



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

T.121

(07/96)

SÉRIE T: ÉQUIPEMENTS TERMINAUX ET
PROTOCOLES DES SERVICES TÉLÉMATIQUES

Modèle générique d'application

Recommandation UIT-T T.121

(Antérieurement «Recommandation du CCITT»)

RECOMMANDATIONS UIT-T DE LA SÉRIE T
EQUIPEMENTS TERMINAUX ET PROTOCOLES DES SERVICES TÉLÉMATIQUES

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (UIT). Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT (Helsinki, 1^{er}-12 mars 1993).

La Recommandation UIT-T T.121, que l'on doit à la Commission d'études 8 (1993-1996) de l'UIT-T, a été approuvée le 3 juillet 1996 selon la procédure définie dans la Résolution n° 1 de la CMNT.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue de télécommunications.

© UIT 1996

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

	<i>Page</i>
1	Domaine d'application 1
2	Références normatives 1
3	Définitions 2
4	Abréviations 2
5	Aperçu général..... 2
5.1	Application d'utilisateur 7
5.2	Entité protocolaire d'application 8
5.3	Gestionnaire de ressources d'application..... 8
5.4	Élément de service d'application 9
6	Modèle générique d'application..... 10
6.1	Initialisation 11
6.2	Inscription 11
6.2.1	Session d'enregistrement..... 14
6.2.2	Session de base normalisée 15
6.2.3	Session de base non normalisée..... 15
6.2.4	Session publique..... 21
6.2.5	Session privée..... 24
6.3	Constitution de clés de référentiel..... 26
6.4	Négociation de capacités..... 26
6.5	Sortir d'une session 26
7	Examen des ressources MCS et des services GCC..... 27
7.1	Canaux 27
7.1.1	Canal statique 27
7.1.2	Canal dynamique 27
7.1.3	Canal d'identificateur d'utilisateur 28
7.1.4	Canal attribué 28
7.1.5	Canal privé 30
7.1.6	Problèmes potentiels d'utilisation de canaux MCS 30
7.1.7	Résumé..... 31
7.2	Transfert de données 32
7.3	Jetons 32
7.3.1	Jetons statiques 33
7.3.2	Jetons dynamiques..... 33
7.4	Répertoire d'application..... 34
7.5	Référentiel d'application..... 35
7.6	Présidence 35
7.7	Invocation à distance..... 36
8	Directives relatives aux applications d'utilisateur 36
8.1	Identification des applications d'utilisateur..... 36
8.2	Utilisation de capacités fédérables et non fédérables 36
8.3	Contexte de session 37
8.4	Choix d'une session pour y participer 37
8.5	Contrôle de l'utilisation d'une session 37
8.6	Ajout de nouveaux participants à une session 37
8.7	Extension, modification et réutilisation de protocoles d'application 38

RÉSUMÉ

La présente Recommandation propose aux réalisateurs d'applications et aux réalisateurs de protocoles d'application des conseils pour une utilisation correcte et efficace de l'infrastructure T.120. Elle propose un modèle générique pour une application T.120 qui communique en faisant usage de services T.120 et définit un modèle générique d'application (*generic application template*) qui spécifie l'utilisation des services T.122 (MCS) et T.124 (GCC) habituellement requis par les protocoles d'applications faisant usage de services T.120. La présente Recommandation propose une structure commune aux entités protocolaires d'application, qui garantit que des entités protocolaires d'application pourront coexister au sein d'une même conférence. Les entités protocolaires d'application qui n'observeront pas les opérations définies dans cette Recommandation s'exposeront au risque d'interférer avec d'autres protocoles d'application.

MODÈLE GÉNÉRIQUE D'APPLICATION

(Genève, 1996)

1 Domaine d'application

Tandis que les applications d'utilisateur par elles-mêmes ne font pas l'objet d'une normalisation, les protocoles employés par une application d'utilisateur doivent être normalisés de façon à assurer leur interconnexion. La présente Recommandation fournit un modèle générique pour une application T.120, en vue de définir un cadre général commun (le modèle générique d'application) qui serve de base aux protocoles d'application normalisés. Il fournit aussi, aux réalisateurs d'application d'utilisateur, une assistance sur la manière de tirer profit de l'infrastructure T.120 pour obtenir de l'application le comportement désiré. Les modèles définis dans la présente Recommandation sont dérivés de protocoles d'application pour systèmes à terminaux; les protocoles d'application qui incorporent une composante MCU peuvent nécessiter des dispositions différentes.

Une application d'utilisateur T.120 emploie une ou plusieurs entités protocolaires d'application (*application protocol entity*) pour communiquer avec des applications d'utilisateur de fonctionnalités similaires en d'autres nœuds dans une conférence. Il importe d'observer que, tandis qu'il n'est pas nécessaire que les mises en œuvre soient identiques en chaque nœud pour parvenir à échanger des informations, des groupes d'applications d'utilisateur qui communiquent ensemble doivent employer les mêmes protocoles d'application dans le même but général. Les différences de capacités sont résolues par un mécanisme d'échange de capacités.

Chaque entité protocolaire d'application (APE) comprend deux composants:

- une partie générique, qui comprend des éléments communs à tous les protocoles d'application ou à la plupart d'entre eux; par exemple, l'initialisation et la gestion des ressources. De telles opérations n'ont à être spécifiées qu'une fois (dans la présente Recommandation); chaque spécification de protocole y fera ensuite référence;
- une partie propre à la fonction qui permet l'interconnexion d'applications d'utilisateur de fonctionnalités semblables. Cette partie peut être normalisée, auquel cas elle est spécifiée dans une Recommandation T.120 relative aux protocoles d'application, ou hors norme.

La présente Recommandation décrit les composants identifiés ci-dessus et définit un modèle générique d'application (GAT) (*generic application template*) contenant ces opérations dont on s'attend à ce qu'elles soient communes à la plupart des protocoles d'application T.120. Le modèle générique d'application facilite la tâche du réalisateur d'un protocole d'application et fournit une structure commune pour les entités protocolaires d'application. Cette façon modulaire de procéder permet une identification nette des éléments propres à une fonction, dans une spécification de protocole, et rend possible une réutilisation plus aisée dans d'autres spécifications de protocole. La présente Recommandation définit un protocole rigoureux qui s'exprime par une séquence d'opérations primitives. En outre, la présente Recommandation peut être rendue normative par la référence qui y sera faite dans les Recommandations relatives aux protocoles d'application normalisés. L'utilisation du modèle générique d'application dans la définition de protocoles d'application normalisés est conseillée.

2 Références normatives

Les Recommandations et autres références suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation. Au moment de la publication, les éditions indiquées étaient en vigueur. Toute Recommandation ou autre référence est sujette à révision; tous les utilisateurs de la présente Recommandation sont donc invités à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et autres références indiquées ci-après. Une liste des Recommandations UIT-T en vigueur est publiée régulièrement.

- Recommandation UIT-T T.120 (1996), *Protocoles de données pour conférence multimédia*.
- Recommandation UIT-T T.122 (1993), *Service de communication multipoint pour la définition des services de conférence audiographique et conférence audiovisuelle*.
- Recommandation UIT-T T.124 (1995), *Commande de conférence générique*.
- Recommandation UIT-T T.125 (1994), *Spécification de protocole du service de communication multipoint*.
- Recommandation UIT-T T.50 (1992), *Alphabet international de référence (ancien alphabet international n° 5 ou IA5) – Technologie de l'information – Jeux de caractères codés à 7 bits pour l'échange d'informations*.

3 Définitions

Pour les besoins de la présente Recommandation, les définitions suivantes s'appliquent:

3.1 gestionnaire de ressource d'application: La partie d'une entité protocolaire d'application qui fournit une fonctionnalité générique pour gérer des ressources MCS et GCC.

3.2 élément de service d'application: La partie d'une entité protocolaire d'application qui fournit au protocole d'application une fonctionnalité particulière; par exemple, les formats et séquences de messages.

3.3 session de base non normalisée: Une session de protocole d'application qui peut être localisée par des entités protocolaires d'application (APE) non normalisées sans nécessiter d'intervention de l'utilisateur. En soi, elle est similaire à une session de base normalisée utilisée par des APE normalisées.

3.4 session privée: Une session de protocole d'application dont le nombre de membres est restreint et contrôlé par le créateur de cette session.

3.5 session publique: Une session de protocole d'application dont le nombre de membres n'est pas restreint et qui est habituellement utilisée si la session de base appropriée au protocole est déjà en cours d'utilisation et si un utilisateur désire établir d'autres sessions indépendantes.

3.6 session d'enregistrement: Une session de protocole d'application utilisée par les applications pour annoncer leur présence à la conférence. Une session d'enregistrement utilise la session par défaut T.124, dans laquelle il n'existe pas d'identificateur de session (*Session ID*).

3.7 créateur de session: L'entité protocolaire d'application qui s'inscrit à une nouvelle session de protocole d'application. Le concept de créateur de session ne s'applique qu'aux sessions publiques et privées.

3.8 membre de session: Une entité protocolaire d'application qui s'inscrit à une session existante de protocole d'application. Le concept de membre de session ne s'applique qu'aux sessions publiques et privées.

3.9 session de base normalisée: Une session de protocole d'application qui peut être localisée par les APE normalisées sans nécessiter d'intervention de l'usager. Les identificateurs de sessions de base normalisées sont spécifiés dans la Recommandation T.120.

4 Abréviations

Pour les besoins de la présente Recommandation, les abréviations suivantes sont utilisées:

ARM	Gestionnaire de ressources d'application (<i>application resource manager</i>)
APE	Entité protocolaire d'application (<i>application protocol entity</i>)
ASE	Élément de service d'application (<i>application service element</i>)
GAT	Modèle générique d'application (<i>generic application template</i>)
GCC	Commande générique de conférence (<i>generic conference control</i>)
GCC SAP	Point d'accès au service GCC (<i>GCC service access point</i>)
MCS	Service de communication multipoint (<i>multipoint communication service</i>)
MCSAP	Point d'accès au service MCS (<i>MCS service access point</i>)
MCU	Unité de commande multipoint (<i>multipoint control unit</i>)
PDU	Unité de données protocolaire (<i>protocol data unit</i>)
SAP	Point d'accès au service (<i>service access point</i>)

5 Aperçu général

La Figure 1 illustre le modèle de système T.120 dans lequel les applications d'utilisateur emploient des protocoles d'application tant normalisés que non normalisés de façon à communiquer avec leurs homologues d'autres nœuds au sein d'une même conférence. Le domaine d'application d'une application d'utilisateur est limité aux fonctions locales (par exemple l'interface utilisateur) qui ne touchent pas à la communication entre applications d'utilisateur homologues. Chaque application d'utilisateur fait usage d'une ou plusieurs entités protocolaires d'application (APE) qui mettent en œuvre des protocoles d'interfonctionnement comme montré dans la Figure 2. Dans le modèle décrit ici, une APE peut

être encore subdivisée en deux éléments: un Gestionnaire de ressources d'application (ARM) et un élément de service d'application (ASE). L'ARM est chargé de la gestion générique de ressources MCS et GCC, tandis que l'ASE fournit l'interfonctionnement de fonctionnalités spécifiques; par exemple, un ASE peut fournir une fonctionnalité universelle de transfert de fichiers suivant les critères établis dans T.127, ou une fonctionnalité d'échange d'images fixes suivant les critères établis dans T.126.

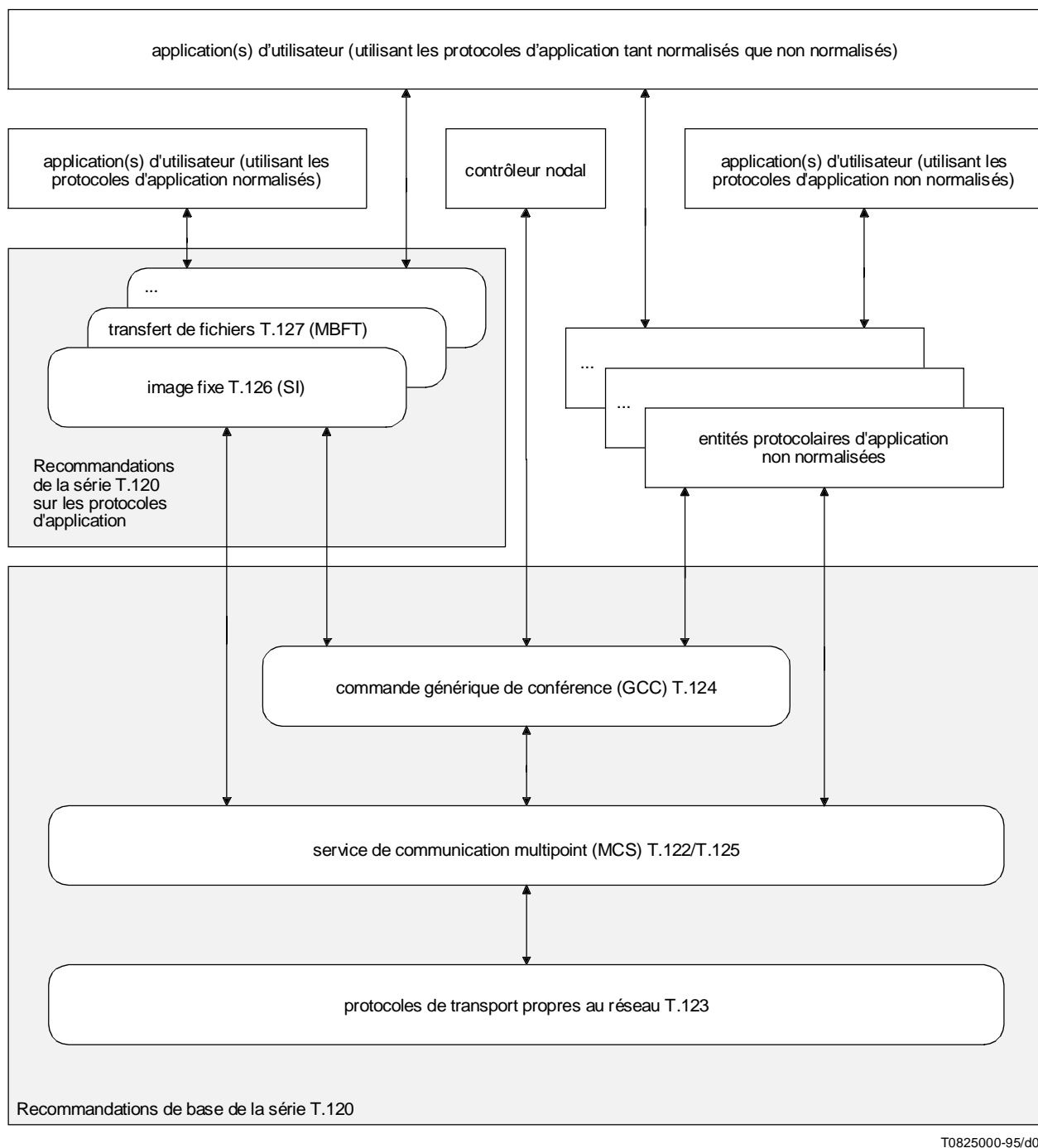
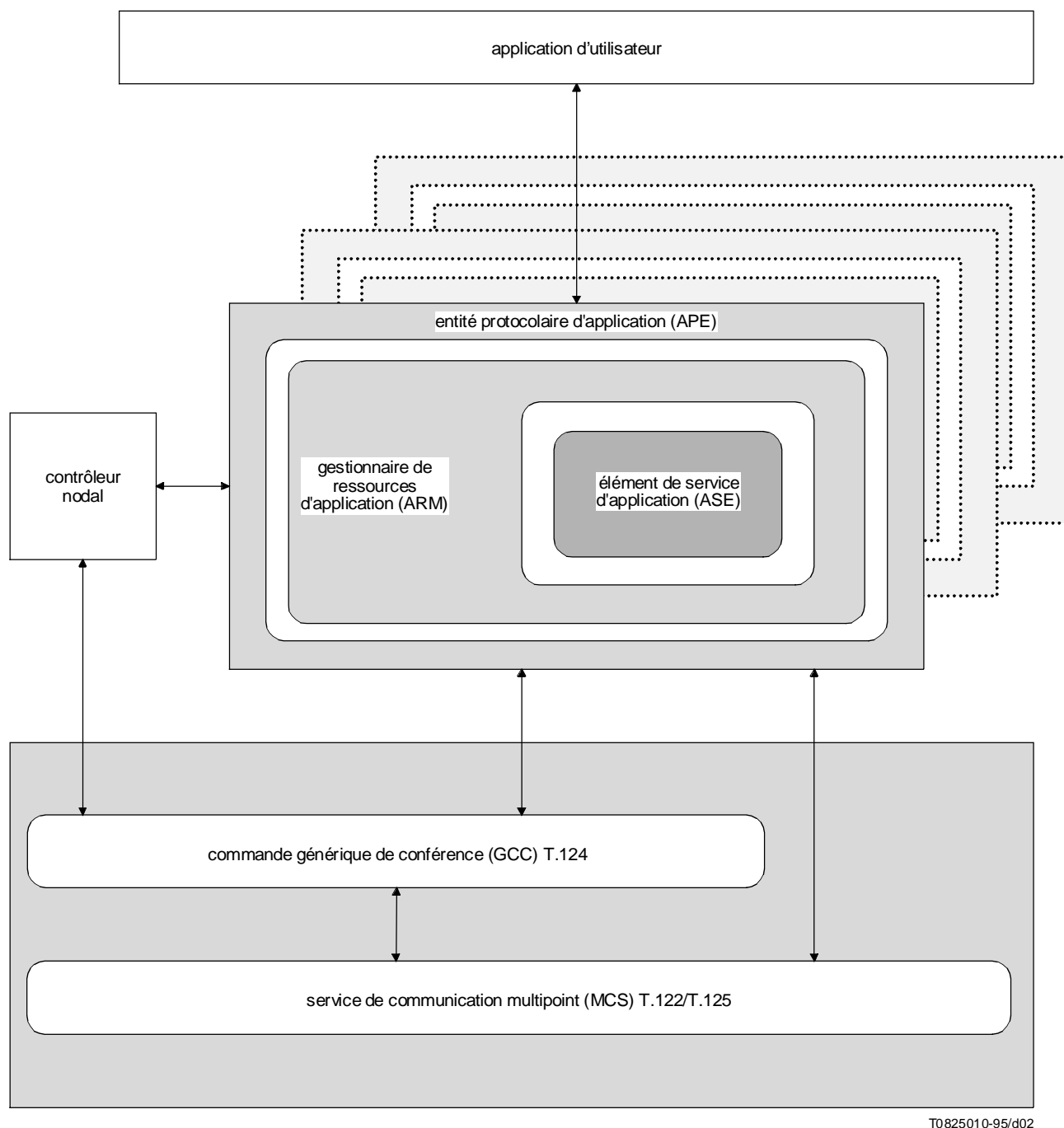


