Groupe de travail Réseau
**Request for Comments : 3928**
Catégorie : En cours de normalisation

Traduction Claude Brière de L'Isle

R. Megginson, éd., Netscape Communications Corp.
M. Smith, Pearl Crescent, LLC
O. Natkovich, Yahoo
J. Parham, Microsoft Corporation
octobre 2004

# Protocole de mise à jour de client (LCUP) pour le protocole léger d'accès à un répertoire (LDAP)

**Statut de ce mémoire**
Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.



**Résumé**
Le présent document définit le protocole de mise à jour de client (LCUP, *LDAP Client Update Protocol*) du protocole léger d'accès à un répertoire (LDAP, *Lightweight Directory Access Protocol*). Le protocole est destiné à permettre à un client LDAP de se synchroniser avec le contenu d'une arborescence d'informations d'un répertoire (DIT, *directory information tree*) mémorisées par un serveur LDAP et d'être notifié des changements de ce contenu.

## Table des Matières

## 1. Vue d'ensemble

Le protocole LCUP est destiné à permettre aux clients LDAP de se synchroniser avec le contenu mémorisé par les serveurs LDAP.

Les problèmes traités par le protocole incluent :

- Les clients mobiles qui tiennent une copie locale en lecture seule des données du répertoire. Lorsque il est hors ligne, le client utilise sa copie locale des données. Lorsque le client se connecte au réseau, il se synchronise avec le contenu actuel du répertoire et peut facultativement recevoir des notifications sur les changements qui se sont produits pendant qu'il est en ligne. Par exemple, un client de messagerie peut tenir une copie locale du répertoire d'adresses professionnelles qu'il synchronise avec la copie maîtresse chaque fois qu'il se connecte au réseau d'entreprise.

- Les applications destinées à synchroniser des mémorisations de données hétérogènes. Une application de méta répertoire, par exemple, va restituer périodiquement une liste d'entrées modifiées à partir du répertoire, construire les changements et les appliquer à une mémorisation de données étrangère.

- Les clients qui ont besoin de faire certaines actions lorsque une entrée du répertoire est modifiée. Par exemple, un dépôt de messagerie électronique peut vouloir effectuer une tâche de "création de boîte aux lettres" lorsque l'entrée d'une nouvelle personne est ajoutée à un répertoire LDAP et une tâche de "suppression de boîte aux lettres" lorsque l'entrée d'une personne est supprimée.

Les problèmes non considérés sont :

- La synchronisation de serveur de répertoire à serveur de répertoire. L'IETF développe un protocole de duplication LDAP, appelé LDUP [RFC3384], qui est spécifiquement conçu pour traiter ce problème.

Il y a actuellement plusieurs protocoles qui sont utilisés pour la synchronisation de serveur à client LDAP. Bien que chaque protocole traite des besoins d'un groupe particulier de clients (par exemple, clients en ligne ou clients hors ligne) aucun ne satisfait aux exigences de tous les clients dans le groupe cible. Par exemple, un client mobile qui était hors ligne et veut se mettre à jour avec le serveur et rester à jour lorsque il est connecté ne peut être facilement pris en charge par aucun des protocoles existants.

LCUP est conçu de telle façon que le serveur n'ait pas besoin de conserver les informations d'état spécifiques des clients individuels. Le serveur peut avoir besoin de conserver des informations d'état supplémentaires sur les modifications d'attributs, les entrées supprimées, et les entrées déplacées/débaptisées. Les clients sont chargés de mémoriser les informations sur leur état de mise à jour par rapport au contenu du serveur. La conception de LCUP évite d'avoir besoin d'accords de mise à jour spécifiques de LCUP entre le client et le serveur avant l'utilisation de LCUP. Le client décide quand et à partir d'où récupérer les changements. La conception de LCUP exige des clients qu'ils initient la session de mise à jour et "tirent" les changements du serveur.

