuming the CCo function. If the CCo identifies a STA in the Topology Table that ranks higher, the CCo shall initiate the CCo handover procedure to that STA. Refer to Section 7.5.

If there is a tie in the rank of STAs in the network for choice of successor CCo, the CCo may select one of the tied STAs at random to become the new CCo. The current CCo shall not hand over the CCo role to any STA that is ranked below its own rank. The current CCo may hand over the CCo role to any STA that has a rank that is the same as its own rank. Under such conditions, implementations must ensure that CCo functionality does not continuously transfer between STAs that have the same rank as the CCo.

#### CCo Capability

Every STA shall perform the mandatory functions required of a STA. However, manufacturers may differentiate STAs based on implementation attributes or other criteria (e.g., differentiated CCo-capability as described below). The CCo capability of each STA shall be classified into four categories (refer to Section 4.4.3.15.4.6.2):

1. **Level-0 CCo:** Level-0 CCo does not support QoS (i.e., cannot schedule any contention-free allocations). The mandatory functions of a Level-0 CCo are:

* AC Line Cycle Synchronization (refer to Section 5.1.1).
* All mandatory functions defined in Chapter 7, except Bandwidth Management (Section 7.8).
* CSMA-Only mode of operation with Passive Coordination (Chapter 8).
* All mandatory HomePlug 1.0.1 Coexistence functions defined in Chapter 9.

1. **Level-1 CCo**: Level-1 CCo supports QoS when operating in Uncoordinated mode. The mandatory functions of a Level-1 CCo are:

* All mandatory functions of Level-0 CCo.
* All of the mandatory Bandwidth Management functions defined in Section 7.8

1. **Level-2 CCo**: Level-2 CCo Capable STA supports Coordinated mode of operation. The mandatory functions of a Level-2 CCo are:

* All mandatory functions of a Level-1 CCo.
* Coordinated mode of operation in the presence of a Neighbor Network.

1. **Level-3 CCo:** Level-3 CCo is a future-generation CCo.

An AVLN with a Level-x CCo is also referred to as Level-x AVLN throughout the specification. All STAs shall be capable of operating as a STA in a Level-0, Level-1, or Level-2 AVLN. The CCo capability of each STA shall be provided to the CCo in the CC\_ASSOC.REQ message at the time of association. Every STA shall also declare its CCo capability in its Discover Beacon transmission. The CCo shall maintain the CCo capability of each associated STA in the network in its Topology Table. The CCo capability of the CCo of an AVLN is indicated in all Central, Proxy, and Discover Beacons.

#### Order for Selection of CCo

The order of precedence used by the Auto Selection function to identify the most suitable STA in the AVLN to assume the role of CCo is shown in Table 7‑3.

1. STA capability is the highest criterion for ranking STAs. This criterion is mandatory. A STA with Level-1 CCo capability is ranked higher than a STA with Level-0 capability and so on.
2. The number of STAs in the Discovered STA List of a STA is the next-highest criterion in ranking a STA’s suitability. This criterion is optional. The STA in the network that supports bi-directional Links with the maximum number of STAs provides the best coverage and may be deemed suitable to be a CCo. The STA in the Topology Table with the largest number of STAs in its Discovered Station List should be ranked the highest in this criterion. Ties would be broken by giving preference to the STA that was in the Discovered Station List of the most other STAs.
3. The number of networks discovered by a STA is the next most important ranking criterion. This criterion is optional. The STA in the network that discovers the largest number of neighbor networks may be deemed suitable to be a CCo to coordinate with the neighbor networks. The STA in the Topology Table with the largest number of entries in the Discovered Network List is preferred.

Table 7‑3: Order of Precedence in Selection of CCo

|  |  |  |
| --- | --- | --- |
| Order | Criteria | Note |
| 1 | User-appointed CCo (Optional) | If the user-appointed STA accepts the CCo function, this STA remains the CCo. |
| 2 | CCo Capability (Mandatory) | Level-3 CCo ranks the highest, followed by Level-2 CCo, followed by Level-1 CCo, followed by Level-0 CCo. |
| 3 | Number of discovered STAs in the Discovered Station List (Optional) | Higher is preferred |
| 4 | Number of discovered networks in the Discovered Network List (Optional) | Higher is preferred |

## Transfer/Handover of CCo Functions

The transfer of the CCo function from the current CCo to another STA (or the new CCo) in the network is shown in Figure 7‑18. The handover may be initiated when the user has appointed a new CCo or a new CCo is selected by the Auto-Selection process. The handover might not be initiated if any other change involving a Beacon countdown is in progress (e.g., Schedule change, NEK change, Beacon relocation, etc.)

Every CCo shall support “hard handover”: every CCo-capable STA shall be able to take over the role of the current CCo when requested and start transmitting Beacons for the network when the handover countdown expires.

The current CCo and the new CCo may optionally exchange CSPEC and BLE information about active Connections in the network during the handover process so that the new CCo may be able to maintain uninterrupted service at the agreed upon QoS level for those Connections. This optional process is called “soft handover.”

The following steps describe the handover process:

1. The current CCo shall send a CC\_HANDOVER.REQ message to the STA, requesting it to assume the role of the CCo. The message indicates whether a soft handover is requested.
2. The STA responds with a CC\_HANDOVER.CNF message, where the STA either accepts the transfer of the CCo function or declines to do so.

* If the STA rejects the request, the handover process is deemed to have failed. If the new CCo selection was based on the Auto-Selection process, the CCo may start a new CCo handover process with another station in the AVLN.
* If the STA accepts the request, the current CCo shall continue with the following steps.

1. The current CCo shall set the Handover-In-Progress (HOIP) bit in the Beacon to indicate that handover is in progress. When this bit is set, STAs and neighbor CCos shall wait before sending new association requests or bandwidth requests to the current CCo. Refer to Section 4.4.3.11.
2. If it is a soft handover, the current CCo shall send a **CC\_LINK\_INFO.IND** message to the new CCo to transfer the CSPEC and BLE information about all Global Links in the AVLN. The new CCo acknowledges the proper reception of this message using **CC\_LINK\_INFO.RSP**. If it is a hard handover, **CC\_LINK\_INFO.IND/RSP** messages are not exchanged.
3. The CCo shall initiate a transfer of relevant network information to the new CCo via the CC\_HANDOVER\_INFO.IND message. The message includes an indication that this message is transmitted as part of a CCo handover, the identity of the Backup CCo (if any) (refer to Section 7.9), and the list of associated and authenticated STAs in the network. The new CCo shall send a **CC\_HANDOVER\_INFO.RSP** message to indicate successful reception of **CC\_HANDOVER\_INFO.IND**.
4. The current CCo shall begin a handover countdown using the CCo Handover BENTRY (refer to Section 4.4.3.15.4.9).
5. When the countdown expires, the current CCo shall stop transmitting the Beacon and the new CCo shall take over the Beacon transmission.
6. The new CCo transmits the Beacon Time Stamp in its Beacons based on either its own STA\_Clk or NTB\_STA from the old CCo (refer to Section 5.5). Since the Network Time Base used by the new CCo can be different than the Network Time Base of the old CCo, all STAs in the AVLN with active CIDs should apply a correction to their timing based on the observed difference between the two BTSs in the last transmission of the old Beacon and the first transmission of the new Beacon.
7. If it is a soft handover, the new CCo shall also maintain the persistent schedule of the last Beacon transmitted by the “old” CCo.
8. The new CCo shall also reset the HOIP bit in the Beacon.
9. All STAs in the AVLN shall renew their TEIs (refer to Section 7.3.2.1.2) subsequent to a CCo handover.



Figure 7‑18: Transfer of CCo Function

|  |
| --- |
| Informative Text  It might not be possible to maintain uninterrupted service at the agreed-upon QoS level, even if soft handover is used. Until the new CCo obtains detailed information about CSPECs, line cycle-dependent TMs, etc., it will not have sufficient information to make correct scheduling decisions. There may also be connectivity issues that will need to be resolved (e.g., appointment of new PCos) for proper communication of schedules and connection requirements. |

## Discover Process

### Overview

The Discover Process is a periodic, low-overhead background process that is ongoing within the network where each associated and authenticated STA takes turn in transmitting a Discover Beacon as instructed by the CCo.

The purposes of the Discover Process are:

* To allow the CCo and STAs to determine the identity and capability of other STAs in the network. Each STA creates and updates its Discovered STA List as an output of this function.
* To allow the CCo to discover networks that it cannot detect directly. Each STA creates and updates a Discovered Network List as an output of this function.
* The Discovered STA Lists and Discovered Network Lists of the CCo and of all STAs are used by the CCo to create the Topology Table which is used by the CCo in the CCo-selection process.
* To allow HSTAs (i.e., STAs that cannot directly detect the Central Beacons transmitted by the CCo, but can detect Discover Beacons transmitted from certain STAs) to communicate with the CCo and to associate and authenticate with the network with the STA transmitting the Discover Beacon acting as a proxy to relay MMEs.

#### Discover Beacons

A Discover Beacon is a special type of Beacon that is transmitted by an associated and authenticated STA in the network during the Discover Process. It contains information including the NID of the network, the TEI, the MAC address (using the MAC Address BENTRY, Section 4.4.3.15.4.4), and the number of discovered STAs and networks and the CCo capability (using the Discovered Info BENTRY, Section 4.4.3.15.4.6) of the transmitting STA.

The Discover Beacon contains a copy of the Regions BENTRY and Schedule BENTRIES of the Central Beacon. Schedule BENTRIES provide the locations of the persistent and non-persistent CSMA allocations as well as all the CF allocations in the Beacon Period. HSTAs that can detect the Discover Beacon can use this information to exchange association and authentication messages with the CCo, using the STA transmitting the Discover Beacon as a relay. Refer to Section 7.7 for more details on association of HSTAs.

Each STA shall update its Discovered STA List and Discovered Network List when it detects a Discover Beacon from another STA.

The CCo shall interpret the Discovered Info BENTRY contained in the Discover Beacon. If the BENTRY indicates that the content of the Discovered STA List or Discovered Network List has been updated recently, the CCo may choose to query the STA transmitting the Discover Beacon for the latest Discovered STA List and Discovered Network List.

#### Discovered STA List and Discovered Network List

Each STA shall record the identity and attributes of every STA (from its own network and from different networks) whose Discover Beacons it can decode correctly. This information is maintained in the Discovered STA List. The Discovered STA List contains the MAC address of the STA that was heard, a flag to indicate whether the discovered STA is associated with the same or a different network, and the Short Network Identifier (SNID) (refer to Section 4.4.1.4) of the network with which the discovered STA is associated. The CCo, PCo, and Backup CCo capability and corresponding status of the transmitting STA shall also be recorded. The Signal Level and Average BLE may optionally be recorded. This optional information may be used by the CCo to determine whether or not to coordinate with another CCo or Group. The Discovered STA list shall be updated every time the STA receives a Discover Beacon from another STA. An example of a Discovered STA List for STA A in Figure 7‑19 is {MAC ADDRESS(CCo), MAC ADDRESS(B), MAC ADDRESS(C), MAC ADDRESS(E)}. STA D is not in this list, as STA A cannot detect transmissions accurately from STA D.

Each STA shall also maintain a Discovered Network List. This list shall be updated when the STA receives and decodes a Central, Proxy, or Discover Beacon with an NID (refer to Section 4.4.3.1 ) that is different from the NID of its own network. Each entry of the Discovered Network List contains the NID, SNID, network mode, hybrid mode flag, number of Beacons Slots (refer to Section 4.4.3.7), and start time of the Beacon Region of that network relative to the start of the Beacon Period.

An aging mechanism shall also be implemented to remove stale entries from the Discovered STA List and Discovered Network List. An entry from the Discovered STA List shall be removed if a Discover Beacon or other transmission from that STA has not been detected for at least Discovered\_List\_Expire\_Time. An entry from the Discovered Network List shall be removed if a Central, Proxy, or Discover Beacon or other transmission from that network has not been detected for at least Discovered\_List\_Expire\_Time.

***Note***: For each unique network discovered, the STA may have received Central, Proxy and Discover Beacons from more than one STA of that network. Even if this is the case, there shall be only one entry for that network in the Discovered Network List.

#### Topology Table

The CCo maintains a Topology Table, which is a composite of the Discovered STA Lists and the Discovered Network Lists of all the STAs and HSTAs associated and authenticated with the CCo, together with the CCo’s own Discovered STA List and Discovered Network List. The Topology Table shall contain the MAC addresses of all STAs and the Network Identifiers of all networks discovered by every STA and HSTA associated and authenticated with the CCo.

The CCo shall use its Topology Table to make decisions such as identifying HSTAs, identifying suitable PCos and establishing PxNs, and determining which STA can best fulfill the role of the PCo. The CCo may also use the Topology Table to determine the scope of a broadcast (i.e., which STAs can receive the BCAST/MCAST from the broadcasting STA) and whether bi-directional Connections can be established between STAs requesting such a point-to-point or point-to-multipoint Connection.

The CCo may use the Topology Table to try to avoid interfering with a neighbor network with which it is not coordinating directly. If a STA in the network reports a non-coordinating network in its Discovered Network List, it is recommended that the CCo refrain from scheduling any contention-free allocations that overlap with the Beacon Region or the minimum CSMA Region of any non-coordinating network. Refer to Section 5.2.5

Figure 7‑19 and Table 7‑4 show an example of a Topology Table for the AVLN “Network 1”.

Table 7‑4: Example of Topology Table

|  |  |  |
| --- | --- | --- |
| List of Associated and Authenticated STAs | Discovered STA Lists | Discovered Network Lists |
| MAC ADDRESS(CCo) | {MAC ADDRESS(A), MAC ADDRESS(B), MAC ADDRESS(C)} | {NID(NCo)} |
| MAC ADDRESS(A) | {MAC ADDRESS(CCo), MAC ADDRESS(B), MAC ADDRESS(C), MAC ADDRESS(E)} | {empty} |
| MAC ADDRESS(B) | {MAC ADDRESS(CCo), MAC ADDRESS(A), MAC ADDRESS(C), MAC ADDRESS(D), MAC ADDRESS(E)} | {empty} |
| MAC ADDRESS(C) | {MAC ADDRESS(CCo), MAC ADDRESS(A), MAC ADDRESS(B), MAC ADDRESS(D)} | {empty} |
| MAC ADDRESS(D) | {MAC ADDRESS(B), MAC ADDRESS(C), MAC ADDRESS(E)} | {empty} |
| MAC ADDRESS(E) | {MAC ADDRESS(A), MAC ADDRESS(B), MAC ADDRESS(D)} | {empty} |

#### Discover Period

The CCo shall schedule each and every associated and authenticated STA in the network and the CCo itself to transmit a Discover Beacon at least once in MaxDiscoverPeriod.