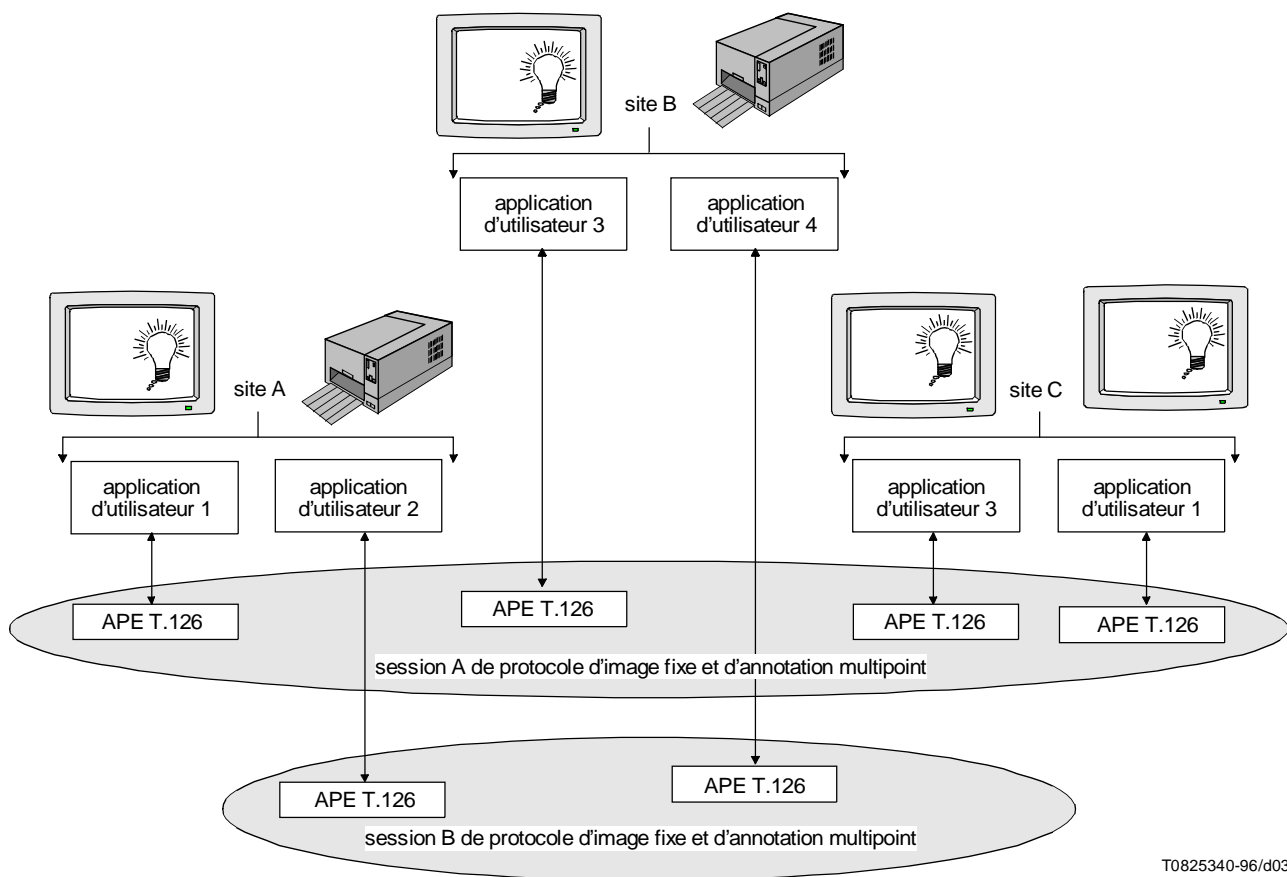
FIGURE 1/T.121
Modèle de système T.120



T0825010-95/d02

FIGURE 2/T.121
Modèle d'entité protocolaire d'application T.120 générique

D'éléments homologues de protocoles d'application qui communiquent entre eux, on dit qu'ils participent à la même *session* de protocole d'application. On notera que plus d'une application d'utilisateur, sur un site donné, peut participer à la même session. Cela entraîne de multiples instances du protocole d'application, c'est-à-dire de multiples APE, une par application d'utilisateur. Il est également possible d'avoir de multiples sessions simultanées utilisant le même protocole d'application. Encore une fois, chaque application d'utilisateur participant nécessite sa propre APE. Un scénario d'exemple est représenté dans la Figure 3.



T0825340-96/d03

FIGURE 3/T.121

Exemple d'applications multiples sur un site utilisant le même protocole

Une session se compose d'un groupe d'entités homologues de protocole d'application qui emploient un protocole d'application commun pour communiquer entre elles. Chaque session a un identificateur de session dont la valeur est égale à celle d'un identificateur de canal MCS (*MCS Channel ID*), habituellement un identificateur utilisé pour transférer des données au sein du groupe d'APE. Dans les modèles présentés ici, le canal associé à l'identificateur de session est le premier canal à recevoir un branchement, bien que ce ne soit pas une condition requise. Il est connu comme canal d'identificateur de session (*Session Identifier Channel*). Une application d'utilisateur qui emploie de multiples APE peut prendre part à de multiples sessions du protocole d'application. Chaque session pourrait avoir un identificateur unique de session mais ce n'est pas nécessaire. Des entités protocolaires d'application de types différents participent à des sessions différentes, même si elles ont le même identificateur de session (*Session ID*). Un scénario d'exemple est représenté dans la Figure 4. Pour chaque protocole d'application, il est possible d'avoir une seule session de protocole d'application sans identificateur de session; on y fait référence comme à la session d'enregistrement (*registration session*).

Il existe les types de session suivants:

session d'enregistrement: peut être utilisée par les protocoles d'application normalisés et non normalisés. L'objet d'une session d'enregistrement est de mettre les applications à même d'annoncer leur présence à la conférence, sans devoir devenir actives. Cela permet aux applications de déterminer quels sont les nœuds qui, dans la conférence, ont des fonctionnalités semblables, tout en consommant un minimum de ressources. La session d'enregistrement permet aussi aux applications d'annoncer les fonctionnalités spécifiques de leur protocole d'application à la conférence en tant que capacités non fédérables. Les applications resteront habituellement inscrites à la session d'enregistrement pour la durée de la conférence, de façon à recevoir notification de tout changement des fonctionnalités annoncées par les autres nœuds. Une application inscrite à la session d'enregistrement pour un protocole d'application spécifique reçoit des informations sur toutes les sessions de la conférence utilisant ce protocole d'application particulier;

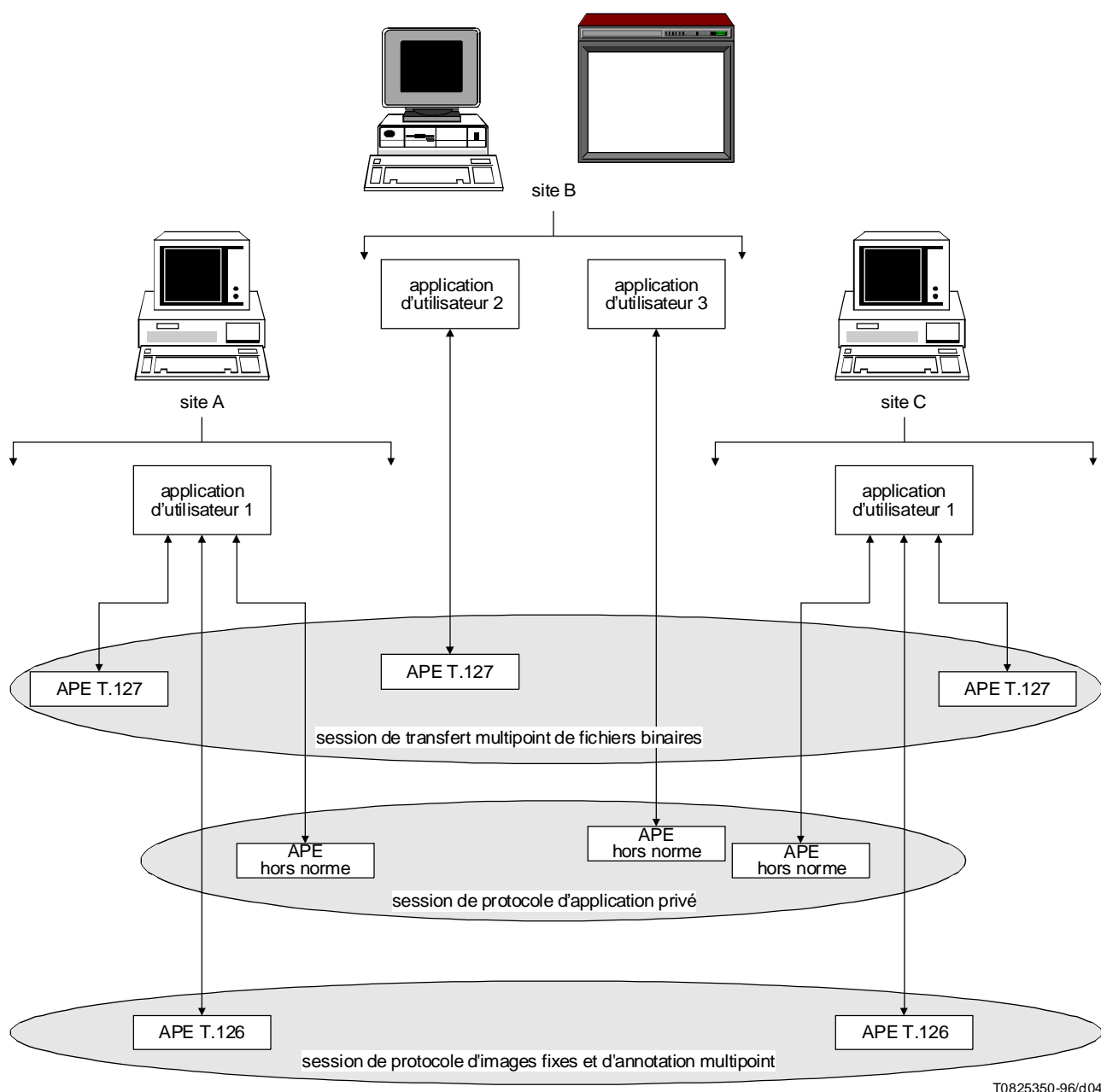


FIGURE 4/T.121
Exemple d'applications utilisant de multiples protocoles

L'utilisation d'une session d'enregistrement pour l'activité effective d'une application ne fait pas partie de ces modèles et est déconseillée. Ceci assure une nette ségrégation entre applications actives et applications en sommeil. Cette ségrégation est importante parce que:

- toute capacité fédérable annoncée par des applications inactives peut avoir un effet défavorable sur l'ensemble des capacités de la session,
- et que les entrées de référentiel périmées dans une session d'enregistrement résultant de l'activité d'une application sont susceptibles de persister jusqu'à la fin de la conférence.

Pour chaque protocole d'application dans une conférence il peut y avoir au plus une session d'enregistrement. Celle-ci est caractérisée par le fait d'avoir une clé de session (*Session Key*) composée de la clé de protocole d'application appropriée sans aucun identificateur de session (*Session ID*);

session de base normalisée: réservée à l'utilisation par des protocoles d'application normalisés. L'accès à une session de base normalisée n'est pas limité et tous ceux qui prennent part à la session ont le même statut, celui de membre (*session member*); ils peuvent se brancher et sortir à volonté. Puisque l'identificateur de session pour une session de base normalisée d'un protocole d'application normalisé est prédéfini (dans T.120), une application peut se joindre à la session sans aucune intervention de l'utilisateur. Une session de base normalisée utilise un canal statique comme canal d'identificateur de session. L'ensemble des ressources à identifier au cours du processus d'inscription à une session de base normalisée peut inclure d'autres canaux et jetons statiques ou dynamiques, comme spécifié par le protocole d'application. Des canaux et jetons dynamiques peuvent être employés si des ressources additionnelles sont requises en sus de cet ensemble de base.

Habituellement, la session de base normalisée devrait avoir la préférence pour une application utilisant un protocole d'application normalisé et qui désire communiquer avec toutes les applications homologues au sein d'une conférence. Dans une conférence, pour chaque protocole d'application normalisé, il existe au plus une session de base normalisée;

session de base non normalisée: réservée à l'utilisation par des protocoles d'application non normalisés. La session de base non normalisée permet aux applications qui emploient un protocole hors norme commun de localiser la même session de manière indépendante, sans requérir d'intervention de l'utilisateur pour sélectionner une session spécifique. Habituellement, la session de base non normalisée sera le choix préférentiel pour une application utilisant un protocole d'application non normalisé et qui désire communiquer avec toutes les applications homologues dans une conférence. La session de base non normalisée utilise, comme identificateur de session (*Session ID*), un canal assigné (c'est-à-dire dynamique multidestinataire) dont l'identité se découvre en utilisant le référentiel d'application GCC (*GCC application registry*). Une conséquence de la participation à la session de base non normalisée est le branchement sur la session d'enregistrement correspondante. Pour chaque protocole d'application non normalisé, dans une conférence, il ne peut y avoir plus d'une session de base non normalisée. Une fois que tous les participants ont quitté une session de base non normalisée, celle-ci cesse d'exister et devra, si besoin est, être recréée par la suite;

session publique: peut être utilisée par des protocoles d'application soit normalisés, soit non normalisés. Une session publique est habituellement utilisée si la session de base correspondant au protocole est déjà en cours d'utilisation et si un usager désire établir d'autres sessions indépendantes pour communiquer avec toutes les applications homologues au sein d'une conférence. L'accès à une session publique n'est pas contrôlé. Une session publique a un créateur désigné. Celui-ci est l'APE qui attribue le canal d'identificateur de session publique et qui est chargée de l'enregistrement des ressources pour la session; tous les autres participants sont membres de la session. Tout participant peut se joindre à une session publique et la quitter à son gré; ils peuvent, à titre facultatif, choisir d'inviter d'autres nœuds à prendre part à la session en utilisant le mécanisme d'appel GCC-Application-Invoke. Une session publique peut demeurer après que le créateur de la session l'a quittée ou même l'a détachée. Une fois que tous les participants ont quitté une session publique, personne ne peut s'y joindre car la session cesse d'exister;

session privée: peut être utilisée par des protocoles d'application soit normalisés, soit non normalisés. Une session privée est habituellement employée lorsqu'un utilisateur désire restreindre l'appartenance à une session de protocole d'application à un sous-ensemble sélectionné de participants à la conférence. Une session privée a un créateur désigné. Celui-ci est l'APE qui organise la session privée et qui est chargée de l'enregistrement des ressources pour la session. La participation à la session privée est contrôlée par le créateur de session, qui doit explicitement inviter d'autres applications à devenir membres de la session. Une session privée cesse d'exister après que le créateur de la session l'a dissoute ou détachée.

5.1 Application d'utilisateur

Une application d'utilisateur s'occupe de ces tâches qui sont sans effet direct sur l'interfonctionnement (par exemple, l'interface utilisateur) et qui peuvent donc être spécifiques à un produit et à une plate-forme. L'influence de l'application d'utilisateur se fait sentir sur les autres sites à travers les protocoles d'application qu'elle utilise.

Une application d'utilisateur dépend des services d'une ou plusieurs entités protocolaires d'application (APE) pour communiquer avec des applications d'utilisateur homologues sur d'autres nœuds. Elle ne communique pas avec le MCS ou la GCC; ceci est effectué par la (les) APE. L'interface entre une application d'utilisateur et son (ses) APE relève de la compétence locale et est hors du domaine d'application de la présente Recommandation.

Pour chaque APE qu'emploie l'application d'utilisateur, l'application d'utilisateur définit le sous-ensemble des capacités de cette APE qu'il faudra annoncer. Elle choisit aussi le type de session (session de base normalisée, de base non normalisée, publique, privée ou d'enregistrement) lorsqu'elle crée une nouvelle instance de cette APE.

L'application d'utilisateur communique avec l'APE appropriée pour déclencher une session d'application en précisant les capacités d'application et le mode de fonctionnement pour cette session. Une fois qu'une session a été établie, toutes les transactions particulières au protocole d'application sont effectuées par l'APE pour le compte de l'application d'utilisateur.

5.2 Entité protocolaire d'application

Une entité protocolaire d'application (APE) est employée par une application d'utilisateur pour permettre à celle-ci de communiquer avec autres applications d'utilisateur de fonctionnalités semblables, c'est-à-dire des applications d'utilisateur qui emploient le même protocole d'application dans le même but. Par exemple, des applications d'utilisateur qui nécessitent une fonctionnalité universelle de transfert de fichiers doivent employer une entité protocolaire d'application qui utilise le protocole défini dans la Recommandation T.127 et la clé de protocole d'application spécifiée dans la Recommandation T.127 pour garantir l'échange d'informations. Les APE peuvent utiliser, soit des protocoles d'application normalisés, soit des protocoles non normalisés.

Une APE a deux composants théoriques, comme le montre la Figure 2: un gestionnaire de ressources d'application (ARM) et un élément de service d'application (ASE). L'ARM fournit les fonctionnalités génériques, communes à de nombreux protocoles d'application normalisés, tandis que l'ASE fournit les fonctionnalités particulières à son protocole d'application correspondant.

Dans le modèle général, une entité protocolaire d'application est caractérisée par les attributs suivants:

- un unique gestionnaire de ressources d'application (ARM);
- un unique élément de service d'application (ASE);
- un unique SAP de GCC;
- interaction avec un contrôleur nodal (hors du domaine d'application des Recommandations de la série T.120);
- un ou plusieurs MCSAP;
- un seul identificateur d'utilisateur MCS (*MCS User ID*) inscrit;
- une liste de capacités (incluant les capacités fédérables et non fédérables);
- un seul identificateur de session (*Session ID*) (les APE homologues communiquant entre elles partagent la même Session ID. L'absence d'un identificateur de session indique que l'APE est inscrite dans la session d'enregistrement pour le protocole d'application);
- une clé de protocole d'application.

5.3 Gestionnaire de ressources d'application

Le gestionnaire de ressources d'application (ARM) est chargé de la gestion des ressources GCC et MCS pour le compte de l'ASE au sein de son APE. L'ARM fournit les services suivants:

- réaction aux indications provenant de la GCC (par exemple, autorisation d'inscription);
- inscription de son APE auprès de la GCC;
- obtention de poignées (*handles*) auprès de la GCC;
- attachement à un domaine MCS pour obtenir un identificateur unique d'utilisateur MCS (*MCS User ID*) pour son APE;
- branchement sur des canaux statiques;
- identification de canaux assignés et branchement sur ces canaux, en utilisant le référentiel GCC et le MCS;
- organisation de canaux privés et admission d'APE homologues sur de tels canaux;
- branchement sur tout canal privé auquel son APE a été admise;
- identification et obtention de jetons auprès du référentiel GCC;
- identification et attribution de paramètres par utilisation du référentiel GCC;
- annulation, dans le référentiel, de toute entrée redondante dont l'APE est responsable;
- invocation d'APE homologues sur d'autres nœuds;
- traitement des comptes rendus de répertoire d'application pour déterminer l'identité et la liste de capacités d'application négociées courantes de nœuds homologues;
- réception des indications GCC-Conductor-Assign (attribution de présidence) et GCC-Conductor-Release (abandon de présidence);
- notification à son ASE de modifications dans l'état de la présidence.

5.4 Elément de service d'application

L'élément de service d'application (ASE) fournit, à l'application d'utilisateur, les fonctionnalités propres au protocole d'application, avec les ressources obtenues par l'ARM. Par exemple, dans T.127, l'ASE fournit une fonctionnalité à usage général de transfert de fichiers. Son fonctionnement est indépendant du type (c'est-à-dire statique ou dynamique) et de l'identité des jetons et des canaux qui lui sont passés. L'ASE obtient de son ARM l'identité des ressources à utiliser. Dans le cas de transactions sur des canaux privés, l'application d'utilisateur doit préciser l'identificateur de nœud GCC (*GCC node ID*) des nœuds avec lesquels la communication est nécessaire.

L'ASE fournit les services suivants:

- envoi et réception des PDU propres au protocole d'application;
- saisie et libération de jetons et détermination du statut des jetons au moyen du MCS;
- branchement sur les canaux et sortie des canaux à travers l'ARM.

Le Tableau 1 définit la subdivision des primitives utilisées par une APE entre ses composantes ASE et ARM. Les primitives GCC non référencées dans ce tableau sont traitées par le contrôleur nodal.