Les opérations de LCUP sont soumises aux politiques administratives et de contrôle d'accès appliquées par le serveur.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

## 2. Applicabilité

LCUP fonctionnera au mieux si les conditions suivantes sont satisfaites :

1) Le serveur mémorise une partie de l'historique des états ou des informations de changement pour réduire la quantité de trafic réseau nécessaire pour les synchronisations incrémentaires. L'équilibre optimal entre l'état de serveur et le trafic réseau varie selon les mises en œuvre et les scénarios d'utilisation, et est donc laissé entre les mains des mises en œuvre.

2) Le client ne peut pas être supposé comprendre le modèle d'informations physique (attributs virtuels, attributs de fonctionnement, sous entrées, etc.) mis en œuvre par le serveur. Des optimisations seraient possibles si de telles hypothèses pouvaient être avancées.

3) Les changements de métadonnées et les changements de noms et suppressions de grandes sous arborescences sont très rares. LCUP fait ces hypothèses afin de réduire la complexité exigée du client pour traiter ces opérations particulières, bien que, quand elles surviennent, elles puissent résulter en un grand nombre de messages de mises à jour incrémentaires ou en une resynchronisation complète.

## 3. Spécification des éléments du protocole

Les paragraphes qui suivent définissent les nouveaux éléments requis pour l'utilisation de ce protocole.

### 3.1 Considérations d'ASN.1

Les éléments du protocole sont décrits en utilisant l'ASN.1 [X.680]. Le terme "codé en BER" signifie que l'élément est à coder en utilisant les règles de codage de base [X.690] avec les restrictions détaillées au paragraphe 5.1 de la [RFC2251]. Tout l'ASN.1 du présent document utilise des étiquettes implicites.

### 3.2 Identifiants uniques universels

Les noms distinctifs peuvent changer, et ne sont donc pas fiables comme identifiants. Un identifiant unique universel (UUID, *Universally Unique Identifier*) DOIT être utilisé pour identifier de façon univoque les entrées utilisées avec LCUP. L'UUID fait partie de la valeur de la commande Sync Update *(mise à jour de synchronisation)* décrite plus loin, qui est retournée avec chaque résultat de recherche. Le serveur DEVRAIT fournir l'UUID comme un attribut de fonctionnement d'une seule valeur de l'entrée (par exemple, "entryUUID"). On RECOMMANDE que le serveur fournisse un moyen de faire des recherches efficaces (c'est-à-dire, indexées) pour les valeurs d'UUID, par exemple, en utilisant un filtre de recherche (comme entryUUID=<valeur d'UUID>) pour rechercher et restituer rapidement une entrée fondée sur son UUID. Les serveurs DEVRAIENT utiliser un format d'UUID comme spécifié dans [UUID]. L'UUID utilisé par LCUP est une valeur du type ASN.1 suivant :

```
LCUPUUID ::= CHAINE D'OCTETS
```

### 3.3 Schéma et mouchard LCUP

Le protocole LCUP utilise un mouchard *(cookie)* pour contenir l'état des données du client par rapport aux données du serveur. Chaque format de mouchard est identifié de façon univoque par son schéma. Le schéma LCUP est une valeur du type ASN.1 suivant :

```
LCUPScheme ::= LDAPOID
```

C'est l'OID qui identifie le format de la valeur du mouchard LCUP. L'OID de schéma, comme tous les identifiants d'objet, DOIT être unique pour un certain schéma de mouchard. La valeur du mouchard peut être opaque ou elle peut être exposée aux clients LCUP. Pour les schémas de mouchard qui exposent leur valeur, la forme préférée de documentation est une RFC. On s'attend à ce qu'il y ait un ou plusieurs schémas de mouchard en voie de normalisation lorsque le format de valeur est exposé et décrit en détail.

Le mouchard LCUP est une valeur du type ASN.1 suivant :

```
LCUPCookie ::= CHAINE D'OCTETS
```

Ce sont les données réelles qui décrivent l'état des données du client. Cette valeur peut être opaque, ou sa valeur peut avoir un

format bien connu, selon le schéma.

