UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T Q.814

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(02/2000)

SÉRIE Q: COMMUTATION ET SIGNALISATION

Spécifications du système de signalisation n° 7 – Interface Q3

Spécification d'un agent interactif d'échange informatisé de données

Recommandation UIT-T Q.814

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE Q

COMMUTATION ET SIGNALISATION

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DES SYSTÈMES DE SIGNALISATION N° 4 ET N° 5	Q.120–Q.249
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 6	Q.250–Q.309
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R1	Q.310–Q.399
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R2	Q.400–Q.499
COMMUTATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.849
Généralités	Q.700
Sous-système transport de messages	Q.701–Q.709
Sous-système commande des connexions sémaphores	Q.711–Q.719
Sous-système utilisateur de téléphonie	Q.720–Q.729
Services complémentaires du RNIS	Q.730–Q.739
Sous-système utilisateur de données	Q.740–Q.749
Gestion du système de signalisation n° 7	Q.750–Q.759
Sous-système utilisateur du RNIS	Q.760–Q.769
Sous-système application de gestion des transactions	Q.770–Q.779
Spécification des tests	Q.780–Q.799
Interface Q3	**Q.800–Q.849**
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1699
PRESCRIPTIONS ET PROTOCOLES DE SIGNALISATION POUR LES IMT-2000	Q.1700–Q.1799
RNIS À LARGE BANDE	Q.2000–Q.2999

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

Recommandation UIT-T Q.814

Spécification d'un agent interactif d'échange informatisé de données

Résumé

La présente Recommandation UIT-T définit la spécification technique d'un module de protocole de couche Session appelé agent interactif de communications électroniques. Il peut être utilisé comme point de référence d'interface dans un modèle RGT pour l'échange asynchrone de données entre des entités d'application homologues. L'agent interactif (IA) prend en charge le déroulement de transactions par échange informatisé de données (EDIFACT ou ASC X12 EDI) quasiment en temps réel. De plus, la présente Recommandation UIT-T définit l'architecture, la conception, la structure et l'enchaînement des opérations pour les fonctions commerciales *normales*[1] et *hautement prioritaires*[2] utilisant la sécurité de la couche de transport (TLS).

Source

La Recommandation Q.814 de l'UIT-T, élaborée par la Commission d'études 4 (1997-2000) de l'UIT-T, a été approuvée le 4 février 2000 selon la procédure définie dans la Résolution 1 de la CMNT.

[1] *Priorité normale* – Une fonction commerciale de priorité normale correspond par exemple à une transaction concernant une demande de commande.

[2] *Hautement prioritaire* – Une fonction commerciale hautement prioritaire correspond par exemple à une transaction concernant une demande de renseignements interactive.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'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'étude 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 1 de la CMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

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.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

TABLE DES MATIÈRES

Introduction

La présente Recommandation UIT-T définit les spécifications relatives à un agent interactif (IA, *interactive agent*) d'échange informatisé de données. L'agent IA prend en charge le déroulement de transactions par échange informatisé de données (EDI, *electronic data interchange*) entre des entités homologues. Il met en correspondance les transactions EDI dans la couche de transport. De manière plus spécifique, il sert d'interface avec la sécurité de la couche de transport (TLS, *transport layer security*) pour les demandes d'établissement et de fermeture de sessions TCP sûres (c'est-à-dire en prenant en charge l'authentification d'entités homologues, l'intégrité et la confidentialité) et le transport sûr de messages EDI. L'agent IA fournit également une fonctionnalité de commande de flux de base.

Recommandation UIT-T Q.814

Spécification d'un agent interactif d'échange informatisé de données

1 Domaine d'application

La présente Recommandation UIT-T fournit une spécification pour l'agent interactif (IA) d'échange informatisé de données. L'agent IA prend en charge le déroulement de transactions par échange informatisé de données (EDIFACT/ASC X12 EDI) sur un réseau de protocole de commande de transmission/protocole Internet (TCP/IP, *transmission control protocol/Internet protocol*) utilisant la sécurité de la couche de transport (TLS). Le présent document spécifie l'architecture générale de l'agent IA, la syntaxe des formats de message à utiliser, les règles de codage pour les messages et les transformations de sécurité applicables.

2 Références

2.1 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée.

– Recommandation UIT-T Q.815 (2000), *Spécification d'un module de sécurité pour la protection globale des messages*.

– Recommandation UIT-T X.509 (1997) | ISO/CEI 9495-8:1998, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: cadre d'authentification*.

– Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.

– Recommandation UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels*.

– Recommandation UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes*.

– Recommandation UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un*.

– Recommandation UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives*.

Société Internet/Groupe de travail d'ingénierie Internet (*Internet Society/Internet Engineering Task Force*):

– RFC 2246, *The TLS Protocol Version 1.0*.

2.2 Référence informatives

– Directory Implementors Guide (Version 12) (1999).

3 Définitions

La présente Recommandation UIT-T définit les termes suivants:

3.1 protocole de transfert d'agent interactif (IATP, *interactive agent transfer protocol*): le protocole IATP est utilisé entre des agents interactifs homologues voulant effectuer des transactions/envoyer des messages d'échanges informatisé de données via le protocole de commande de transmission/protocole Internet (TCP/IP) en utilisant la sécurité de la couche de transport.

3.2 traducteur EDI: un traducteur EDI est généralement un programme ou un module logiciel qui convertit des formats et des représentations de données privés en formats et en représentations normalisés, tels que ceux spécifiés dans l'ISO 9735 ou l'ANSI ASC X.12, et réciproquement.

3.3 agent interactif *(IA, interactive agent)*: l'agent interactif prend en charge le déroulement de transactions (UN/EDIFACT ou ASC X12 EDI) par échange informatisé de données entre des entités homologues. L'agent IA fonctionne comme une interface entre son utilisateur direct (généralement un traducteur ou un module de sécurité EDIFACT/ASC X12 EDI) et la sécurité de la couche de transport. Plusieurs approches concernant l'implémentation peuvent être envisagées, allant d'une simple interface de programme d'application (API, a*pplication program interface)* jusqu'à un programme autonome. L'agent IA est décrit dans la Recommandation UIT-T Q.814 et le module de sécurité est décrit dans la Recommandation UIT-T Q.815.

3.4 sécurité de la couche de transport (TLS, *transport layer security*): le protocole TLS fournit en option la confidentialité des communications. Ce protocole permet aux applications client/serveur de communiquer de manière à empêcher toute écoute illégale, altération ou intrusion. Le protocole TLS permet également une authentification d'homologues et une intégrité de circulation des données efficaces.

4 Abréviations

La présente Recommandation UIT-T utilise les abréviations suivantes:

IA agent interactif (*interactive agent*)

IATP protocole de transfert d'agent interactif (*interactive agent transfer protocol*)

MD résumé de message (*message digest*)

SHA-1 algorithme de hachage de sécurité, révision 1 (*secure hashing algorithm, revision 1*)

SM module de sécurité (*security module*)

TLS sécurité de la couche de transport (*transport layer security*)

WAN réseau régional (*wide area network*)

5 Conventions

Les conventions suivantes sont utilisées dans la présente Recommandation UIT-T:

Le terme *EDI*, tel qu'il est utilisé dans la présente Recommandation UIT-T, se réfère à l'un quelconque ou à tous les termes suivants:

- UN/EDIFACT, tel que défini par le Département Commerce de l'ONU/CEE, et adopté par le Comité Technique ISO/TC 154
- EDIFACT, tel que défini dans l'ISO 9735

NOTE – EDI, tel que défini dans l'ANSI ASC X12 est également inclus.