### Procedures

The Discover Process consists of the following steps:

* At least once in every Discover Period, the CCo shall schedule an opportunity to transmit a Discover Beacon for every associated and authenticated STA in the network (including HSTAs) and the CCo itself. In Uncoordinated and Coordinated modes, a special contention-free allocation (refer to Section 5.2.1.4.1) is provided using Non-Persistent Schedule BENTRY to specify the location where a Discover Beacon shall be transmitted. In CSMA-Only mode, the STA sends a Discover Beacon using CSMA/CA channel access at CAP2. The Discover BENTRY (refer to Section 4.4.3.15.4.2) is used to identify the STA that transmits the Discover Beacon. The CCo shall not include more than one Discover BENTRY in each Central Beacon.
* The STA identified by the Discover BENTRY shall construct and broadcast a Discover Beacon when scheduled by the CCo.
* Every STA that receives a Discover Beacon and is able to correctly decode the Beacon shall update its own Discovered STA List with the MAC address of the STA transmitting the Discover Beacon. The receiving STA shall also record the CCo capability of the STA transmitting the Discover Beacon. Every STA may also update its Discovered STA List every time it decodes a transmission from another STA successfully, in addition to the updates based on receipt of Discover Beacons.
* Periodically, the CCo shall query all STAs associated with the CCo, including HSTAs, to obtain their individual Discovered STA Lists and Discovered Network Lists. The CCo shall construct and update its Topology Table using this information. The CCo sends a CC\_DISCOVER\_LIST.REQ message to a STA to query for its Discovered STA List and Discovered Network List. The STA responds with the information in a CC\_DISCOVER\_LIST.CNF message. STAs in the AVLN shall also send the CC\_DISCOVER\_LIST.IND message to the CCo in an unsolicited manner when they detect a new neighboring AVLN.

## Proxy Networking with GREEN PHY Extensions to Support Routing and Repeating

Proxy Networking in GREEN PHY is an extension from Proxy Networking in HomePlug AV 1.1. The extensions add support for multiple levels of Proxy Coordinators and also support routing and repeating functionality.

Every STA in the AVLN must be able to communicate directly or indirectly with the CCo. All STAs are free to communicate with other STAs in the same AVLN, provided the power line channel characteristics between the two communicating STAs permit communication, but such communication is not required.

A Hidden STA (HSTA) is a STA that cannot communicate directly with the AVLN’s CCo and, as a result, must communicate indirectly, relaying messages via a proxy. By definition, an HSTA cannot hear the Beacons transmitted by the CCo; it can, however, infer the existence of the AVLN from Discover Beacons transmitted by other STAs or Proxy Beacons transmitted by Proxy Coordinators (PCos). The HSTA uses information from these Beacons to learn the identity of the AVLN and to identify a Proxy STA (PSTA), see below, or existing PCo to relay messages to/from the CCo. A PSTA or PCo that cannot communicate directly with the AVLN’s CCo shall route messages to the CCo based on the LRT.

Once the CCo receives a message and becomes aware of the HSTA, it appoints a STA (possibly the PSTA) as a PCo. From that point on, all messages will be relayed between the HSTA and the CCo via the PCo, routing the relay messages based on the LRT when necessary.

A Proxy Network (PxN) is a network established by the CCo when it appoints a PCo to support one or more HSTAs. A Proxy Network (PxN) is always associated with an existing AVLN and is a wholly contained part of the AVLN. It consists of a PCo (appointed by the CCo) and one or more HSTAs. PxNs are strictly the concern of the AVLN to which they belong. Multiple networks that are neighbors may each independently support PxNs.

All operations of the HSTAs are controlled by the CCo with the PCo serving as a relay. Since the HSTA must obtain information contained in the CCo’s Beacon, the PCo shall transmit a Proxy Beacon (refer to Section 7.7.4) once every Beacon Period. In Uncoordinated and Coordinated modes, the CCo shall specify TDMA allocations in which the Proxy Beacons need to be transmitted. During CSMA-Only mode, the Proxy Coordinator should transmit the Proxy Beacon using CSMA/CA at CAP3 as soon as the Central Beacon from CCo is received.

Figure 7‑19 shows an example of Network 1 containing a PxN. Within the PxN, STA B is the PCo, and STAs D and E are HSTAs.



Figure 7‑19: Proxy Network Created By Network 1

Proxy networking is an optional feature. Proxy Networking includes STAs functioning as Hidden Stations, Proxy Stations, Proxy Coordinators, and Central Coordinators. All these features are optional. In the remainder of this section, the terms Hidden Station, Proxy Station, Proxy Coordinator, and Central Coordinator refer to STAs that support the optional Proxy networking feature. Each STA shall advertise in its Discover Beacon whether it supports Proxy Networking.

When a STA announces in its Discover Beacon that it is unable to support Proxy Networking, HSTAs or the CCo shall not transmit CC\_RELAY MMEs or the CP\_PROXY\_APPOINT MMEs to that station.

### Identification of Hidden Stations

If the STA cannot hear a CCo’s Beacon, but can determine the existence of the AVLN from Discover Beacons or Proxy Beacons, it knows it is an HSTA. If the HSTA hears a Proxy Beacon, it shall request that PCo to relay its messages to the CCo. If the HSTA does not hear a Proxy Beacon, it shall select a STA from among those from which it has heard Discover Beacons, and ask that STA to serve as a PSTA and relay its initial message to the CCo. The HSTA should select a PCo or STA with the lowest Plevel.

When the CCo receives the first relayed message from a NewHSTA, it recognizes that there is a new HSTA that wishes to associate with the AVLN. The method by which the CCo receives the association request message from the HSTA is described in Section 7.7.2. The accurate decryption and interpretation of the association request message from the HSTA informs the CCo of the presence of an HSTA in the network.

Before responding to the HSTA’s association request message, the CCo shall appoint a PCo to support the HSTA as described in 7.7.3.

### Association of Hidden Station

The association of a new STA which is in range with the CCo is described in Section 7.3.2. When the new STA (or more appropriately a New Hidden STA (NewHSTA)) is out of range of the CCo, it shall perform the association described below and shown in Figure 7‑20.

Note: The fundamental association messaging is identical for both STAs and HSTAs. The only difference for a NewHSTA is the encapsulation of the messages within CC\_RELAY.REQ/IND messages and the insertion of the CP\_PROXY\_APPOINT.xxx messaging between the CC\_ASSOC.REQ and the CC\_ASSOC.CNF messages.

The new STA shall encapsulate the CC\_ASSOC.REQ MME within a CC\_RELAY.REQ MME and send it to the PCo or the PSTA. If the HSTA does not know the CCo’s TEI or MAC address, it shall set the FDA and FTEI field to the broadcast MAC address and TEI respectively. Otherwise, it shall set the FDA and FTEI field to the MAC address and TEI of the CCo.

The PCo or the PSTA shall decapsulate the CC\_ASSOC.REQ MME, re-encapsulate it inside a CC\_RELAY.IND MME. The values of the OSA and OTEI fields inside the CC\_RELAY.IND MME are set to the MAC address of the NewHSTA and the TEI used by the NewHSTA (which is 0x00). The PCo shall send the CC\_RELAY.IND MME to the CCo. Normally the PCo would use the FDA and FTEI fields of the CC\_RELAY.REQ MME to address the CC\_RELAY.IND MME but in this case, the NewHSTA could not supply them. The PCo can identify the destination by virtue of the fact the OTEI field is 0x00 (new STA) and recognize that the only STA that can be addressed when the STEI = 0x00 is the CCo (refer to Section 7.3.2.1).

The CCo shall extract the Payload field of the CC\_RELAY.IND MME and process the association request.

If it determines to accept the association request, the CCo shall appoint a PCo, as described in Section 7.7.3. The messaging for PCo appointment shall occur before the CCo replies to the association request.

After appointing a PCo, the CCo shall create a CC\_ASSOC.CNF MME and encapsulate it inside a CC\_RELAY.REQ MME. The FDA and FTEI fields inside the CC\_RELAY.REQ MME are set based on the OSA and OTEI fields inside the CC\_RELAY.IND MME. The CCo shall send the CC\_RELAY.REQ MME to the newly appointed PCo.

If the association request is rejected, the CC\_ASSOC.CNF MME shall be encapsulated as just described and sent to the PSTA that relayed the original CC\_ASSOC.REQ MME.

If the association request is accepted, the CC\_ASSOC.CNF MME contains the TEI assigned to the new STA and the lease time.

The PCo or the PSTA shall extract the CC\_ASSOC.CNF MME from the CC\_RELAY.REQ MME, encapsulates it inside a CC\_RELAY.IND MME and sends it to the NewHSTA. The FDA and FTEI fields inside the CC\_RELAY.REQ MME provide the required addressing information to send the CC\_RELAY.IND MME.



Figure 7‑20: HSTA Association

### Instantiation of Proxy Network

A PxN is established when the CCo appoints a PCo to support one or more HSTAs. The PxN consists of the PCo and the HSTA(s) that it supports.

A PxN is established when:

* The CCo learns of the presence of an HSTA through relay of a CC\_ASSOC.REQ MME from a PSTA. There are two sub-cases:
* A STA that cannot hear the CCo joins the AVLN for the first time.
* A STA that is already authenticated becomes unable to hear the CCo because of changing channel conditions.
* The CCo decides to change the PCo because:
* The PCo STA is leaving (or has left) the AVLN, or
* The CCo determines that another STA is better suited to being the PCo for this particular PxN.

The CCo may also choose to reassign HSTAs from one PxN to another PxN. The CCo should attempt to minimize the number of PxNs that it creates. This may require it to reconfigure PxNs (e.g., combine two or more PxNs into a single PxN) as the Discovery Process progresses.

#### Selecting a PCo

The determination of which STA to assign as a PCo is an implementation decision. The only requirement is that the PCo must be able to communicate bi-directionally with both the CCo and the HSTA.

Initially, the PSTA may be the only choice for the PCo of a NewHSTA. Once the NewHSTA has been admitted to the network and begun partaking in the Discovery Process, the CCo may determine that another STA is better suited (e.g., because it can support more HSTAs than the current PCo).

The CCo assigns a STA to the role of PCo by sending it a CP\_PROXY\_APPOINT.REQ MME, giving it the particulars of the assignment. The designated STA has the option of accepting or refusing the PCo service role. It shall notify the CCo of its decision via the CP\_PROXY\_APPOINT.CNF MME.

It the STA accepts the PCo role and if the ReqType = Add, the PCo shall notify the HSTA(s) it has assumed responsibility for via the PH\_PROXY\_APPOINT.IND.

#### PCo-Required Tasks

The PCo shall relay management messages between its assigned HSTA(s) and the CCo. The PCo is only required to relay MMEs which are encapsulated in a CC\_RELAY.REQ MME addressed to it. The PCo shall de-encapsulate the MMEs and re-encapsulate them in a CC\_RELAY.IND MME and send them to their destination, routing them based on the Local Routing Table (LRT – see Section 5.8.1.1) when necessary.

The PCo shall also broadcast a Proxy Beacon (Section 7.7.4) once during each Beacon Period in an allocation specified by the CCo and identified by a GLID. When a PCo cannot reliably detect the CCo, it shall select a PCo with the lowest Plevel that it can reliably detect. In Uncoordinated and Coordinated modes, the selected PCo shall specify TDMA allocations in which the Proxy Beacons need to be transmitted. During CSMA-Only mode, the PCo should transmit the Proxy Beacon using CSMA/CA at CAP3 as soon as the Proxy Beacon from the selected PCo is received.

### Proxy Beacons

A Proxy Beacon is a special type of Beacon that is transmitted by a PCo once every Beacon Period. It provides timing and schedule information for the HSTAs in the PxN. It contains the TEI of the transmitting STA (the PCo) and the Network ID of the network. It also contains any other BENTRYs found in the current Beacon, with the exception that the PCo may optionally omit an entire BENTRY (refer to Table 4-66).

### Provisioning the NMK to Hidden Stations

If the HSTA and the UIS are within range of each other, the provisioning of NMK to a HSTA is the same as described in Section 7.10.3. The HSTA may receive the broadcast message a second time, relayed from its PCo; it shall ignore this duplicate message.

If the HSTA and the UIS are not in rage of each other, the PCo’s Relay functions will support NMK provisioning automatically.

All Encryption Key management methods rely on Encrypted Payloads, so they may be relayed freely and securely without intermediate knowledge of any encryption keys. Furthermore, all of the NMK provisioning methods begin with a broadcast CSMA message, which the PCo will encapsulate and relay to its HSTAs, as described in Section 7.7.7.

If the HSTA elects to respond to the broadcast message, the HSTA shall encapsulate its response (the encrypted payload MME) in a CC\_RELAY.REQ MME and send it to the PCo which shall relay it to the UIS. The UIS shall process the MME encapsulated in the CC\_RELAY.IND MME it receives as if it had been received directly except that it shall encapsulate its reply in a CC\_RELAY.REQ MME that it sends to the PCo. The HSTA and the UIS shall continue encapsulating their messages in the CC\_RELAY.REQ MMEs for the duration of the protocol run. The PCo shall act as a relay for these messages.

### Provisioning NEK for Hidden Stations (Authenticating the HSTA)

There is no difference in the process for an HSTA joining the AVLN than for any other STA except that the messages in the protocol run (Section 7.10.4) shall be encapsulated in CC\_RELAY.xxx MMEs and relayed via the PCo.

Once the HSTA has been authenticated, the CCo shall send a CP\_PROXY\_APPOINT.REQ (ReqType=Update) to tell the PCo that the HSTA is authenticated.

An HSTA may communicate with any other STA in the AVLN (subject to powerline channel characteristics) once the HSTA is associated and been authenticated by the CCo via the PCo.