D'autres utilisations de la valeur de mouchard LCUP sont décrites plus loin.

### 3.4 Contexte LCUP

Une partie de la DIT qui est activée pour LCUP est appelée un contexte LCUP. Un serveur peut prendre en charge un ou plusieurs contextes LCUP. Par exemple, un serveur avec deux contextes de désignation peut prendre en charge LCUP dans un contexte de désignation mais pas dans l'autre, ou prendre en charge différents schémas de mouchard LCUP dans chaque contexte de désignation. Chaque contexte LCUP PEUT utiliser un schéma de mouchard différent. Une recherche LCUP ne franchira pas une limite de contexte LCUP, mais va plutôt retourner un message SearchResultReference, dont l'URL LDAP spécifiera le même hôte et accès que ceux actuellement recherchés, et avec le baseDN réglé au baseDN du nouveau contexte LCUP. Le client est alors responsable de la production d'une autre recherche en utilisant le nouveau baseDN, et éventuellement un mouchard différent si ce contexte LCUP utilise un mouchard différent. Le client est responsable du maintien d'une transposition de l'URL LDAP en son mouchard correspondant.

### 3.5 Codes de résultat LDAP supplémentaires définis par LCUP

Les mises en œuvre de la présente spécification DEVRONT reconnaître les valeurs supplémentaires de resultCode suivantes. Les noms et les numéros de code de résultat LDAP définis dans le tableau suivant ont été alloués par l'IANA selon la [RFC3383].

lcupResourcesExhausted (113) : le serveur est à court de ressources
lcupSecurityViolation (114) : le client est soupçonné d'actions malveillantes
lcupInvalidData (115) : schéma invalide ou mouchard fourni par le client
lcupUnsupportedScheme (116) : le schéma de mouchard est un OID valide mais n'est pas accepté par ce serveur
lcupReloadRequired (117) : indique que les données du client doivent être réinitialisées. Cette raison est retournée si le serveur ne contient pas d'informations suffisantes pour synchroniser le client ou si les données du serveur ont été rechargées depuis la dernière session de synchronisation.

Les utilisations de ces codes sont décrites ci-dessous.

### 3.6 Commande Demande Sync

La commande Demande Sync est une commande LDAP [RFC2251] (paragraphe 4.1.2) où le controlType est l'identifiant d'objet 1.3.6.1.1.7.1 et la controlValue, une CHAINE D'OCTETS, contient une syncRequestControlValue codée en BER.

```
syncRequestControlValue ::= SEQUENCE {
  updateType          ENUMERATION {
                syncOnly      (0),
                syncAndPersist (1),
                 persistOnly   (2) },
  sendCookieInterval   [0] ENTIER    FACULTATIF,
  scheme               [1] LCUPScheme FACULTATIF,
  cookie               [2] LCUPCookie FACULTATIF
}
```

sendCookieInterval : le serveur DEVRAIT renvoyer le mouchard dans la valeur de commande Sync Update (définie plus loin) pour chaque numéro de sendCookieInterval des PDU SearchResultEntry et SearchResultReference retournées au client. Par exemple, si la valeur est 5, le serveur DEVRAIT renvoyer le mouchard dans la valeur de commande Sync Update tous les cinq résultats de recherche retournés au client. Si cette valeur est absente, zéro ou moins que zéro, le serveur choisit l'intervalle.

La commande Sync Request n'est applicable qu'au message searchRequest. L'utilisation de cette commande est décrite plus loin.

