

NOTE 3 La condition suspecte se produit uniquement si le nœud modérateur et au moins un autre nœud n'ont pas reçu l'un des trois paquets de demande de service de l'attribut de réseau de définition de diffusion. Cela se produit uniquement en cas d'événement bruyant sérieux (durant au moins 3 NUT).

Figure 20 – Traitement de la modification de réseau synchronisée

7.4 Objet de planification

7.4.1 Vue d'ensemble

L'objet de planification doit exister dans les dispositifs à l'origine de la connexion (CO). L'objet de planification doit être utilisé par un logiciel de planification de liaison (LSS) pour

- a) lire les informations de connexion planifiée;
- b) écrire les informations de planification;
- c) offrir un mot de passe (clé logicielle) à l'auteur de la connexion, qui doit être utilisé pour authentifier l'accès du dispositif CO à une liaison particulière.

Une session LSS avec l'objet de planification doit permettre de planifier une liaison particulière. Si un dispositif CO détient des connexions sur plusieurs liaisons, plusieurs sessions LSS doivent être nécessaires pour planifier la totalité des connexions. Les résultats de la session de planification doivent être sauvegardés par le dispositif CO (y compris le mot de passe CO calculé par le LSS).

Étant donné que le LSS peut être intégré au logiciel de programmation d'un dispositif CO, l'objet de planification doit fournir les commandes de prise en charge d'un logiciel de programmation intégré ou séparé. De manière spécifique, lorsque le LSS écrit des données de planification dans l'objet de planification, il peut le faire sous condition ou forcer l'écriture. Les données de planification doivent être écrites sous condition si le logiciel de programmation et le LSS n'ont pas été intégrés, et l'objet de planification doit uniquement utiliser les données de planification fournies si un autre logiciel de programmation n'a pas modifié les informations de connexion pendant la session LSS. Une écriture forcée peut être utilisée avec un LSS/logiciel de programmation intégré si l'outil intégré peut garantir qu'aucun autre logiciel de programmation n'a modifié les informations de connexion depuis la dernière lecture pendant la session de planification en cours.

Toutes les communications adressées à l'objet de planification doivent utiliser le service fixe UCMM (Unconnected Message Manager) ou un service de connexion à balise générique actif vers l'objet Routeur de message.

7.4.2 Attributs de classe

L'objet de planification doit prendre en charge les attributs de classe spécifiés dans le Tableau 34. Le service Get_Attribute_All doit renvoyer les attributs de niveau de classe dans l'ordre numérique croissant.

Tableau 34 – Attributs de classe de l'objet de planification

ID attribut	Nécessité dans la mise en œuvre	Règle d'accès	Nom	Type de données	Description d'attribut
0x01	Requise	Get	Revision	UINT	Première révision, valeur = 1
0x02	Requise	Get	MaxInstances	UDINT	Nombre maximal d'objets de planification pris en charge
0x03	Requise	Get	NumInstances	UDINT	Nombre d'objets de planification instanciés

7.4.3 Attributs d'instance

L'objet de planification doit prendre en charge les attributs d'instance spécifiés dans le Tableau 35. Le service Get_Attribute_All doit renvoyer les attributs de niveau d'attribut dans l'ordre numérique croissant.

Tableau 35 – Attributs d'instance de l'objet de planification

ID attribut	Nécessité dans la mise en œuvre	Règle d'accès	Nom	Type de données	Description d'attribut
0x01	Requise	Get/Set	Route	STRUCT de taille variable	Route entre le CO et la liaison en cours de planification
			NumSegments	UINT	Nombre de segments du chemin
			Segments	MATRICE d'UINT	Matrice de segments
0x02	Requise	Get/Set	TimeOut	UDINT	Délai d'attente du chien de garde (µs) (valeur par défaut = 60 s)
0x03	Facultatif	Get/Set	État du contrôleur	UINT	bit 0 (exécution/programme) = 0, peut être programmé = 1, contrôle le bit d'E-S bit 1 (distant/local) = 0, le bit 0 ne doit pas être modifié à distance = 1, le bit 0 peut être modifié à distance bits 2 à 15 réservés et doivent être nuls

7.4.4 Services communs

7.4.4.1 Généralités

Outre les services décrits dans le présent paragraphe, l'objet de planification doit prendre en charge le Get_Attribute_All standard adressé à l'instance de classe (instance zéro). La mise en œuvre d'un objet de planification particulier peut éventuellement prendre en charge d'autres attributs de service standard.