### Exchange of MMEs Through a PCo

Only MMEs destined to or originated from a CCo shall be relayed through a PCo. Data transmission is not permitted using the CC\_RELAY.IND/RSP **MMEs**.

Other MMEs may be routed as defined in Section 5.8

All CC\_RELAY.REQ MMEs shall be encrypted with the NEK unless the transmitting STA is not yet part of the Network. STAs that are not yet part of the network are permitted to send CC\_RELAY.REQ MMEs unencrypted.

The PCo shall encrypt each CC\_RELAY.IND MME using the NEK unless it is directed to a station that has not yet been authenticated into the AVLN. The PCo shall broadcast any unencrypted CC\_RELAY.IND MME that contains a CM\_ENCRYPTED\_PAYLOAD.IND. This is necessary so that the UIS can monitor the Protocol Run for an MITM attack. The HSTA shall not accept a CC\_RELAY.IND MME that contains an CM\_ENCRYPTED\_PAYLOAD.IND message unless it is broadcast.

### Transitioning from Being a STA to Being an HSTA

It is possible that an existing STA may no longer be able to decode the CCo’s Beacons reliably. This might occur because channel conditions have changed significantly. It might also occur after a CCo handover if the existing STA is no longer within range of the new CCo. When this happens, the existing STA shall re-associate with the CCo through a PSTA (or existing PCo). The procedure is the same as the association procedure for a HSTA (Section 7.7.2), except the ReqType field in the CC\_ASSOC.REQ message (refer to Section ‎11.2.28) shall indicate that it is a renewal request. Since the message was delivered to the CCo in a CC\_RELAY.IND MME, the CCo shall assign a PCo to support the new HSTA and then process the TEI renewal request normally.

### Transitioning from Being an HSTA to Being a STA

It is possible that an existing HSTA can decode the CCo’s Beacons reliably. This might occur because channel conditions have changed significantly. It might also occur after a CCo handover if the existing HSTA is within range of the new CCo. When this happens, the existing STA shall re-associate with the CCo directly. The procedure is the same as the association procedure for a STA (Section 7.3.2) except that the ReqType field in the CC\_ASSOC.REQ message (refer to Section 11.2.28) shall indicate that it is a renewal request.

Since the message was received directly from a STA that the CCo knows as hidden, the CCo shall remove the STA from the PCo’s list of supported HSTAs using the CP\_PROXY\_APPOINT.REQ message with a ReqType of “Delete.” If the PCo now has no more HSTAs assigned to it, the CCo shall shut down the PxN, as described in Section 7.7.11.

### Recovering from the Loss of a PCo

It is possible for the PCo to drop out of the network without warning, either due to equipment failure or because the user unknowingly unplugged the STA that was serving as PCo. In this case, the HSTA will observe that it is no longer receiving Proxy Beacons from its PCo. When its schedule expires, it shall select another PSTA (or PCo) and reassociate with the CCo using the procedure described in Section 7.7.8. The CCo shall notice that the TEI of the relay device is different than the TEI of the HSTA’s assigned PCo and shall assign the HSTA to a new PCo.

### Proxy Network Shutdown

Once established, the PxN shall continue until the PCo is instructed by the CCo to halt PCo functions. Instruction for this occurs via the CP\_PROXY\_APPOINT.REQ message with a ReqType = Shutdown.

The CCo shall shutdown the PxN when all of the HSTAs assigned to the PCo have:

* Disassociated,
* Had their TEI leases expire,
* Been transferred to another PCo, or
* Transitioned from being an HSTA to a STA

### Proxy Network Limitations

Proxy Networking suffers from the following limitation:

Some bandwidth-management metrics (e.g., BLEs, queue depth) are in the Frame Control; the CCo will not hear them and will not be able to provide full bandwidth-management services. A STA may provide the CCo with BLE information using the **CC\_BLE\_UPDATE**.IND message sent via relay.

## Bandwidth Manager

The main functions performed by the Bandwidth Manager in the CCo are:

* **Scheduling and Bandwidth Allocation to Connections**: The CCo receives requests from STAs in the network, requesting bandwidth assignments for Connections. In response, the CCo must assign a Global Connection ID (GLID) and schedule allocations to the Connection. The traffic characteristics, QoS guarantees, MAC services, and MAC parameters specific to a Connection are defined in the Connection Specification (CSPEC). Sounding and Channel Estimation results are used by the Bandwidth Manager in making allocations to Connection requests.
* **Admission Control**: When the CCo receives the connection establishment or connection-reconfiguration requests from a STA, the Bandwidth Manager must determine whether there is adequate bandwidth available to support the request, without compromising the QoS of existing Connections. The Bandwidth Manager is responsible for either accepting or rejecting the requests.
* **Beacon Period Configuration and Beacon Transmission**: The Bandwidth Manager must determine the allocations within a Beacon Period. It assembles and broadcasts the Beacon once every Beacon Period.

### Connection Specification (CSPEC)

Connection Specification contains the set of parameters that define the characteristics and QoS expectations of a Connection. Connections can be either unidirectional or bi-directional. For bi-directional Connections, the Connection Specification for both the Forward Link and Reverse Link is contained in the CSPEC.

The format of the CSPEC is shown in Table 7‑5. The first two octets of the CSPEC indicate the length (in octets) of the CSPEC information to follow.

The CSPEC of each Link is composed of two parts:

* The Connection Information (CINFO)
* The QoS and MAC parameters (QMP)

Table 7‑5: Format of Connection Specification (CSPEC)

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Octet Number | Field Size (Octets) | Definition |
| CSPEC\_LEN | 0 - 1 | 2 | Length of CSPEC, including the 2-octet CSPEC\_LEN field  0x0000 = 0 octets, and so on |
| CINFO (Forward) | - | 1 or 5 | Forward Connection Information |
| CINFO (Reverse) | - | 1 or 5 | Reverse Connection Information |
| QMP (Forward) | - | Var | Forward QoS and MAC Parameters  Only present if Connection requires a Forward Link |
| QMP (Reverse) | - | Var | Reverse QoS and MAC Parameters  Only present if Connection requires a Reverse Link |

CINFO identifies the attributes of the Connection and the MAC and PAL operations required by the Connection at the source and destinations STAs. The format of the CINFO fields is shown in Table 7‑6. A separate CINFO field is required for the forward and reverse directions of the Connection.

The CINFO, QoS, and MAC parameter fields specifically apply to the forward or Reverse Links, as indicated in the CSPEC.

Table 7‑6: Format of Connection Information (CINFO)

| Field | Octet | Field Size (Octets) | Description | Reconfigurable |
| --- | --- | --- | --- | --- |
| Valid CINFO | 0 | 1 | 0x00 = CINFO is not valid.  0x01 = CINFO is valid.  0x02 - 0xFF = reserved  Valid CINFO shall be set to 0x00 if the corresponding Link is not present.  If Valid CINFO is set to 0x00, the remaining field in the CINFO and the corresponding QMP fields are not present in the CSPEC. | No |
| MAC  Service Type | 1 | 1 | 0x00 = contention-free service. In this case, connection setup shall fail if the requested Global Link(s) cannot be established.  0x01 = contention-based service. In this case, connection setup shall use local link(s).  0x02 = contention-free service preferred. In this case, connection will use Global Links if the network is operating in Uncoordinated mode or Coordinated mode and the Global Link(s) can be established. Otherwise, the connection will use contention based service.  0x03 - 0xFF = reserved  Refer to Section 5.2.3.5.3 for details. This field is only present when Valid CINFO is set to 0x01. | No |
| User Priority | 2 | 1 | For contention-based service, this field indicates the channel access priority of the Connection.  For contention-free services and contention-free preferred services, this field indicates the channel access priority to be used by packets belonging to this Link when they are transmitted in CSMA allocations.  This field is only present when Valid CINFO is set to 0x01.  0x00 = CAP0,  0x01 = CAP1,  0x02 = CAP2,  0x03 = CAP3,  0x04 - 0xFF = reserved | Yes |
| Arrival Time Stamp to HLE (ATS) | 3 | 1 | 0x00 = ATS should not be passed to the HLE.  0x01 = ATS should be passed to the HLE (at the receiver) for each MSDU.  0x02 - 0xFF = reserved  This field is only present when Valid CINFO is set to 0x01. | No |
| Smoothing | 4 | 1 | 0x00 = smoothing is not requested.  0x01 = if supported, receiver shall activate smoothing function / delay compensation function.  0x02 - 0xFF = reserved  This field is only present when Valid CINFO is set to 0x01. | No |

The QoS and MAC parameters identify the QoS requirements (delay, jitter, data rates), as well as MAC parameters that are specific to the particular Connection. The QoS parameters are generated by the Connection Manager using the Auto-Connect function or through PAL-specific primitives exchanged between the higher layer applications and the CM.

Each QoS and MAC parameter field consists of a Forward/Reverse (F/R) field, a Length (LEN) field, and a 1-octet Field Identifier (FID) field, followed by the Body of the QoS and MAC Parameter field. Table 7‑7 shows the format of a QoS and MAC parameter field.

The QMPs exchanged between the HLE and Connection Manager are shown in Table 7‑8.

The QoS and MAC parameters exchanged between two CMs include the parameters shown in Table 7‑8 and Table 7‑9.

The QoS and MAC parameters exchanged between the CM and CCo are shown in Table 7‑10.

Table 7‑7: Format of QoS and MAC Parameter Field in CSPEC

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Octet Number | Field Size (Octets) | Description |
| Forward/Reverse (F/R) | 0 | 1 | 0x00 = forward (from source to receiver)  0x01 = reverse (from receiver to source)  0x02 - 0xFF = reserved |
| Length (LEN) | 1 | 1 | Length of the Body Field, in Octets  0b0000000 = 0 octets, and so on. |
| Field Identifier (FID) | 2 | 1 | Identifier of the QoS and MAC parameter field |
| Body | - | Var | Data of the QoS and MAC parameter field |

| Table 7‑8: QoS and MAC Parameter Fields Exchanged between HLE and CM, and between CMs | | | | | |
| --- | --- | --- | --- | --- | --- |
| CSPEC Field | FID | LEN (Octets) | Descriptions | Reconfigurable |
| Delay Bound | 0x00 | 4 | Maximum amount of time allowed to transport an MSDU, measured from the time the MSDU arrives at the Convergence Layer SAP of the transmit station until the time is transmitted or retransmitted successfully across the powerline network and delivered out of the Convergence Layer SAP of the receive station(s). Unit is microseconds.  0x00000000 = 0 microseconds, and so on.  The maximum allowed value is 10 seconds. | Yes |
| Jitter Bound | 0x01 | 4 | Maximum difference in the delay experienced by an MSDU. Delay is measured from the time the MSDU arrives at the Convergence Layer SAP of the transmit station until it is successfully delivered out of the Convergence Layer SAP of the receive station(s). Unit is microseconds.  0x00000000 = 0 microseconds and so on.  The maximum allowed value is 10 seconds. | Yes |
| Average MSDU Size | 0x02 | 2 | Average MSDU Payload Size in octets (refer to Section 12.3.2.1.1)  0x0000 = 0 octets and so on. | Yes |
| Maximum MSDU Size | 0x03 | 2 | Maximum MSDU Payload Size in octets. If this parameter is not specified, a value of “Default Maximum MSDU Size” is assumed.  0x0000 = 0 octets and so on. | Yes |
| Average Data Rate | 0x04 | 2 | The average application data rate specified at the CL SAP that is required for transport of MSDUs belonging to this Link. This does not include the MAC and PHY overhead incurred in transferring the MSDU. Units in multiples of 10 Kilobits per second (kbps).  0x0000 = 0 Kbps, 0x0001 = 10 Kbps and so on. | Yes |
| Minimum Data Rate | 0x05 | 2 | The minimum application data rate specified at the CL SAP that is required for transport of MSDUs belonging to this Link. This does not include the MAC and PHY overhead incurred in transferring the MSDU. Units in multiples of 10 Kilobits per second (kbps).  0x0000 = 0 Kbps, 0x0001 = 10 Kbps and so on. | Yes |
| Maximum Data Rate | 0x06 | 2 | The maximum application data rate specified at the CL SAP that is required for transport of MSDUs belonging to this Link. This does not include the MAC and PHY overhead incurred in transferring the MSDU. Units in multiples of 10 Kilobits per second (kbps).  0x0000 = 0 Kbps, 0x0001 = 10 Kbps and so on. | Yes |
| Maximum Inter-TXOP time | 0x07 | 2 | Maximum time allowed between two transmission opportunities (TXOPs) on the medium for this Link. Unit is microseconds.  0x0000 = 0 microseconds. | Yes |
| Minimum Inter-TXOP time | 0x08 | 2 | Minimum time allowed between two transmission opportunities (TXOPs) on the medium for this Link. Unit is microseconds.  0x0000 = 0 microseconds | Yes |
| Maximum Burst Size | 0x09 | 2 | Maximum size of a single contiguous burst of MSDUs that is generated by the application at the maximum rate. Units in octets.  0x0000 = 0 octets | Yes |
| Exception Policy | 0x0a | 1 | 0x00 = terminate the Connection.  0x01 = reconfigure the Connection.  0x02 to 0xFF = reserved. | Yes |
| Inactivity Interval | 0x0b | 4 | Maximum duration of time a Connection is allowed to remain inactive without transporting any application data before the CM can release the allocation. The units are milliseconds.  0x00000000 = indefinite inactivity interval (i.e., Connection should be considered active until explicitly terminated).  0x00000001 = 1 millisecond and so on.  The maximum allowed value is 60 seconds. | Yes |
| MSDU Error Rate | 0x0c | 2 | MSDU error rate requested. It is expressed as x•10-y. The value of x is specified in the most-significant 8 bits in unsigned integer format. The value of y is specified in the least-significant 8 bits in unsigned integer format. | Yes |
| CLST | 0x0d | 1 | Convergence Layer SAP Type This field supports negotiation of Connections using Convergence Layer SAPs other than the 802.3 SAP. If this field is not present, the 802.3 SAP is assumed.   | **CLST Value** | **Interpretation** | | --- | --- | | 0x00 | IEEE 802.3 SAP | | others | Reserved | | No |
| CDESC | 0x0e | 13 or 37 | Connection Descriptor (refer to Section 7.8.1.1) |  |
| Vendor Specific | 0x0f | Var | Vendor-Specific QoS and MAC information (refer to Section 7.8.1.1) | Yes |
| ATS Tolerance | 0x10 | 2 | Measured variance in value of Arrival Time Stamp (ATS) from the synchronized Network Time Base at the time the ATS is applied to the MSDU arriving at the Convergence Layer SAP of the transmit station. Unit is microseconds.  0x0000 => 0 microseconds and so on. | Yes |
| Smallest Tolerable Average Data Rates | 0x11 | 2 | Smallest Tolerable Average Data Rate indicates the smallest average Data rate at which the application is capable of operating. Units in multiples of 10 Kilobits per second (kbps).  0x0000 = 0 Kbps, 0x0001 = 10 Kbps and so on. | No |
| Original Average Data Rate | 0x12 | 2 | Original Average Data Rate indicates the average data rate at which the application intends to operate when sufficient station and network resources are available. Units in multiples of 10 Kilobits per second (kbps).  0x0000 = 0 Kbps, 0x0001 = 10 Kbps and so on. | No |