TABLEAU 1/T.121

Primitives utilisées par un ARM ou un ASE

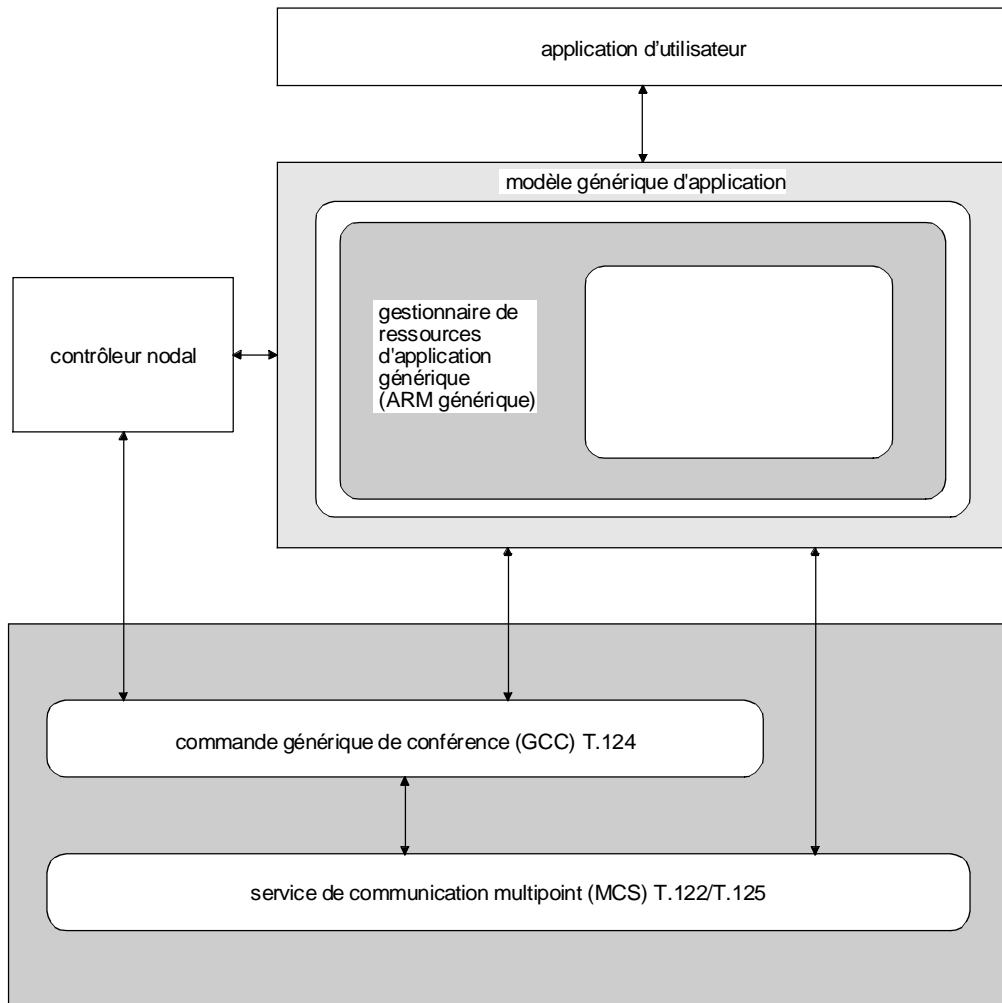
Primitive	ARM	ASE	Contrôleur nodal
GCC-Registry-Register-Channel (enregistrement de canal)	✓		
GCC-Registry-Assign-Token (attribution de jeton)	✓		
GCC-Registry-Set-Parameter (fixation de paramètre)	✓		
GCC-Registry-Retrieve-Entry (recherche d'entrée)	✓		
GCC-Registry-Delete-Entry (annulation d'entrée)	✓		
GCC-Registry-Monitor (surveillance)	✓		
GCC-Registry-Allocate-Handle (allocation de poignée)	✓		
GCC-Conference-Roster-Inquire (consultation de répertoire de conférence)	✓		✓
GCC-Conductor-Assign (attribution de présidence)	Indication		✓
GCC-Conductor-Release (abandon de présidence)	Indication		✓
GCC-Conductor-Please (demande de présidence)			✓
GCC-Conductor-Give (cession de présidence)			✓
GCC-Conductor-Inquire (interrogation du président)	✓		✓
GCC-Conductor-Permission-Ask (demande d'autorisation du président)	Indication		✓
GCC-Conductor-Permission-Grant (octroi d'autorisation du président)	Indication		✓
GCC-Application-Permission-To-Enroll (autorisation d'inscription à une application)	✓		
GCC-Application-Enroll (inscription d'application)	✓		
GCC-Application-Roster-Report (compte rendu de répertoire d'application)	✓		✓
GCC-Application-Roster-Inquire (consultation de répertoire d'application)	✓		✓
GCC-Application-Invoke (invocation d'application)	Demande		✓
MCS-Attach-User (rattachement d'utilisateur)	✓		
MCS-Detach-User (détachement d'utilisateur)	✓		
MCS-Channel-Join (branchement sur le canal MCS)	✓		
MCS-Channel-Leave (sortie de canal MCS)	✓		
MCS-Channel-Convene (constitution de canal d'invitation)	✓		
MCS-Channel-Disband (dissolution de canal)	✓		
MCS-Channel-Admit (admission sur canal)	✓		
MCS-Channel-Expel (rejet de canal)	✓		
MCS-Send-Data (envoi de données)		✓	
MCS-Uniform-Send-Data (envoi uniforme de données)		✓	
MCS-Token-Grab (saisie de jeton)		✓	
MCS-Token-Inhibit (interdiction de jeton)		✓	

TABLEAU 1/T.121 (*fin*)**Primitives utilisées par un ARM ou un ASE**

Primitive	ARM	ASE	Contrôleur nodal
MCS-Token-Give (cession de jeton)		✓	
MCS-Token-Please (demande de jeton)		✓	
MCS-Token-Release (libération de jeton)		✓	
MCS-Token-Test (test de jeton)		✓	

6 Modèle générique d'application

Le modèle générique d'application (GAT) est conçu pour fournir une fonctionnalité générique de gestion de ressources aux protocoles d'application, tant normalisés que hors norme. En soi, il comporte un gestionnaire de ressources d'application (l'ARM générique) mais ne comporte pas d'élément de service d'application (ASE) – voir la Figure 5. Les spécifications de protocoles d'application distinctes détaillent le fonctionnement de leurs ASE respectifs et identifient, parmi les services offerts par l'ARM générique, ceux qui sont requis par les ASE. Elles n'ont pas à décrire le fonctionnement de l'ARM, mais devraient en revanche faire référence à la présente Recommandation. L'ARM générique fournit tous les services décrits au 5.3.



T0825040-95/d05

FIGURE 5/T.121

Modèle générique d'application

Une description des fonctionnalités offertes par un ARM générique est proposée ci-après.

6.1 Initialisation

Un ARM est responsable de l'inscription de son entité protocolaire d'application pour le compte de l'application d'utilisateur, et de l'obtention des ressources nécessaires requises par l'ASE. Pour chaque protocole d'application employé, l'application d'utilisateur doit spécifier, à l'ARM correspondant, les paramètres suivants:

- si elle veut ou non s'inscrire (aucun autre paramètre n'est requis si l'application d'utilisateur ne veut pas s'inscrire);
- la clé du protocole d'application;
- le type de la session (session d'enregistrement, de base normalisée, de base non normalisée, publique ou privée);
- l'identificateur de session (*Session ID*) (omis en cas de déclaration de prise en charge d'un protocole à l'occasion de la session d'enregistrement, d'inscription à une session de base non normalisée ou de création d'une nouvelle session publique ou privée);
- la liste des jetons dynamiques nécessaires;
- la liste des canaux assignés nécessaires;
- la liste des canaux privés nécessaires (et à titre facultatif, des nœuds à inviter sur ces canaux);
- si elle est à même d'agir en tant que président de la session de l'application ou non (s'il y a plus d'une application sur un site participant à la même session, le fournisseur GCC local devra déterminer laquelle deviendra présidente);
- la liste des capacités d'application prises en charge (comme défini dans la spécification du protocole d'application). Les capacités fédérables devront être omises pour la session d'enregistrement mais devront être incluses pour toutes les autres sessions.

Le Tableau 2 précise les paramètres qui sont transmis à l'ARM pour chaque protocole d'application employé par l'application d'utilisateur.

6.2 Inscription

Pour participer à une conférence, une application d'utilisateur requiert une APE (consistant en un ARM et un ASE) pour chaque protocole qu'elle emploie. La méthode de création des APE est une affaire locale, hors du domaine d'application de la présente Recommandation. Chaque APE doit donc, à son tour, établir un GCC SAP qui lui permettra de communiquer avec le fournisseur de GCC à ce nœud; encore une fois, la façon dont on y parvient relève de la compétence locale. De même, chaque APE doit établir un MCSAP pour lui permettre de communiquer avec son fournisseur MCS local.

Lorsque le nœud se branche sur une conférence ou lorsqu'une APE établit un GCC SAP à la suite de son branchement sur une ou plusieurs conférences, le fournisseur local de GCC le notifie à toutes les APE de ce nœud en émettant une indication GCC-Application-Permission-To-Enroll (autorisation d'inscription à une application) avec Grant/Revoke (indicateur d'octroi) positionné sur Grant (octroi). Pour chaque protocole employé par l'application d'utilisateur, l'ARM correspondant émettra alors une demande GCC-Application-Enroll (inscription d'application), que l'application d'utilisateur désire ou non s'inscrire à ce moment. Si l'application d'utilisateur ne désire pas inscrire une APE, l'ARM correspondant précisera l'identificateur de conférence et positionnera Enroll/Un-enroll (indicateur d'inscription) sur Un-enroll (non-inscription) dans la demande GCC-Application-Enroll (inscription d'application). Aucun autre paramètre n'est requis. L'application d'utilisateur peut donner ordre à son ou à ses ARM de s'inscrire par la suite et à tout moment, à moins que l'autorisation ne soit retirée par la réception d'une indication GCC-Application-Permission-To-Enroll avec Grant/Revoke positionné sur Revoke (retrait).

Une application d'utilisateur peut vouloir recevoir des informations sur toutes les sessions en cours avant de décider si elle participera à une session existante ou en créera une nouvelle. Elle peut le faire en donnant ordre à son ou ses ARM de s'inscrire à l'état inactif dans la session d'enregistrement, comme décrit au 6.2.1.

Si l'application d'utilisateur veut avertir de sa présence les applications homologues sur d'autres nœuds mais ne veut pas participer immédiatement à une session, elle peut ordonner à son ARM de s'inscrire à l'état inactif dans la session d'enregistrement sans MCS User ID (identificateur d'utilisateur MCS). Cela permet à l'application d'utilisateur de déclarer la prise en charge d'un ou plusieurs protocoles sans consommer de ressources MCS.

TABLEAU 2/T.121

Paramètres d'une APE

Paramètre	Description
Enroll/Un-enroll (indicateur d'inscription)	Ce paramètre est positionné à ENROLL (inscription) si l'application d'utilisateur désire s'inscrire immédiatement, et à UN-ENROLL (non-inscription) si l'application d'utilisateur désire retirer l'inscription. S'il est positionné à UN-ENROLL, aucun autre paramètre n'est requis.
Application-protocol-key (clé de protocole d'application)	La clé de protocole d'application identifie le protocole d'application utilisé dans une session de protocole d'application. De multiples sessions du protocole d'application ayant le même objet sont identifiées en utilisant la même clé de protocole d'application (mais des identificateurs de session différents). Une clé de protocole d'application est soit un identificateur d'objet ASN.1 appartenant à une Recommandation, à une Norme ou à un protocole hors norme, soit un identificateur non normalisé utilisant les conventions de codage de la Recommandation T.124 (laquelle utilise les indicatifs de pays de T.35 comme H.221).
SessionType (type de session)	<p>Ce paramètre peut prendre l'une des cinq valeurs suivantes:</p> <p><i>enregistrement</i>: la session d'enregistrement permet, à une application d'utilisateur, d'annoncer sa présence à la conférence et offre aussi un moyen de localiser la session de base non normalisée. Aucun identificateur de session (<i>Session ID</i>) n'est requis;</p> <p><i>de base normalisée</i>: cette valeur indique que l'ARM devra s'inscrire en utilisant une clé de session composée de la clé du protocole d'application et du paramètre sessionID (identificateur de session). Il utilisera le ou les canaux statiques prédéfinis et le(s) jeton(s) statique(s) tels que définis dans les spécifications du protocole d'application, et tels qu'assignés dans la Recommandation T.120 comme ensemble de ressources de base. Ce type de session n'est valable que pour des protocoles d'application normalisés;</p> <p><i>de base non normalisée</i>: cette valeur indique que l'ARM devrait en premier lieu s'inscrire à la session d'enregistrement pour déterminer l'identificateur de session (sessionID) assigné à la session de base non normalisée en consultant le référentiel GCC (GCC Registry). Si un identificateur de session n'existe pas déjà, il en est assigné un par l'ARM. L'ARM devra alors s'inscrire à la session de base non normalisée en utilisant une clé de session (Session Key) composée de la clé du protocole d'application et du paramètre sessionID. Les membres d'une session de base non normalisée peuvent déterminer les identificateurs de jeton et de canal au moyen du référentiel GCC. Ce type de session n'est valable que pour des protocoles d'application non normalisés;</p> <p><i>publique</i>: cette valeur indique que l'ARM devra s'inscrire en utilisant une clé de session composée de la clé du protocole d'application et du paramètre sessionID. Si le paramètre sessionID est omis, l'ARM en attribue un. Toutes les ressources en canaux et en jetons sont dynamiques et sont attribuées par le créateur de la session publique en utilisant les mécanismes MCS-Channel-Join (branchement sur le canal MCS) et GCC-Registry-Assign-Token (attribution de jeton), respectivement. Les participants à une session publique peuvent déterminer les identificateurs de jeton et de canal au moyen du référentiel GCC;</p> <p><i>privée</i>: cette valeur indique que l'ARM devra s'inscrire en utilisant une clé de session composée de la clé de protocole d'application et du paramètre sessionID. Si le paramètre sessionID est omis, l'ARM en attribue un. Toutes les ressources en canaux et en jetons sont dynamiques et sont attribuées par l'ARM de l'organisateur privé en utilisant les mécanismes MCS-Channel-Convene (constitution de canal d'invitation) et GCC-Registry-Assign-Token (attribution de jeton), respectivement. L'ARM qui crée la session peut alors admettre, sur le ou les canaux, des APE homologues, aux nœuds dont les identificateurs nœuds GCC (GCC Node Id) figurent dans le paramètre protocolaire admitList (liste des admis).</p> <p>Un ARM de participant privé (private Member ARM) doit attendre que son APE soit admise par l'organisateur des canaux privés MCS avant d'essayer de se brancher sur ceux-ci.</p>
SessionID (identificateur de session)	<p>Ce paramètre est utilisé pour différencier les ressources utilisées par de multiples sessions du protocole qui peuvent exister simultanément dans le même domaine MCS.</p> <p>Le paramètre sessionID doit être précisé si l'application d'utilisateur désire prendre part à une session de base normalisée ou une session existante, publique ou privée. Il est omis si l'application d'utilisateur veut créer une nouvelle session privée ou publique, participer à une session de base non normalisée, ou si l'application d'utilisateur désire s'inscrire à la session d'enregistrement sans consommer de ressources MCS. Exception faite de la session d'enregistrement, l'identificateur de canal MCS (<i>MCS Channel ID</i>) assigné comme canal de démarrage est utilisé comme sessionID, puisqu'il est garanti unique dans le domaine de la conférence.</p>

TABLEAU 2/T.121 (*fin*)**Paramètres d'une APE**

Paramètre	Description
AdmitList (liste des admis)	SessionType (type de session) = private (privé) et sessionID (identificateur de session) omis liste des identificateurs nodaux GCC (<i>GCC NodeID</i>) correspondant aux nœuds où les APE homologues devront être admises sur les canaux organisés en mode privé. Sinon omis
ResourceList (liste des ressources)	C'est une liste de ressources (canaux et jetons) requises par l'application d'utilisateur. SessionType (type de session) = de base normalisée: ResourceList se compose des identificateurs MCS de canaux statiques sur lesquels se branchera l'APE et des jetons statiques à utiliser par l'APE. Elle peut, facultativement, contenir une liste d'identificateurs de ressource (<i>Resource Id</i>) pour déterminer les identificateurs MCS de jetons et canaux dynamiques au moyen du référentiel GCC. Ce paramètre peut aussi spécifier le nombre requis de poignées de référentiel (<i>Registry handles</i>) et inclure une liste d'identificateurs de ressource pour les paramètres de référentiel ainsi que la valeur de ces paramètres (si opportun). SessionType = publique, privée ou de base non normalisée: ResourceList se compose d'une liste d'identificateurs de ressource pour déterminer les identificateurs MCS de jetons et canaux dynamiques au moyen du référentiel GCC. Il peut aussi spécifier le nombre requis de poignées de référentiel (<i>Registry handles</i>) et inclure une liste d'identificateurs de ressource pour les paramètres de référentiel ainsi que la valeur de ces paramètres (si opportun). Si l'identificateur de session est omis (c'est-à-dire si l'APE est organisateur de session), l'ARM est chargé d'attribuer et d'enregistrer tous les canaux et jetons précisés dans la liste des ressources. Si l'identificateur de session est présent (c'est-à-dire si l'APE est membre de la session), l'ARM est chargé de rechercher tous les canaux et jetons. Dans les deux cas, l'identificateur de ressource et le MCS Channel ID (identificateur de canal MCS) ou le Token ID (identificateur de jeton) sont transmis par l'ARM à son ASE. SessionType = enregistrement omis
Conducting Operation Capable (capacité d'opération de présidence)	Positionné à vrai si l'APE est capable de devenir président de la session en mode présidé.
Non-Collapsing Capabilities List (liste de capacités non fédérables)	Toute capacité non fédérable spécifiée par le protocole d'application et prise en charge par l'ASE.
Application Capability List (liste de capacités d'application)	Toute capacité fédérable spécifiée par le protocole d'application et prise en charge par l'ASE.
Expected Capability List (liste de capacités prévues)	SessionType (type de session) = privée et sessionID (identificateur de session) omis: indique toutes les capacités d'application qui doivent être prises en charge par les APE appelées, permettant au créateur d'une session privée de préciser un ensemble de capacités minimales pour cette session. Sinon omis.

Lorsque l'application décide de devenir active, chaque ARM doit adopter un identificateur d'utilisateur MCS (MCS User ID). Un nouvel MCS User ID est acquis par l'ARM en émettant une demande MCS-Attach-User (rattachement d'utilisateur) vers le fournisseur MCS, en utilisant l'indication GCC-Application-Permission-To-Enroll (autorisation d'inscription) comme Domain Selector (sélecteur de domaine). Sur réception d'une confirmation de succès de MCS-Attach-User (rattachement d'utilisateur) en réponse, un ARM se branchera sur le canal User ID (identificateur d'utilisateur) indiqué, en émettant une demande MCS-Channel-Join (branchement sur le canal MCS), et fournira User ID à l'ASE correspondant.

Les opérations consécutives sont fonction du type de session. On remarquera que les processus décrits doivent être répétés pour chaque APE employé par une application d'utilisateur. Il faut observer qu'une caractéristique commune aux squelettes du modèle (à l'exception des squelettes de session d'enregistrement) est que le canal Session ID (identificateur de session) est le canal de démarrage, que son type détermine le type de la session, et qu'il reçoit toujours les branchements pendant la séquence de déclenchement d'une APE.

Les procédures définies pour l'inscription à une session et pour le branchement sur le canal de démarrage doivent être suivies dans l'ordre spécifié pour chacun des types de session décrits. Si un protocole d'application ne requiert l'utilisation d'aucune autre ressource (canaux, jetons ou paramètres de référentiel), un ARM peut omettre l'opération d'inscription à l'état inactif. Là où un protocole d'application utilise des jetons, des paramètres de référentiel ou des canaux dynamiques supplémentaires, la présente Recommandation n'impose aucune exigence relative à l'ordre d'attribution ou d'identification de ces ressources. Si un protocole d'application utilise des poignées (*handles*) GCC, celles-ci peuvent être allouées pendant le processus d'inscription d'application et, par la suite, à tout moment de la durée utile d'une session.

Afin de réduire au minimum la période d'initialisation, il est permis d'identifier parallèlement de multiples canaux en entrelaçant les opérations requises pour chaque canal. Par exemple, un créateur de session pourra choisir de constituer tous les canaux d'invitation nécessaires puis d'enregistrer tous ces canaux, plutôt que de constituer et d'enregistrer chaque canal un par un. Pour tout canal donné, cependant, la séquence d'opérations définie dans la présente Recommandation devra être respectée.