Le Tableau 1 du 6.2.2 utilise les conventions suivantes:

M = obligatoire (*mandatory*)

O = facultatif (*optional*)

Toutes les lignes de code en *langage C* figurant dans la présente Recommandation UIT-T ne sont indiquées qu'à titre d'exemple.

6 Architecture et caractéristiques de service

6.1 Architecture

L'agent IA fonctionne comme une interface entre son utilisateur direct (généralement un traducteur EDIFACT ou ASC X12 EDI) et la couche de transport. (Voir Figure 1.) La sécurité de base des transactions EDI est fournie par le protocole TLS. Des possibilités de sécurité supplémentaires (par exemple non-répudiation) peuvent être fournies par un module de sécurité séparé qui effectue des transformations de sécurité sur des messages EDI complets. Un tel module de sécurité peut également être un utilisateur direct de l'agent IA, comme illustré à la Figure 2.

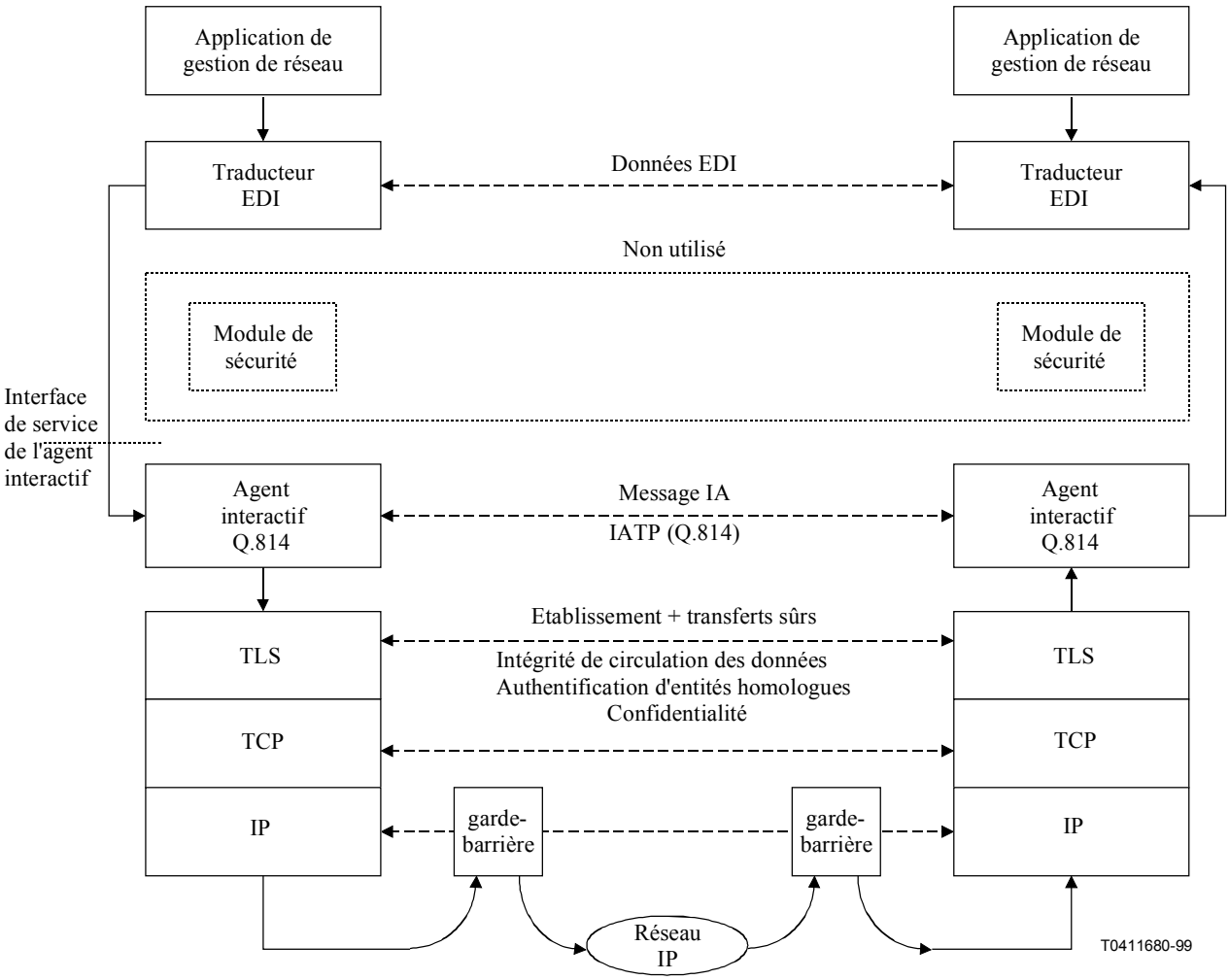

Figure 1/Q.814 – Relations pendant la circulation des messages (sans module de sécurité)