L'objet de planification doit prendre en charge les services communs spécifiés dans le Tableau 36.

Tableau 36 – Services communs de l'objet de planification

Code de service	Nécessité dans la mise en œuvre		Nom du service	Description du service
	Classe	Instance		
0x01	Requise	Facultatif	Get_Attribute_All	Renvoie une liste prédéfinie des attributs de cet objet. Modifie tous les attributs définissables
0x02	N/A	Facultatif	Set_Attribute_All	Modifie tous les attributs définissables
0x03	Facultatif	Facultatif	Get_Attribute_List	Renvoie une liste d'attributs spécifiés
0x04	N/A	Facultatif	Set_Attribute_List	Modifie une liste d'attributs définissables
0x08	Requise	Facultatif	Create	Permet d'instancier un objet de planification
0x09	N/A	Requise	Delete	Permet de mettre fin à une session de planification LSS
0x0E	Facultatif	Facultatif	Get_Attribute_Single	Renvoie le contenu de l'attribut spécifié
0x10	N/A	Facultatif	Set_Attribute_Single	Modifie un seul attribut

Une mise en œuvre peut prendre en charge la définition de l'attribut au niveau de l'instance de la route, mais si ce n'est pas le cas, il ne doit pas être possible de distinguer le comportement interne d'un service delete suivi d'un service create contenant la nouvelle route.

7.4.4.2 Create

Le service Create doit être utilisé pour instancier un objet de planification sur le CO. Dans le message Create, le client (LSS) doit inclure une route pointant du CO vers la liaison à planifier. La route vers la liaison doit être composée de segments de port et ne doit pas contenir de segments de réseau. L'objet de planification doit répondre par un numéro d'instance auquel toutes les communications ultérieures doivent être adressées pendant cette session de planification, et par les mots de passe et chemin CO déjà enregistrés.

La nouvelle instance doit se supprimer elle-même si les communications avec son client sont coupées. Les communications avec une instance doivent être surveillées à l'aide d'un temporisateur dont la valeur de délai d'attente (par défaut = 60 s) peut être attribuée par l'un des services d'attribut de définition facultatifs. Le temporisateur d'une instance doit être réinitialisé lorsqu'il reçoit un message. Si le temporisateur expire, l'instance doit être supprimée en annulant toutes les modifications en attente. La valeur de délai d'attente doit être un entier de 32 bits exprimé en µs.

Si le service Create s'adresse au niveau de classe (instance zéro), le numéro d'instance de la nouvelle instance créée doit être choisi automatiquement par l'objet de planification. Pour les besoins du débogage, l'objet de planification peut éventuellement permettre au service Create d'être adressé à un numéro d'instance particulier.

L'objet de planification doit prendre en charge au moins une instance. Si l'objet de planification admet l'existence de plusieurs instances, il doit s'assurer de l'intégrité de chacune d'elles. La valeur de l'attribut d'instance 0x01 doit être fournie avec le service Create.

```

class Scheduling_Create_Request: public MR_Request
{
    UINT    number_of_attributes;    // set to 1
    InstanceAttribute1 attribute1;
}

class Scheduling_Create_Response: public MR_Response
{
    UDINT    instance_number;
    UDINT    cop;                // Connection Originator Password (COP)
    UINT     path_size;          // in words, range 0 to 255
    UINT     path[];             // array of port segments from the scheduled link
                                // to the connection originator
                                // no key segments, no network segments
                                // include the MAC ID of the link hop
};

```

La réponse doit contenir l'un des codes d'état figurant dans le Tableau 37.