Une APE ne peut être considérée comme faisant partie d'une session que lorsque son ARM reçoit une indication de compte rendu de répertoire d'application GCC (*GCC-Application-Roster-Report*) dans laquelle son identificateur d'entité (*Entity ID*) est présent. Une APE ne devra pas soumettre de données à une session avant de recevoir une telle indication de compte rendu de répertoire. Toutefois, les APE devraient être prêtes à recevoir dans une mémoire tampon toutes les données reçues pour une session entre l'inscription à l'état actif dans une session et la réception d'un compte rendu de répertoire où l'identificateur d'entité APE (*APE Entity ID*) est présent. Cela est dû à ce qu'une autre APE peut avoir reçu une indication de compte rendu de répertoire contenant le nouveau participant et pourrait soumettre, à la session, des données qui atteindraient le nouveau participant avant le compte rendu de répertoire correspondant. Les protocoles d'application peuvent spécifier comment doivent être traitées ces données tamponnées.

6.2.1 Session d'enregistrement

Si l'application d'utilisateur désire avertir de sa présence les applications homologues sur d'autres nœuds mais ne désire pas prendre immédiatement part à une session, elle peut donner ordre à son ARM de s'inscrire à la session d'enregistrement sans identificateur d'utilisateur MCS (*MCS user ID*). Cela permet à l'application d'utilisateur de déclarer la prise en charge d'un ou plusieurs protocoles d'application sans consommer de ressources MCS. Elle reçoit aussi des informations sur toutes les sessions en cours pour chacun des protocoles d'application annoncés. Chaque ARM émet une demande d'inscription d'application GCC (*GCC-Application-Enroll*) avec l'indicateur d'inscription (*Enroll/Un-enroll*) mis à «inscription» (*Enroll*) et l'indicateur d'activité (*active/inactive*) mis à «inactif» (*inactive*), en spécifiant la clé de session (*Session Key*) de son protocole correspondant sans identificateur de session (*Session ID*). Voir la Figure 6.

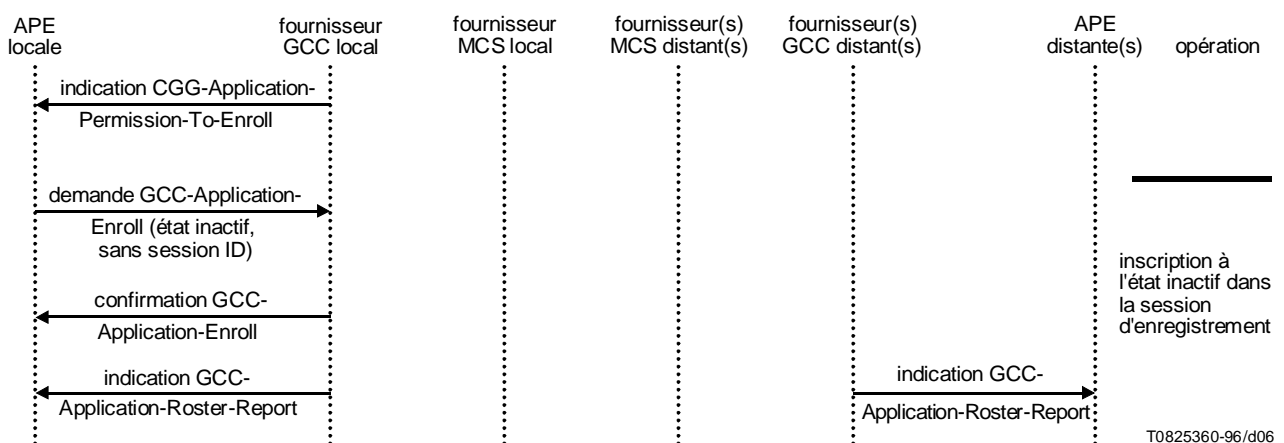


FIGURE 6/T.121

Inscription à la session d'enregistrement pour déclarer la prise en charge d'un protocole

6.2.2 Session de base normalisée

Après s'être branché sur son canal d'identificateur d'utilisateur (*User ID Channel*), l'ARM se branchera sur des canaux statiques comme établi par les spécifications du protocole d'application correspondant. Une fois reçue la confirmation positive du branchement sur ces canaux, l'ARM inscrira l'APE à l'état inactif en émettant une demande GCC-Application-Enroll (inscription d'application) vers le fournisseur GCC, avec les paramètres précisés dans le Tableau 3.

Après avoir reçu une indication de compte rendu de répertoire d'application (*GCC-Application-Roster-Report*) avec une entrée correspondant à son APE, l'ARM devra attribuer tout canal dynamique assigné obligatoire en vertu des spécifications du protocole. L'ARM devra émettre une primitive de demande de recherche d'entrée de référentiel (*GCC-Registry-Retrieve-Entry*) pour chaque canal de cette nature, afin de déterminer s'il a déjà été enregistré ou non. Si le paramètre de résultat renvoyé dans la confirmation de *GCC-Registry-Retrieve-Entry* est «succès» (*«successful»*), l'ARM tentera de se brancher sur le canal indiqué dans le paramètre «élément de référentiel» (*registry item*) en émettra une demande de branchement *MCS-Channel-Join*. Si le paramètre de résultat renvoyé dans *GCC-Registry-Retrieve-Entry* a la valeur «entrée non trouvée» (*«entry not found»*), l'ARM tentera d'attribuer un nouveau canal en émettant une demande de branchement *MCS-Channel-Join* où l'identificateur de canal (*channel ID*) sera égal à 0. La confirmation de branchement *MCS-Channel-Join* renvoyée, en cas de succès, contient l'identificateur du canal (*channel ID*) assigné, que l'ARM tentera d'enregistrer dans la session de base normalisée en émettant une demande d'enregistrement de canal (*GCC-Registry-Register-Channel*). A la réception de la confirmation d'enregistrement, l'ARM devra examiner le paramètre de résultat. Si le résultat est «succès», l'ARM ne prendra aucune autre mesure. Si le résultat est «l'indice existe déjà» (*«index already exists»*), c'est que le canal a déjà été enregistré par une autre APE; son identificateur de canal est fourni dans le paramètre «élément de référentiel». Dans ce cas, l'ARM devra, en premier lieu, quitter le canal qu'il avait précédemment assigné, en émettant une demande de sortie de canal (*MCS-Channel-Leave*) spécifiant l'identificateur de canal renvoyé dans la confirmation de branchement (*MCS-Channel-Join*). Il se branchera ensuite sur le canal indiqué dans le paramètre «élément de référentiel» en émettant une demande de branchement (*MCS-Channel-Join*).

Si le protocole d'application rend obligatoire l'identification, par une APE, de tout jeton dynamique avant de s'interconnecter avec ses homologues, l'ARM peut déterminer leur identité au moyen du référentiel GCC (*GCC Registry*). Pour chaque identificateur de ressource de jeton (*Token Resource ID*) spécifié par le protocole d'application, l'ARM émettra une demande d'attribution de jeton (*GCC-Registry-Assign-Token*) au fournisseur GCC en utilisant les paramètres spécifiés dans le Tableau 7. Si le paramètre de résultat renvoyé dans la confirmation d'attribution de jeton a la valeur «succès» ou la valeur «l'indice existe déjà», l'identificateur de jeton (*Token ID*) contenu dans l'élément de référentiel de la primitive de confirmation renvoyée est utilisé comme identificateur de jeton pour le jeton correspondant à l'identificateur de ressource (*Resource ID*) utilisé dans la clé de référentiel. L'ARM transmettra à son ASE l'identificateur de ressource et l'identificateur de jeton associé.

Si le protocole d'application exige qu'une APE attribue ou identifie les paramètres de référentiel avant de s'interconnecter avec ses homologues, l'ARM devra émettre une demande de fixation de paramètre (*GCC-Registry-Set-Parameter*) vers le fournisseur GCC.

Une fois que toutes les ressources nécessaires à la session auront été identifiées, l'ARM s'inscrira à l'état actif en émettant une demande d'inscription d'application (*GCC-Application-Enroll*). L'indicateur d'activité sera positionné sur «actif» (*Active*), l'identificateur de session (*Session ID*) sera précisé comme partie de la clé de session (*Session Key*), le canal de démarrage sera spécifié statique (*Static*) et les listes de capacités seront fournies. Voir les Figures 7 et 8.

6.2.3 Session de base non normalisée

La session d'enregistrement peut être utilisée pour fournir l'accès à une session de base non normalisée. Après le branchement sur son canal d'identificateur d'utilisateur (*User ID Channel*), l'ARM s'inscrira dans la session d'enregistrement en émettant une demande d'inscription (*GCC-Application-Enroll*) avec l'indicateur d'activité (*Active/Inactive*) mis à «inactif» et en spécifiant la clé de session (*Session Key*) sans aucun identificateur de session.

Après avoir reçu une indication de compte rendu (*GCC-Application-Roster-Report*) avec une entrée correspondant à son APE, l'ARM tentera de créer la session de base non normalisée en émettant tout d'abord une demande de branchement (*MCS-Channel-Join*) où l'identificateur de canal (*Channel ID*) a la valeur 0. La confirmation de branchement renvoyée, en cas de succès, contient l'identificateur de canal assigné que l'ARM tentera d'enregistrer, au cours de la session d'enregistrement, comme canal de démarrage pour la session de base non normalisée. L'ARM émettra une demande d'enregistrement de canal (*GCC-Registry-Register-Channel*) utilisant une clé de référentiel (*Registry Key*) constituée de la clé de session (sans identificateur de session) et de l'identificateur de ressource (*Resource ID*) «de base» (*«BASE»*) codés comme quatre octets consécutifs conformément à la Recommandation T.50. A la réception de la confirmation de l'enregistrement, l'ARM examinera le paramètre de résultat.

Si le résultat est «l'indice existe déjà» («*index already exists*»), c'est que la session de base non normalisée a déjà été enregistrée par une autre APE; son identificateur de session est fourni dans le paramètre «élément de référentiel» (*Registry Item*). Dans ce cas, l'ARM quittera d'abord le canal qu'il avait précédemment assigné, en émettant une demande de sortie de canal (*MCS-Channel-Leave*) spécifiant l'identificateur de canal renvoyé dans la confirmation de branchement. Puis l'ARM assumera le rôle de membre de session et s'inscrira à la session de base existante en suivant le processus décrit pour un membre de session publique au 6.2.4. Ce processus est représenté dans la Figure 9. Une tentative de branchement sur le canal de démarrage identifié dans la confirmation d'enregistrement de canal (*GCC Registry-Register-Channel*) peut échouer si le paramètre «élément de référentiel» sur une session non normalisée qui n'existe plus (c'est-à-dire si le canal de démarrage de la session n'existe plus); dans telles circonstances, l'ARM pourra tenter de réutiliser cette session qui n'existe plus, en émettant une demande d'annulation d'entrée (*GCC-Registry-Delete-Entry*) utilisant la même clé de référentiel que précédemment. A la réception de la confirmation d'annulation d'entrée correspondante, l'ARM examinera le paramètre de résultat et, s'il a la valeur «succès» («*successful*»), l'ARM pourra répéter le processus de création d'une session de base non normalisée.

Si le paramètre de résultat de la confirmation d'enregistrement (*GCC-Registry-Register-Channel*) est «succès» («*successful*»), l'ARM assumera la responsabilité de créateur pour la session de base non normalisée comme représenté dans la Figure 10. Il devra s'inscrire à l'état inactif à la nouvelle session de base comme spécifié dans le Tableau 3.

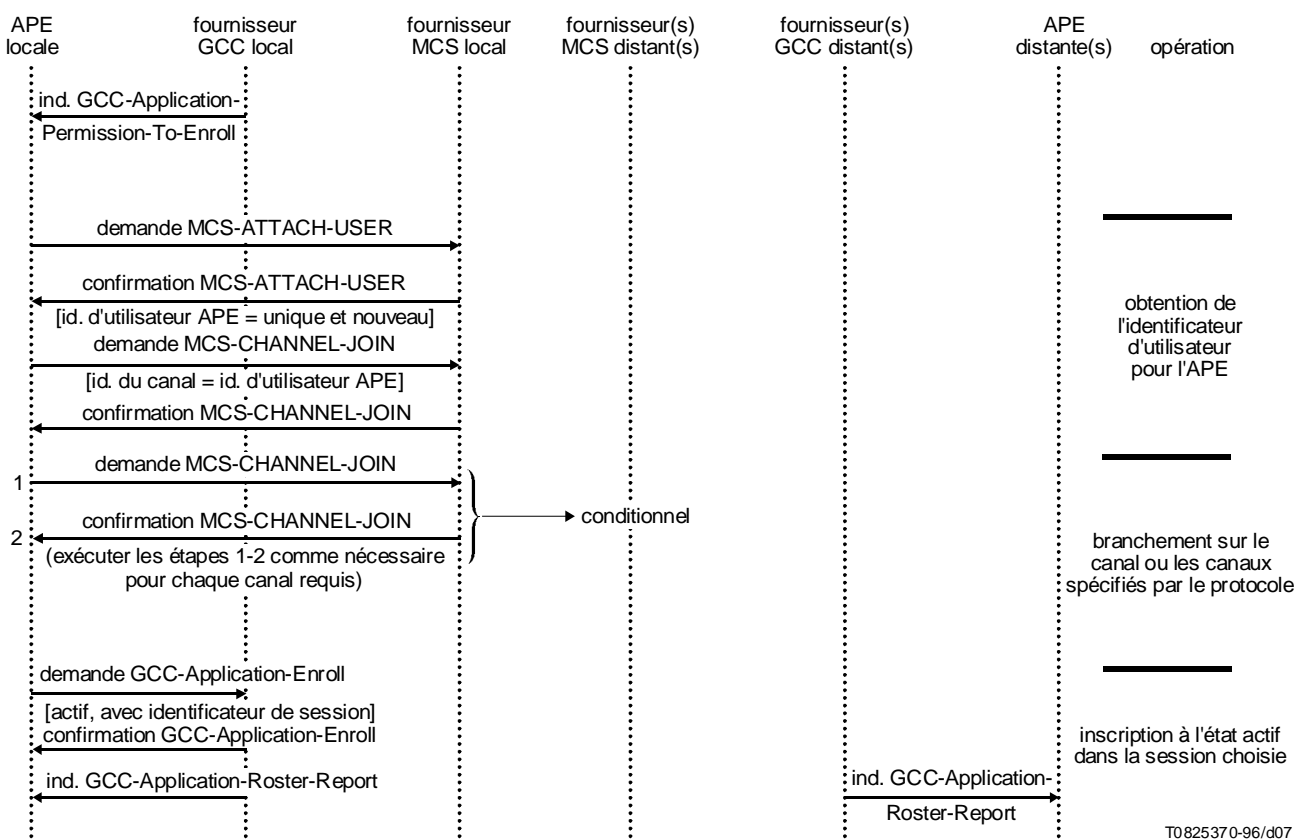


FIGURE 7/T.121

**Exemple de séquence de déclenchement de protocole de session de base normalisée –
Cas n° 1: utilisation de ressources exclusivement statiques**

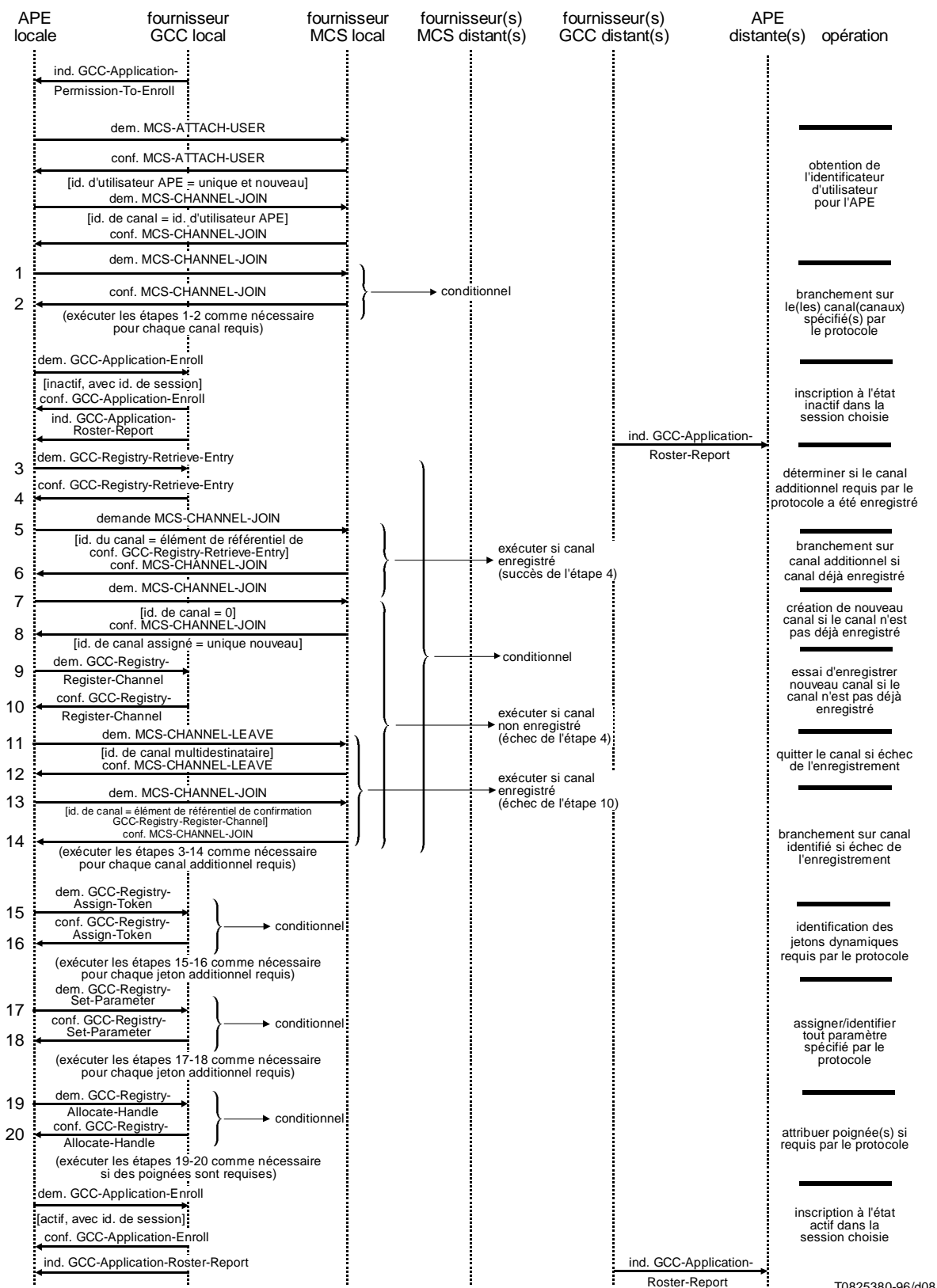


FIGURE 8/T.121

Exemple de séquence de déclenchement de protocole de session de base normalisée –
Cas n° 2: utilisation de ressources statiques et dynamiques

TABLEAU 3/T.121

Paramètres pour la demande GCC-Application-Enroll (inscription d'application)