### 3.7 Commande de mise à jour Sync

La commande Sync Update est une commande LDAP [RFC2251] (paragraphe 4.1.2) où le controlType est l'identifiant d'objet 1.3.6.1.1.7.2 et la controlValue, une CHAINE D'OCTETS, contient une syncUpdateControlValue codée en BER.

```
syncUpdateControlValue ::= SEQUENCE {
  stateUpdate   BOOLEEN,
  entryUUID     [0] LCUPUUID FACULTATIF, -- EXIGÉ pour les entrées --
  UUIDAttribute [1] AttributeType FACULTATIF,
  entryLeftSet  [2] BOOLEEN,
  persistPhase  [3] BOOLEEN,
  scheme        [4] LCUPScheme FACULTATIF,
  cookie        [5] LCUPCookie FACULTATIF
}
```

Le champ UUIDAttribute contient le nom ou l'OID de l'attribut que le client devrait utiliser pour effectuer des recherches d'entrées sur la base de l'UUID. Le client devrait être capable de l'utiliser dans un filtre de recherche d'égalité, par exemple, "(<uuid attribute>=<valeur d'UUID d'entrée>)" et devrait être capable de l'utiliser dans les listes d'attributs de la demande de recherche pour retourner sa valeur. Le champ UUIDAttribute peut être omis si le serveur ne prend pas en charge les recherches sur les valeurs d'UUID.

La commande Sync Update n'est applicable qu'aux messages SearchResultEntry et SearchResultReference. Bien que entryUUID soit FACULTATIF, il DOIT être utilisé avec les messages SearchResultEntry. L'utilisation de cette commande est décrite plus loin.

### 3.8 Commande Sync Done

La commande Sync Done est une commande LDAP [RFC2251] (paragraphe 4.1.2) où le controlType est l'identifiant d'objet 1.3.6.1.1.7.3 et la controlValue contient une syncDoneValue codée en BER.

```
syncDoneValue ::= SEQUENCE {
  scheme      [0] LCUPScheme FACULTATIF,
  cookie      [1] LCUPCookie FACULTATIF
}
```

La commande Sync Done n'est applicable qu'au message SearchResultDone. L'utilisation de cette commande est décrite plus loin.

## 4. Usage et flux du protocole

### 4.1 Demandes de recherche LCUP

Un client initie une synchronisation ou une session de recherche persistante avec un serveur en attachant une commande Sync Request à un message LDAP searchRequest. La spécification de recherche détermine la partie de l'arborescence d'informations de répertoire (DIT, *directory information tree*) avec laquelle le client souhaite se synchroniser, l'ensemble d'attributs auquel il s'intéresse et la quantité de données que le client veut recevoir. La commande Sync Request contient la spécification de la demande du client.

Si il y a une condition d'erreur, le serveur DOIT immédiatement retourner un message SearchResultDone avec le resultCode réglé à un code d'erreur. Ce tableau transpose une condition en son comportement correspondant et son code de résultat.

| **Condition** | **Comportement ou code de résultat** |
|---|---|
| La commande Sync Request n'est pas prise en charge | Le serveur se comporte selon le paragraphe 4.1.2 de la [RFC2251] – précisément, si la criticité de la commande est FAUX, le serveur va traiter la demande comme une demande de recherche normale. |
| Schéma non pris en charge | lcupUnsupportedScheme |
| Un champ de valeur de la commande est invalide (par exemple, updateType illégal, ou le schéma n'est pas un OID valide, ou le mouchard est invalide). | lcupInvalidData |
| Le serveur est à bout de ressources | lcupResourcesExhausted |
| Le serveur soupçonne le client de comportement malveillant (connexions/déconnexions fréquentes, etc.) | lcupSecurityViolation |
| Le serveur ne peut mettre le client à jour (les données du serveur ont été rechargées, ou un autre changement empêche la convergence). | lcupReloadRequired |

#### 4.1.1 Synchronisation initiale et resynchronisation complète

Pour une synchronisation initiale ou une resynchronisation complète, les champs de la commande Sync Request DOIVENT être spécifiés comme suit :

updateType : DOIT être réglé à syncOnly ou syncAndPersist.
sendCookieInterval : PEUT être établi.
scheme : PEUT être établi – s'il est établi, le serveur DOIT utiliser le schéma spécifié ou retourner lcupUnsupportedScheme (voir ci-dessus) - sinon, le serveur PEUT utiliser tout schéma qu'il prend en charge.
cookie : NE DOIT PAS être établi.