Tableau 37 – Descriptions de l'erreur d'état de Create

État général	État étendu	Description de l'erreur
0x02	0x0001	Ressources insuffisantes — le nombre maximal d'instances d'objet de planification existe déjà
	0x0002	Ressources insuffisantes — mémoire insuffisante sur le dispositif CO
0x04	N/A	Erreur de syntaxe du chemin
0x08	N/A	Service non mis en œuvre (les instances non nulles étant facultatives)
0x10	N/A	Impossible d'ouvrir une autre instance suite à des conflits internes
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée
0x0D	N/A	L'objet existe déjà (lorsqu'une instance non nulle est spécifiée)
0x1C	N/A	Pénurie de listes d'attributs — il manque l'attribut requis
0xDO	N/A	Aucune connexion planifiée sur cette liaison

7.4.4.3 Delete

Le service Delete doit être utilisé pour mettre fin à une session de planification du LSS. Il permet de désallouer toutes les ressources d'une instance spécifiée de l'objet de planification. Toutes les modifications en attente qui n'ont pas encore été engagées doivent être perdues. Les connexions rompues par des commandes antérieures dans la session LSS ne doivent pas nécessairement être restaurées. Le service Delete ne doit pas être pris en charge au niveau de la classe. La réponse de ce service doit être spécifiée conformément au Tableau 38.

Tableau 38 – Description d'erreur d'état pour Delete et Kick_Timer

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x04	N/A	Erreur de syntaxe du chemin
0x05	N/A	Instance non définie
0x08	N/A	Service non mis en œuvre (si dirigé vers l'instance 0)

7.4.5 Services spécifiques à la classe

7.4.5.1 Généralités

L'objet de planification doit prendre en charge les services spécifiques à la classe suivants spécifiés dans le Tableau 39.

Tableau 39 – Services spécifiques à la classe de l'objet de planification

Code de service	Nécessité dans la mise en œuvre		Nom du service	Description du service
	Classe	Instance		
0x4B	N/A	Requise	Kick_Timer	Réinitialisation du temporisateur dans l'objet de planification
0x4C	N/A	Requise	Read	Lit les informations relatives aux connexions planifiées
0x4D	N/A	Requise	Conditional_Write	Réécrit les informations de planification sur le dispositif CO
0x4E	N/A	Facultatif	Forced Write	Réécrit les informations de planification sur le dispositif CO
0x4F	N/A	Requise	Change Start	Lance une modification des données de planification
0x50	N/A	Requise	Break Connections	Interrompt toutes les connexions affectées après de nouvelles données de planification
0x51	N/A	Requise	Change Complete	Indique que toutes les connexions planifiées ont été interrompues
0x52	Requise	N/A	Restart Connections	Interrompt toutes les connexions qui utilisent une liaison particulière

7.4.5.2 Kick_Timer

Le service kick_timer doit réinitialiser le temporisateur dans une instance de l'objet de planification sans affecter l'état de cet objet. Le LSS doit émettre quatre commandes par période de délai d'attente. Par exemple, le LSS doit émettre une demande de service toutes les 15 s si la valeur de délai d'attente est de 60 s par défaut. Tous les autres services de l'objet de planification, dirigés vers cette instance de planification, doivent également réinitialiser le temporisateur de manière à utiliser le service Kick_Timer uniquement si le client (LSS) souhaite conserver une instance de l'objet de planification active alors qu'il n'a rien à dire.

Le temporisateur d'une instance doit être réinitialisé lorsqu'il reçoit un message. Toutefois, son compte à rebours ne doit pas démarrer tant que la réponse correspondante n'a pas été envoyée. Ceci ne doit permettre la mise en œuvre que d'une seule tâche. La réponse de ce service doit être spécifiée conformément au Tableau 38.