Paramètre	Contenu		
Conference ID (identificateur de conférence)	Fourni par l'indication GCC-Application-Permission-To-Enroll (autorisation d'inscription à une application).		
Session Key (clé de session)	Composé de la clé du protocole d'application et de Session ID (identificateur de session).		
Application User ID (identificateur d'utilisateur d'application)	Fourni par la confirmation MCS-Attach-User (rattachement d'utilisateur).		
Active-Inactive (indicateur d'activité)	Actif lorsqu'il indique que l'ARM a identifié l'ensemble des ressources qui doivent être identifiées dans l'exécution du processus d'inscription et s'est branché sur tous les canaux requis par le protocole. Inactif pour la session d'enregistrement et dans le cas des sessions de base normalisée, de base non normalisée, publique ou privée quand l'inscription précède l'identification de l'ensemble des ressources qui doivent être identifiées dans l'exécution du processus d'inscription.		
Conducting Operation Flag (indicateur de présidence)	Positionné sur vrai si l'APE est à même de devenir le président de session en mode présidé. Cet indicateur ne doit pas être positionné si l'indicateur d'activité (<i>Active/Inactive</i>) est positionné sur <i>Inactive</i> (état inactif).		
Start-Up Channel (canal de démarrage)	Paramètre optionnel qui spécifie comme suit le type de session: <table border="1"> <tr> <td>Type de session: <i>enregistrement</i> <i>de base normalisée</i> <i>de base non normalisée</i> <i>publique</i> <i>privée</i></td><td>Canal de démarrage: <i>omis</i> <i>statique</i> <i>multidestinataire dynamique</i> <i>multidestinataire dynamique</i> <i>privé dynamique</i></td></tr> </table>	Type de session: <i>enregistrement</i> <i>de base normalisée</i> <i>de base non normalisée</i> <i>publique</i> <i>privée</i>	Canal de démarrage: <i>omis</i> <i>statique</i> <i>multidestinataire dynamique</i> <i>multidestinataire dynamique</i> <i>privé dynamique</i>
Type de session: <i>enregistrement</i> <i>de base normalisée</i> <i>de base non normalisée</i> <i>publique</i> <i>privée</i>	Canal de démarrage: <i>omis</i> <i>statique</i> <i>multidestinataire dynamique</i> <i>multidestinataire dynamique</i> <i>privé dynamique</i>		
Non-Collapsing Capabilities List (liste de capacités non fédérables)	Toutes les capacités non fédérables spécifiées par le protocole d'application et prises en charge par l'ASE.		
Application Capability List (liste de capacités d'application)	Toutes les capacités fédérables spécifiées par le protocole d'application et prises en charge par l'ASE. Omis si l'indicateur d'activité (<i>Active/Inactive</i>) est positionné sur <i>Inactive</i> (état inactif).		
Enroll-Un-enroll (indicateur d'inscription)	ENROLL (inscription)		

Si des canaux additionnels sont obligatoires en vertu des spécifications du protocole, l'ARM émettra une primitive de demande de recherche d'entrée (*GCC-Registry-Retrieve-Entry*) pour chacun de ces canaux afin de déterminer s'il a déjà été enregistré. La clé de référentiel (*Registry Key*) utilisée pour enregistrer ces ressources devra inclure l'identificateur de session (*Session ID*) de la session de base non normalisée comme partie de la clé de session. Si le paramètre de résultat renvoyé dans la confirmation de recherche d'entrée a la valeur «succès» (*«successful»*), l'ARM tentera de se brancher sur le canal indiqué dans le paramètre élément de référentiel (*Registry Item*) en émettant une demande de branchement (*MCS-Channel-Join*). Si le paramètre de résultat renvoyé dans la recherche d'entrée est «entrée non trouvée» (*«entry not found»*), l'ARM essaiera d'assigner un nouveau canal en émettant une demande de branchement avec l'identificateur de canal égal à 0. La confirmation de branchement renvoyée, en cas de succès, contient l'identificateur du canal assigné, que l'ARM tentera d'enregistrer dans la session de base non normalisée en émettant une demande d'enregistrement de canal (*GCC-Registry-Register-Channel*). A la réception de la confirmation de l'enregistrement du canal, l'ARM examinera le paramètre de résultat. Si le résultat est «succès», l'ARM ne prendra aucune autre mesure. Si le résultat est «l'indice existe déjà» (*«index already exists»*), c'est que le canal a déjà été enregistré par une autre APE; son identificateur de canal est fourni dans le paramètre élément de référentiel. Dans ce cas, l'ARM devra d'abord quitter le canal qu'il avait précédemment assigné, en émettant une demande de sortie de canal (*MCS-Channel-Leave*) spécifiant l'identificateur de canal renvoyé dans la confirmation de branchement. Il se branchera ensuite sur le canal indiqué dans le paramètre élément de référentiel en émettant une demande de branchement (*MCS-Channel-Join*).

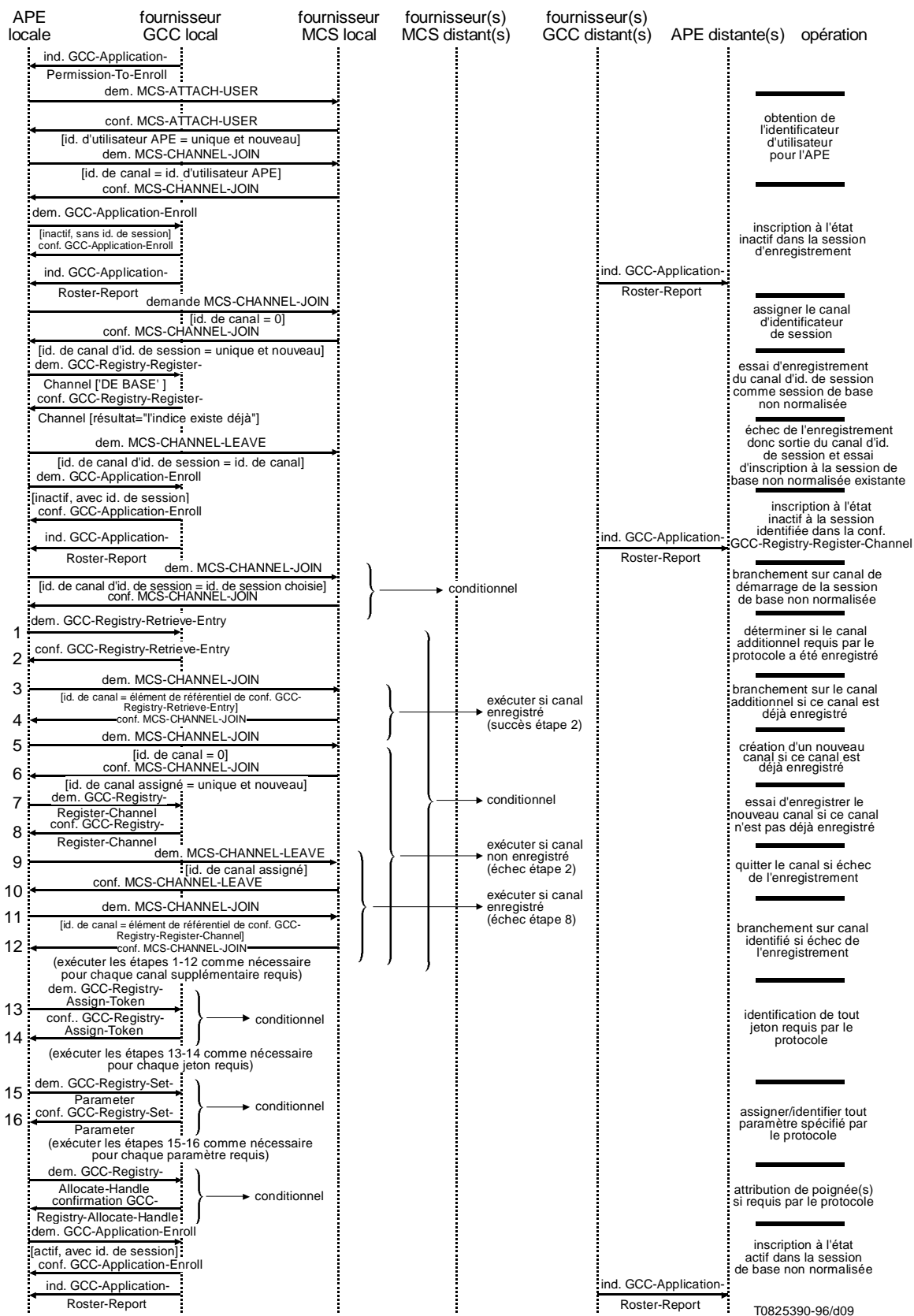


FIGURE 9/T.121

Exemple de séquence de déclenchement de protocole de session de base non normalisée –
Cas n° 1: session de base déjà enregistrée

Si le protocole d'application impose qu'une APE connaisse l'identité des jetons avant de s'interconnecter avec ses homologues, l'ARM pourra déterminer leur identité par le référentiel GCC. Pour chaque identificateur de ressource de jeton spécifié par le protocole d'application, l'ARM émettra une demande d'attribution de jeton (*GCC-Registry-Assign-Token*) vers le fournisseur GCC en utilisant les paramètres spécifiés dans le Tableau 7. Si le paramètre de résultat renvoyé dans la confirmation d'attribution de jeton est «succès» ou «l'indice existe déjà», l'identificateur de jeton (*Token ID*) contenu dans l'élément de référentiel de la primitive de confirmation renvoyée est utilisé comme identificateur de jeton pour le jeton correspondant à l'identificateur de ressource (*Resource ID*) utilisé dans la clé de référentiel. L'ARM transmettra, à son ASE, l'identificateur de ressource et l'identificateur de jeton qui lui est associé.

Si le protocole d'application exige qu'une APE assigne ou identifie les paramètres de référentiel avant de s'interconnecter avec ses homologues, l'ARM devra émettre une demande de fixation de paramètre (*GCC-Registry-Set-Parameter*) vers le fournisseur GCC.

Une fois que toutes les ressources requises pour la session auront été identifiées, l'ARM s'inscrira à l'état actif en émettant une demande d'inscription d'application (*GCC-Application-Enroll*) avec les paramètres spécifiés dans le Tableau 3.

6.2.4 Session publique

Après s'être branché sur son canal d'identificateur d'utilisateur (*User ID Channel*), l'ARM examinera le paramètre *sessionID* (identificateur de session) pour décider, s'il est présent, si elle participera à une session publique existante (comme membre public) ou, s'il est omis, d'en créer une nouvelle (comme créateur public). Voir les Figures 11 et 12.

Si le paramètre *sessionID* est présent, l'ARM s'inscrira dans la session indiquée en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec l'indicateur *Active/Inactive* (indicateur d'activité) positionné sur *Inactive* (état inactif), et en précisant la clé de session avec l'identificateur de session requis. Il émettra alors une demande *MCS-Channel-Join* (branchement sur le canal MCS), précisant l'identificateur de session de la session choisie comme paramètre d'identificateur de canal.

Si le protocole d'application nécessite le branchement sur de multiples canaux avant qu'une APE puisse être considérée comme participante à l'état actif dans une session, alors l'ARM tentera d'identifier ces canaux et de se brancher sur eux en utilisant les services du référentiel GCC et du MCS, respectivement. Il est possible de se brancher sur des canaux additionnels quand la session d'application est en cours. De même, si un protocole d'application exige qu'une APE connaisse l'identité des jetons ou les paramètres du référentiel avant de dialoguer avec ses homologues, l'ARM peut déterminer leur identité à travers le référentiel GCC.

Une fois que toutes les ressources nécessaires à la session auront été identifiées, l'ARM devra s'inscrire à l'état actif dans la session élue en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec les paramètres précisés dans le Tableau 3.

Si le paramètre *sessionID* est omis, l'ARM s'inscrira dans une nouvelle session publique. L'ARM devra d'abord attribuer à la session un canal assigné en émettant une demande *MCS-Channel-Join* (branchement sur le canal MCS) où *Channel ID* (identificateur de canal) = 0. La confirmation *MCS-Channel-Join* renvoyée, s'il y a succès, contient le *Channel ID* attribué. Ce *Channel ID* est aussi utilisé comme identificateur de session pour l'APE. L'ARM s'inscrira alors à l'état inactif dans cette session en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec les paramètres précisés dans le Tableau 3.

La responsabilité d'acquérir l'ensemble de ressources qu'il est nécessaire d'identifier comme partie du processus d'inscription à une session publique incombe à l'ARM qui a créé cette session. Si la spécification de protocole exige des jetons, des paramètres de référentiel ou des canaux supplémentaires, l'ARM devra les attribuer et les enregistrer auprès de la GCC comme nécessaire. La clé de référentiel utilisée pour enregistrer ces ressources comprendra l'identificateur de session comme partie de la clé de session. On remarquera que l'ARM ne doit pas essayer d'utiliser le *GCC-Registry* (référentiel GCC) tant qu'il n'a pas reçu une indication *GCC-Application-Roster-Report* (compte rendu de répertoire d'application) contenant une entrée correspondant à sa propre APE.

Une fois que toutes les ressources ont été attribuées avec succès, l'ARM s'inscrira à l'état actif en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec les paramètres précisés dans le Tableau 3.

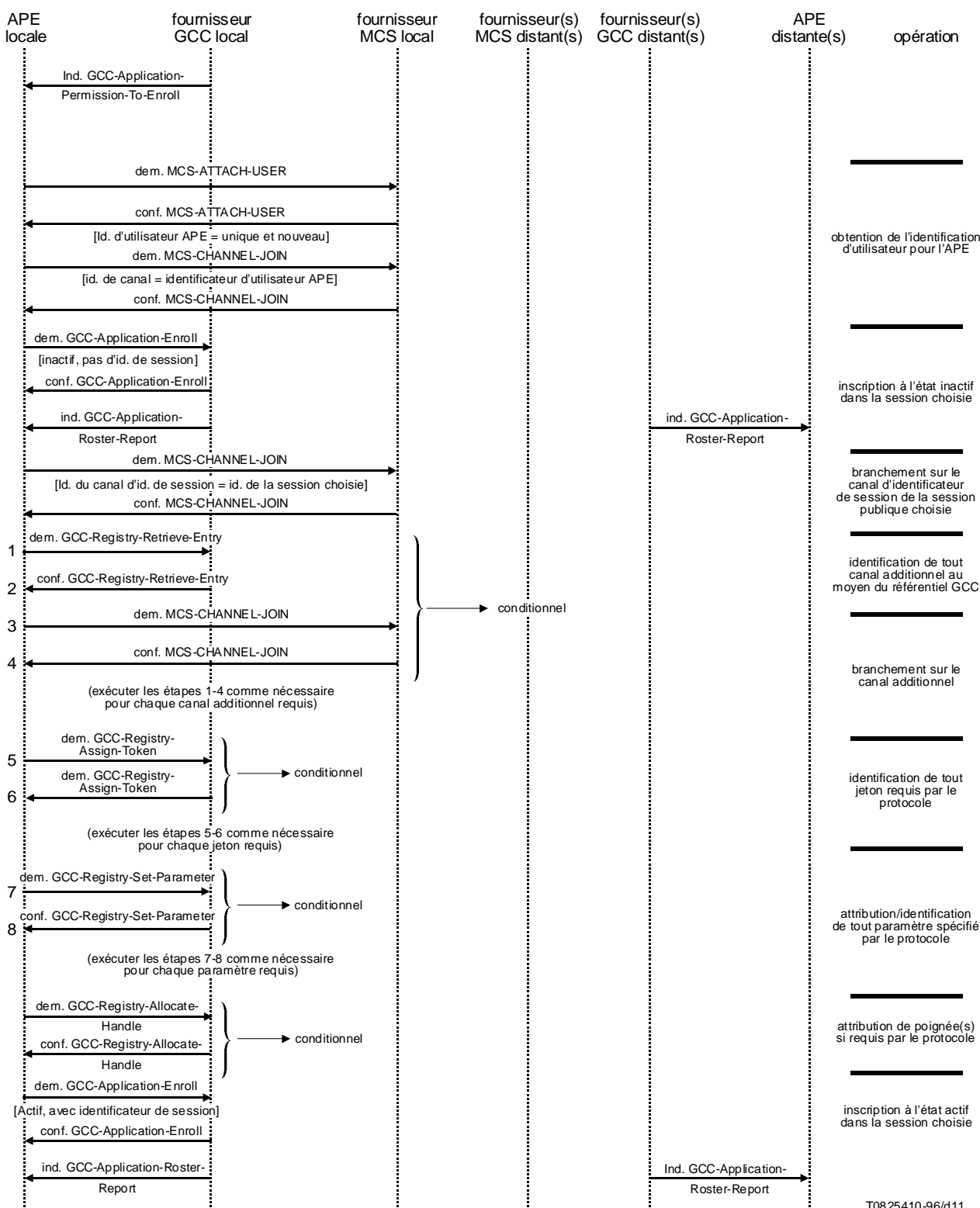


FIGURE 11/T.121

Exemple de séquence de déclenchement d'un protocole de session publique –
Cas n° 1: membre de la session

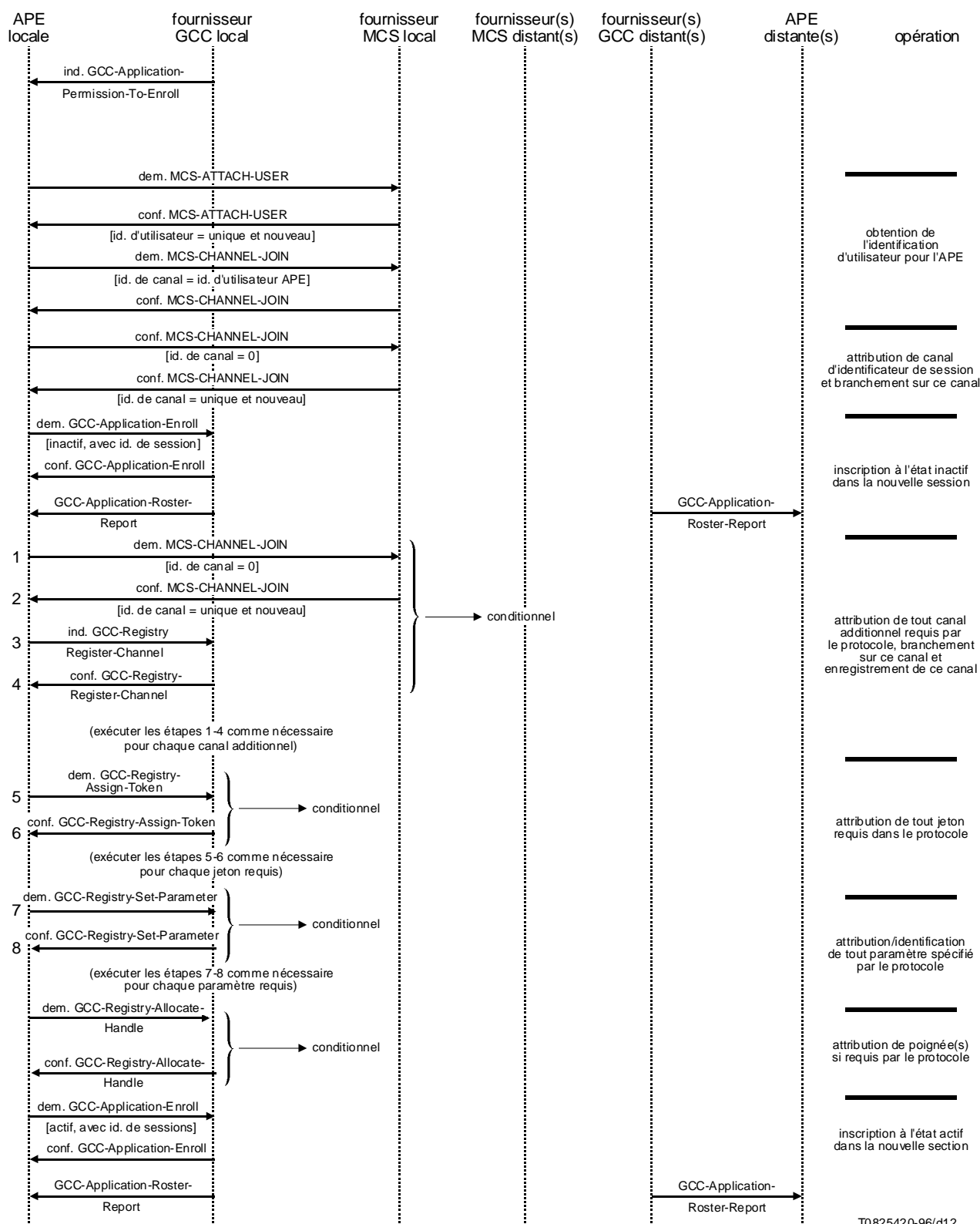


FIGURE 12/T.121

Exemple de séquence de déclenchement de protocole de session publique –
Cas n° 2: créateur de session

6.2.5 Session privée

Après s'être branché sur son canal d'identificateur d'utilisateur (*User ID Channel*), l'ARM examinera le paramètre *sessionID* pour décider, s'il est présent, de participer à une session privée existante (comme membre privé) ou, s'il est omis, de créer une nouvelle session (comme organisateur privé). Voir la Figure 13.