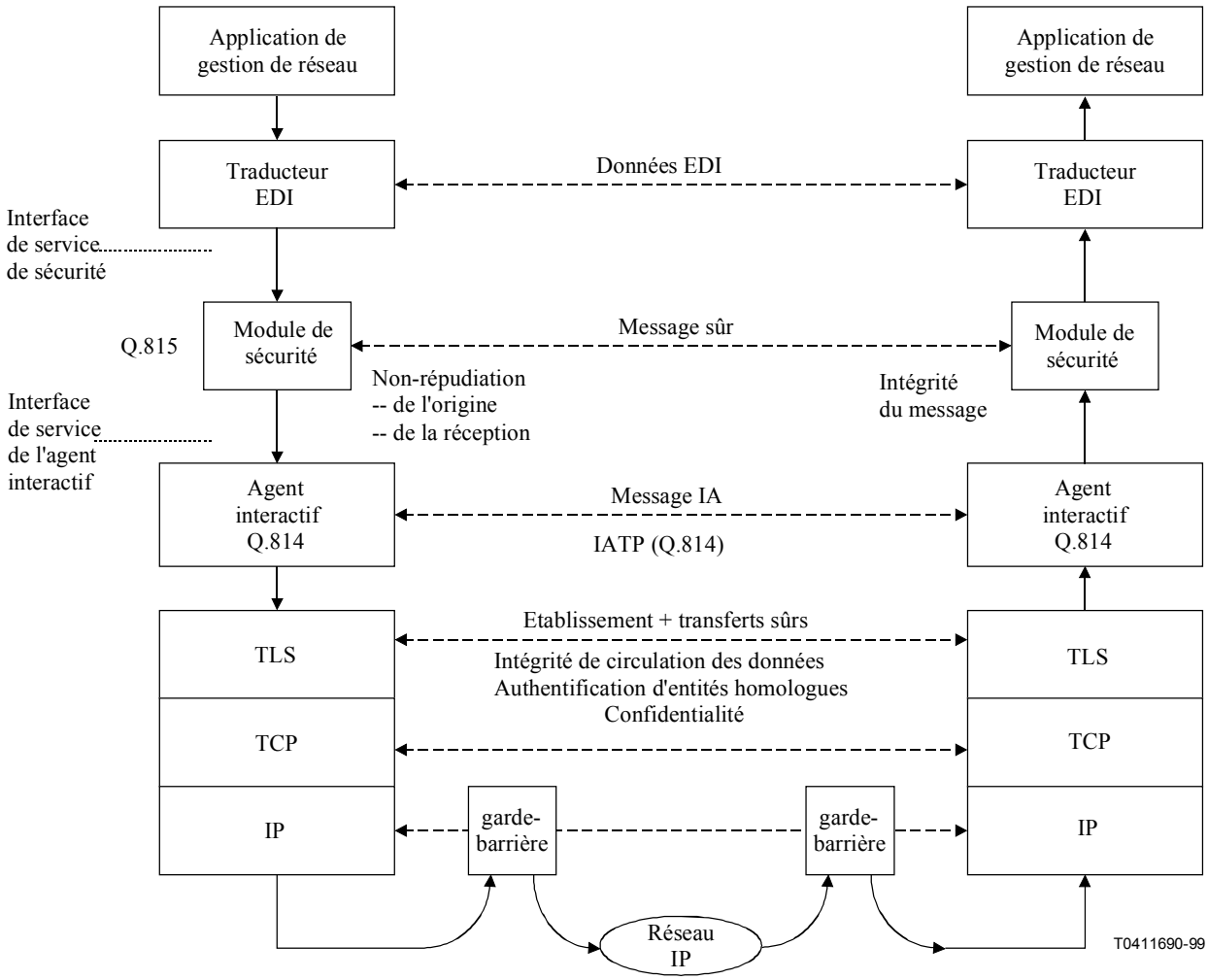

Figure 2/Q.814 – Relations pendant la circulation des messages (avec services de sécurité des messages)

La structure sous-jacente de l'agent IA est une configuration client/serveur symétrique dans laquelle les fonctions client et serveur sont nécessaires à chaque implémentation. Voir Figure 3.

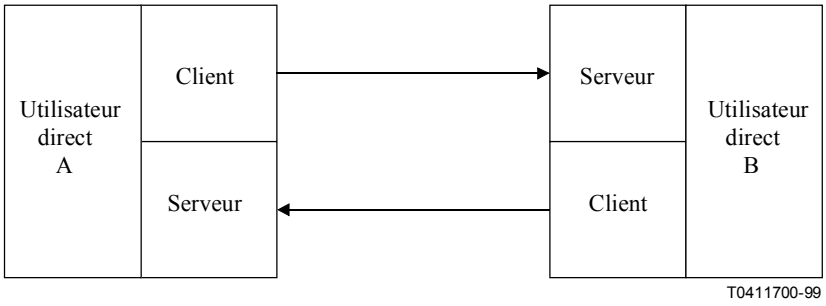

Figure 3/Q.814 – Architecture de l'agent interactif

NOTE – L'architecture client/serveur symétrique peut ne pas être appropriée pour certains types d'applications *en temps quasi réel*.

L'agent IA remplit des fonctions de la couche Session du modèle OSI à sept couches. Les fonctions de la couche Session fournies par l'agent IA comprennent l'établissement, la gestion et la fermeture de sessions de communications entre des entités homologues. L'agent IA peut également effectuer la conversion d'identités EDIFACT/ASC X12 EDI en adresses de réseau et gérer la session de la couche de transport. A l'issue d'une session, l'agent IA détermine si la session doit être clôturée ou suspendue dans un état permettant sa *reprise*.

6.2 Caractéristiques de service

L'agent IA peut présenter les services suivants à son utilisateur direct ou à des agents IA homologues:

- nom de destinataire IA (*IA recipient name*);
- priorité IA (*IA priority*);
- message de base IA (*IA basic message*);
- message avancé IA (*IA enhanced message*);
- commande IA (*IA control*);
- journalisation IA (*IA logging*).

6.2.1 Eléments de service

6.2.1.1 Nom de destinataire IA

L'utilisateur direct fournit l'identité du destinataire du message transféré à l'agent IA (par exemple l'identité du partenaire commercial dans un message EDI fondé sur ASC X12). L'agent IA convertit cette valeur en une adresse de la couche de transport.

6.2.1.2 Priorité IA

L'utilisateur direct fournit un indicateur de priorité du message à l'agent IA. L'agent IA gère des messages *normaux* et *hautement* prioritaires. L'indicateur de priorité est converti en une adresse d'accès de destination et combiné avec l'adresse de la couche de transport afin d'établir une demande de connexion de la couche Session.

6.2.1.3 Message de base IA

L'utilisateur direct fournit le contenu d'un message de données non avancé à l'agent IA.

6.2.1.4 Commande IA

L'utilisateur direct demande à l'agent IA de transmettre des informations de commande de session à l'entité homologue.

6.2.1.5 Message avancé IA

L'utilisateur direct fournit un message de données avancé à l'agent IA.

6.2.1.6 Journalisation (question locale)

L'utilisateur direct peut spécifier le mécanisme, les classifications et le contenu des données à journaliser.

6.2.2 Classifications des éléments de service

Voir Tableau 1.

Tableau 1/Q.814 – Classifications des éléments de service

Classe de service	Origine	Réception
Nom de destinataire IA	M	M
Priorité IA	O	M
Message de base IA	O[a)]	M
Message avancé IA	O[a)]	M
Commande IA	O	M
Journalisation IA (question locale)	O	O
[a)] Au moins l'un des deux doit être choisi.		

7 Circulation des données

Le présent paragraphe illustre l'établissement d'une session entre un client et un serveur. La Figure 4 illustre l'établissement d'une session TLS complète. La Figure 5 illustre l'établissement d'une session abrégée, également appelée session "reprise".