7.4.5.3 Read

Le LSS doit utiliser le service Read pour lire les informations relatives aux connexions planifiées que le dispositif CO souhaite créer sur une liaison particulière. Les informations de connexion doivent provenir d'un objet d'application dans le dispositif CO.

Une demande de service Read doit contenir un paramètre UDINT requis qui spécifie l'indice de connexion sur lequel démarrer la lecture. L'objet de planification doit répondre à un service Read avec autant d'informations de connexion que ne peut en contenir un Lpacket de réponse. Les indices de connexion d'un Lpacket doivent être classés dans l'ordre croissant (les intervalles doivent être admis). Le champ d'état étendu de la réponse doit indiquer s'il

existe des connexions dotées d'indices plus élevés dans le dispositif CO. Un client (LSS) doit continuer à soumettre des demandes **read** avec des indices de démarrage plus élevés tant qu'il n'a pas lu toutes les informations de connexion de la liaison choisie. Un état étendu 0 doit indiquer qu'il existe plus de connexions dans le dispositif. Un état étendu 1 doit indiquer que toutes les connexions ont été lues.

La réponse doit être composée d'un UINT indiquant le nombre de connexions décrites dans cette réponse, suivi d'une série d'entrées décrivant chacune une connexion répondant aux critères de demande de connexion (la route vers la liaison). Chaque entrée doit être associée à un indice de données de planification unique, choisi et devant être utilisé principalement par le dispositif CO. Le dernier de ces indices peut être incrémenté par le LSS pour poursuivre la demande si elle est trop longue pour une réponse.

Les autres paramètres fournis dans l'entrée de connexion doivent décrire les tailles de connexion, les types de connexion (multidiffusion/point à point) et les vitesses de mise à jour des connexions (RPI). Les informations relatives à la route d'entrée de connexion doivent être composées de la route entre la liaison et le module cible pour la connexion comprenant les points de connexion du module cible. Il n'est pas utile que le LSS utilise réellement les informations relatives au port.

Cette route doit également contenir les valeurs en cours attribuées à la connexion dans les segments de réseau. Les segments de réseau doivent contenir les informations relatives à l'API (intervalle Lpacket) et à la NUT de départ.

```
class Scheduling_Read_Request: public MR_Request
{
    UDINT first_connection_index;
}

class Scheduling_Read_Response: public MR_Response
{
    UINT number_of_connections;
    Scheduling_Connection_Description desc[];
}

class Scheduling_Connection_Description
{
    UDINT connection_index;
    UINT O2T_parameters;
    UINT T2O_parameters;
    UDINT O2T_RPI;
    UDINT T2O_RPI;
    UINT path_size;           // in words, range 0 to 255
    UINT path[];             // array of path segments
                                // first segment is always O=>T schedule segment
                                // second is always port segment onto the link
                                // third is always T=>O schedule segment
                                // remaining path including logical segments
                                // and other port segments if multihop connection
};
```

L'outil de planification LSS doit utiliser les segments logiques du chemin cible afin de déterminer si plusieurs demandes de connexion utilisent réellement un producteur multipoint. La règle concernant tous les objets (autre l'objet Assembly) doit stipuler qu'une correspondance de classe, d'instance, de point de connexion T=>O et de granularité plus fine (attribut ou membre, par exemple) constitue un producteur unique et doit être planifiée une seule fois pour les deux connexions. Étant donné que l'objet Assembly compare une instance et un point de connexion, une correspondance d'une instance de chemin de connexion ne doit pas déterminer le caractère unique du producteur.

Par exemple, ces deux chemins de connexion spécifient le même producteur (instance 0x88 de l'objet Assembly):

"20 04 24 03 2C 99 2C 88";

"20 04 24 03 2C 77 2C 88";

Ces deux chemins de connexion spécifient des producteurs différents, étant donné que le segment de classe logique ne correspond pas:

"20 77 24 03 2C 99 2C 88";

"20 77 24 AA 2C 99 2C 88".