Si le paramètre *sessionID* est présent, l'ARM s'inscrira à l'état inactif dans la session indiquée en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec *Active/Inactive* (indicateur d'activité) positionné à *Inactive* (inactif) et en spécifiant la clé de session avec l'identificateur de session requis. L'ARM devra alors attendre jusqu'à ce qu'il ait reçu une indication *MCS-Channel-Admit* (admission sur canal) de l'ARM homologue sur le nœud qui organise la session privée. Il essaie alors de se brancher sur le canal indiqué dans cette primitive en émettant une demande *MCS-Channel-Join* (branchement sur le canal MCS) et informe l'ASE correspondant du succès du branchement sur le canal. L'identificateur de canal est utilisé comme identificateur de session.

Si le protocole d'application requiert le branchement sur de multiples canaux avant qu'une APE puisse être considérée comme participante active à une session, alors l'ARM attendra d'être admis sur ces canaux puis se branchera sur eux en conséquence. Il est possible de se brancher sur de nouveaux canaux quand la session d'application est en cours. De même, si un protocole d'application exige qu'une APE connaisse l'identité des jetons ou des paramètres de référentiel avant de dialoguer avec ses homologues, l'ARM peut déterminer leur identité au moyen du référentiel GCC.

L'ARM s'inscrira alors à l'état actif en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec les paramètres spécifiés dans le Tableau 3.

Si le paramètre *sessionID* est omis, l'ARM tentera de créer une nouvelle session privée. L'ARM organisateur privé constituera d'abord un canal privé d'invitation en émettant une demande *MCS-Channel-Convene* (constitution de canal d'invitation). En cas de succès, la confirmation *MCS-Channel-Convene* (constitution de canal d'invitation) renvoyée contient l'identificateur du canal attribué. L'identificateur de canal devient aussi identificateur de session. L'ARM se branchera alors sur ce canal en émettant une demande *MCS-Channel-Join* (branchement sur le canal MCS). L'ARM s'inscrira alors à l'état inactif dans la session en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec les paramètres précisés dans le Tableau 3.

La responsabilité d'acquiescer l'ensemble de ressources qui doivent être identifiées comme partie du processus d'inscription à une session privée incombe à l'ARM qui a créé cette session. Si la spécification de protocole exige des jetons, des paramètres de référentiel ou des canaux supplémentaires, l'ARM devra les créer et les enregistrer auprès de la GCC comme nécessaire. La clé de référentiel utilisée pour enregistrer ces ressources comprendra l'identificateur de session comme partie de la clé de session. On remarquera qu'il peut ne pas être nécessaire d'enregistrer les canaux privés pour permettre, aux membres d'une session privée, de déterminer la fonction de chaque canal. Ces informations peuvent être acheminées plus efficacement si un protocole d'application rend obligatoire l'ordre dans lequel l'organisateur émet les primitives de demande d'admission (*MCS-Channel-Admit*) vers les membres de la session. L'enregistrement de canaux privés n'est cependant pas exclu.

Une fois que toutes les ressources auront été attribuées avec succès, l'ARM s'inscrira à l'état actif dans la session en émettant une demande *GCC-Application-Enroll* (inscription d'application) avec les paramètres précisés dans le Tableau 3.

L'ARM essaiera alors d'inviter d'autres nœuds à participer à la session privée, en émettant une demande *GCC-Application-Invoke* (invocation d'application) spécifiant une liste de *GCC Node ID* (identificateurs nœuds GCC) à inviter ou *NULL* (nul) (indiquant que tous les nœuds de la conférence doivent être invités) comme paramètre de nœuds de destination. Il devra aussi spécifier *Dynamic Private* (privé dynamique) comme canal de démarrage dans l'entrée de protocole d'application.

L'ARM organisateur devra attendre alors jusqu'à ce qu'il ait reçu une indication *GCC-Application-Roster-Report* (compte rendu de répertoire d'application) contenant les *MCS User ID* (identificateur d'utilisateur MCS) des APE à inviter à la session privée. Il convient de prévoir qu'une APE pourra ne pas être activée sur un ou plusieurs nœuds invités. L'ARM émettra alors une demande *MCS-Channel-Admit* (admission sur canal) pour le canal de démarrage, précisant les identificateurs d'utilisateurs MCS des APE à inviter comme liste de *MCS User ID*. La liste des invités est fournie par l'application d'utilisateur. Ce processus se répète alors pour tout canal additionnel spécifié par le protocole.

6.3 Constitution de clés de référentiel

Pour déterminer l'identité d'un canal ou jeton dynamique à travers le référentiel GCC, un ARM doit constituer une clé de référentiel qui se compose de Session Key (clé de session) pour la session en cours et d'un Resource ID (identificateur de ressource). L'identificateur de ressource est un OCTET STRING (chaîne d'octets) ASN.1 qui décrit le canal ou le jeton particuliers dont l'identificateur est demandé. La construction d'identificateurs de ressource pour les canaux et les jetons sera définie dans la spécification de protocole d'application.

6.4 Négociation de capacités

Les APE utilisent le mécanisme d'inscription de l'application pour la négociation de capacités. Le paramètre de liste de capacités d'application (*Application Capabilities List*) de la primitive de demande d'inscription (*GCC-Application-Enroll*) est utilisé pour spécifier la liste des capacités prises en charge par l'APE locale.

La liste de capacités d'application incluse dans la demande GCC-Application-Enroll se compose d'une liste de capacités qui devront être annoncées comme prises en charge par l'APE. Le résultat de la procédure de négociation de capacités est rendu visible à l'ARM par la réception d'une indication de compte rendu de répertoire d'applications (*GCC-Application-Roster-Report*) envoyée par le fournisseur GCC. Le compte rendu de répertoire d'applications inclut le répertoire d'applications pour les APE homologues dans la conférence indiquée, c'est-à-dire les APE qui ont désigné la même clé de session. Le répertoire d'applications inclut une liste des nœuds pour lesquels une APE homologue s'est inscrite. Pour chaque nœud, la liste contient l'identification d'utilisateur GCC (*GCC User ID*) de ce nœud, et l'identification d'utilisateur d'application de l'APE homologue sur ce nœud. Le répertoire d'applications inclut aussi un numéro d'instance, un indicateur signalant si de nouveaux nœuds ont été ajoutés depuis la dernière instance, un indicateur signalant si des nœuds ont été retirés depuis la dernière instance, un indicateur signalant si la liste des capacités d'application (*Application Capabilities List*) a été mise à jour depuis la dernière instance et, si tel est le cas, la nouvelle liste de capacités d'application. Dans le cas de l'inscription d'une nouvelle APE, la liste de capacités d'application est toujours mise à jour, puisque les instances antérieures de la liste ne sont pas disponibles pour cette APE.

Lors de sa première inscription, un ARM devra ignorer les indications de compte rendu de répertoire (*GCC-Application-Roster-Report*) reçues où l'APE n'est pas incluse (c'est-à-dire où il y a aucune entrée dans le répertoire d'applications qui ait l'identificateur d'utilisateur GCC du nœud local et l'identificateur d'entité protocolaire d'application (*Application Protocol Entity ID*) de l'APE locale tels qu'indiqués dans la confirmation d'inscription (*GCC-Application-Enroll*). Une fois qu'un répertoire comprenant l'APE locale a été reçu, cette APE est alors considérée comme faisant partie de la conférence et peut poursuivre, en examinant le répertoire pour déterminer comment poursuivre.

La liste de capacités d'application reçue comme partie intégrante de l'indication de compte rendu (*GCC-Application-Roster-Report*) correspond aux listes de capacités d'application fédérées de toutes les APE homologues inscrites. C'est-à-dire que cette liste inclut une entrée pour chaque capacité présentée par une quelconque APE homologue. Pour chaque entrée, elle contient l'identificateur de capacité (*Capability ID*), le nombre des APE homologues (APE locale comprise) qui avaient annoncé cette capacité au cours de leur procédure d'inscription et, dans le cas de capacités de type MIN ou MAX, la valeur minimale ou maximale du paramètre chez toutes les APE homologues qui avaient annoncé cette capacité. Pour chaque capacité, les règles employées pour déterminer le résultat de l'échange de capacités sont spécifiées par le protocole d'application.

A tout moment, pendant qu'une APE est inscrite à une conférence, l'APE peut recevoir, du fournisseur GCC, des indications de compte rendu (*GCC-Application-Roster-Report*) supplémentaires indiquant que le contenu du répertoire a changé. Cela peut être dû à l'inscription, à la conférence, de nouvelles APE homologues, à ce que des APE homologues sont sorties de la conférence ou à ce que des APE homologues ont modifié leurs renseignements d'inscription.

Si, à quelque moment, l'APE locale désire indiquer un changement dans sa liste de capacités d'application, elle peut se réinscrire. Cela se fait par réémission d'une demande d'inscription (*GCC-Application-Enroll*) vers le fournisseur GCC avec l'indicateur d'inscription (Enroll/Un-enroll) mis à «inscription» (*Enroll*) et la liste de capacités d'application mise à jour, ainsi que les autres paramètres normalement inclus dans la demande d'inscription. Le résultat en est, potentiellement, une modification au répertoire d'applications ayant pour résultat la réception d'une indication de compte rendu (*GCC-Application-Roster-Report*) par l'APE locale aussi bien que par toutes les autres APE homologues dans la conférence.

6.5 Sortir d'une session

Une application d'utilisateur qui désire sortir d'une session le signalera à l'ARM. Avant l'annulation d'inscription à la session, l'ARM est chargé de l'annulation de toutes les entrées de référentiel de GCC qu'il peut avoir créées, sous réserve que celles-ci ne soient plus nécessaires au reste des participants à la session. La méthode pour déterminer cela est spécifiée par le protocole d'application. L'ARM sort de la session en émettant une demande GCC-Application-Enroll (inscription d'application) avec Enroll/Un-enroll (indicateur d'inscription) positionné sur Un-enroll (non-inscription). Aucun autre paramètre n'est nécessaire.

A tout moment, si l'ARM reçoit une indication GCC-Application-Roster-Report (compte rendu de répertoire d'applications) dans laquelle il n'est plus inclus (c'est-à-dire si son identificateur d'entité est absent), l'ARM émettra immédiatement une demande MCS-Detach-User (détachement d'utilisateur) pour se détacher de la conférence spécifiée. L'APE est plus considérée comme inscrite dans la conférence à ce moment mais peut tenter de se réinscrire dans la conférence en répétant les étapes de son initialisation.

A tout moment, si l'ARM reçoit une indication GCC-Application-Permission-To-Enroll (autorisation d'inscription à une application) avec Grant/Revoke (indicateur d'octroi) positionné sur Revoke (retrait), il émettra immédiatement une demande MCS-Detach-User (détachement d'utilisateur) pour se détacher de la conférence spécifiée. L'APE n'est plus considérée comme inscrite à la conférence à ce moment et ne tentera pas de se réinscrire.

Si l'ARM est un membre d'une session privée et s'il reçoit une indication GCC-Application-Roster-Report (compte rendu de répertoire d'applications) pour la session où le créateur de session n'est plus inclus (c'est-à-dire si son identificateur d'entité est absent), la session est censée avoir cessé d'exister. L'ARM émettra aussitôt une demande GCC-Application-Enroll (inscription d'application) avec Enroll/Un-enroll (indicateur d'inscription) positionné sur Un-enroll (non-inscription) pour se détacher de la session spécifiée. Aucun autre paramètre n'est nécessaire.

7 Examen des ressources MCS et des services GCC

Le présent article propose un examen de l'utilisation des ressources MCS (canaux et jetons) et du GCC Registry (référentiel GCC). Ces thèmes sont traités de façon plus complète dans les Recommandations T.122, T.125 et T.124. On remarquera que, tandis que la responsabilité d'obtenir l'ensemble initial de ressources pour une session incombe au créateur de la session, tout membre de la session peut, à tout moment, acquérir des ressources supplémentaires, si les spécifications du protocole le permettent.

Les éléments de service d'application (ASE) devraient être conçus de telle sorte que, dans la mesure du possible, leur fonctionnement soit indépendant du type de ressource (c'est-à-dire statique ou dynamique) disponible pour la session.

7.1 Canaux

Les canaux MCS sont utilisés pour la distribution de données dans une conférence. Les canaux MCS se divisent en deux types, *statique* et *dynamique*, ce dernier ayant trois variétés, à savoir: User ID Channel (canal d'identificateur d'utilisateur), Assigned Channel (canal attribué) et Private Channel (canal privé) (voir 8.4/T.122). Un ARM s'inscrit aux canaux requis par son application d'utilisateur et en sort en utilisant les services *channel join* (branchement sur canal) et *channel leave* (sortie de canal), respectivement.

7.1.1 Canal statique

Canal dont le MCS Channel ID (identificateur de canal MCS) est dans l'intervalle 1-1000. Les canaux statiques sont réservés à l'utilisation par des protocoles d'application normalisés tels que T.126 et T.127. Les canaux statiques sont attribués à leurs protocoles respectifs dans T.120; chaque canal statique a un but précis, comme décrit dans la Recommandation de protocole d'application correspondante. Un protocole d'application normalisé peut utiliser un ou plusieurs canaux statiques. Pour chaque protocole d'application normalisé, un des canaux sera désigné comme *canal de démarrage*; c'est le canal dont le MCS Channel ID est utilisé comme identificateur de session pour la session de base normalisée de ce protocole.

Les canaux statiques ne peuvent être utilisés que dans la session de base normalisée du protocole d'application normalisé correspondant. Habituellement, toutes les APE participant à une telle session s'y branchent et ils sont ainsi utilisés pour la diffusion générale de données à la session.

L'ARM examinera le paramètre Resource List (liste de ressources) afin de déterminer s'il lui faut se brancher sur un canal statique quelconque. Pour chaque canal statique identifié dans Resource List, l'ARM émettra une demande MCS-Channel-Join (branchement sur le canal MCS) précisant le Channel ID (identificateur de canal) approprié. En cas de succès, l'ARM transmettra l'identificateur de canal à son ASE.

7.1.2 Canal dynamique

Canal dont le MCS Channel ID (identificateur de canal MCS) est dans l'intervalle 1001-65535. Les canaux dynamiques forment une réserve de ressources de canaux, disponible à toutes les sessions au sein d'une conférence. La fonction d'un canal dynamique n'est pas prédéfinie, mais déterminée par une APE lorsqu'elle émet une demande de canal dynamique. Il y a trois variétés de canal dynamique: User ID Channel (canal d'identificateur d'utilisateur), Assigned Channel (canal attribué) et Private Channel (canal privé).

7.1.3 Canal d'identificateur d'utilisateur

Canal MCS recevant le branchement d'une seule APE. Un canal MCS User ID identifie une APE de façon univoque et peut servir à envoyer des données exclusivement à cette APE. Un User ID (identificateur d'utilisateur) est nécessaire lorsqu'une APE entreprend de s'inscrire à l'état actif. Pour acquérir un User ID unique, l'ARM émettra une demande MCS-Attach-User (rattachement d'utilisateur) vers le fournisseur MCS, en utilisant la Conference ID (identificateur de conférence) contenu dans l'indication GCC-Application-Permission-To-Enroll (autorisation d'inscription) comme Domain Selector (sélecteur de domaine). A la réception d'une confirmation de succès de MCS-Attach-User (rattachement d'utilisateur) en réponse, l'ARM se branchera sur le User ID Channel indiqué, en émettant une demande MCS-Channel-Join (branchement sur le canal MCS). Voir la Figure 14.

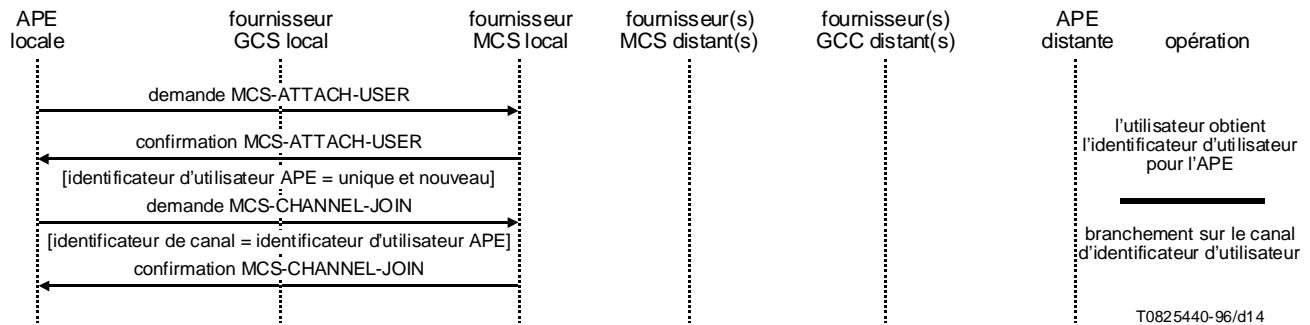


FIGURE 14/T.121

Obtention d'un canal d'identificateur d'utilisateur MCS

7.1.4 Canal attribué

Canal MCS qui peut être utilisé pour la diffusion générale d'informations à tous les participants à une session. L'accès à ces canaux n'est pas réglementé; c'est-à-dire que toute APE peut, à tout moment, se brancher sur un canal attribué ou en sortir. Si toutes les APE sortent d'un canal attribué, il cesse d'exister et son Channel ID (identificateur de canal) peut être attribué de nouveau pour une utilisation différente; c'est là un piège potentiel. Voir la Figure 15.

Les canaux attribués peuvent être utilisés tant dans les sessions de type statique que dans les sessions de type multidestinataire. Leur utilisation dans des sessions de type privé n'est pas recommandée car toute APE peut se brancher sur un canal attribué si elle est avertie de l'existence du canal.

Le canal de démarrage d'une session publique ou de base non normalisée est un canal attribué.

Si des canaux attribués additionnels sont nécessaires après l'établissement d'une session, l'ARM tentera d'abord de créer un nouveau canal attribué en émettant une primitive de demande MCS-Channel-Join (branchement sur le canal MCS) avec Channel ID (identificateur de canal) égal à 0. La confirmation MCS-Channel-Join renvoyée, en cas de succès, contient le Channel ID attribué qui devra être utilisé comme canal attribué. L'ARM peut alors tenter d'enregistrer le canal attribué en émettant une demande GCC-Registry-Register-Channel (enregistrement de canal) avec les paramètres donnés dans le Tableau 4. A la réception de la confirmation GCC-Registry-Register-Channel (enregistrement de canal) correspondante, l'ARM devra examiner le paramètre Result (résultat). S'il a la valeur «successful» (réussite), l'ARM notifiera à son ASE le Channel ID attribué au canal attribué. Un paramètre Result avec la valeur «index already exists» (le pointeur existe déjà) indique qu'un autre ARM a déjà enregistré un canal attribué avec la même Registry Key (clé de référentiel). Dans ce cas, l'ARM devra sortir du canal qu'il avait précédemment créé, en émettant une demande MCS-Channel-Leave (sortie de canal MCS). Il tentera alors de se brancher sur le canal attribué enregistré par l'autre ARM, en émettant une demande MCS-Channel-Join précisant le Channel ID renvoyé dans le paramètre Registry Item (élément de référentiel) de la confirmation GCC-Registry-Register-Channel (enregistrement de canal) en tant que Channel ID. En cas de succès, l'ARM transmettra à son ASE le Resource ID (identificateur de ressource) et le Channel ID correspondant. Si l'essai échoue et si le résultat est «canal inconnu» («no such channel»), l'ARM pourra tenter de nouveau le processus de création d'un canal. Dans tous les autres cas, l'ARM devra indiquer à l'ASE qu'il n'a pas pu attribuer un canal. Voir également le Tableau 5.