| Table 7‑9: Additional QoS and MAC Parameter Fields Exchanged between Two CMs | | | | | |
| --- | --- | --- | --- | --- | --- |
| QoS and MAC Parameter Field (CM-CM) | FID | LEN (Octets) | Descriptions | Reconfigurable |
| Rx Window Size | 0x13 | 2 | Receive window size in number of 512-octet segments.  0x0000 = 0 Receive Window Size and so on. | No |
| Smoothing Buffer Size | 0x14 | 3 | The smoothing buffer size in octets that is required to support the Link at the transmitter and receiver. If this field is not present and smoothing is requested, the default buffer size is chosen to be the product of Delay (in seconds) and Average Data Rate (in bits per second)  0x000000 = 1 octet and so on. | No |
| Bidirectional Burst | 0x15 | 1 | This parameter shall only be present in QMPs of Local Link that belong to a bidirectional Connection, and whose traffic is intended to be transmitted as part of Reverse SOF of the Bidirectional Bursts initiated by the other Link in the connection. The presence of this indicates that the payload of this Local Link has to be transmitted as part of Reverse SOF of Bidirectional Bursts.  A value of 0x00 is used when both links are local links.  A value of 0x01 indicates that the Bidirectional Bursts in CFP will always end with a SACK.  A value of 0x02 indicates that the Bidirectional Burst in CFP may end with a Reverse SOF.  All other values are reserved. | No |

| Table 7‑10: QoS and MAC Parameter Fields between CM and CCo | | | | | |
| --- | --- | --- | --- | --- | --- |
| QoS and MAC Parameter Field (CM-CCo) | FID | LEN (Octets) | Descriptions | Reconfigurable |
| TXOPs per Beacon Period | 0x80 | 1 | The number of uniformly spaced TXOPs requested per Beacon Period.  0x00 = 1 TXOP per Beacon Period  0x01 = 2 TXOPs per Beacon Period  0x02 = 3 TXOPs per Beacon Period  0x03 = 4 TXOPs per Beacon Period  0x04 - 0xFF = reserved | No |
| Average Number of PBs per TXOP | 0x81 | 2 | The average number of 520-octet PHY Blocks per TXOP required for transporting MSDUs belonging to this Link.  0x0000 = 0 PBs per TXOP and so on. | Yes |
| Minimum Number of PBs per TXOP | 0x82 | 2 | The minimum number of 520-octet PHY Blocks per TXOP required for transporting the MSDUs belonging to this Link.  0x0000 = 0, and so on | Yes |
| Maximum Number of PBs per TXOP | 0x83 | 2 | The maximum number of 520-octet PHY Blocks per TXOP required for transporting the MSDUs belonging to this Link.  0x0000 = 0, and so on | Yes |
| PPB\_Threshold | 0x84 | 2 | The Pending PHY Block (PPB) threshold indicates the threshold of Pending PBs at which the Link requires extra bandwidth to clear the backlog.  If there is sufficient bandwidth available, the CCo should provide Extra Allocation when the PPB threshold is exceeded.  0x0000 = 0 PBs and so on | Yes |
| Surplus Bandwidth | 0x85 | 2 | Surplus Bandwidth (refer to Section 7.8.1.4) | No |
| Exception Policy | 0x0a | 1 | 0x00 = terminate the Connection.  0x01 = reconfigure the Connection.  0x02 to 0xFF = reserved | Yes |
| CDESC | 0x0e | 13 or 37 | Connection Descriptor (refer to Section 7.8.1.1) | Yes |
| Vendor Specific | 0x86 | Var | Vendor-Specific QoS and MAC information (refer to Section 7.8.1.1) | Yes |
| Smallest Tolerable Average Number of PBs per TXOP | 0x87 | 2 | Smallest Tolerable Average Number of PBs per TXOP indicates the smallest average number of PBs per TXOP at which the application is capable of operating.  0x0000 = 0 PBs per TXOP and so on. | No |
| Original Average Number of PBs per TXOP | 0x88 | 2 | Original Average Number of PBs per TXOP indicates the average number of PBs per TXOP at which the application intends to operate when sufficient station and network resources are available.  0x0000 = 0 PBs per TXOP, and so on | No |
| Bidirectional Burst | 0x89 | 1 | This parameter indicates that this Global link will be used for Bidirectional Bursting for a Local Link that is part of this connection (refer to Section 5.4.7).  0x00 = bidirectional burst will always end with a SACK.  0x01 = bidirectional burst may end with a Reverse SOF.  All other values are reserved. | No |

#### Connection Descriptor (CDESC)

The QoS and MAC Parameters of the CSPEC exchanged between the HLE and CM, between CMs, and between CM and CCo can optionally include a Connection Descriptor (CDESC). A Connection Descriptor is a set of fields that defines the Connection to the HLEs (refer to Table 7‑11). It is used by UPnP QoS and possibly other HLEs. This field is for HLE use only; the AV system merely passes the information to all involved STAs (including the CCo) so the HLE can later construct a list of the active Connections without having to query every STA.

There shall be at most one CDESC per CSPEC (even if the Connection is bidirectional) and the Forward/Reverse field of the QMP field containing the CDESC shall have no meaning and shall be ignored by the receiving entities.

Table 7‑11: Format of the Body of Connection Descriptor

| CDESC Field | Octet Number | Field Size (Octets) | Descriptions |
| --- | --- | --- | --- |
| IP Version | 0 | 1 | IP protocol version  0x00 = IP Version 4  0x01 = IP Version 6  0x02 - 0xFF = reserved |
| Source IP Addr | - | 4 or 16 | IP Address of Source HLE  4 Octets Long for IP v4, 16 Octets Long for IP v6 |
| Source IP Port | - | 2 | IP Port number (corresponding to the Protocol Type) of Source HLE |
| Destination IP Addr | - | 4 or 16 | IP Address of Destination HLE  4 octets long for IP v4, 16 octets long for IP v6 |
| Destination IP Port | - | 2 | IP Port number (corresponding to the Protocol Type) of Destination HLE |
| Protocol Type | - | 1 | IP Protocol Type (e.g., TCP, UDP) |

#### Vendor-Specific QoS and MAC Parameters

The QoS and MAC Parameters of the CSPEC exchanged between the HLE and CM, between CMs and between CM and CCo may optionally include vendor-specific parameters. A vendor-specific QoS and MAC parameter has a Field Identifier (FID) of 0xFF. The first 3 octets of the Data field of this parameter contain the IEEE-assigned Organizationally Unique Identifier (OUI) assigned to the vendor as shown in Table 7‑12 and specified in Section 11.7.

Table 7‑12: Format of the Body of Vendor-Specific MAC and QoS Parameter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Octet Number | Bit Number | Field Size (Bits) | Definition |
| OUI | 0 | 7 - 0 | 8 | OUI first octet |
| 1 | 7 - 0 | 8 | OUI second octet |
| 2 | 7 - 0 | 8 | OUI third octet |
| Vendor Defined | -- | -- | -- | Vendor defined |

#### Ordering of Fields within the CSPEC

CSPECs exchanged between CMs and between CM and CCo shall obey the following rules:

* If the CSPEC contains both Forward Link and Reverse Link CSPECs, the Forward Link QMP field shall be presented before the Reverse Link.
* Within the CSPEC of each Link, the QMP fields shall be arranged in ascending order of the Field Identifier values. For example, if delay and jitter are both exchanged for the Forward Link between two CMs, the delay parameter appears before the jitter parameter in the Forward Link CSPEC.

#### Surplus Bandwidth

Surplus Bandwidth indicates the excess amount of bandwidth required to support the Link relative to the Average Number of PBs per Transmit operation. A value of 0x00 indicates that no surplus bandwidth is required. A value of 0x01 indicates that one PB per Transmit Operation amount of surplus bandwidth is required and so on.

The CCo shall use surplus Bandwidth during the initial admission control procedure. A Connection shall be rejected if the Average number of PBs per Transmit Operation along with the requested Surplus Bandwidth cannot be allocated. The CCo shall use the following parameters for converting PBs to allocation time requirements:

MaxFL\_AV value of 2501.12 μsec shall be used in AV-only Mode. In Hybrid mode, the HomePlug 1.0-compatible Frame Length (FL\_AV) shall be used (refer to Section 9.4).

RIFS\_AV value of RIFS\_AV\_default shall be used.

CFIFS\_AV value of 140 μsec shall be used.

Tone Map boundaries as in Section 5.2.6.4 shall be obeyed.

Bursting shall be assumed with a maximum of four MPDUs per Burst.

#### Minimum Set of QoS and MAC Parameters

If contention-free service is requested in the MAC Service Type parameter of CINFO, the following minimum set of QoS and MAC parameters between the HLE and the CM and between the two CMs shall be specified:

* Average Data Rate
* At least one of Delay Bound and Maximum Inter-TXOP Time

The minimum additional set of QoS and MAC parameters that shall be exchanged between the two CMs are:

* RX Window Size

If contention-free service is requested in the MAC Service Type parameter of CINFO, the following minimum set of QoS and MAC parameters between the CM and the CCo shall be specified:

* TXOPs Per Beacon Period
* Average Number of PBs per TXOP

#### CSPEC Reconfigurability

Most, but not all, CSPEC fields may be modified (reconfigured) over the life of the Connection. There are a few that shall not be modified once the Connection is established. Whether a field can be modified is indicated in the “Reconfigurable” column of Table 7‑8, Table 7‑9, and Table 7‑10. Connection modification request will be rejected if the reconfigured CSPEC cannot be supported.

#### Connection Specification (CSPEC) for GREEN PHY

A station implementing only GREEN PHY is not required to implement this feature (i.e., only priority “connectionless” links are required to be implemented by a GREEN PHY station).

### Scheduler and Bandwidth Allocation

The CCo implements scheduling algorithms for the CFP. These algorithms make bandwidth assignments in the form of time grants. The assignments are carried in the Persistent and Non-Persistent Schedule BENTRYs that are broadcast in the Beacon.

The scheduling algorithms make updates to the schedules based on the following events:

* Requests for new Links from STAs within the network.
* Request for Link reconfigurations from existing Links within the network.
* Changes to the capacity of existing Links as a result of changes to the physical channel.

Figure 7‑21 contains a finite state machine diagram describing the life cycle of a Global Link. The lifecycle of the Link typically begins when the CCo receives a CC\_LINK\_NEW.REQ message from a station in the network.

The CC\_LINK\_NEW.REQ message contains the CSPEC and CINFO parameters for a Connection that requires use of the CFP. It also includes Bit Load Estimates (BLEs) for the physical channel between the STAs requesting the Link. These BLEs are based on communications between the STA that occurred during the CP.



Figure 7‑21: Global Link Life Cycle

The CCo performs admission control on the Link based on the BLE. If there is sufficient bandwidth available to support the Link, the CCo sends the CC\_LINK\_NEW.CNF message indicating success and containing the GLID for the Link(s) to both STAs. At this point, the Link is active and carrying user data.

If the admission-control function returns failure to allocate, the CCo sends a CC\_LINK\_NEW.CNF message indicating failure along with a Proposed CSPEC to both STAs involved in the connection setup.

While the Link is active, the CCo “sniffs” on the FC fields transmitted in the delimiter by the STA. The SOF FC contains a BLE field and a PPB field. The CCo uses the contents of the BLE field to update its estimates of the channel capacity vs. line cycle. It uses the PPB field to determine whether to increase or decrease persistent allocation and/or to provide non-persistent allocation for the Link.

As Figure 7‑21 shows, the CCo may update schedule (i.e., Persistent and Non-Persistent allocations) based on one of the following events:

* The STA may request a change in the QoS parameters of a Link. In this case, if sufficient bandwidth is available, the CCo updates the Schedule; otherwise, the modification request is rejected.
* The CCo decides that a change to the schedule is needed based on the PPB and BLE fields.

Schedule updates are indicated to all stations in the network using Persistent and Non-Persistent BENTRYs in the Beacon.

#### 7.8.2.1 Scheduler and Bandwidth Allocation for GREEN PHY

A station implementing only GREEN PHY is only required to be Level-0 capable. Therefore, HomePlug GREEN PHY Stations are not required to support Scheduling and Bandwidth Allocation functionality.

### Connection Admission Control

The Connection Admission Control procedure ensures that the station and network resources are not over allocated, thus ensuring the QoS guarantee on admitted Connections.

CMs shall execute the Connection Admission Control procedure whenever a new Connection is requested or an existing Connection is modified. CMs may also continuously monitor existing Connections for adherence to the negotiated traffic characteristics (traffic policing) and QoS guarantees. Violation of the CSPEC parameters may cause the CM to reconfigure or tear down an existing Connection.

The CCo shall execute the admission control procedure when a new Global Link is requested or an existing Global Link is modified. The CCo shall also continuously monitor all existing Global Links and it may modify or terminate Connections if the available bandwidth is not sufficient to satisfy the CSPEC requirements.

#### 7.8.3.1 Connection Admission Control for GREEN PHY

A station implementing only GREEN PHY is only required to be Level-0 capable. Therefore, HomePlug GREEN PHY Stations are not required to support Connection Admission Control functionality.