Si la demande réussit, le client va recevoir des résultats comme décrit au paragraphe 4.2 "Réponses de recherche LCUP".

#### 4.1.2 Synchronisation incrémentaire ou mise à jour de synchronisation

Pour une synchronisation incrémentaire ou de mise à jour, les champs de la commande Sync Request DOIVENT être spécifiés comme suit :

updateType : DOIT être réglé à syncOnly ou syncAndPersist.
sendCookieInterval : PEUT être établi.
scheme : DOIT être établi.
cookie : DOIT être établi.

Le client DEVRAIT toujours utiliser le dernier mouchard *(cookie)* qu'il a reçu du serveur.

Si la demande a réussi, le client va recevoir des résultats comme décrit au paragraphe 4.2 "Réponses de recherche LCUP".

#### 4.1.3 Seulement persistante

Pour les demandes de recherche seulement persistantes, les champs de la demande Sync DOIVENT être spécifiés comme suit :

updateType : DOIT être réglé à persistOnly.
sendCookieInterval : PEUT être établi.
scheme : PEUT être établi - s'il est établi, le serveur DOIT utiliser le schéma spécifié ou retourner lcupUnsupportedScheme (voir ci-dessus) - sinon, le serveur PEUT utiliser tout schéma qu'il prend en charge.
cookie : PEUT être établi, mais le serveur DOIT l'ignorer.

Si la demande a réussi, le client va recevoir des résultats comme décrit au paragraphe 4.2 "Réponses de recherche LCUP".

### 4.2 Réponses de recherche LCUP

En réponse à la demande LCUP du client, le serveur retourne zéro, une ou plusieurs PDU SearchResultEntry ou SearchResultReference qui tiennent dans la spécification du client, suivies par une PDU SearchResultDone. Le comportement est celui spécifié au paragraphe 4.5 de la [RFC2251]. Chaque PDU SearchResultEntry ou SearchResultReference contient aussi une commande Sync Update qui décrit l'état LCUP de l'entrée retournée. La PDU SearchResultDone contient une commande Sync Done. Les paragraphes qui suivent spécifient les comportements qui s'ajoutent à ceux du paragraphe 4.5 de la [RFC2251].

#### 4.2.1 Réponses d'information de Sync Update

Le serveur peut utiliser la commande Sync Update pour retourner des informations qui ne se rapportent pas à une entrée particulière. Il PEUT faire cela à tout moment pour retourner un mouchard au client, ou pour informer le client que la phase de synchronisation d'une recherche syncAndPersist est achevée et que la phase persistante a commencé. Il PEUT faire cela durant la phase persistante même si aucune entrée n'a changé qui aurait normalement déclanché une réponse. Pour faire cela, il est EXIGÉ de retourner ce qui suit :

- Une PDU SearchResultEntry avec le champ objectName réglé au nom distinctif (DN, *Distinguished Name*) du baseObject de la demande de recherche et avec une liste d'attributs vide.
- Une valeur de commande Sync Update avec les champs réglés comme suit :
  stateUpdate : DOIT être réglé à VRAI
  entryUUID : DEVRAIT être réglé à l'UUID du baseObject de la demande de recherche
  entryLeftSet : DOIT être réglé à FAUX.
  persistPhase : DOIT être FAUX si la recherche est dans la phase de synchronisation d'une demande, et DOIT être VRAI

si la recherche est dans la phase persistante.
UUIDAttribute : DEVRAIT n'être établi que si c'est le premier résultat retourné ou si l'attribut a changé.
scheme : DOIT être établi si le mouchard est établi et si le format de mouchard a changé ; autrement, il PEUT être omis.
cookie : DEVRAIT être établi.

Si le serveur veut simplement retourner un mouchard au client, il devrait le retourner comme ci-dessus avec le champ mouchard établi.

Durant une demande syncAndPersist, le serveur DOIT retourner (comme ci-dessus) immédiatement après l'envoi de la dernière entrée de la phase de synchronisation et avant l'envoi de la première entrée de la phase persistante. Dans ce cas, le champ persistPhase DOIT être établi à VRAI. Cela permet au client de savoir que la phase de synchronisation est achevée et que la phase persistante commence.