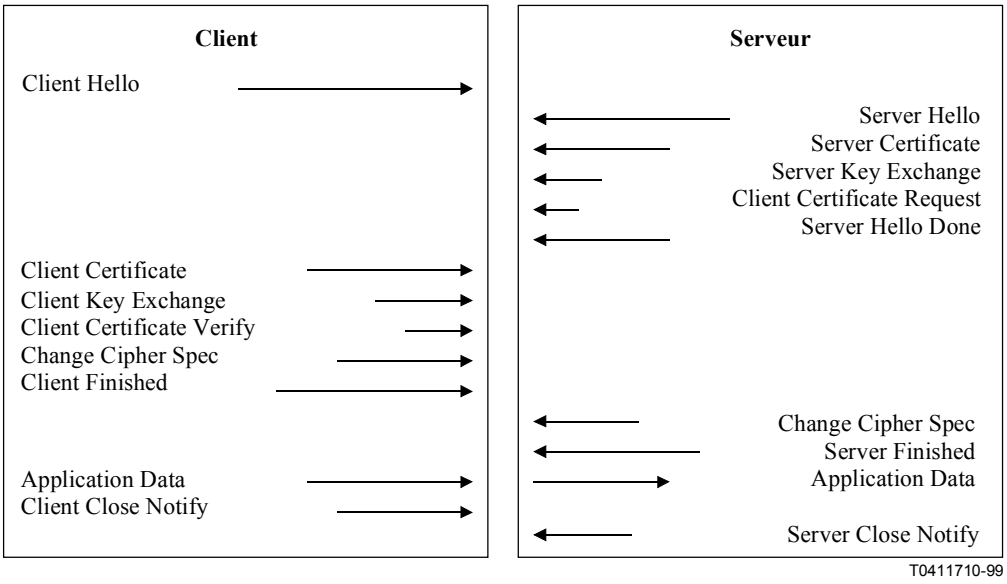

Figure 4/Q.814 – Circulation des messages IA – Etablissement d'une session TLS complète

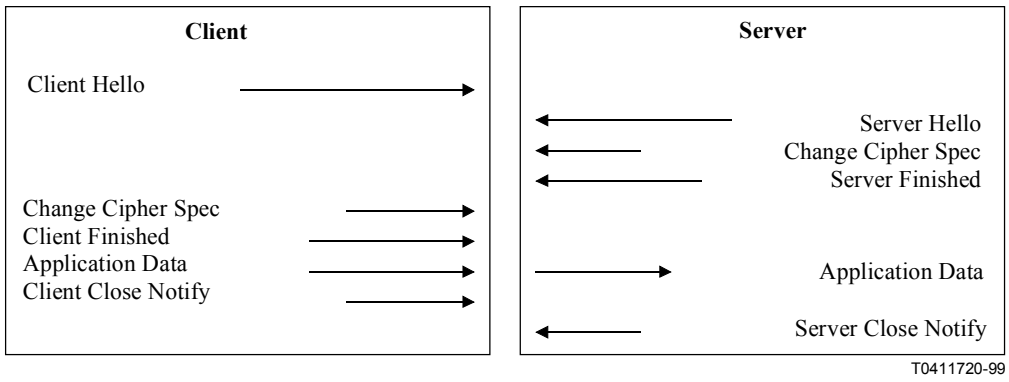

Figure 5/Q.814 – Circulation des messages IA – Etablissement d'une session TLS abrégée

8 Messages IA

Afin de répondre à la fois aux besoins commerciaux et aux besoins de sécurité d'une entité, deux types de message ont été définis. Ces types de message sont les suivants:

- message de base (intégrité de la circulation des données et/ou chiffrement de confidentialité);
- message avancé (intégrité du message complet ou non-répudiation).

8.1 Définitions des formats de message

Les exigences relatives aux formats de message discret sont liées à chacun des deux formats de message définis ci-dessus. De plus, un troisième format a été défini pour la communication de l'état/de la commande entre les agents interactifs, à savoir le message d'état IA. Chacun de ces formats est codé conformément à la définition de la notation de syntaxe abstraite numéro un (ASN.1).

Les règles de codage distinctives (DER) pour la notation ASN.1 doivent être utilisées pour coder les messages IA. Seule la méthode de codage de champs de longueur utilisant des formes de longueur définie doit être utilisée.

8.2 Définitions des syntaxes de message

L'agent interactif peut être utilisé pour transporter des données EDIFACT, ASC X12 EDI et/ou toute autre forme de données de chaîne de caractères. Les définitions de la notation ASN.1 pour chacun des types de message gérés sont données dans le présent paragraphe.

La syntaxe générale se présente comme suit:

```
IaMessage ::= CHOICE {
        basicMessage        BasicMessage,       -- Message de base
        iaStatusMessage     IaStatusMessage,    -- Message d'état de commande IA
        enhancedMessage     EnhancedMessage     -- Message avancé de module de sécurité
}
```

8.2.1 Message de base

Le message de base est choisi pour acheminer le texte des messages IA dans l'un ou l'autre des deux répertoires de caractères.

```
BasicMessage ::= CHOICE {
basicMessage1     GeneralString,
basicMessage2      IA5String
}
```

8.2.2 Message d'état/de commande IA

Le message d'état peut être utilisé pour permettre l'échange de commande de la circulation ou de conditions d'erreurs entre des agents IA (voir 8.3).

```
IaStatusMessage ::= BIT STRING ( SIZE (32) )
```

8.2.3 Message avancé

La syntaxe du message avancé est choisie pour acheminer de manière transparente des messages qui ont été codés par une fonction de sécurité de couche supérieure. Généralement, ce service est fourni par la Recommandation UIT-T Q.815.

```
EnhancedMessage ::= OCTET STRING        -- Message avancé de module de sécurité
```

8.3 Format détaillé de message d'état IA

Le message d'état IA comprend quatre octets contenant des informations sur l'état. Un résumé des valeurs du message d'état IA est présenté dans le Tableau 2. Les quatre octets transportant les informations sur l'état sont structurés comme suit:

8.3.1 Premier Octet

Utilisé pour les conditions d'état portant sur une interface étendue de l'agent IA (c'est-à-dire TCP/IP et TLS).

Structure de codage

Cet octet doit être interprété comme deux valeurs hexadécimales dans la gamme 00 à FF.

Les valeurs de cet octet ne doivent être définies que dans la présente Recommandation UIT-T.

Les valeurs actuellement définies comprennent:

00 = Ignoré

08 = Demande à l'homologue de cesser toute transmission sur le flux de données entrant en raison de problèmes avec l'interface WAN

Aucune autre valeur n'est actuellement définie.

8.3.2 Second Octet

Utilisé pour les conditions d'état portant sur les systèmes dans le sens descendant sur l'interface locale (c'est-à-dire le traducteur EDI ou autre utilisateur direct associé, récepteur de l'agent IA).

Structure de codage

Cet octet doit être interprété comme deux valeurs hexadécimales dans la gamme 00 à FF.

Les valeurs de cet octet ne doivent être définies que dans la présente Recommandation UIT-T.

Les valeurs actuellement définies comprennent:

00 = Ignoré

08 = Demande à l'homologue de cesser toute transmission sur le flux de données entrant en raison de problèmes de communication entre l'agent IA et le traducteur EDI ou autres systèmes de traitement dans le sens descendant.

Aucune autre valeur n'est actuellement définie.

8.3.3 Troisième et quatrième octets

Lorsque les octets un, deux, trois et quatre sont tous à zéro, cela doit signifier "Message d'essai".

Lorsque les octets un et deux sont à zéro, les entités homologues peuvent définir les octets trois et quatre. Les entités homologues peuvent librement définir les valeurs de ces octets. Il est recommandé que les valeurs utilisées soient deux valeurs hexadécimales par octet.

Lorsque l'octet un ou l'octet deux n'est pas à zéro, les valeurs des octets trois et quatre sont alors réservées pour une normalisation future.

8.3.4 Message d'essai spécial

Un message contenant quatre octets de valeur hexadécimale 00 00 00 00 est défini comme un message NULL qui peut être utilisé comme message d'essai. Aucune action ne doit être entreprise en réponse à la réception d'un tel message.