### Beacon Period Configuration

The Bandwidth Manager must determine the allocations within a Beacon Period. If Neighbor Networks are present, it must ensure that allocations within its network are compatible with the Neighbor Networks in its INL.

The Bandwidth Manager assembles the portion of the Beacon payload dealing with bandwidth allocation.

### Bandwidth Allocation for GREEN PHY

Green PHY Preferred (GPP) allocations are based on the presence of Green PHY stations in the AVLN. Green PHY stations may treat the GPP allocation to have forever persistence (i.e., always present). Thus, Green PHY stations may transmit in GPP allocations without processing Persistent and Non-Persistent Schedules in the Central Beacon.



Figure 7‑22: Green PHY Preferred CSMA allocation immediately following the Beacon Region.

|  |
| --- |
| Informative Text  Future HomePlug AV CCo operating in Uncoordinated Mode provides Green PHY Preferred (GPP) allocations for Green PHY traffic. Future HomePlug AV CCo operating in Coordinated Mode provides GPP allocations for Green PHY Traffic provided it is not Coordinating with a HomePlug AV 1.1 CCo. The duration of the GPP allocation is 7% of the Beacon Period. GPP allocation immediately follows the Beacon Region. Future HomePlug AV CCo may use Persistent or non-Persistent schedule BENTRIES to indicate the GPP allocations. Future HomePlug AV CCo provides a CSMA allocation of at least MinCSMARegion duration immediately following the GPP allocation. |

## Backup CCo and CCo Failure Recovery

Each CCo-capable STA may optionally implement the “Backup CCo” function. When such a STA is appointed by the current CCo as the Backup CCo of a network, and when the current CCo suffers from a failure, the Backup CCo shall execute the CCo failure recovery procedure.

### Backup CCo

The CCo may optionally designate a STA as the Backup CCo by sending a CC\_BACKUP\_APPOINT.REQ message with the Appoint/Release flag set to 0x00 (Appoint). The STA that is identified shall respond with a CC\_BACKUP\_APPOINT.CNF message. Depending on whether the STA supports the Backup CCo function, it may accept or reject the request.

The function of the Backup CCo is to assume the role of the CCo in the event of a CCo failure.

The Backup CCo may be selected by an analysis of the Topology Table using the criteria defined in Table 7‑3. The Backup CCo is not selected on a periodic basis like the Auto-selection function for the CCo. The CCo may evaluate its Topology Table to find an alternate Backup CCo only when the current Backup CCo disassociates from the network.

When the CCo role is handed over to a new STA, the new CCo shall be informed of the identity of the Backup CCo (if any) in the CC\_HANDOVER\_INFO.IND message. The new CCo may choose to appoint a new STA as the Backup CCo. When the new CCo chooses a different STA to be the Backup CCo, it shall inform the current Backup CCo via the CC\_BACKUP\_APPOINT.REQ message with the Appoint/Release flag set to 0x01 (Release), prior to appointing a new Backup CCo.

The CCo should keep the Backup CCo up-to-date with the list of associated and authenticated stations in the AVLN by sending **CC\_HANDOVER\_INFO.IND** message with the Reason Code set to 0x01 (Update of network information to Backup CCo). The Backup CCo shall send a **CC\_HANDOVER\_INFO.RSP** message to indicate successful reception of **CC\_HANDOVER\_INFO.IND**.

The current CCo may also send **CC\_LINK\_INFO.IND** message(s) to the Backup CCo to transfer the CSPEC and BLE information about all Global Links in the AVLN. The Backup CCo shall acknowledge the proper reception of this message using **CC\_LINK\_INFO.RSP**.

### CCo Failure Recovery

It is possible for the existing CCo to drop out of the network without warning either because of equipment failure or because the user unknowingly unplugged the STA that was serving as the CCo.

If a Backup CCo is appointed in the network and if the Backup CCo does not receive any Central Beacons from the CCo for (Max\_Missed\_Beacon) Beacon Periods, and during the same time period, the Beacon Detect Flag (refer to Section 4.4.1.5.2.5) in the Frame Controls of all transmissions indicate Beacons are not detected, the Backup CCo shall assume the role of the new CCo, and perform the following steps:

1. The Backup CCo shall transmit a Central Beacon in the same Beacon Slot used by the old CCo once every Beacon Period.
2. The Backup CCo may maintain the persistent schedule of the last received Beacon.
3. The Backup CCo shall include the MAC Address BENTRY in at least the first 10 Central Beacons it transmits following a CCo failure. This will enable STAs in the AVLN to determine that a Backup CCo has taken control of the AVLN.
4. The Backup CCo may request the CSPEC and BLE of all active contention-free Links in the schedule from each STA involved using the CC\_LINK\_INFO.REQ/CNF message exchange.
5. The Backup CCo may appoint a new Backup CCo.
6. The Backup CCo shall perform normal CCo operations.
7. All STAs in the network shall adjust their clocks based on the new Network Time Base being transmitted by the Backup CCo.
8. All STAs in the AVLN shall renew their TEIs (refer to Section 7.3.2.1.2) once the Backup CCo starts sending the Central Beacons.

All STAs in the AVLN shall reinitiate the power-on network procedure (refer to Section 7.1) when a CCo failure occurs and the Backup CCo (if any) fails to takeover as the new CCo. A STA should wait for at least CCo\_Failure\_Time before reinitiating the power-on network procedure.

## Security

### Security Overview

As with HomePlug AV, HomePlug GREEN PHY security performs two functions:

* Controlling access to the AVLN. This function is described in Section 7.10.3.
* Securing the privacy of data transferred on the AVLN. This function is described in Section 7.10.6.1.

Security is provided by means of a single encryption algorithm and a single hash function:

* 128-bit AES encryption (refer to Section 7.10.6)
* SHA-256 secure hash function

Security is also enhanced through the use of nonces (Section 7.10.7.3), which help to prevent unauthorized replays of MMEs.

Access to the AVLN and ability to participate in the AVLN is provided by encryption keys and passwords.

|  |
| --- |
| Informative Text  With the exception of the encryption of PBs in the MAC/PHY using the NEK, all security-related activities (e.g., payload encryption and key management) take place in an entity that may be thought of as the “Security Layer.” This entity is implementation dependent and is assumed to lie in the Control Plane. The Security Layer is responsible for generating the CM\_ENCRYPTED\_PAYLOAD.IND messages. |

### Encryption Keys, Pass Phrases, Nonces, and Their Uses

As with HomePlug AV, all the encryption keys used in HomePlug GREEN PHY are 128-bit AES keys. These may be machine-generated or they may be based on pass phrases. Password” and “pass phrase” both describe the same object. The term “pass phrase” is preferred, but the term “password” is also preserved because of its use in acronyms such as “DPW” and “NPW.” Keys based on pass phrases come in two varieties:

* Device Passwords (DPWs)
* Network Passwords (NPWs)

In addition, as with HomePlug AV, GREEN PHY uses Hash Keys (longer, machine-generated strings) for the UKE protocol, and nonces for freshness and association between messages in a protocol run.

#### Device Access Key (DAK)

The Device Access Key (DAK) is unique to a STA. Each STA is provided with a unique DAK during manufacture. Another STA may—if it knows a particular STA’s DAK ⎯ use that DAK to encrypt a message intended only for the particular STA. Upon receipt, such a message is treated as equivalent to the direct entry of the NMK from HLE. The DAK shall never be reset and it shall never be sent over the medium, even in encrypted MMEs. Support for the DAK is mandatory.

|  |
| --- |
| Informative Text  Implementers may choose to provide a means to change a compromised DAK using some type of firmware upgrade methodology. |

#### Device Password (DPW)

As part of the packaging of the product (perhaps as a label on the back or bottom of the product), the user shall be provided with a Device Password (DPW) ⎯ a password that will uniquely generate the new STA’s unique DAK via the standard hashing algorithm defined in Section 7.10.7.1. The DPW is the value that is actually entered by the user during Authorization using the DAK (refer to Section 7.10.3.4).

#### Network Membership Key (NMK)

The Network Membership Key (NMK) is used by a STA to prove its membership in an AVLN (or a sub-AVLN); i.e., its right to join (participate in) a sub-AVLN. Thus the NMK defines the sub-AVLN. The user may designate the NMK(s) — by entering an NPW — or may elect machine generation (which is more secure). Hashing the NMK produces the default NID offset, but the NMK may be associated with a non-default NID in some cases. Support for two NMKs (one present and one future) and their associated NIDs is mandatory.

The NMK is associated with a Security Level (SL). The NMK’s SL defines the SL of the sub-AVLN and must be the same for all sub-AVLNs in an AVLN; the SL determines what key distribution methods are allowed for that NMK. The SL associated with an NMK is passed to the STA as part of the NID in the **CM\_SET\_KEY.REQ** MME or in the **APCM\_SET\_KEY.REQ** primitive.

#### Network Password (NPW)

The Network Password (NPW) is the value that generates the NMK when it is run through the hashing function described in Section 7.10.7.1.

#### Network Encryption Key (NEK)

During normal operation, most messages are encrypted using the Network Encryption Key (NEK), which shall only be generated by the CCo and is never exposed to the user. Support for two NEKs (one present and one future) is mandatory.

The NEK shall only be set by a CCo internally generating it randomly (refer to Section 7.10.7.2), or by the NEK-provisioning processes, whereby the CCo provides a STA with the NEK in an MME encrypted with the NMK. The NEK is not known by a STA until the STA has completed the authentication process (refer to Section 7.10.4) and joined the AVLN.

#### Temporary Encryption Key (TEK)

The Temporary Encryption Key (TEK) is an AES key that is used to encrypt messages on a temporary private channel between two STAs. It may be distributed using the receiver’s DAK, or generated by the Unicast Key Exchange (UKE) protocol (protected from standard equipment by unicast, and possibly at the signal level by tone map modulation). It can be used over unauthenticated channels (i.e., it may be distributed without proof of freshness using the DAK, or it may be generated using UKE). It shall not be distributed to more than one STA, and both sender and receiver must discard it after no more than Max\_TEK\_Lifetime. If the TEK was exchanged using UKE, it may be used only until a protocol message with PMN=0xFF is sent or until the protocol aborts (by either STA). Once a message with PMN=0xFF has been sent, the TEK established at the beginning of that protocol run shall no longer be used. Support for at least one TEK is mandatory. Refer to Section 7.10.7.2 for generation of random AES keys.

#### Nonces

Nonces are pseudo-random numbers (i.e., the sequence of values that they take for a given station are unpredictable) that are used only once. Practically, as the number of bits in a nonce is finite, they may repeat, but the same nonce should never be used with the same encryption key; that is, changing the encryption key in essence clears the set of unusable nonces.

Nonces are used to prevent replay attacks (when a STA receives a nonce that it recently generated, it can be assured that the message it received was composed recently also). They may also be used to associate messages in a protocol run, though this is accomplished in through the use of Protocol IDs, Protocol Run Numbers, and Protocol Message Numbers. A station need only generate one nonce per run of a protocol (i.e., the value of My Nonce for that STA may remain constant for a run of a protocol), but a new nonce should be generated for each new protocol run. This reflects the purpose of nonces to provide a STA with a quantity it believes to be freshly generated (to defeat replay attacks) and to use for association of messages within the protocol run. Refer to Section 7.10.7.3 for generation of nonces.

### Methods for Authorization (NMK Provisioning)

An AVLN is defined as a set of stations that share a common NID and CCo with which they share a common NMK and Security Level. Typically, all STAs in an AVLN can communicate with each other and share a common NMK and Security Level, but the CCo may form separate sub-AVLNs within an AVLN, each with its own NMK.

Before a new STA can participate in an AVLN, it must determine whether to join an existing AVLN or form a new one. The determination of whether to join a given AVLN consists of three decisions:

1. At what level of security does the user wish the AVLN to operate?
2. Does the STA want to authorize the AVLN? (More precisely: does the STA’s owner want to have the STA join the AVLN?)
3. Does the AVLN want to authorize the STA’s membership request? (More precisely: does the AVLN’s owner want to allow the STA to join the AVLN?)

These decisions must occur before a new STA can fully participate in the selected AVLN. The first decision determines the operating mode of the AVLN and the ways in which its NMK(s) may be distributed. All three decisions implicitly require user participation and approval. It is possible for all three decisions to be made simultaneously by a single action of the user. Several alternative methods of NMK Provisioning are provided to enable these UEs. These methods are specified in this section.

Regardless of whether the three processes occur separately or concurrently, the end result — if successful — is that the STA possesses a Network Membership Key (NMK) that it can use now and in the future to join the AVLN.

In general, NMK Provisioning will occur once and the authorization will be permanent. It is possible for the STA to be expelled from an AVLN by providing a different NMK to all stations except the STA being expelled, which would invalidate the STA’s NMK. Subsequent distribution and use of a new NEK(s) then excludes the expelled STA from secure communication within the AVLN.

|  |
| --- |
| Informative Text  It is the implementer’s responsibility to inform the user if a STA cannot connect to an AVLN because of a Security Level mismatch. If the user needs to change the Security Level for a STA to join an AVLN, the user should be so advised. This situation can arise when the user enters the NPW on multiple devices or uses a mix of mechanisms to distribute NMKs. |

There are three methods for NMK Provisioning; each has its own merits and shortcomings, which are described with the UEs that these methods support (refer to Section 13.2). These methods are covered in the following sections:

* NMK Obtained by Direct Entry (Section 7.10.3.3)
* NMK Obtained From AVLN Using DAK Encryption (Section 7.10.3.4)
* NMK Obtained From AVLN Using UKE (Section 7.10.3.5)

Support for all methods is mandatory. Note, however, that the user experience(s) supported by a method will not be available to the user in the absence of a suitable user interface on the STA. Depending upon the method used, the STA might or might not need to engage in AVLN communication prior to obtaining the NMK. The Security Level of the AVLN may also restrict the methods that may be used for NMK distribution.

The choice of the preferred method for obtaining an NMK from among those supported by an implementer is left to the user. NMK Provisioning is closely linked to Security and Privacy. It is expected that the user will determine which method to use based upon the method’s ease of use and the user’s perceived need for security and privacy.