La réponse d'état de ce service doit être spécifiée conformément au Tableau 40.

Tableau 40 – Descriptions de l'erreur d'état de Read

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x04	N/A	Erreur de syntaxe du chemin
0x05	N/A	Instance non définie
0x08	N/A	Service non mis en œuvre (si dirigé vers l'instance 0)
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée

7.4.5.4 Conditional_Write

Le service Conditional_Write doit être utilisé par le LSS pour réécrire les informations de planification dans le dispositif CO. La réception de ce message doit indiquer que le LSS planifie de nouveau des indices particuliers des informations de connexion. Un dispositif CO peut commencer par interrompre les connexions en cours associées aux indices de connexion ou attendre la réception d'une demande de service d'interruption des connexions consécutive.

Si la nouvelle planification fournie par le LSS correspond exactement pour une connexion donnée, le dispositif CO peut ignorer cette partie de la demande d'écriture. Une mise en œuvre de l'objet de planification peut répondre par une erreur de ressource indisponible si elle reçoit plus de mises à jour que demandées pour plusieurs connexions (par l'intermédiaire de demandes Read). Une mise en œuvre peut également vérifier les demandes d'écriture en double et donc autoriser les nouvelles tentatives (plus de demandes d'écriture que de lecture). Les indices de connexion peuvent ne pas être dans l'ordre.

La différence entre le service d'écriture conditionnelle et le service d'écriture forcée doit être que, dans le premier cas, le dispositif CO doit être requis pour vérifier la fraîcheur des informations de connexion préalablement envoyées au LSS. Si les données ne sont pas fraîches, le dispositif CO ne doit pas utiliser les données de planification pour cet indice de connexion. Dans le cas du service d'écriture forcée, il n'est pas utile que le dispositif CO procède à une vérification. Informations de connexion « fraîches » s'entend qu'elles n'ont pas été modifiées depuis leur envoi au LSS à l'aide de la commande Read.

La demande doit être une matrice de planifications de connexion.

```
class Scheduling_Write_Request: public MR_Request
{
    UINT    number_of_connection_schedules;
    struct {
        UDINT connection_index;
        USINT  O2T_schedule;
        USINT  T2O_schedule;
    } schedules[];
};
```

L'état de la réponse de ce service doit être spécifié conformément au Tableau 41.

Tableau 41 – Descriptions de l'erreur d'état de Conditional_Write

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x03	N/A	Valeur non valide — La requête contient un indice incorrect
0x04	N/A	Erreur de syntaxe du chemin
0x05	N/A	Instance non définie
0x08	N/A	Service non mis en œuvre (si dirigé vers l'instance 0)
0x0C	N/A	État incorrect (Change_Start non reçu)
0x10	N/A	Conflit d'état du dispositif (le dispositif ne peut pas interrompre la connexion)
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée
0x15	N/A	L'instance n'est pas en mesure de recevoir d'autres demandes d'écriture

7.4.5.5 Forced_Write

Le service Forced_Write doit être analogue au service Conditional_Write, sauf que le LSS doit garantir avoir fourni des informations de connexion valides. Les paramètres de demande doivent être identiques à ceux du service Conditional_Write. L'état de la réponse de ce service doit être spécifié conformément au Tableau 42.

Tableau 42 – Descriptions de l'erreur d'état de Forced_Write

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x04	N/A	Erreur de syntaxe du chemin
0x05	N/A	Instance non définie
0x08	N/A	Service non mis en œuvre (si dirigé vers l'instance 0)
0x0C	N/A	État incorrect (Change_Start non reçu)
0x10	N/A	Conflit d'état du dispositif (le dispositif ne peut pas interrompre la connexion)
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée
0x15	N/A	L'instance n'est pas en mesure de recevoir d'autres demandes d'écriture

7.4.5.6 Change_Start

Le service Change_Start doit être utilisé pour lancer une modification de données de planification dans le contrôleur. Le contrôleur doit allouer suffisamment de mémoire pour les nouvelles données, le mot de passe et le chemin CO. Le service **Conditional_Write** ou **Forced_Write** doit être utilisé pour écrire les données dans le contrôleur.