8.3.5 Message non valide

Le message suivant est non valide et doit être ignoré s'il est reçu:

08 08 xx xx

Tableau 2/Q.814 – Valeurs de message d'état IA

Octet 1	Octet 2	Octet 3	Octet 4	Utilisation
00	00	00	00	Message d'essai
08	00	00	00	Encombrement des couches inférieures
00	08	00	00	Encombrement des couches supérieures
08	00	xx	xx	Réservé pour utilisation future
00	08	xx	xx	Réservé pour utilisation future
00	00	yy	yy	Défini par l'utilisateur
08	08	xx	xx	Message non valide ignoré
yy	00	xx	xx	Réservé
00	yy	xx	xx	Réservé
NOTE 1 – xx = 00-FF. NOTE 2 – yy = 01-FF.				

9 Spécifications du client

Le présent paragraphe définit le traitement associé au processus du client de l'agent interactif.

A la réception de données provenant de son utilisateur direct sur l'interface de service IA (par exemple le traducteur EDIFACT/ASC X12 EDI ou le module de sécurité), le client suit les

étapes suivantes. Toutefois, avant d'appliquer ce processus, l'agent IA détermine si une demande de cessation de transmission en suspens a été reçue, voir 11.3.

9.1 Détermination de l'adresse de destination IP

Sur la base de l'identité de l'entité homologue et d'autres informations pertinentes, l'adresse IP correspondante et le numéro d'accès sont déterminés pour l'acheminement des données au serveur approprié.

NOTE – Le choix du *numéro d'accès* détermine lequel des deux niveaux de sécurité sera utilisé. Voir le paragraphe 12 concernant les attributions d'accès.

9.2 Connexion au serveur

La séquence d'étapes suivante résume cette opération. Des détails sur les étapes sont fournis à la suite du résumé:

- allocation de la structure de données TLS et allocation de mémoire, si nécessaire
- ouverture d'un point de connexion (*Open Socket*);
- envoi du message TLS de bienvenue du client (*TLS Client Hello*);
- envoi du certificat du client au serveur (*Client's Certificate to Server*);
- échange de clé du client (*Client Key Exchange*);
- envoi du message de vérification du certificat du client (*Client Certificate Verify*);
- exécution des modifications des spécifications de chiffre (*Change Cipher Specs*);
- envoi du message de fin du client (*Client Finished*).

Les détails suivants sont fournis pour chacune des étapes identifiées ci-dessus:

9.2.1 Allocation de la structure de données TLS et allocation de mémoire

Une structure de données est utilisée pour stocker les données de chiffrement associées à une connexion individuelle, y compris certificats, références à des rappels et diverses autres données. Une structure de données indépendante doit être attribuée et conservée pour chacune des connexions.

9.2.2 Ouverture d'un point de connexion

Une application du client crée un point de connexion et établit ensuite une connexion avec un service spécifié dans une structure (par exemple *sockaddr_in)*. L'exemple suivant représente une opération d'ouverture d'un point de connexion en langage "C":

```
int     tcpopen(host,service)
char   *service, *host;
{  int    unit;
   struct  sockaddr_in    sin;
   struct  servent        *sp;
   struct  hostent        *hp;
   if ((sp=getservbyname(service,"tcp")) == NULL) then error…
   if ((hp=gethostbyname(host)) == NULL) then error...
   bzero((char *)&sin, sizeof(sin))
       etc...
   if ((unit=socket(AF_INET,SOCK_STREAM,0)) < 0) then error...
   if (connect(unit,&sin,sizeof(sin)) < 0) then error...
   return(unit);
}
```

Le résultat reçu est un *descripteur de fichier (file descriptor)* connecté à un processus du serveur. Ceci représente une voie de communication sur laquelle peut être appliqué un protocole spécifique à une application.

9.2.3 Envoi du message TLS de bienvenue du client

La fonction envoie la date/l'heure actuelles du client, l'identificateur de session, la liste de suite de chiffres, la liste d'algorithmes de compression, une structure de donnée aléatoire et un paramètre de version du client. Ce paramètre spécifie la ou les versions du protocole TLS qui peuvent être utilisées pour la connexion. Il convient de le régler à une valeur de {3.1} pour le protocole TLS. Après avoir envoyé un message *Client Hello,* le client doit attendre jusqu'à ce qu'il reçoive un message *Server Hello* en réponse.

9.2.4 Envoi du certificat du client au serveur

Le client envoie son certificat numérique au serveur.

9.2.5 Echange de clé du client

Ce message établit la clé-test secrète de 48 octets, effectue son chiffrement avec la clé publique du serveur et envoie les résultats dans un message-test confidentiel chiffré.

9.2.6 Envoi du message de vérification du certificat du client

Ce message est utilisé pour fournir une vérification explicite du certificat du client. Ce message est envoyé uniquement si le certificat du client envoyé peut être signé. Il est envoyé immédiatement après le message *client key exchange*.

9.2.7 Modification des spécifications de chiffre

Ce message est envoyé pour informer le système homologue que les enregistrements ultérieurs seront protégés selon les nouvelles *spécifications de chiffre (cipher specs)* et clés négociées.

9.2.8 Envoi du message de fin du client

Le client envoie alors le message *Client Finished* immédiatement après un message *Change Cipher Specs* pour vérifier que les processus d'échange de clés et d'authentification ont été concluants. Ce message est le premier message envoyé protégé avec les algorithmes, clés et clés secrètes négociés. Le client est prêt à envoyer des données d'application.

9.3 Envoi de données d'application au serveur

Après avoir établi la connexion avec le serveur, le client code le message en utilisant des règles DER et en transmettant le message au serveur comme un flux de données. Ce processus est réalisé au moyen de la fonction d'écriture TLS, comme spécifié par la boîte à outils ou la bibliothèque TLS utilisée.

9.4 Journalisation des transmissions

Des entrées de journalisation sont créées pour enregistrer la transmission de données au serveur. L'ensemble minimal de données à journaliser comprend les éléments suivants:

- date/heure;
- identificateur de message unique (par exemple le segment ISA si ASC X12 EDI);
- adresse IP distante et numéro d'accès de l'homologue;
- indicateur succès/échec avec les interfaces.

9.5 Déconnexion du client

Le client et le serveur doivent tous deux savoir que la connexion prend fin. Le client débute la séquence suivante après avoir écrit toutes les données à l'homologue:

- Exécution de la fonction de notification de fermeture de la session TLS (*TLS Close Notify*):

  cette fonction ferme la session TLS avec l'homologue. Aucune autre transmission n'est prévue. Le client doit attendre la réponse du serveur avant d'effectuer les étapes suivantes de cette séquence.

- Exécution de la fonction de fermeture du point de connexion (*Socket Close*)

  le client ferme le point de connexion avec la fonction *close()*, en utilisant le descripteur du point de connexion comme paramètre.

- Exécution de la fonction de suppression de la structure de données TLS (*TLS Data Structure Delete*):

  cette fonction libère généralement les ressources utilisées par la connexion TLS;

  cette étape est effectuée uniquement si la session doit être fermée, et n'est pas exécutée pour les sessions qui peuvent être reprises ultérieurement. Selon l'implémentation, il peut être nécessaire de supprimer l'allocation de mémoire utilisée par la structure de données après utilisation.