Regardless of how the NMK is obtained, the STA shall store the NMK and its Security Level in its non-volatile memory. If the NID associated with the NMK is not based on the default NID offset, the NID shall also be stored in non-volatile memory. The NMK shall be replaced with a pseudo-random value and associated with its default NID upon reset.

#### Security Level

The Security Level of the NMK defines both the Security Level of the AVLN and the methods that can be used to distribute the NMK. The SL is encoded as part of the NID so that AVLNs at different SLs cannot be considered to be the same AVLN even if the rest of their NID is the same.

Table 7‑13 shows the interpretation of the 2-bit SL value.

Table 7‑13: Security Level Interpretation

|  |  |
| --- | --- |
| Security Level Value | Interpretation |
| 0b00 | Simple Connect. The NMK may have been exchanged using UKE. |
| 0b01 | Secure Security Level. The NMK must not have been exchanged using UKE. |
| 0b10-0b11 | Reserved. |

Table 7‑14: Security Level and NMK Provisioning.

|  |  |  |
| --- | --- | --- |
| Key Status | Description | NMK Provisioning Allowed |
| NMK-SC | Simple Connect, NMK randomly generated by MAC or set by HLE | Direct (from HLE), UKE or DAK |
| NMK-HS | Secure Security Level, NMK hashed from password – Password can be application generated for the first station, but the user would have to record the password or use DAK provisioning for all additional stations. The STA ignores local Add/Join button presses (i.e., Add, Join button presses and UKE distribution require an explicit UI Security Level change, or reset). | Direct (from HLE) or DAK |

|  |
| --- |
| Informative Text  The Security Level applies primarily to Encryption Key Management protocols. It is assumed that all data plane traffic and — with a few exceptions specified herein — all MMEs will be encrypted with the NEK. STAs are not required to accept and should not process any unencrypted data plane traffic and should not be assumed to do so. STAs are not required to accept unencrypted control plane traffic outside of the exceptions specified herein, and should not be assumed to do so. STAs that do accept other unencrypted traffic do so at their own risk. |

##### Secure Security Level

An NMK-HS shall be associated with Secure Security Level. A STA shall only accept a new NMK-HS that is either set by direct entry by the host or sent using the DAK-based distribution method (refer to Section 7.10.3.4). These are termed Secure key distribution methods. It shall not distribute an NMK-HS in its possession except by using the DAK of another STA.

A STA shall neither accept nor distribute an NMK-HS using Unicast Key Exchange (UKE). A STA with an NMK-HS shall ignore local Add/Join button presses.

##### Simple Connect Security Level

An NMK-SC shall be associated with Simple Connect Security Level. A STA shall only accept a new NMK-SC that it generates, or is sent using UKE or its DAK, or that it receives from the HLE using the APCM\_SET\_KEY.REQ primitive. The NMK-SC should be generated randomly by the first STA to become the CCo of a Simple Connect Security Level AVLN. A STA in Simple Connect may use UKE or the DAK of another STA to distribute NMK-SCs.

Just because a STA is in Simple Connect SL does not mean that it will provide its NMK-SC or accept a new NMK-SC from another STA using UKE. The HLE must place the STA in the SC-Join state to accept an NMK-SC from another STA using UKE, and must place the STA in the SC-Add state to provide its NMK-SC to another STA using UKE.

Note: An Unassociated STA may be placed in the SC-Add state so that its NMK-SC will be used for the AVLN. This may be desirable if the STA has the NMK of an existing AVLN whose other STAs are not currently present.

If a STA in the SC-Join state observes another STA in the SC-Join state and it determines it should become the CCo, it shall generate a new NMK-SC, enter the SC-Add state, and provide its new NMK-SC to the other STA using UKE, as specified in Section 7.3.4.4.

Note: The HLE of a device may put the STA into the SC-Join state without user intervention under some circumstances. Conversely, the user may explicitly put any STA (including one already in an AVLN, even the CCo of an AVLN) into SC-Join state, causing it to discard its current NMK (even if it is an NMK-SC) and seek another STA in SC-Add or SC-Join. A STA that was previously in an AVLN must not use the NMK from that AVLN if it has been placed into SC-Join, even it if automatically enters the SC-Add state as described above.

##### Changing the Security Level

A STA that changes Security Level shall discard the previous NMK.

A Station in Secure Security Level may only be changed to Simple Connect SL by receipt of an NMK-SC in the DAK-based distribution method (refer to Section 7.10.3.4), or by the HLE. If it is changed to Simple Connect SL by the HLE (e.g., by resetting the STA), it shall generate a new NMK. In either case, it shall discard the previous NMK-HS.

A STA in Simple Connect SL may only be changed to Secure SL by receipt of an NMK-HS in the DAK-based distribution method (refer to Section 7.10.3.4) or by the HLE. If it is changed to Secure SL by the HLE, the HLE shall supply the new NMK-HS. In either case, the STA shall discard the previous NMK-SC.

#### Preloaded NMK

All devices are required to have an NMK at all times, so at a minimum, each device must have a random NMK. However, a vendor may replace the random NMK with an NMK derived from an NPW of the vendor’s choice. This NMK must be set at the Secure Security Level. Alternatively, a vendor may cause several devices shipped as a set to have the same randomly generated NMK set at the Simple Connect Security Level. These options allow vendors to ship products that connect to each other out of the box for user convenience.

#### Direct Entry of the NMK

Direct entry only implies that the HLE somehow obtains an NMK that it intends for the STA to use, and passes it to the Convergence Layer along with the NID (including the Security Level) using the APCM\_SET\_KEY.REQ primitive or the CM\_SET\_KEY.REQ message over the H1 interface. The CL shall store it for current and future use in joining the specified AVLN. The HLE may obtain the NMK from a set of (NID,NMK) pairs that it has stored, from user entry of an NPW(and generation of the default NID), from key exchange using some higher layer authentication mechanism or by other means (refer to Section 7.10.3.6 and Section 13.2).

As with HomePlug AV, if a HomePlug GREEN PHY Device is provided with a suitable user-interface mechanism, it may permit the user to enter the Network Password directly into the STA, from which the NMK-HS and default NID are generated. The user-interface mechanism may also allow the user to enter a non-default NID. The HLE will obtain the Network Password (NPW) and Secure Security Level from the user, generate the NMK-HS (as described in Section 7.10.7.1) and default NID offset, and pass the NMK-HS with Security Level of the NID set to Secure to the CL via the APCM\_SET\_KEY.REQ primitive or the CM\_SET\_KEY.REQ message. Whether the HLE retains the NPW and NID for future use is an implementation issue.

A STA that changes the NMK in this manner shall leave its current AVLN (if any) and become an Unassociated STA. It may join an existing AVLN or form a new one if it detects another STA with the same NMK-SL (refer to Section 7.3.4.1 and Section 7.3.4.2).

#### Distribution of NMK Using DAK

The user may enter the DPW of a STA into any UIS on the AVLN (or into an Unassociated STA). The HLE will derive the DAK and pass it to the STA, possibly with the NMK and NID to distribute, using the APCM\_AUTHORIZE.REQ primitive. The STA shall use the DAK to perform Payload Encryption of a CM\_SET\_KEY.REQ MME containing a Temporary Encryption Key (TEK) and transmit the encrypted payload to the STA in a CM\_ENCRYPTED\_PAYLOAD.IND MME using multi-network broadcast (refer to Section 5.4.3.1). The protocol ID used shall be PID=0x02 (refer to Section 11.5.2.2) and the PMN shall be 0x01. The UIS shall periodically broadcast this message until it either receives a response or times out.

When the new STA receives and correctly decrypts the payload of this message using its DAK, as indicated in the PEKS, it cannot tell if the message is a replay or not. The new STA shall use the TEK supplied to respond with a CM\_SET\_KEY.CNF MME containing a new nonce in a CM\_ENCRYPTED\_PAYLOAD.IND MME unicast to the sender indicating that it has a matching DAK.

By receiving the TEK-encrypted payload in the response, the UIS knows that the new STA has a matching DAK and has the TEK. The UIS shall wait until both it and the New STA are associated with the same AVLN (known from the TEI Map information) before resuming the DAK-encrypted NMK key distribution protocol.

Note: A STA receiving a DAK-encrypted TEK does not need to disassociate from its current AVLN until it has correctly received the NMK later in the protocol. This is to facilitate its return to the current AVLN should the protocol fail.

Both STAs have obtained CCo capability and AVLN status information from the invitation messages exchanged. If the UIS is part of an AVLN, the new STA (i.e., the one whose DAK was used to encrypt the first MME) shall associate with the AVLN of the UIS. If the UIS determines it will become the CCo (as defined in Section 7.4.1), it shall begin transmitting Central Beacons and the new STA shall associate with the CCo (UIS). If the UIS is an Unassociated STA and the new STA is determined to become the CCo, the new STA shall become the CCo and the UIS shall associate with it.

Once the two STAs are associated with the same AVLN, the key distribution protocol resumes. The UIS shall send the new STA the NMK in a CM\_SET\_KEY.REQ MME as an encrypted payload encrypted with the DAK in a unicastCM\_ENCRYPTED\_PAYLOAD.IND MME, including the nonce received in the previous CM\_SET\_KEY.CNF MME. The new STA shall receive this and obtain the NID (including the Security Level) and NMK, which it now knows to be fresh because of the nonce. The new STA shall confirm its correct receipt of the NMK by using it to encrypt the CM\_SET\_KEY.CNF MME sent as an encrypted payload in a unicast CM\_ENCRYPTED\_PAYLOAD.IND MME in response.

By virtue of receiving the third message in the protocol correctly (encrypted with its DAK and containing a new NMK and NID), the STA will know that the user wants it to join the AVLN from which the NMK was received and shall join it immediately, even if it is already participating in another AVLN. A STA shall set its security mode to the Security Level provided in the NID sent in the **CM\_SET\_KEY.REQ** MME when it receives the NMK via DAK-based encryption and should discard its current NMK.

Note: The STA should not discard its current NMK(s) upon receipt of a DAK-encrypted CM\_SET\_KEY.REQ MME, as this may be a replay. Should the protocol abort before reception of the UIS’ NMK, the new STA shall disassociate from the UIS AVLN, reassociate (if necessary) with its previous AVLN, and resume use of its current NMK.

If the protocol completes successfully, the STA (i.e., the one that does not have the NEK) shall disassociate and associate (if not already associated with the correct AVLN) and authenticate using the NMK, as described in Section 7.3.3.

The entire process for an Unassociated STA forming a new AVLN is shown in Figure 7‑10 (refer to Section 7.3.4.2) and for an AVLN STA adding a new STA, the process is shown in Figure 7‑14 (refer to Section 7.3.5.2). Absent receipt of a response to its initial invitation, the UIS need not send an abort message. A STA that receives an abort message with matching PRN from the other STA shall terminate the protocol on its end unless it has already completed it. In particular, a New STA that correctly receives the NMK may attempt to authenticate with that NMK regardless of whether it receives an abort message later.

#### Distribution of NMK Using Unicast Key Exchange (UKE)

If both the new STA and an existing STA have suitable user interfaces — alphanumeric entry is not a requirement— the new STA may, upon invitation, establish a private channel with the UIS using Unicast Key Exchange (UKE) method, establish a shared TEK and then obtain the NMK via the TEK-secured communications channel. This also requires that the AVLN be in Simple Connect (SC) Security Level, one STA be in the SC-Join state and the other be in the SC-Add state.

Note: The user may explicitly place a STA already in an AVLN in SC-Join state, causing it to leave its current AVLN and seek another to join (even if it later becomes the CCo as provided below). If the UKE protocol aborts, the STA may resume using the existing NMK. Likewise, an Unassociated STA may be placed in the SC-Add state in order that it continue to use its current NMK (e.g., when this is one previously used in an AVLN, but the other STAs are not currently present).

When an Unassociated STA in the SC-Join state observes another STA in the SC-Join state and determines that it should become the CCo, it shall enter the SC-Add state, generate a new random NMK-SC, and become the CCo of a new AVLN, as specified in Section 7.3.4.4. Once the user has directed the STA to join an AVLN (possibly with a single button press at the STA) and has directed the AVLN to adopt a new STA (possibly with a single button press at a designated STA already on the AVLN), the AVLN can give the STA the NMK. Two stages are involved in this process.

1. First, a private channel is established using Unicast Key Exchange (UKE) method.
2. Second, the network STA (the one with the NMK to distribute) sends the new STA the NMK over the private channel.

|  |
| --- |
| Informative Text  There is some chance that a new STA joins the wrong AVLN, or that an AVLN adopts the wrong new STA. In the former case, the new STA may be directed to join a different AVLN. In the latter case, the AVLN may expel the incorrectly adopted new STA. In extreme cases, the user may have to abandon UKE and use one of the other, more secure methods to distribute the NMK to the desired stations. |

Once a shared AES key is established, it shall be used until the protocol is aborted or the TEK is superseded by additional AES encryption keys, subject to limitations (e.g., TEK timeout – refer to Section 7.10.2.6).

Note: Two STAs can use this method to establish a private AES key likely to be known only to themselves and distribute a new NMK-SC, even if they already share a common AES key known to other STAs (e.g., an NMK or an NEK).

To use UKE, the two STAs should receive some positive indication from the user that they are to join the same AVLN at the Simple Connect SL. This requires both STAs to be at the Simple Connect SL, and implies that they both have NMK-SCs. The two STAs first detect each other (using CM\_SC\_JOIN.REQ/CNF MMEs) and either associate with an existing AVLN or form a new AVLN (if both are Unassociated STAs). Once the two STAs are associated with the same AVLN (known from the TEI Map updates from the CCo), they should establish channel adapted tone maps (refer to Section 5.2.6) and then start the UKE protocol.

UKE requires that the STAs use unicast communications to derive the common TEK. Each STA contributes a secret Hash Key during an exchange of unencrypted, unicast CM\_GET\_KEY.REQ/CNF MMEs with Requested Key Type=HashKey with the PID=0x03, PMN=0x01 and 0x02 respectively. The STA in SC-Add sends the CM\_GET\_KEY.REQ MME, and the STA in SC-Join responds with the CM\_GET\_KEY.CNF MME containing the PEKS chosen by the STA in SC-Join. The two Hash Keys are concatenated and hashed to produce the shared, unauthenticated TEK as defined in Section 7.10.7.2. The secrets (Hash Keys) exchanged by the STAs shall be pseudorandom strings of length 384 octets.