Le mot de passe CO doit être utilisé lorsqu'un contrôleur est activé, puis commence à rétablir les connexions planifiées. Le CO doit s'assurer que le chemin associé au mot de passe CO (voir réponse Create ou demande Change_Complete) correspond à celui du chemin entre le réseau et le dispositif CO. S'il correspond, le CO doit s'assurer que ce mot de passe CO correspond à celui stocké dans l'objet Keeper de la liaison, avant d'établir des connexions planifiées sur ladite liaison.

La demande `Change_Start` ne doit comporter qu'un seul paramètre. Le paramètre doit être un UINT compris entre 0 et 255 qui spécifie la taille du mot de passe et du chemin CO à fournir dans le service `Change_Complete`. Les mises en œuvre peuvent l'utiliser pour allouer une mémoire tampon de réception pour le mot de passe et le chemin CO.

L'état de la réponse de ce service doit être spécifié conformément au Tableau 43.

Tableau 43 – Descriptions de l'erreur d'état de `Change_Start`

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x02	0x0002	Ressources insuffisantes — mémoire insuffisante sur le dispositif CO
0x04	N/A	Erreur de syntaxe du chemin
0x05	N/A	Instance non définie
0x08	N/A	Service non mis en œuvre (si dirigé vers l'instance 0)
0x10	N/A	Conflit d'état du dispositif (le dispositif ne peut pas allouer suffisamment de mémoire, par exemple)
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée

7.4.5.7 Break_Connections

Le service `Break_Connections` doit être utilisé après avoir écrit les nouvelles données de planification dans le contrôleur afin d'interrompre toutes les connexions concernées. Le service doit répondre uniquement après l'interruption de toutes les connexions. Si des connexions sont en cours, elles doivent également être interrompues avant de répondre à ce service.

Le service `Break_Connections` de l'objet de planification doit interrompre toutes les connexions spécifiées dans les services d'écriture de données de planification. Un LSS peut toujours écrire des données de planification pour toutes les connexions (interrompant toutes les connexions sur la liaison), mais un autre LSS peut être optimisé pour limiter le nombre de connexions interrompues.

Lors de la réception du service `Break_Connections` de l'objet de planification, outre l'interruption des connexions modifiées, le CO doit attribuer une valeur nulle (zéro) aux données planifiées actives pour les connexions qu'il interrompt, afin éviter de les rétablir tant que le service `Change_Complete` des données de planification n'a pas été reçu. Si le service `Change_Complete` n'est jamais reçu (les communications avec le LSS ayant été interrompues), ces connexions ne doivent pas pouvoir être ouvertes tant qu'elles n'ont pas été de nouveau planifiées. La réponse à ce service doit être spécifiée conformément au Tableau 44.

Tableau 44 – Descriptions de l'erreur d'état de Break_Connections

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x04	N/A	Erreur de syntaxe du chemin
0x05	N/A	Instance non définie
0x08	N/A	Service non mis en œuvre (si dirigé vers l'instance 0)
0x0C	N/A	Conflit d'état de l'objet
0x10	N/A	Conflit d'état du dispositif (le dispositif ne peut pas interrompre les connexions)
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée

7.4.5.8 Change_Complete

Le service Change_Complete doit être utilisé pour indiquer que toutes les connexions planifiées ont été interrompues et qu'il est admis d'établir les nouvelles connexions planifiées. La demande contient les nouveaux mot de passe et chemin CO utilisés pour la liaison qui a été planifiée. Le CO doit s'assurer que le chemin associé au mot de passe CO (voir réponse Create ou demande Change_Complete) correspond à celui du chemin entre le réseau et le dispositif CO. S'il correspond, le CO doit s'assurer que ce mot de passe CO correspond à celui stocké dans l'objet Keeper de la liaison, avant d'établir des connexions planifiées sur ladite liaison.

```
class Scheduling_Change_Complete_Request: public MR_Request
{
    UDINT  cop;           // Connection Originator Password (COP)
    UINT    path_size;     // in words, range 0 to 255
    UINT    path[];        // array of port segments from the scheduled link
                          // to the connection originator
                          // no key segments, no network segments
                          // include the MAC ID of the link hop
};
```

NOTE Le chemin inverse de cette demande permet au LSS de détecter les modifications dans la position de réseau du CO planifié pendant les dernières sessions de planification.

L'état de la réponse de ce service doit être spécifié conformément au Tableau 45.

Tableau 45 – Descriptions de l'erreur d'état de Change_Complete

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x03	N/A	Valeur non valide (les nouveaux mot de passe et chemin CO ne sont pas correctement mis en forme)
0x04	N/A	Erreur de syntaxe du chemin
0x05	N/A	Instance non définie
0x08	N/A	Service non mis en œuvre (si dirigé vers l'instance 0)
0x10	N/A	Conflit d'état du dispositif
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée
0x0C	N/A	État incorrect

7.4.5.9 Restart_Connections

Le service Restart_Connections peut uniquement être envoyé à l'instance de classe de l'objet de planification (instance zéro). Il doit être utilisé pour interrompre toutes les connexions qui utilisent une liaison particulière. Suite à l'interruption de toutes les connexions, le dispositif CO doit tenter de les rétablir. Le paramètre de demande de ce service doit être structuré comme l'attribut d'instance 0x01. L'état de la réponse de ce service doit être spécifié conformément au Tableau 46.

Tableau 46 – Descriptions de l'erreur d'état de Restart_Connections

État général	État étendu	Description de l'erreur
0x00	N/A	Réussite
0x04	N/A	Erreur de syntaxe du chemin
0x08	N/A	Service non mis en œuvre (s'il est adressé à une instance non nulle)
0x10	N/A	Conflit d'état du dispositif (impossible d'interrompre les connexions)
0x13	N/A	Données de demande insuffisantes — la demande est trop courte ou tronquée
0xD0	N/A	Aucune connexion planifiée sur cette liaison

7.4.6 Session de planification classique

Une session de planification classique doit utiliser ces services:

- a) Create;
- b) Read;
- c) Write;
- d) Change_Start;
- e) Break_Connections;
- f) Conditional_Write;
- g) Forced_Write;
- h) Change_Complete.

La procédure ci-dessous décrit l'utilisation des services dans une session de planification classique:

- 1) Le LSS doit commencer à envoyer un message **create** à une instance zéro de l'objet de planification. Ce message doit contenir un chemin entre le CO et la liaison. Ce chemin doit déterminer les connexions présentant un intérêt pour le LSS.
- 2) L'objet de planification doit créer une instance de lui-même au service de ce LSS et répondre par le numéro d'instance auquel toutes les communications ultérieures doivent être envoyées.
- 3) L'instance qui vient d'être créée doit réinitialiser un temporisateur de 60 s dès la réception d'un message. Si ce temporisateur expire, l'instance doit se supprimer elle-même en annulant toutes les modifications en cours. En fonctionnement normal, le LSS doit envoyer une commande à l'objet de planification au moins une fois toutes les 15 s, en permettant de supprimer quelques messages sans supprimer inutilement une instance.
- 4) Le LSS peut éventuellement envoyer des commandes **read** à l'instance qui vient d'être créée afin de transférer les informations de connexion depuis le dispositif CO. Les informations de connexion transmises doivent contenir les indices utilisés pour identifier de manière unique les connexions en interne dans le dispositif CO.