Si le message *Close Notify* du serveur n'est pas reçu, le client en déduit une condition d'erreur et effectue les étapes suivantes:

- exécution de la fonction de fermeture du point de connexion (*Socket Close*);
- exécution de la fonction de suppression de la structure de données TLS (*TLS Data Structure Delete*) et indication que la session ne peut pas être reprise;
- notification d'une défaillance de communication à l'application de l'utilisateur.

10 Spécifications du serveur

Le présent paragraphe définit le traitement associé au processus du serveur de l'agent interactif.

Au début du processus du serveur, celui-ci effectue la séquence d'étapes suivante:

10.1 Initialisation du serveur

Pour accepter les connexions, un point de connexion est créé et relié à un accès de service. Une file d'attente pour les connexions entrantes est spécifiée et les connexions sont acceptées comme illustré dans cet exemple de fragment de code en langage C:

```
struct servent          *sp;
struct sockaddr_in sin,from;
if ((sp=getservbyname(service,"tcp")) == NULL) then error...
sin.sin_family=etc...
if ((s=socket(AF_INET,SOCK_STREAM,0)) < 0) then error...
if (bind(s, &sin, sizeof(sin)) < 0) then error...
if (listen(s,QUELEN) < 0) then error...
for (;;) {
   if ((g=accept(f,&from,&len)) < 0) then error...
   if (!fork()) {
      child handles request...
         ...and exits
      exit(0);
   }
   close(g);   /* parent releases file */
}
```

Lorsque le processus et l'accès sont reliés, le serveur attend des demandes de connexion au niveau de l'accès. A la réception d'une demande de connexion, un point de connexion est créé. Par exemple, lorsqu'un programme en multitraitement est utilisé, les connexions sont établies et le processus crée un *processus enfant* pour gérer cette demande de service. Le *processus parent* continue d'attendre et d'accepter d'autres demandes de service.

10.2 Acceptation de la connexion du client

Sur la base de l'identité de l'entité homologue et autres informations pertinentes (qui pourraient comprendre l'adresse IP distante et le numéro d'accès local), l'acheminement des données est généralement déterminé selon le traducteur EDIFACT/ASC X12 EDI approprié ou le module de sécurité.

10.3 Etablissement de la lecture du message

La séquence d'étapes suivante résume cette opération. Des détails sur les étapes sont fournis à la suite du résumé:

- allocation de la structure de données TLS et allocation de mémoire;
- liaison de la structure de données TLS au point de connexion;
- envoi du message TLS de bienvenue du serveur (*TLS Serveur Hello*);
- envoi du certificat du serveur au client (*Server's Certificate to Client*);
- échange de clé du serveur (*Server Key Exchange*);
- envoi de la demande de certificat du client (*Client Certificate Request*);
- envoi du message de confirmation de bienvenue du serveur (*Server Hello Done*);
- exécution des modifications des spécifications de chiffre (*Execute Change Cipher Specs*);
- envoi du message de fin du serveur (*Server Finished*).

Lorsqu'une demande de connexion entrante est reçue, le processus *d'écoute* du serveur transmet la demande de connexion au logiciel de *traitement* des connexions. Ce logiciel effectue les étapes suivantes:

10.3.1 Allocation de la structure de données TLS et allocation de mémoire

Une structure de données est utilisée pour stocker les données de chiffrement associées à une connexion individuelle, y compris certificats, références à des rappels et diverses autres données. Une structure de données indépendante doit être attribuée et conservée pour chacune des connexions.

10.3.2 Liaison de la structure de données TLS au point de connexion

Cette action connecte logiquement la structure de données TLS au point de connexion sur lequel la demande de connexion entrante est en attente. Les étapes réelles effectuées pour accomplir cette liaison varient en fonction de la boîte à outils TLS et du logiciel de gestion de point de connexion utilisés.

10.3.3 Envoi du message TLS de bienvenue du serveur

Cette fonction envoie l'identificateur de session (soit un nouveau soit la valeur envoyée par le client dans le cas d'une reprise de session), une suite unique de chiffres choisie, un algorithme de compression unique choisi, une structure de données aléatoire (différente de celle du client) et un paramètre de version du serveur. Ce paramètre spécifie la ou les versions du protocole TLS qui seront utilisées pour la connexion. Ce paramètre doit avoir une valeur de {3.1} pour le protocole TLS. Ce message est envoyé en réponse à un message *Client Hello*.

10.3.4 Envoi du certificat du serveur au client

Le serveur envoie le contenu de son certificat X.509 version 3 au client.

10.3.5 Echange de clé du serveur

Ce message doit être envoyé immédiatement après le message de certificat du serveur, si nécessaire. Le message d'échange de clé du serveur est envoyé uniquement lorsque le message de certificat du serveur ne contient pas suffisamment de données pour permettre au client d'échanger une clé-test secrète. Ceci est applicable uniquement si une méthode d'échange de clé RSA_EXPORT a été choisie et si la longueur de la clé publique dans le certificat du serveur est supérieure à 512 bits.

10.3.6 Envoi de demande de certificat du client

Cette fonction demande un certificat au client. Ce message doit être envoyé immédiatement après le message d'échange de clé du serveur (s'il est envoyé), ou le message de certificat du serveur.

10.3.7 Envoi du message de confirmation de bienvenue du serveur

Cette fonction informe le client que le serveur a fini d'envoyer le message Server Hello et les messages associés. Après envoi de ce message, le serveur attend une réponse du client.

10.3.8 Exécution des modifications des spécifications de chiffre

Ce message demande au système homologue d'activer l'ensemble des paramètres de chiffrement reçu le plus récent.

10.3.9 Envoi du message de fin du serveur

Le serveur envoie ensuite le message *Server Finished* immédiatement après le message *Change Cipher Specs* pour vérifier que les processus d'échange de clé et d'authentification ont été concluants. Ce message est le premier message envoyé protégé avec les algorithmes, clés et clés secrètes négociés. Le côté serveur de la connexion est maintenant prêt à recevoir des données.

10.4 Traitement de lecture TLS

Ce processus nécessite d'effectuer la *lecture (read)* TLS initiale de deux octets. Le premier octet contient une représentation DER d'une étiquette de notation ASN.1 identifiant le type de message IA, comme spécifié dans le paragraphe 8. Le second octet est un octet de longueur définie selon les règles DER de forme courte ou le premier octet de longueur définie selon les règles DER de forme longue, représentant le nombre d'octets contenus dans le reste du champ de longueur.

Si le second octet est un octet de longueur DER de forme courte, il représente la longueur du reste du message IA actuel.

Si le second octet est un octet de longueur DER de forme longue, une seconde *lecture* du nombre indiqué d'octets est nécessaire pour déterminer le nombre restant d'octets dans le champ de longueur définie de forme longue. Ces octets restants représentent la longueur totale du reste du message IA.