The first two messages in UKE are sent unencrypted. These messages shall be unicast and should be sent using channel-adapted tone maps. These messages should not be retransmitted (that is, if the PB containing a Hash Key is not properly received, a new Hash Key should be generated before retransmitting; alternatively, if either of the two initial messages fails to deliver on the first attempt, then the protocol may be aborted and resumed with a different PRN).

Once both STAs have the TEK derived from the Hash Keys, the STA in SC-Add shall send the STA in SC-Join the NMK and the NID (which shall indicate SC Security Level) in a CM\_SET\_KEY.REQ MME, containing the nonce received in the previous message, as an encrypted payload encrypted with the TEK with PID=0x03 and PMN=0x03 in a unicast CM\_ENCRYPTED\_PAYLOAD.IND MME. If the NID received with the NMK in the CM\_SET\_KEY.REQ MME does not indicate that the SL is SC, the receiver shall abort the protocol by sending an unencrypted CM\_ENCRYPTED\_PAYLOAD.RSP with Result set to Abort. Otherwise, the STA in SC\_Join shall respond with a CM\_SET\_KEY.CNF MME, containing the nonce just received, encrypted with the NMK with PID=0x03 and PMN=0xFF in a unicast CM\_ENCRYPTED\_PAYLOAD.IND MME. This verifies that the NMK was received and concludes the UKE portion of the process.

The new STA shall then go on to authenticate using the NMK to obtain the NEK from the CCo. The entire process for Unassociated STAs is described in Section 7.3.4.3 and Section 7.3.4.4. For one Unassociated STA and one STA in an AVLN, it is described in Section 7.3.5.3.

#### Distribution of NMK Using Other Key Management Protocols

As an alternative to the methods described in this specification for Authorization, and as with HomePlug AV, GREEN PHY fully supports the use of HLE standards such as 802.1x, EAP, and SNMP.

All messages sent by the HLE for key distribution shall be transmitted as CM\_ENCRYPTED\_PAYLOAD.IND MMEs with PID=0x04. The 16-octet field used for the IV in encrypted messages shall be used for the UUID (see references [9], [10]) of the protocol sending the messages. This field shall remain constant for the duration of the protocol run. The UUID field may be used by the HLE to distinguish between application protocols using this lower level transport mechanism.

The entire “encrypted” payload portion of the message shall not be interpreted by the sending or receiving STA, and is thus available to the HLE to carry its protocol messages. In particular, the STA shall not check the PID, PRN, or PMN fields across the “encrypted” and “unencrypted” parts of the MME. References to PID, PRN, and PMN in this section only concern the “unencrypted” fields of the MME.

The STA shall use the ODA and its knowledge of TEIs to determine whether or not to broadcast a message it receives from the HLE with PID=0x04. The OSA and/or ODA in these messages might not belong to a STA. All messages in the same protocol run shall use the same PRN (randomly selected by the HLE application at the start of the protocol run), and the first message shall have PMN=0x01. Subsequent messages in the protocol run shall increment the PMN until the last message, which shall have PMN=0xFF. A STA that receives a valid CM\_ENCRYPTED\_PAYLOAD.IND MME with PID=0x04 from the powerline medium shall pass the entire CM\_ENCRYPTED\_PAYLOAD.IND MME to its HLE for processing.

If a STA receives (from its HLE or from the powerline medium in response to a message it sent) a valid CM\_ENCRYPTED\_PAYLOAD.IND MME with PID=0x04 and PMN=0x02, it may choose to associate with the same AVLN of the other STA or form a new AVLN as described in Section 7.3.4 and Section 7.4.1. In particular, this is desirable when one or both of the STAs are unassociated. Sufficient information is present from the messages (the SNID and TEI) that the STAs can communicate using unicast without association, but communication may be more reliable if they are associated with the same AVLN.

The HLEs that generate and process these messages are responsible for properly setting the PRN and incrementing the PMN. A STA that transmits one of these messages using MNBC may repeat that message unchanged multiple times for reliability purposes. A STA that receives duplicate messages (i.e., with the same PID, PRN, and PMN) may discard the duplicate copies silently. The HLE must be able to manage reception of duplicate messages. It is up to the HLE to generate the response to any message received.

When the HLE sends the last message in the protocol, it shall use PMN=0xFF, regardless of the success or failure of the protocol. If the protocol succeeds in providing an NMK to the new STA, the HLE for the STA receiving the NMK shall use the APCM\_SET\_KEY.REQ primitive to set the NMK and NID (including the SL) of its STA, as described in Section 7.10.3.3. When both STAs have the NMK, both should also have the same NID, and the STAs should form an AVLN as described in Section 7.3.4.1 or expand an existing AVLN as described in Section 7.3.5.1.

#### Changing the NMK

If for some reason (e.g., a rogue STA has managed to get into the AVLN) the user suspects a hostile environment, the user may force a change to the NMK. Essentially, this redefines the AVLN and the NID may change along with the NMK. The user shall change the NMK (and typically, the NID) for each STA that held the compromised NMK individually, using direct entry, the DAK if it is available (Section 7.10.3.4), or UKE (Section 7.10.3.5). In the latter case, only an NMK-SC may be distributed.

Changing the NMK may change the NID, which is expected to be a disruptive event. That is, while the STAs in the AVLN are receiving the new NMK, leaving the old AVLN with the compromised NMK and NEK, and joining the new AVLN, they might not be able to sustain QoS levels for ongoing connections. In fact, they may have to terminate existing connections and establish new connections in the new AVLN. The degree to which the STA and the HLE shield applications and the user from these interruptions of services is both implementation and vendor dependent.

### NEK Provisioning

The NEK is always provided by the CCo, and always encrypted by the NMK. The NEK shall not be set using any other encryption key or by an unencrypted MME. The NEK is never passed over the H1 interface (neither to the host from the STA nor from the host to the STA). NEK provisioning can be initiated in one of two ways:

* A new STA joins an AVLN and is given the NEK (with PID=0x00, Section 7.10.4.1)
* The CCo determines to change the NEK for part or all of the AVLN ((with PID=0x01, Section 7.10.4.2)

#### Provision NEK for new STA

NEK provisioning for a new STA is called Authentication and is described in Section 7.3.3.

#### Provision NEK for Part or All of the AVLN

The CCo may update the NEK at any time, and is required to do so as specified in Section 7.10.2.5. It does so using the procedure shown in Figure 7‑23. The PID shall be set to 0x01 for all messages in this protocol. The CCo will provision each STA individually to get a positive ACK from each STA. After all active STAs have been provisioned with the new NEK and EKS, the CCo shall cause the change to be effective by putting an Encryption Key Change BENTRY in the Beacon for the appropriate number of Beacon Periods. If a STA that belongs to an AVLN detects an Encryption Change BENTRY in that AVLN’s Beacon and the STA does not have the new NEK, it shall request the new NEK from the CCo using the authentication procedure specified in Section 7.3.3. The STA shall not use the new NEK until indicated by the CCo in the Encryption Key Change BENTRY in the Central or Proxy Beacon.

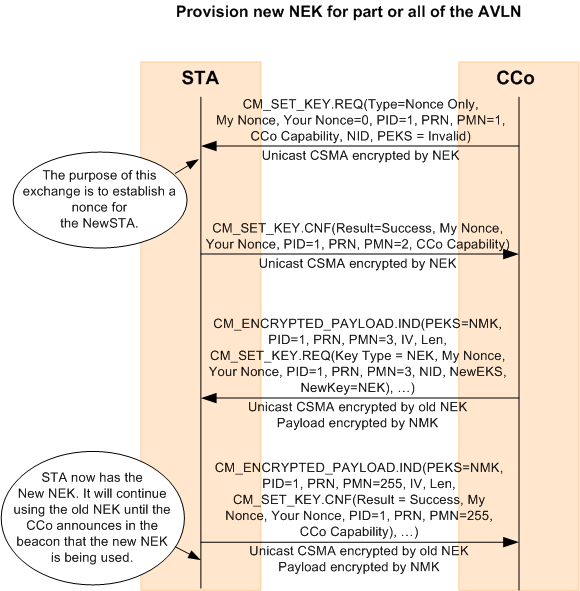


Figure 7‑23: Provision NEK for Part or All of the AVLN

### Encryption Key Uses and Protocol Failures

Encryption keys are used for specific purposes, and improper use of these keys is one of several forms of protocol failure.

The NEK is used exclusively for encrypting PBBs (i.e., at the PHY level); that is, only PBBs are encrypted with the NEK, and nothing else may be encrypted with the NEK. All data plane traffic and most control traffic will be between authenticated STAs in an AVLN, and those PBBs will be encrypted using the current NEK for that AVLN. A few messages may be sent unencrypted at the PHY level. For more information, refer to Table 11‑5 in Section 11.1.8.

A failure to decrypt a PBB using the NEK cannot be detected by the PHY; it can only use the PBCS to check that the encrypted PBB was demodulated properly, then decrypt whatever it receives. Proper decryption of the PBB can only be checked by the ICV at the MAC Frame level. Incorrect MAC Frames shall be discarded silently by the receiver, and it is up to the HLE to recover from this loss.

All other keys (DAK, NMK, TEK) are used to encrypt the payload of CM\_ENCRYPTED\_PAYLOAD.IND MMEs, which are usually sent unencrypted at the PHY level. (Figure 7‑23 is an anticipated exception.) How a STA responds to a CM\_ENCRYPTED\_PAYLOAD.IND MME depends on how it was sent as well as the ability of the STA to decrypt it and to validate it.

When the PID is 0x04, only the unencrypted fields are checked, and the entire CM\_ENCRYPTED\_PAYLOAD.IND MME shall be passed from the HLE to the PHY or from the PHY to the HLE without interpreting the “encrypted” (or “uninterpreted”) portion (see Figure 7‑24 and Figure 7‑25).

A STA that receives a CM\_ENCRYPTED\_PAYLOAD.IND MME shall attempt to decrypt the payload using the key indicated in the PEKS, if it is valid and the PID is not 0x04. Proper decryption is determined by using the last octet to ascertain the length of the Random Filler, then decoding the encapsulated MME for the MME length, and then checking the CRC as well as the Protocol Identifier (PID), PRN, and PMN (refer to Section 7.10.8). These last three should match their unencrypted values exactly. Lastly, if a TEK is used, it must not have expired (refer to Section 7.10.2.6). Failure to decrypt a CM\_ENCRYPTED\_PAYLOAD.IND MME properly shall cause the recipient to respond with a CM\_ENCRYPTED\_PAYLOAD.RSP MME indicating failure, unless the CM\_ENCRYPTED\_PAYLOAD.IND MME was sent with a broadcast address ODA. Note that a broadcast DTEI may be necessary even when the transmission is unicast in the ODA, so the DTEI cannot be used to determine whether a response is needed.

If the CM\_ENCRYPTED\_PAYLOAD.IND MME is decrypted correctly, other checks must be performed at the level of the protocol. At a minimum:

* The PID, PRN, and PMN must be appropriate.
* The key indicated in the PEKS and the encapsulated MME type must be appropriate for the PID and PMN.
* The nonces, if any, in the encapsulated MME must be valid.

The PID must be between 0x00 and 0x04. All protocol runs shall start with the first message having PMN=0x01 and end with the last message having PMN=0xFF. If the PMN is greater than 0x01, the PRN must be already known and the Your Nonce field in the current message must match the My Nonce field in the previous message in that protocol run; otherwise, the protocol run is aborted. The PMN must be in the appropriate range for the PID, and must be the next PMN expected in that run of the protocol (the special PMN value of 0xFF is the sole exception and is used to indicate the last message). Once a STA sends or receives a message with PMN=0xFF, it shall ignore all further messages with that PID and PRN. It is recommended that at least 16 of the most recently used sets of {PID,PRN} pairs are stored. Except for PID=0x04, the key indicated by the PEKS must be of the correct type for that PID and PMN, and the MME must be of the correct type for that PID and PMN

A CM\_ENCRYPTED\_PAYLOAD.IND MME with an incorrect PID, an unknown PRN with PMN greater than 0x01, an out-of-sequence PMN, or an encapsulated MME of the wrong type for the PID and PMN shall be ignored. An exception for duplicate MMEs is noted below. A correct MME type that has other defects may be ignored or, if it is a request, may cause a corresponding confirm to be sent indicating failure. When a confirm MME indicating failure is sent, it shall be sent as the body of an encrypted payload MME that does not encrypt its payload (indicated by PEKS=0xF; refer to Section 11.5.2.1).

The higher protocol must be able to withstand the loss of a message in the protocol sequence. This is done by setting a timer for a time by which a response is expected, and retransmitting the last message if no response has been received by that time. Since these protocol messages are exchanged using CSMA, and may also have to traverse several layers of processing by the recipient, the timer should be set to account for these latencies. When a duplicate of the previous received message in a protocol run is received, then a duplicate of the message sent in response to it (if any) must be sent.

### AES Encryption Algorithm and Mode

AES Encryption may be performed either at the PHY Block level, as described in Section 5.4.7.1, and/or within a CM\_ENCRYPTED\_PAYLOAD.IND MME. Both modes are mandatory.

#### PHY Block-Level Encryption

As with HomePlug AV, GREEN PHY uses the 128-bit Advanced Encryption Standard (AES) algorithm in CBC Mode. The encryption method is described in Section 5.4.7.1.

The 4-bit Encryption Key Select (EKS) in the Frame Control is an index of the encryption key used for encrypting the PBBs. EKS values are defined in Section 4.4.1.5.2.8. The values instruct the STA about how to encrypt/decrypt the message, including use of no encryption at all.

#### Payload-Level Encryption

Payload level encryption shall be identical to PHY Level encryption with the following exception:

* Only a portion of the payload is encrypted.
* The PEKS (called the PEKS to distinguish it from the EKS) identifying the Encryption Key and, if AES encryption is used, the Initialization Vector are carried in the unencrypted portion of the payload.
* The bit ordering is different.

The payload shall be padded to a 128-bit boundary as needed. The padding shall be random bits.