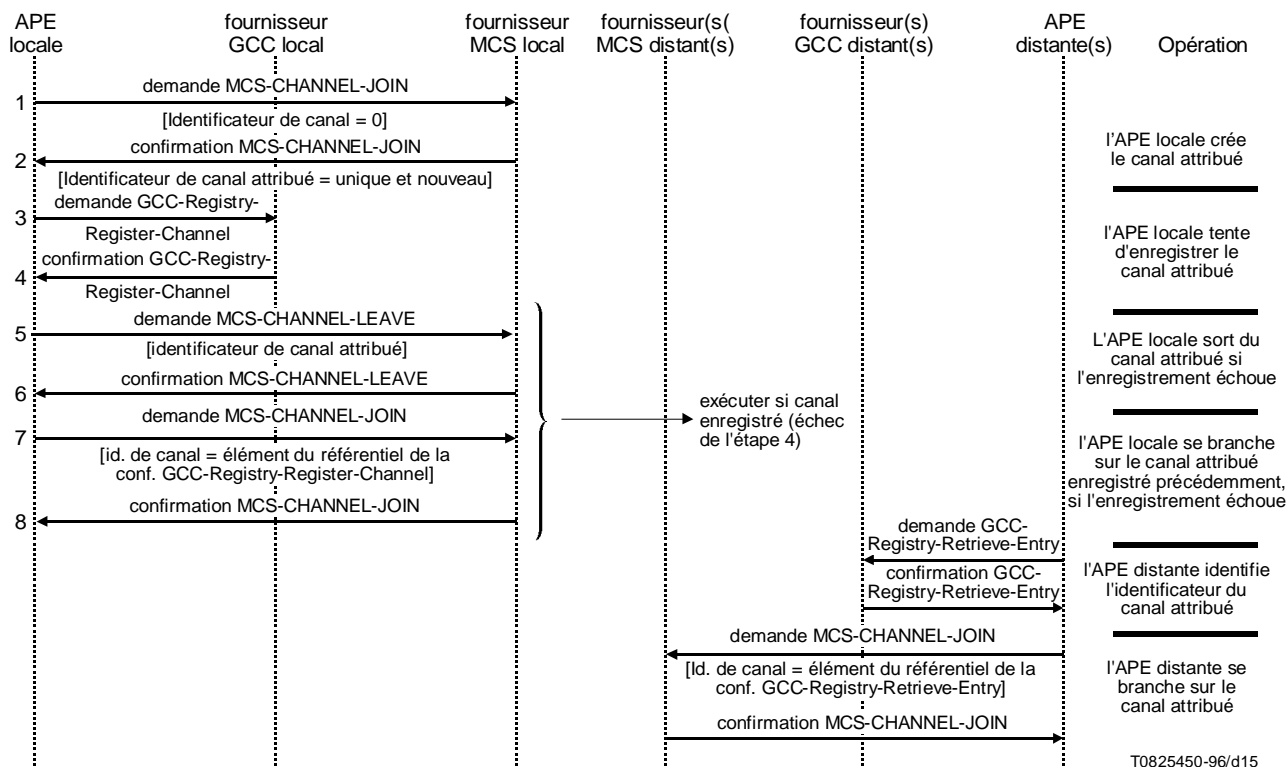


FIGURE 15/T.121

Création et enregistrement d'un canal attribué

TABEAU 4/T.121

Paramètres de demande GCC-Registry-Register-Channel (enregistrement de canal)

Paramètre	Contenu
Conference ID (identificateur de conférence)	Fourni par l'indication GCC-Application-Permission-To-Enroll (autorisation d'inscription à une application).
Registry Key (clé de référentiel)	Clé de référentiel constituée comme décrit au 6.3.
Channel ID (identificateur de canal)	Identificateur de canal renvoyé dans la confirmation MCS-Channel-Join (branchement sur le canal MCS).

TABEAU 5/T.121

Paramètres de demande GCC-Registry-Retrieve-Entry (recherche d'entrée)

Paramètre	Contenu
Conference ID (identificateur de conférence)	Fourni par l'indication GCC-Application-Permission-To-Enroll (autorisation d'inscription à une application).
Registry Key (clé de référentiel)	Clé de référentiel constituée comme décrit au 6.3.

7.1.5 Canal privé

Un canal privé est un canal MCS avec un groupe d'utilisateurs autorisés. Il peut être utilisé pour la distribution de données à un sous-ensemble sélectionné de participants au sein d'une session. L'accès à un canal privé est dirigé par l'APE qui a organisé le canal. Cette APE est dénommée Channel Manager (gestionnaire de canal) et a les privilèges suivants:

- le privilège d'admettre d'autres APE sur le canal privé;
- le privilège de rejeter d'autres APE du canal privé;
- le privilège de dissoudre le canal privé.

Les canaux privés peuvent être utilisés dans des sessions de base normalisées, de base non normalisées, publiques ou privées.

Le canal de démarrage d'une session privée est un canal privé.

Si l'application d'utilisateur organisatrice désire retirer une ou plusieurs APE d'entre les membres du canal privé, le Convenor ARM (ARM d'organisateur) émettra une demande MCS-Channel-Expel (rejet de canal) spécifiant les MCS User ID (identificateurs d'utilisateur MCS) des ASE à rejeter comme liste de MCS User ID.

Dès que l'application d'utilisateur au nœud d'organisation indique qu'elle ne se sert plus du canal privé, le Convenor ARM doit dissoudre le canal en émettant une demande MCS-Channel-Disband (dissolution de canal) spécifiant le canal privé comme Channel ID (identificateur de canal). Tous les ARM pour l'instant branchés sur le canal privé reçoivent alors une indication MCS-Channel-Expel (rejet de canal) leur notifiant que le canal privé n'est plus disponible à l'utilisation.

Si des canaux privés additionnels sont nécessaires après l'établissement d'une session, l'ARM suivra la procédure décrite pour l'organisateur ou pour le membre, comme il conviendra. Voir la Figure 16.

7.1.6 Problèmes potentiels d'utilisation de canaux MCS

- i) Le MCS ne fournit aucune indication aux membres d'un canal quand une APE se branche sur un canal ou en sort. C'est un problème, particulièrement dans l'utilisation de canaux privés, puisque le gestionnaire de canal privé ne reçoit aucune confirmation du branchement ou du défaut de branchement des APE invitées sur le canal privé. Un moyen possible (tel qu'il est mis en œuvre dans la Recommandation T.127) pour tourner ce problème est que les APE confirment qu'elles se sont branchées sur le canal privé en envoyant une PDU à l'User ID Channel (canal d'identificateur d'utilisateur) du gestionnaire de canal.
- ii) La prudence est recommandée dans l'utilisation des canaux attribués, puisque ces canaux cessent d'exister quand tous les utilisateurs qui s'y sont branchés sortent. Aucune indication n'est fournie quand cela survient. Puisque le MCS peut réutiliser les canaux dynamiques, une APE ne devrait pas tenter d'utiliser un canal attribué à moins d'être certaine de son existence et de son utilisation à ce moment. La présence d'une entrée dans GCC Application Registry (référentiel d'application GCC) pour un canal attribué ne garantit pas que le canal existe ou que l'utilisation indiquée par le référentiel soit encore en cours.
- iii) Les données en cours de transit sur un canal privé quand le gestionnaire de canal en sort, le dissout ou rejette les membres peuvent ne pas atteindre leur destination d'origine, du fait que le MCS peut rejeter des membres du canal avant la remise de ces données. L'utilisation du service d'interdiction de jeton MCS permet aux membres du canal de signaler au gestionnaire de canal l'aboutissement d'une transaction, mettant ainsi le gestionnaire de canal en mesure d'éviter des pertes de données causées par inadvertance lors d'un rejet prématuré de membres du canal. Il faut que chaque membre de canal interdise un jeton désigné avant que la transaction puisse commencer, et qu'il libère ce jeton à l'exécution de la transaction. Les protocoles d'application doivent garantir que tous les membres du canal ont interdit le jeton pertinent avant que la transaction commence. Sinon, les protocoles d'application peuvent définir un échange de PDU entre le gestionnaire de canal et les membres du canal qui garantisse que toutes les données ont été remises avant qu'un canal privé soit dissous.
- iv) Les données en transit sur tout canal peuvent ne pas atteindre leur destination d'origine si l'APE émettrice se détache avant la remise de ces données. L'APE émettrice peut éviter cela en utilisant le remède décrit dans iii).

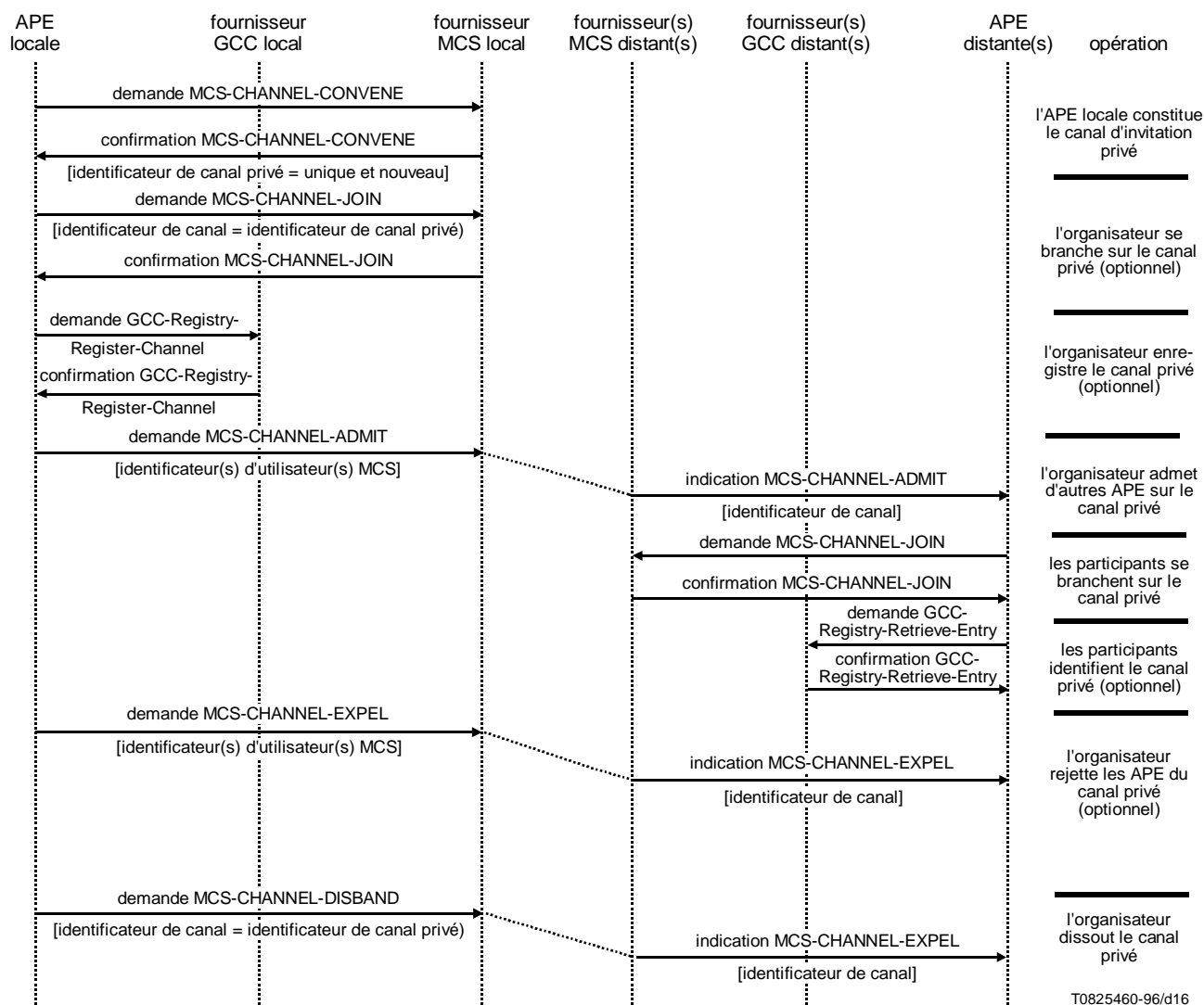


FIGURE 16/T.121
Gestion d'un canal privé

7.1.7 Résumé

Les canaux statiques sont utilisés par la session de base normalisée de protocoles d'application normalisés pour la diffusion générale de données.

Un User ID Channel (canal d'identificateur d'utilisateur) est nécessaire pour chaque APE active. Il peut être utilisé pour envoyer des données exclusivement à cette APE.

Les Assigned Channels (canaux attribués) sont utilisés dans n'importe quel type de session pour la diffusion générale de données.

Les Private Channels (canaux privés) sont utilisés dans une session privée pour la diffusion générale de données. Ils peuvent aussi être utilisés dans toute session pour une distribution sélective de données.

Le choix de canaux ouvert aux protocoles d'application non normalisés pour une utilisation générale est entre canaux attribués et canaux privés. Le présent paragraphe compare les avantages et les inconvénients de chacun de façon à aider le réalisateur de protocole d'application à faire le choix approprié. Voir le Tableau 6.

Privé

- Durée utile connue, déterminée par le gestionnaire de canal.
- Accès réglementé.
- Cesse d'exister si le gestionnaire de canal quitte la session.

Attribué

- Durée utile inconnue.
- Accès non réglementé.
- Peut persister après que le créateur a quitté la session.

TABLEAU 6/T.121

Utilisation de canaux MCS

Type de canal	Type de session			
	De base normalisée	De base non normalisée	Publique	Privée
Statique	✓	7	7	7
ID d'utilisateur	✓	✓	✓	✓
Attribué	✓	✓	✓	●
Privé	✓	✓	✓	✓
✓ Permis 7 Non permis ● Déconseillé				

7.2 Transfert de données

Il existe deux mécanismes disponibles pour la distribution de données, à savoir: *Send Data* (envoi de données) et *Uniform Send Data* (envoi uniforme de données). Les deux techniques transfèrent des données à toutes les APE branchées sur un canal MCS.

Send Data fournit la remise de données à chaque APE branchée sur un canal MCS par le trajet le plus court possible, minimisant ainsi trafic de réseau et temps d'attente. L'ordre des données reçues à chaque APE peut différer et l'expéditeur des données ne reçoit pas de copie des données.

Uniform Send Data fournit une remise de données mises en séquence, c'est-à-dire qu'il garantit que chaque APE branchée sur un canal MCS reçoit des données dans le même ordre. L'expéditeur des données reçoit une copie des données si l'expéditeur est branché sur ce canal.

Une APE doit se brancher sur un canal MCS pour *recevoir* des données sur ce canal. Une APE peut *envoyer* des données à tout canal MCS, sans tenir compte de ce qu'il est ou non branché sur ce canal, sauf pour les canaux privés, où une APE doit être admise sur le canal avant d'essayer d'envoyer des données au canal.

7.3 Jetons

Les jetons fournissent un moyen pour mettre en œuvre un accès exclusif aux ressources au sein d'une conférence. Par exemple, pour empêcher que des commandes contradictoires soient émises vers un dispositif (comme une caméra) par des utilisateurs différents, il est souhaitable de limiter l'accès à ce dispositif à un seul utilisateur en toute circonstance donnée. Cela peut être obtenu en associant un jeton au dispositif et en prescrivant, dans le protocole d'application correspondant, que seul le détenteur présent du jeton est autorisé à émettre des commandes de contrôle vers ce dispositif.

La manipulation des jetons est prise en charge par plusieurs services MCS:

Le service *token grab* (saisie de jeton) autorise un seul utilisateur à détenir exclusivement un jeton spécifié. Les utilisateurs peuvent utiliser le service *token test* (essai de jeton) pour déterminer le statut d'un jeton à tout moment, et peuvent demander le jeton au détenteur avec le service *token please* (demande de jeton). Le détenteur du jeton peut transférer le contrôle d'un jeton à un autre utilisateur spécifié avec le service *token give* (cession de jeton) ou rendre à un jeton un statut de disponibilité générale avec le service *token release* (libération de jeton). Les services de demande de jeton et de cession de jeton permettent qu'un jeton soit transféré entre deux usagers sans possibilité de saisie de ce jeton par un troisième usager.

Les jetons peuvent aussi être utilisés pour la coordination d'événements entre de multiples utilisateurs, à travers le service *token inhibit* (interdiction de jeton). Les utilisateurs peuvent indépendamment interdire et libérer le même jeton. Par exemple, si l'on désire savoir à quel moment tous les utilisateurs ont fini de recevoir et de traiter un transfert de fichier volumineux, tous les utilisateurs interdiront le même jeton au début de l'opération et chaque utilisateur individuel libérera le jeton quand il aura terminé le processus. Pour éviter des situations de compétition, il est nécessaire de s'assurer que tous les utilisateurs ont interdit le jeton avant de faire un essai pour vérifier que le jeton est devenu non interdit. Tout utilisateur pourra tester le jeton à volonté pour déterminer si le jeton est libre, ce qui signifie que tous les utilisateurs ont terminé le traitement.

Les jetons peuvent être soit statiques soit dynamiques. Le MCS traite tous les jetons identiquement; la distinction entre les jetons statiques et dynamiques se fait dans la GCC, qui utilise le Registry (référentiel) pour attribuer des jetons dynamiques. Toutes les opérations sur des jetons sont effectuées par l'ASE en utilisant les services de jetons MCS.

7.3.1 Jetons statiques

Jeton avec MCS Token ID (identificateur de jeton MCS) compris dans l'intervalle 1-16383. Les jetons statiques sont réservés à l'utilisation par des protocoles d'application normalisés tels que T.126 et T.127. Les jetons statiques sont attribués à leurs protocoles respectifs dans la Recommandation T.120; chaque jeton statique a un objet précis comme décrit dans la Recommandation de protocole d'application correspondante. Un protocole d'application normalisé peut utiliser zéro jetons statiques ou davantage.

Les jetons statiques ne peuvent être utilisés que dans la session de base normalisée du protocole d'application normalisé correspondant.

L'ARM n'est aucunement impliqué dans l'utilisation de jetons statiques.

7.3.2 Jetons dynamiques

Jeton avec MCS Token ID compris dans l'intervalle 16384-65535. Les jetons dynamiques constituent une réserve de ressources de jetons accessible à toutes les sessions au sein d'une conférence. La fonction d'un jeton dynamique n'est pas prédéfinie mais est déterminée quand il est attribué par la GCC. Les jetons dynamiques peuvent être utilisés dans tous les types de session. Voir la Figure 17.

L'ARM devra examiner le paramètre Resource List (liste de ressources) pour déterminer s'il lui faut identifier des Token ID (identificateur de jeton) pour quelque jeton dynamique. La même méthode peut être utilisée indépendamment du fait que l'ARM sera organisateur de session ou membre de la session. Pour chaque Token Resource ID (identificateur de ressource de jeton) précisé dans Resource List, l'ARM émettra une demande GCC-Registry-Assign-Token (attribution de jeton) vers le fournisseur GCC en utilisant les paramètres spécifiés dans le Tableau 7. Si le paramètre Result renvoyé dans la confirmation GCC-Registry-Assign-Token (attribution de jeton) a la valeur «successful» (réussite) ou la valeur «index already exists» (le pointeur existe déjà), le Token ID contenu dans le Registry Item (élément de référentiel) de la primitive de confirmation renvoyée est utilisé comme Token ID pour le jeton qui correspond au Resource ID utilisé dans la clé de référentiel. L'ARM transmettra à son ASE le Resource ID et le Token ID associé.

On remarquera qu'il n'est pas nécessaire d'utiliser GCC-Registry-Retrieve-Entry (recherche d'entrée) pour identifier les jetons, puisque GCC-Registry-Assign-Token (attribution de jeton) renvoie l'identificateur de jeton, sans tenir compte de ce que le jeton existait ou non avant que la demande soit faite.

Les jetons dynamiques additionnels nécessaires pendant une session peuvent être attribués par tout participant à cette session, sous réserve de toute contrainte définie par la spécification de protocole d'application pertinente.