Il convient alors de procéder à une *lecture* finale pour *lire* tous les octets composant le reste du message IA. Cette troisième lecture peut être effectuée en une opération de *lecture* unique ou en une série de *lectures* partielles dont la somme est égale à la valeur totale. Toutes les *lectures* dans le présent paragraphe sont effectuées au moyen de la fonction de *lecture* TLS, comme spécifié par la boîte à outils ou la bibliothèque TLS utilisée.

Si la *lecture* TLS échoue en raison de ressources insuffisantes, l'agent IA déclenche des procédures de commande de flux, comme spécifié au 11.3.

10.5 Déconnexion du serveur

Le client et le serveur doivent tous deux savoir que la connexion prend fin. Le client signale au serveur qu'il a effectué la transmission du message en cours en envoyant un message *Client Close Notify*. Le serveur doit effectuer le traitement de lecture TLS en attente, il doit ensuite exécuter les procédures suivantes:

- Exécution de la fonction de notification de fermeture de la session TLS (*TLS Close Notify*):

  cette fonction ferme la session TLS avec l'homologue. Aucune autre communication n'est prévue. Cette fonction est généralement effectuée lorsque le client a envoyé le message *Client Close Notify* et lorsque toutes les données entrantes ont été lues à partir du flux de données entrantes.
- Exécution de la fonction de fermeture du point de connexion (*Execute Socket Close*):

  le serveur ferme le point de connexion avec la fonction *close()*, en utilisant le descripteur du point de connexion comme paramètre.
- exécution de la fonction de suppression de la structure de données TLS (*TLS Data Structure Delete*):

  cette fonction libère généralement les ressources utilisées par la connexion TLS. Cette étape est effectuée uniquement si la session doit être fermée, elle n'est pas exécutée pour les sessions qui peuvent être reprises ultérieurement. Selon l'implémentation, il peut être nécessaire de supprimer l'allocation de mémoire utilisée par la structure de données après utilisation.

10.6 Analyse du message reçu

Dès qu'un agent IA reçoit un message émanant d'un autre agent IA, ce message est analysé. Lors de l'analyse du message, l'étiquette initiale est examinée pour déterminer la syntaxe normalisée à utiliser pour décoder et interpréter le reste du message.

10.7 Transfert de données à l'utilisateur immédiat (Traducteur/Module de sécurité)

Lorsque le message est de type "de base", il convient de transmettre les données d'utilisateur contenues dans le message à l'utilisateur immédiat de l'agent IA (en général un traducteur EDIFACT/ASC X12 EDI) pour tout traitement supplémentaire nécessaire. Le mécanisme de ce processus au niveau de l'agent IA sort du cadre de la présente Recommandation UIT-T.

Si le message est de type "avancé", il convient de transmettre son contenu au module de sécurité concerné pour toute analyse supplémentaire et toutes validations de sécurité nécessaires.

Si le transfert de données à l'utilisateur immédiat de l'agent interactif échoue, l'agent IA peut déclencher des procédures de commande de flux, comme spécifié au 11.3.

10.8 Journalisation des réceptions

Il convient de créer des entrées de journalisation après réception de données par le serveur. Il convient que l'ensemble minimal de données à journaliser comprenne les éléments suivants:

- date/heure de réception;
- identificateur de message unique (par exemple segment ASC X12 EDI ISA);
- adresse IP distante et numéro d'accès local;
- indicateurs de succès/échec de la réception et transfert à l'utilisateur immédiat (généralement un traducteur EDIFACT/ASC X12 EDI ou un module de sécurité).

11 Exigences de fonctionnement

11.1 Sécurité

Deux entités homologues en communication doivent convenir de manière bilatérale du niveau à employer pendant l'échange IA et de la méthode de gestion du niveau convenu de sécurité.

Les directives suivantes s'appliquent à l'utilisateur de services de sécurité de la couche de transport (TLS):

- une authentification efficace des entités homologues, fondée sur un chiffrement par clé publique, doit être fournie pour toutes les associations;
- les clients IA comme les serveurs IA sont supposés échanger des certificats numériques;
- la clé secrète de session est chiffrée au moyen de la clé publique du récepteur;
- l'utilisation du chiffrement des messages est facultative, mais recommandée;
- SHA1 est l'algorithme de résumé recommandé pour les fonctions d'intégrité de blocs de transmission TLS;
- si la protection de la confidentialité TLS est choisie, la norme de chiffrement des données pour le mode de chaînage de blocs de chiffre (DES-CBC, *data encryption standard in the cipher block chaining*) est recommandée pour le chiffrement par clé symétrique;
- toute entité utilisant un agent IA doit obtenir un certificat par clé publique auprès d'un agent CA acceptable pour les entités homologues;
- les certificats doivent être compatibles avec la X.509 version 3.

NOTE 1 – Les sessions pouvant être reprises ne présentent aucun autre risque pour la sécurité.

NOTE 2 – Voir la Recommandation UIT-T Q.815 pour de plus amples détails sur les spécifications relatives à la sécurité au niveau des messages.

11.2 Certificats numériques

L'architecture de l'agent IA implique que les deux parties échangent des certificats numériques lors de l'établissement d'une session TLS. Après réception d'un certificat émanant d'un homologue, il convient de transmettre les informations contenues dans le certificat de la couche TLS (couche de transport) via l'agent IA jusqu'au module de sécurité où elles seront conservées et utilisées pour des opérations de sécurité avancées telles que la non-répudiation. La même paire de clés et le même certificat utilisés pour l'authentification du système homologue seront également utilisés pour toute autre signature numérique exigée.

Il convient que les entités homologues choisissent des autorités de contrôle des certificats de confiance, acceptables pour les deux parties. Il convient que seuls les certificats émis par ces autorités soient échangés ou cités dans les signatures numériques.

Lorsque des certificats numériques ou des listes de certificats utilisent le type de données de notation ASN.1 **`UTCTime`**, la procédure suivante doit être suivie:

Avant d'utiliser une valeur de **`Time`** pour une quelconque opération de comparaison, et si la syntaxe de **`Time`** a été choisie comme le type **`UTCTime`**, la valeur du champ Année à deux chiffres doit être rationalisée en une valeur Année à quatre chiffres comme suit:

- si la valeur à deux chiffres est comprise entre 00 et 49 inclus, ajouter 2000 à cette valeur.
- si la valeur à deux chiffres est comprise entre 50 et 99 inclus, ajouter 1900 à cette valeur.

NOTE 1 – L'utilisation du type de données **GeneralizedTime** peut empêcher l'interfonctionnement avec des applications ignorant la possibilité de choisir entre **UTCTime** et **GeneralizedTime**. Il incombe aux personnes spécifiant les domaines dans lesquels seront utilisés les certificats définis dans la présente Spécification d'Annuaire, par exemple des groupes d'établissement de profils, de déterminer quand le type **GeneralizedTime** peut être utilisé. Le type **UTCTime** ne doit en aucun cas être utilisé pour des dates au-delà de 2049.

NOTE 2 – Cette procédure a été élaborée pour résoudre le problème du bug de l'an 2000 créé par la notation ASN.1 qui a défini **UTCTime** avec uniquement deux caractères pour l'année.