##### CM\_ENCRYPTED\_PAYLOAD.IND Message Encryption

If encryption is specified in the CM\_ENCRYPTED\_PAYLOAD.IND message, the encrypted portion of the payload shall be encrypted using 128-bit AES-algorithm in CBC Mode. Refer to Section 5.4.7.1.

Section 13.5 provides an example that shows payload encryption. It should eliminate any confusion about bit ordering, initialization vectors, etc.

Note: The bit ordering for the Encrypted Payload and corresponding Initialization Vector and Encryption Key is different than the bit ordering defined in Section 5.4.7 for PHY Block Body.

###### Encrypted Payload Encryption Bit Order

The MSB of the first octet of the Encrypted Payload shall correspond to bit number 0 of the AES encoder defined in Section 3.1 of reference [2] Federal Information Processing Standards Publication 197.

###### Encrypted Payload Initialization Vector Bit Order

The MSB of the first octet of the IV shall correspond to bit number 0 of the AES encoder defined in Section 3.1 of reference [2] Federal Information Processing Standards Publication 197.

When PID=0x04, this field shall be used as a Unique Universal Identifier (UUID, see references [9], [10]). The MSB of the first octet of this field shall correspond to the MSB of octet 0 of the UUID.

###### Encrypted Payload Encryption Key Bit Order

The MSB of the first octet of the Encryption Key shall correspond to bit number 0 of the AES encoder defined in Section 3.1 of reference [2] Federal Information Processing Standards Publication 197.

### Generation of AES Encryption Keys

#### Generation from Passwords

The HP-AV privacy function may at times require the generation of an encryption key from a password. In all such cases, the mechanism for creating a key from a password shall be the PBKDF1 function, as shown in the PKCS #5 v2.0 standard, Password-based Cryptography Standard, using SHA-256 as the underlying hash algorithm. The iteration count used to calculate the key shall be 1000. The salt value shall be 0x0885 6DAF 7CF5 8185 for DAKs and 0x0885 6DAF 7CF5 8186 for NMK-HSs. After the 1000th iteration, the leftmost 16 octets of the SHA-256 output (as described in FIPS-180-2 change notice) shall be used as the AES encryption key. The first octet of the output corresponds to octet 0 of the AES encryption key. The bit ordering of the AES encryption key within an octet is dependent on where it is used. Refer to Section 7.10.6.2.1.3 and Section 5.4.5.4.

HP-AV passwords (DPW and NPW) shall be limited to strings of ASCII characters chosen from the range ASCII[32] to ASCII[127]. The length of a DPW shall be between 16 and 64 characters inclusive. The length of a NPW shall be between 8 and 64 characters inclusive. Users shall be provided a warning by the user interface that the entered NPW may not be secure when the user enters a NPW less than 24 characters in length. Passwords are not sent to the Convergence Layer and it is recommended that they not be retained. DPWs should be pseudo-randomly generated and recommended to be a minimum of 20 characters long. It is recommended that pseudo-random NPWs be a minimum of 16 characters long and user-selected NPWs a minimum of 24 characters long.

#### Automatic Generation of AES Keys

All STAs shall be able to generate Hash Keys (used in UKE) and AES encryption keys (e.g., NEK, NMK, and TEK) directly. The generation of these is implementation dependent. Implementers shall ensure that key and secret generation are based on a good random number generation algorithm. The method used shall be at least as good as the following baseline method.

In the baseline method, a station shall use a keyed cryptographic hash algorithm (such as the one used to generate keys from passwords) with a key distinct to the station and not the same as its DAK. It shall maintain an internal counter that is never exposed. This counter value shall be hashed to obtain a pseudorandom number, and the counter shall be incremented each time a pseudorandom number is generated. RFC 4086 [5] and the book on Applied Cryptography by Schneier [6] may be used for guidance.

The TEK for the UKE method is generated from the Hash Keys using SHA-256 as the underlying hash algorithm and truncating the output. The AES key shall be the leftmost 128 bits of output (as described in FIPS-180-2 change notice), and the input shall be the first Hash Key (from the CM\_GET\_KEY.REQ MME) with the MSB of the least-significant octet of the first Hash Key as the leftmost bit of input to the SHA-256 function, concatenated with the second Hash Key (from the CM\_GET\_KEY.CNF MME) with LSB of the most-significant octet of the second Hash Key as the rightmost bit of input to the SHA-256 function. Also refer to Section 7.10.3.5.

The iteration count used to calculate the key shall be 0 (i.e., hash once). There shall be no salt. After one iteration, the leftmost 16 octets of the SHA-256 output (as described in FIPS-180-2 change notice) shall be used as the AES encryption key. The first octet of the output corresponds to octet 0 of the AES encryption key. The bit ordering of the AES encryption key within an octet is dependent on where it is used. Refer to Section 7.10.6.2.1.3 .

#### Generation of Nonces

Generation of nonces shall be done in a way so that the repetition of a nonce is very unlikely and that the nonces are not predictable. A method at least as good as the one described in Section 7.10.7.2 for automatic generation of AES keys shall be used. A possible way to generate a nonce is to use a counter that is passed through a keyed secure hash function. The counter is incremented each time a nonce is generated and the key for the hash is kept private to the station. The same key and counter may be used for generating nonces and random numbers for different purposes, by using additional information specific to the purpose that is also passed through the hash function.

### Encrypted Payload Message

The CM\_ENCRYPTED\_PAYLOAD.IND management message may be exchanged over the medium unencrypted. As such, they are most useful for the processes that establish or distribute keys. They may also be used by higher layer protocols to perform discovery or key distribution (with PID=0x04 -refer to Section 7.10.3.6).

As with HomePlug AV, within HomePlug GREEN PHY, these are used for five distinct purposes:

* Distribution of the NEK using the NMK based on a request from a STA seeking to authenticate (PID=0x00 – refer to Section 7.10.4).
* Distribution of the NEK using the NMK initiated by the CCo when rotating the NEK (PID= 0x01 – refer to Section 7.10.4)
* Distribution of the NMK using the DAK (PID=0x02 – refer to Section 7.10.3.4).
* Distribution of the NMK using UKE (PID=0x03 – refer to Section 7.10.3.5).
* Execution of higher layer protocols (PID=0x04 – refer to Section 7.10.3.6).

Each of these is given its own PID for use in the PID fields (one unencrypted, one encrypted, except when PID=**0x04**) of the CM\_ENCRYPTED\_PAYLOAD.IND message. The PID value of 0x04 is set aside for higher layer protocols, so that the MAC knows to pass these messages across the H1 interface uninterpreted. For interpretation of PID values, refer to Section 11.5.2.3 and Table 11‑84.

Figure 7‑24 shows an entire Encrypted Payload Message when PID is between **0x00** and **0x03**, including all of its components contributed by MAC Framing and other processes.



Figure 7‑24: Encrypted Payload Message when PID is between 0x00 and 0x03

The PEKS indicates the AES encryption key used to encrypt the payload (refer to Section 11.5.2.1 and Table 11‑82). When used with a higher layer protocol or when a confirm MME indicating failure is sent, a PEKS of No Key (0x0F), may be used to indicate that the payload is not encrypted.

When PID=**0x04**, the 16-octet IV field shall be used for a UUID (see references [9], [10]), and the portion of the CM\_ENCRYPTED\_PAYLOAD.IND MME from Random Filler to RF Length shall not be interpreted by the STA (see Figure 7‑25). This entire portion shall instead be HLE protocol payload (i.e., the eight fields defined for the encrypted portion are instead defined by the higher layer protocol).



Figure 7‑25: Encrypted Payload Message when PID = 0x04

The limitation of MNBC transmissions to a single PB constrains the length of the CM\_ENCRYPTED\_PAYLOAD.IND MME to be no more than 502 octets, and the HLE Payload to be no more than 460 octets, unless the CM\_ENCRYPTED\_PAYLOAD.IND MME is fragmented. Fragmentation at the MAC level is discouraged, however, due to the unreliability of MNBC, and if necessary, should be done by the HLE.

### User Interface Station (UIS)

A UIS is any station that is capable of providing a suitable mechanism (preferably keyboard and display) to enable user interaction with the AVLN and capable of supporting network control and security functions for the AVLN. It might not be physically attached to the AVLN (i.e., it might connect through a bridging STA).

There is no limit to the number of UISs that may exist in the AVLN.

### Resisting Common Security Attacks

#### Man-in-the-Middle (MITM)

A man in the middle attack (MITM) is an attack in which an attacker is able to read, insert and modify at will, messages between two parties without either party knowing that the link between them has been compromised. The attacker must be able to observe and intercept messages going between the two victims. In the absence of countermeasures, Key Exchange protocols are vulnerable to this kind of attack.

As with HomePlug AV, GREEN PHY only admits this attack when UKE is in use. If threat analysis shows this type of attack is likely, then Secure Security Level should be used.

#### Repetition (Replay) Attacks

Repetition attacks involve the attacker capturing a message or sequence of messages and playing it (them) back at a later time. This attack can be thwarted by the use of Nonces. A nonce is a parameter that varies with time. It can be a timestamp or some other special value that changes with time. The use of nonces reduces the likelihood of replay attacks.

When called for in the specification, a nonce shall be created and included (as My Nonce) in an MME from one STA to another. The receiving STA shall take this value and embed it (as Your Nonce) in the MME it sends in response to the original MME. When the STA that created the nonce receives the reply, it knows ⎯ with a high degree of certainty ⎯ that the reply is fresh. A STA sending a nonce in one message is required to verify that the same nonce is returned in the next message of the protocol, if any. A nonce mismatch shall cause the protocol to abort.

Once a particular value has been used as a nonce, it shall not be reused, except within the same protocol run (refer to Section 7.10.2.7). The next time that STA transmits a message that includes a nonce, it shall generate another one. The method used to generate nonces is an implementation decision. Although there is no explicit requirement that a value provided as a nonce not occur again within a specific time interval, implementers shall endeavor to generate nonces using a method that ensures nonce values do not repeat often and that it is not easy to predict the nonce’s value. Refer to Section 7.10.7.3.

### Discussion of Security Mechanisms (Informative)

The highly structured nature of the contents being encrypted makes these messages susceptible to recognizable plaintext attack. That is, an attacker can tell, with a good deal of certainty, when the right key has been guessed from the structure of the decrypted data. However, the key space is 128 bits and flat, and the keys used for data encryption are randomly selected by the CCo. Therefore, finding the right NEK by brute force is a risk we accept, given the threat model (refer to Section 2.4.2.

Keys based on pass phrases are another issue and come in two varieties:

* Device Passwords (DPWs)
* Network Passwords (NPWs)

The specification requires these to be at least 12 characters long, selected from a range of ASCII 32 – 127 (96 characters). The DPW must be randomly chosen by the OEM, so its entropy should be at least 6 bits per character (log2(96) > 6.5). With a 20-symbol DPW, this should translate to at least 120 bits of search space. Only if an OEM uses short or non-random DPWs could this be a problem. Even the shortest (12 symbols) random DPWs should be equivalent to 12x6.5 = 78 bits, which can be brute-forced, but not without great effort. The threat model is for typical individuals, not for well-heeled governments or corporations. If the data requires protection against greater threats than this, higher level security should be applied. The recommended minimum length of 15 characters for DPWs and machine generated NPWs is equivalent to 97.5 bits.

NMKs are used infrequently, and are also 128 bits long. Brute forcing these should be no easier than brute-forcing the NEKs, with the exception of NMKs based on user-chosen NPWs. These NPWs must be at least 12 characters long, and are recommended to be at least 24 characters long. With natural language entropy of 2-3 bits per character, these lengths translate to 48-72 bits of entropy in the key. NPWs of length up to 64 characters are supported, providing 128 bits of entropy in the key for users concerned about weak NPWs.

The use of distinct salts for generation of DAK and NMK-HS means that an attacker cannot use a table generated for one of these types of keys to attack another type of key.

Initialization vectors are either random (in encrypted payload MMEs) or are derived from the frame control, the PB header, and the position of the PB within the MPDU. While it is possible for an IV to repeat in less than the hour maximum between NEK changes, the primary threat in CBC mode is recognized ciphertext. The contents are likely to be different in high-speed multimedia streams, and for MMEs, a random confounder is included to thwart this attack.

TEKs derived through the UKE process are based on secret Hash Keys provided by the initiator and the respondent. These are each 384 octets long, and while they are sent in the clear, they should be sent in unicast messages, which should use channel-adapted tone maps. To intercept one of these correctly, the eavesdropper would at the least have to use non-standard equipment capable of receiving unicast MPDUs not destined for it. If the MPDUs are sent using channel adapted tone maps, the eavesdropper would have to know the tone map used, and have a signal to noise ratio at least as good as the receiver on the majority of the carriers. Otherwise, there are likely to be too many bit errors for the FEC to correct. The systematic bits may be recovered with confidence level, however, leaving the attacker with reasonable certainty in some of the received bits. For the hash function to produce the right key, however, all of the input bits must be correct, so the attacker must guess values for all uncertain bits. The hash key length provides 6144 bits total for hashing, so if the attacker is uncertain about only 1% of these, he still has to search a 61-bit space. To intercept these unicast messages at all, non-standard equipment must be used, and due to attenuation and frequency-selective fading, the attacker would have to have access to the power lines comparable to the user himself.

## Network Power Management

If the CCo of an AVLN does not detect a valid AV Frame Control (excluding Beacon Frame Controls) from STAs that are associated with its AVLN for at least 30 seconds, the CCo shall enter Network Power Saving Mode.

In Network Power Saving Mode, the CCo shall:

* Set the NPSM bit in the Beacon.
* Specify a schedule consisting of only the Beacon Region, a CSMA Region of length equal to minCSMARegion, and a Reserved Region of sufficient size to support any necessary Discover Beacons and Proxy Beacons. The remainder is specified as Stayout region.

Upon detection of a valid AV Frame Control (excluding Beacon Frame Controls) from a STA associated with its AVLN, the CCo shall exit NPSM and reset the NPSM bit in the Beacon. It may then update the schedule to make additional time available in the CSMA Region.

Since a HSTA cannot rely on the CCo detecting its transmissions to exit Network Power Saving Mode, the PCo shall monitor the CSMA Region while in Network Power Saving Mode, and shall send a CP\_PROXY\_WAKE.REQ to the CCo if it detects a valid AV Frame Control.