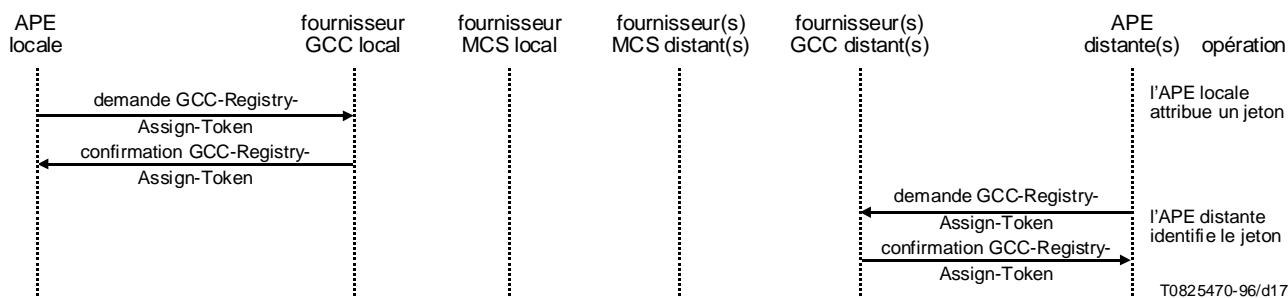


FIGURE 17/T.121

Attribution et identification de jetons dynamiques

TABLEAU 7/T.121

Paramètres d'une demande GCC-Registry-Assign-Token (attribution de jeton)

Paramètre	Contenu
Conference ID (identificateur de conférence)	Fourni par l'indication GCC-Application-Permission-To-Enroll (autorisation d'inscription à une application).
Registry Key (clé de référentiel)	Clé de référentiel constituée comme décrit au 6.3.

7.4 Répertoire d'application

L'ARM est responsable de la surveillance des indications de compte rendu GCC-Application-Roster (répertoire d'application) et de la notification à son ASE de tout changement. Le compte rendu d'entrée au répertoire d'application (*Application Roster Entry Report*) contient la partie du répertoire qui correspond à la session de protocole d'application dans laquelle l'APE est inscrite. Dans le cas d'une APE qui s'est inscrite à l'état inactif sans identificateur de session, le compte rendu de répertoire d'application contiendra la partie du répertoire qui correspond à toutes les sessions d'application ayant la même clé de protocole d'application.

Tant qu'une APE est inscrite dans une conférence, elle peut à tout moment recevoir une indication GCC-Application-Roster-Report (compte rendu de répertoire d'application) additionnelle, lui notifiant une modification dans le contenu du répertoire. Ceci peut être dû à l'inscription de nouvelles APE homologues dans la conférence, à la sortie de conférence d'APE homologues ou à ce que des APE homologues ont modifié leurs informations d'inscription.

Une application d'utilisateur peut changer sa liste de capacités d'application (*Application Capabilities List*) à tout moment en ordonnant à l'ARM approprié de se réinscrire. L'ARM émet alors une demande GCC-Application-Enroll (inscription d'application) comprenant Enroll/Un-enroll (indicateur d'inscription) positionné sur Enroll (inscription), la liste de capacités d'application révisée, et tous les autres paramètres tels qu'ils étaient inclus dans l'inscription initiale à l'état actif. Cela peut entraîner une modification de l'ensemble des capacités fédérables, auquel cas toutes les APE dans la session recevront une indication GCC-Application-Roster-Report (compte rendu de répertoire d'application).

L'effet de tout changement dans l'ensemble des capacités fédérables sur toute transaction ayant lieu au moment de ce changement sera précisé par le protocole d'application correspondant.

7.5 Référentiel d'application

Le référentiel d'application (*Application Registry*) est une base de données centralisée tenue à jour par le principal fournisseur GCC et qui peut être utilisée pour permettre à des APE de déterminer de façon indépendante quelles sont les ressources (telles que canaux dynamiques et jetons) à utiliser pour une session de protocole d'application précise. L'importance du contenu du référentiel est précisée par les protocoles d'application individuels.

Le référentiel d'application fournit des mécanismes pour:

- enregistrer les canaux,
- attribuer des jetons,
- attribuer des poignées (*handles*) globalement uniques, soit individuellement, soit comme bloc de poignées,
- déterminer des paramètres communs pour la session,
- changer et annuler des entrées de référentiel,
- rechercher des entrées de référentiel,
- surveiller les modifications apportées aux entrées du référentiel (y compris la suppression d'entrées).

Chaque entrée de référentiel est identifiée par une clé unique qui identifie la session de protocole d'application à laquelle est associée l'entrée. La clé comprend aussi un Resource ID (identificateur de ressource), défini par le protocole d'application, pour décrire l'utilisation de l'entrée de référentiel dans cette session.

Chaque entrée de référentiel a un propriétaire désigné: l'APE qui a créé l'entrée. Si le propriétaire d'une entrée sort de la conférence, l'entrée n'a plus de propriétaire. Les entrées dotées d'un propriétaire ne peuvent être supprimées que par le propriétaire correspondant; les entrées sans propriétaire peuvent être supprimées par toute APE. Les entrées de canaux et de jetons ne peuvent pas subir de modifications. Les entrées de paramètres dotées d'un propriétaire peuvent être modifiées par le propriétaire correspondant. Les entrées de paramètres sans propriétaire peuvent être modifiées par toute APE; celle-ci devient le nouveau propriétaire de cette entrée. Les entrées de référentiel demeurent pendant la durée utile de la session et les propriétaires d'entrées de référentiel doivent faire preuve de soin pour garantir que ces entrées de référentiel sont annulées quand elles ne sont plus nécessaires. Ceci est particulièrement vrai dans le cas de la session d'enregistrement, où les entrées redondantes sont susceptibles de demeurer indéfiniment, à moins d'être explicitement annulées par le propriétaire. On remarquera que, dans certains cas, il pourra être inapproprié ou impossible pour les APE sortantes de supprimer les entrées qu'elles ont créées.

Toute communication entre une APE et le référentiel d'application est effectuée par l'ARM.

7.6 Présidence

L'ARM est chargé de notifier à son ASE et à son application d'utilisateur le moment où la conférence entre dans le mode présidentiel ou en sort, ainsi que toute modification dans le nœud de présidence. D'autres opérations en mode présidentiel sont effectuées directement par l'application d'utilisateur ou par l'ASE.

Une application d'utilisateur devra émettre les demandes pour que le nœud local devienne président GCC, pour transférer la présidence ou abandonner la présidence, à travers son contrôleur nodal local.

Le *président de session* est une APE au nœud de présidence, dont le Conducting Operation Flag (indicateur de présidence) est mis à «vrai» dans le répertoire d'application pour cette session. Le contrôleur nodal du nœud de présidence fait fonction de président de session par procuration en l'absence d'une telle APE.

Si une APE assume le rôle de président de session, son ASE arbitre entre les demandes d'action des ASE homologues. De telles demandes et les réponses associées seront émises en utilisant des PDU spécifiées par le protocole d'application. Celles-ci peuvent être utilisées pour attribuer des privilèges différents à chaque ASE de la session.

Si un contrôleur nodal devient président de session, il ne peut octroyer ou refuser l'autorisation d'agir que nœud par nœud. Ceci entraîne un contrôle beaucoup plus étroit des ASE. Un ASE peut demander l'autorisation d'agir à travers son contrôleur nodal local, qui, à son tour, émet une GCC-Conductor-Permission-Ask (demande d'autorisation du président). Si le contrôleur nodal présidentiel émet une GCC-Conductor-Permission-Grant (octroi d'autorisation du président) qui inclut le nœud demandeur dans la «List of Nodes Granted Permission» (liste de nœuds autorisés), alors tous les ASE de ce nœud peuvent agir.

Les effets de la présidence sont détaillés par chaque spécification de protocole d'application.

7.7 Invocation à distance

Une application d'utilisateur qui désire invoquer dynamiquement une application homologue à un autre nœud et l'inviter à se brancher sur la session d'application le fera à travers l'ARM. L'application d'utilisateur doit fournir les informations suivantes pour chacun des protocoles d'application qu'elle souhaite utiliser:

- clé de session;
- type du canal de démarrage (statique, dynamique multidestinataire, dynamique privé ou dynamique «ID d'utilisateur»);
- ensemble des capacités minimales;
- liste des nœuds à inviter à la session d'application.

L'ARM émet alors une demande *GCC-Application-Invoke* (invocation d'application) qui précise une liste de GCC Node ID (identificateurs de nœuds GCC) des nœuds à inviter, le type du canal de démarrage, une liste de sessions de protocole d'application auxquelles s'inscrire et, pour chaque protocole d'application, l'ensemble de capacités prévu (c'est-à-dire la capacité minimale).

Les indications d'invocation (*GCC-Application-Invoke*) sont remises au contrôleur nodal qui est responsable de la détermination des APE à activer.

8 Directives relatives aux applications d'utilisateur

Le présent article offre des directives sur l'utilisation de l'infrastructure T.120 afin d'assurer que le comportement des applications soit cohérent et prévisible.

8.1 Identification des applications d'utilisateur

Dans certaines circonstances, il peut être utile de pouvoir déterminer les nœuds qui hébergent une application d'utilisateur particulière ou la gamme des applications d'utilisateur disponibles sur un nœud distant. Ce peut être nécessaire pour l'invocation, locale ou à distance, d'applications d'utilisateur. Une APE peut, à titre facultatif, choisir d'identifier son application d'utilisateur en incluant un paramètre de capacité non fédérable hors norme dans la demande d'inscription d'application (*GCC-Application-Enroll*). En consultant le répertoire d'applications (*Application Roster*) à travers son APE, une application d'utilisateur peut ainsi déceler la présence de la même application d'utilisateur sur d'autres nœuds. Un contrôleur nodal recevant une indication d'invocation d'application (*GCC-Application-Invoke*) peut choisir de consulter les entrées de capacité d'application d'utilisateur dans le répertoire d'applications pour la ou les sessions appropriées, afin de déterminer quelles sont les applications d'utilisateur actives sur d'autres nœuds avant d'activer localement une application d'utilisateur. On remarquera qu'un contrôleur nodal ne sera à même de déceler la présence d'une application d'utilisateur donnée que s'il est informé du paramètre de capacité non fédérable annoncé par cette application.

Des clés de protocole d'application hors norme ne peuvent être utilisées que pour annoncer la prise en charge d'un protocole d'application non normalisé. L'inscription avec une clé de protocole d'application hors norme pour annoncer la présence d'une application d'utilisateur spécifique est fortement déconseillée, du fait que cela empêcherait l'interopérabilité avec d'autres applications d'utilisateur qui emploient le même protocole d'application dans le même but.

8.2 Utilisation de capacités fédérables et non fédérables

Une APE peut utiliser le champ «capacités non fédérables» de la demande d'inscription d'application (*GCC-Application-Enroll*) pour identifier son application d'utilisateur et annoncer les capacités spécifiques de son protocole d'application.

L'utilisation d'un paramètre de capacités non fédérables dans la demande d'inscription d'application (*GCC-Application-Enroll*) est appropriée s'il est besoin d'identifier les APE homologues *précises* qui prennent en charge une capacité donnée, afin de garantir l'interopérabilité. Ce paramètre peut encore être utilisé pour annoncer à la conférence la disponibilité d'une ressource (par exemple, une caméra disponible pour la télécommande). Si l'APE a seulement besoin de déterminer le *nombre* des APE homologues qui prennent en charge une capacité spécifique, alors elle devrait utiliser une capacité fédérable. Pour éviter de rendre le répertoire d'applications (*Application Roster*) inutilement grand, il est fortement recommandé de maintenir le nombre des capacités non fédérables à un minimum absolu.

8.3 Contexte de session

Si une conférence a de multiples sessions simultanées d'un protocole d'application spécifique, il est besoin de distinguer ces sessions de sorte que les utilisateurs puissent choisir, pour y participer, la session la plus appropriée. Habituellement, une session sera utilisée dans un contexte précis qui sera déterminé par les participants à cette session. Par exemple, une session de panneau de téléaffichage (*whiteboard*) T.126 peut être utilisée pour présenter un sujet et une autre pourrait servir à présenter un second sujet. Leur contexte différent permet de distinguer ces sessions.

Un paramètre de référentiel d'application GCC (*GCC Application Registry Parameter*) peut, à titre facultatif, être utilisé pour décrire le contexte d'une session de protocole d'application spécifique. Un identificateur de ressource (*Resource ID*) (construit par codage des 7 caractères du mot «CONTEXT» en octets successifs conformément à la Recommandation T.50) est réservé à cet effet. Le contenu du paramètre de référentiel est une chaîne lisible par l'utilisateur et qui devra être codée comme une chaîne non restreinte d'équivalence binaire (*BMPString*) en utilisant les règles de codage condensées (*Packed Encoding Rules*) dans la variante *Basic Aligned*. La chaîne binaire codée résultante est placée dans la CHAÎNE D'OCTETS (*OCTET STRING*) dans un ordre tel que, pour chaque octet, le bit de tête soit mis dans la position du bit de poids fort et le bit de fin, dans la position du bit de poids faible.

8.4 Choix d'une session pour y participer

Si un utilisateur désire simplement communiquer avec *tout* nœud prenant en charge *le même protocole d'application(s) dans le même but*, alors la session dont l'utilisation est la plus appropriée est:

- la session de base normalisée, pour les protocoles d'application normalisés (voir 8.3);
- la session de base non normalisée, pour les protocoles d'application non normalisés.

Les sessions de base sont destinées aux communications à usage général entre tous les terminaux qui prennent en charge un protocole spécifique (par exemple, la session de base normalisée T.127 sera utilisée pour le transfert de fichiers à usage général).

A remarquer qu'une APE peut être configurée pour s'inscrire dans une session de base sans intervention de l'utilisateur, permettant ainsi aux applications d'utilisateur fondées sur des protocoles normalisés ou hors norme de démarrer automatiquement lors du branchement sur une conférence.

Si plusieurs sessions actives d'un protocole d'application donné sont présentes dans une conférence, un utilisateur peut devoir intervenir. L'utilisateur a besoin d'informations suffisantes pour choisir la ou les sessions les plus appropriées auxquelles participer, ou pour décider s'il est nécessaire d'établir une nouvelle session. Cette décision peut se fonder sur un ou plusieurs d'entre les critères suivants:

- le contexte de session (par exemple, le thème de discussion);
- quels nœuds prennent part à la session;
- l'ensemble des capacités négociées de la session.

8.5 Contrôle de l'utilisation d'une session

Le créateur d'une session peut désirer imposer des contraintes sur une session, afin de s'assurer que des APE de moindre capacité ne s'inscrivent pas dans la session par inadvertance et ne dégradent pas potentiellement les capacités de la session à un niveau jugé inadmissible par le créateur de la session. Si un utilisateur désire imposer un *ensemble minimal de capacités* à une session, une session privée dynamique sera utilisée. Le créateur de la session spécifie les critères de capacités requises dans la demande d'invocation d'application (*GCC-Application-Invoke*).

Si un utilisateur désire indiquer qu'une session doit être utilisée dans un *contexte spécifique*, une session dynamique devra être utilisée et un contexte de session (chaîne de caractères lisible par l'utilisateur) sera fourni dans le référentiel d'application GCC (*GCC Application Registry*) pour cette session.

8.6 Ajout de nouveaux participants à une session

La primitive de demande d'invocation d'application (*GCC-Application-Invoke*) peut être utilisée, à tout moment, par l'organisateur d'une session privée pour inviter à cette session des nœuds supplémentaires (par exemple, des participants tard venus). Tout participant à une session de base normalisée, à une session de base non normalisée ou à une session publique peut, à tout moment, inviter de nouveaux nœuds à cette session en utilisant la demande d'invocation d'application (*GCC-Application-Invoke*). Un contrôleur nodal recevant une indication d'invocation d'application peut choisir de l'ignorer s'il y a déjà, à ce nœud, une APE en état actif dans la session identifiée.

8.7 Extension, modification et réutilisation de protocoles d'application

Les protocoles d'application sont des outils qui peuvent être étendus, modifiés ou personnalisés en vue d'une utilisation spécifique. Pour déterminer si de tels changements aboutissent à un nouveau protocole distinct ou ne sont qu'un simple développement du protocole existant, il faut déterminer si les APE se conformant au protocole d'application modifié sont destinées à s'interconnecter avec des APE conformes au protocole d'application originel. Si l'interconnexion est prévue, on pourra considérer le protocole révisé comme une extension du protocole existant; sinon, c'est un nouveau protocole d'application, auquel il faut attribuer une clé et des ressources de protocole d'application uniques.

Si une APE offre, à un protocole d'application, des extensions privées telles *qu'il est destiné à pouvoir s'interconnecter avec des APE conformes au protocole originel*, il s'inscrira avec la clé de protocole d'application du protocole originel et annoncera ses fonctionnalités étendues en utilisant des capacités fédérables ou non fédérables, conformément aux directives spécifiées au 8.2.

Lorsqu'un protocole d'application est destiné à être utilisé dans un but distinct ou est modifié de façon telle *que les APE se conformant au protocole révisé ne soient pas destinées à pouvoir s'interconnecter avec des APE conformes au protocole originel*, alors le protocole révisé est censé être un nouveau protocole d'application et doit se voir attribuer une clé unique de protocole d'application, pour le différencier du protocole originel. Tous les canaux statiques et les jetons utilisés par le protocole d'application originel ne pourront pas être utilisés par le protocole révisé. Ces règles s'appliquent également si l'utilisation d'un protocole d'application est soumise à de quelconques contraintes. Par exemple, des APE employant les méthodes (les PDU et procédures) de T.127 exclusivement pour le transfert de documents entre des applications de traitement de texte ne s'inscriront pas avec la clé d'application T.127 si elles ne sont pas destinées à pouvoir s'interconnecter avec une APE de transfert de fichiers à usage général; elles n'utiliseront pas davantage les canaux statiques ni les jetons réservés pour la Recommandation T.127.

Pareillement, si plusieurs protocoles d'application doivent collaborer pour réaliser certaines fonctionnalités (par exemple, un protocole d'application de carte de paiement réutilisant les méthodes de T.126 et de T.127), une ou plusieurs nouvelles clés d'application sont requises si les APE composantes sont seulement destinées à s'interconnecter avec d'autres APE qui collaborent dans le même but.

Là où des protocoles d'application sont combinés en vue de réaliser une fonction spécifique, la démarche préférée consiste à définir un seul nouveau protocole d'application, qui est un assemblage des protocoles composants. Cela crée, en fait, une nouvelle APE par fusion des APE préexistantes. Il pourra être nécessaire de résoudre des conflits d'identificateurs de capacité et d'identificateurs de ressource de référentiel GCC (*GCC Registry Resource ID*) entre les protocoles d'application pour la nouvelle APE.

Une autre démarche consiste à garder les APE préexistantes en tant que composantes distinctes, chacune avec une nouvelle clé de protocole d'application pour la distinguer du protocole dont elle a été dérivée. Toutefois, on observera que, dans ce cas, il est nécessaire de définir un moyen d'associer des sessions de protocole d'application qui coopèrent pour réaliser un but commun, de telle sorte que les applications d'utilisateur puissent déterminer la combinaison correcte des sessions auxquelles leurs APE s'inscriront. Pour éviter les doublons, il est suggéré que la liste des sessions associées soit tenue à jour dans le référentiel d'application (*Application Registry*) pour l'une des sessions, qui sera dénommée *session racine*. Il revient au réalisateur du protocole d'application de déterminer quel protocole d'application devrait servir de session racine; un choix approprié pourrait être celui du protocole d'application dont les fonctionnalités correspondent le plus étroitement au but commun prédéfini.

SERIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Réseau téléphonique et RNIS
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission
Série H	Transmission des signaux autres que téléphoniques
Série I	Réseau numérique avec intégration des services
Série J	Transmission des signaux radiophoniques et télévisuels
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Maintenance: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophoniques et télévisuels
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie alphabétique
Série T	Equipements terminaux et protocoles des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Z	Langages de programmation