NOTE 3 – De plus amples informations concernant **UTCTime** dans les certificats et signatures numériques X.509 sont disponibles pour la Recommandation UIT-T X.509 (1995) dans le Directory Implementors Guide de l'UIT-T (Version 11) et incorporées dans la Recommandation UIT-T X.509 (1997).

11.3 Commande de flux

Le mécanisme de transport spécifié par l'agent interactif nécessite d'établir une session TLS individuelle pour chaque message. Les communications s'effectuent principalement dans un sens: du client vers le serveur. Le message d'état IA est un mécanisme qui permet à des entités homologues d'échanger des erreurs et autres types d'informations de commande de flux. Des codes de message spécifiques peuvent être définis par les entités homologues en dehors du domaine d'application de la présente Recommandation UIT-T; voir Tableau 2.

Le serveur peut également refuser une proposition d'établissement de session TLS si les conditions de son côté empêchent le traitement rapide d'un message entrant. Dans ce cas, le client est supposé réessayer après un délai convenu.

Lorsqu'un serveur IA reçoit un message d'état dont la valeur indique un encombrement au niveau supérieur ou inférieur, il demande à son client, par des moyens sortant du cadre de la présente Recommandation UIT-T de cesser la transmission d'autres messages IA destinés à l'expéditeur du message d'état jusqu'à ce qu'une intervention extérieure permette la reprise du traitement.

12 Attributions d'accès

Les attributions d'accès TCP/IP doivent faire l'objet d'un accord entre les entités homologues. Ces accès ne doivent pas forcément être identiques à chaque extrémité de la connexion. Ces accès sont associés à la pile de protocole TCP au sein de la configuration du système.

L'Autorité chargée de l'assignation des numéros Internet (IANA, *Internet assigned number authority*) a enregistré le protocole décrit dans la présente Recommandation UIT-T et lui a attribué le numéro de protocole 117. Le protocole est enregistré comme le "Protocole de transfert d'agent interactif" et son abréviation est ***iatp***.

L'IANA a attribué un numéro d'accès 6999 à ***iatp-normalpri***. L'utilisation de cet accès est recommandée pour les transactions de priorité normale utilisant ce protocole.

L'IANA a attribué le numéro d'accès 6998 à ***iatp-highpri***. L'utilisation de cet accès est recommandée pour les transactions hautement prioritaires utilisant ce protocole.

ANNEXE A

Module de production ASN.1

InteractiveAgent {itu-t(0) recommendation(0) q(17) q814(814) ia(0) messages(0)} DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTER tout

```
IaMessage ::= CHOICE {
        basicMessage        BasicMessage,        -- Message de base
        iaStatusMessage     IaStatusMessage,     -- Message d'arrêt de commande IA
        enhancedMessage     EnhancedMessage      -- Message avancé de module de sécurité
}

BasicMessage ::= CHOICE {

basicMessage1     GeneralString,

basicMessage2      IA5String

}
IaStatusMessage ::= BIT STRING ( SIZE (32) )
EnhancedMessage ::= OCTET STRING          -- Message avancé de module de sécurité
END
```

ANNEXE B

Considérations concernant la conception

La présente annexe identifie les considérations/contraintes de conception qui permettront de créer un agent interactif efficace et résistant.

B.1 Multitraitement/mise en place multiple

La nature et l'utilisation de l'application de l'agent IA impliqueront dans la plupart des cas le fonctionnement simultané d'entités IA concurrentes. Pour obtenir des clients multiples, il peut simplement suffire d'invoquer des instances multiples de logiciels ou de processus de clients IA. D'autres plans peuvent également être utilisés. Les processus de serveur IA sont en général structurés de manière à traiter des demandes de clients multiples et simultanées au moyen de processus tels que la *création d'un processus enfant, la mise en place multiple, le multiplexage,* ou autres technologies comparables.

NOTE – Il convient que chaque serveur IA étudie la possibilité de 16 connexions concurrentes par entité homologue.

B.2 Comparaison entre connexions non permanentes et connexions permanentes

Chaque connexion TLS peut prendre en charge le transfert d'un seul message EDIFACT ou ASC X12 EDI ou de plusieurs de ces messages pendant une même session. Dans le premier cas, cela implique qu'une session n'existe que pendant la durée de transmission d'un seul message, alors que dans le second, une connexion peut être maintenue pendant toute la durée mutuellement convenue par les entités homologues.

La décision d'utiliser l'option "un seul message par session" ou l'option "messages multiples" intervient au stade de la conception, en fonction des exigences de l'application. Pour une application susceptible de s'adresser à un grand nombre de clients, le paradigme "un seul message par session" utilisera probablement au mieux les ressources. En revanche, pour une application qui requiert

l'échange d'un grand nombre de messages entre un petit nombre de clients, il sera probablement plus efficace d'utiliser l'option "messages multiples par session".

B.3 Sessions TLS pouvant être reprises

Pour des raisons de qualité de fonctionnement, les sessions TLS pouvant être reprises sont recommandées. L'établissement d'une session TLS complète nécessite un travail important de la part du processeur, par conséquent, un mécanisme permettant de reprendre une session au moyen d'un sous-ensemble du traitement d'établissement de session TLS doit être prévu.

La durée pendant laquelle une session sera considérée comme *pouvant être reprise* sort du cadre de la présente Recommandation UIT-T. Il convient que les applications permettent la configuration de ce paramètre. Si les sessions doivent être *fermées* en fonction de la mémoire cache d'une session *pouvant être reprise*, il convient de configurer ce paramètre pour déclencher la fermeture après expiration du nombre indiqué de minutes depuis la dernière utilisation de l'identificateur de session. Les valeurs recommandées se situent entre une et trente minutes.

ANNEXE C

Traitement d'erreur/rétablissement

Une session d'agent interactif peut se terminer subitement à cause de défaillances techniques. Dans ce cas, le rétablissement consiste à établir une nouvelle session et à renvoyer la transaction en cours. Il existe cependant un élément communément défini permettant au client de considérer qu'une transaction a été envoyée avec succès: lorsque le message *Close Notify* est envoyé par le serveur. Après cela, le rétablissement se situe au niveau de l'application de bout en bout, qui sera déclenché par la non-réception de l'accusé de réception de transaction (par exemple ASC X12 997 – Accusé de réception fonctionnel) pendant le délai d'attente.

Le rétablissement au niveau de l'application est indiqué par l'expiration du délai d'attente de l'accusé de réception de transaction correspondant (par exemple ASC X12 997 – Accusé de réception fonctionnel). Des délais d'attente distincts peuvent être spécifiés pour les transactions de type normal et hautement prioritaire pour chaque entité homologue.

APPENDICE I

Références non normatives

– ISO 9735:1988, *Echange informatisé de données pour l'administration, le commerce et le transport (EDIFACT) – Règles de syntaxe au niveau de l'application*.

– ANSI ASC X12: American National Standards Institute (ANSI) Accredited Standards Committee X12. Ce comité a été mandaté par l'ANSI en 1979 pour développer des normes uniformes concernant les échanges électroniques de documents commerciaux.

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	**Commutation et signalisation**
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux 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 Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects informatiques généraux des systèmes de télécommunication

Genève, 2001