### 4.2.2 Fréquence de retour des mouchards

Le champ cookie de la valeur de la commande Sync Update PEUT être établi dans tout résultat retourné, aussi bien dans la phase de synchronisation que dans la phase persistante. Le serveur devrait retourner le mouchard au client assez souvent pour que le client se resynchronise dans un délai raisonnable au cas où la recherche serait déconnectée ou terminée d'une autre façon. Le champ sendCookieInterval dans la commande Sync Request est une suggestion au serveur de la fréquence de retour du mouchard dans la commande Sync Update. Le serveur DEVRAIT respecter cette valeur.

Le champ scheme de la valeur de la commande Sync Update DOIT être établi si le mouchard est établi et si le format du mouchard a changé ; autrement, il PEUT être omis.

Certains clients peuvent avoir des connexions non fiables, par exemple, un appareil sans fil ou une connexion de WAN. Ces clients peuvent vouloir s'assurer que le mouchard est retourné souvent dans la valeur de la commande Sync Update, afin que si ils doivent se reconnecter, ils n'aient pas à traiter trop d'entrées redondantes. Ces clients devraient régler le sendCookieInterval dans la valeur de la commande Sync Request à un nombre faible, peut-être même 1. Certains clients peuvent avoir une connexion à bande passante limitée, et peuvent ne pas vouloir recevoir très souvent le mouchard, ou même pas du tout (cependant, le mouchard est toujours renvoyé dans la valeur de la commande Sync Done en cas de réussite). Ces clients devraient régler le sendCookieInterval dans la valeur de la commande Sync Request à un nombre élevé.

Un comportement raisonnable du serveur est de retourner le mouchard seulement quand les données ont changé dans le contexte LCUP, même si le client a spécifié un sendCookieInterval fréquent. Si rien n'a changé, le serveur peut probablement épargner un peu de bande passante en ne retournant pas le mouchard.

### 4.2.3 Définition d'une entrée qui est entrée dans l'ensemble des résultats

Une entrée DEVRA ÊTRE considérée comme étant entrée dans l'ensemble de résultat de recherche du client si une des conditions suivantes est satisfaite :

- Durant la phase de synchronisation pour une opération de synchronisation incrémentaire, l'entrée est présente dans l'ensemble de résultats de recherche mais n'y était pas présente avant ; ce peut être dû à l'ajout de l'entrée via une opération Add LDAP, ou par le déplacement de l'entrée dans l'ensemble de résultats par une opération Modify DN LDAP, ou par une modification de l'entrée qui la fait entrer dans l'ensemble de résultats (par exemple, l'ajout d'une valeur d'attribut qui correspond au filtre de recherche du client) ou par un changement de métadonnées qui fait entrer l'entrée dans l'ensemble de résultats (par exemple, relâcher un contrôle d'accès qui permet à l'entrée d'être visible au client).

- Durant la phase persistante pour une opération de recherche persistante, l'entrée passe dans l'ensemble de résultats de recherche ; ceci peut être causé par l'ajout de l'entrée via une opération Add LDAP, ou par le déplacement de l'entrée dans l'ensemble de résultats par l'opération Modify DN LDAP, ou par une modification de l'entrée qui la fait entrer dans l'ensemble de résultats (par exemple, l'ajout d'une valeur d'attribut qui correspond au filtre de recherche du client) ou par un changement de métadonnées qui cause le passage de l'entrée dans l'ensemble de résultats (par exemple, l'assouplissement d'un contrôle d'accès qui permet que l'entrée soit visible au client).

### 4.2.4 Définition d'une entrée qui a changé

Une entrée DEVRA ÊTRE considérée comme ayant changé si un ou plusieurs des attributs de la liste d'attributs de la demande de recherche ont été modifiés. Par exemple, si la demande de recherche fait la liste des attributs "cn sn uid", et si il y a une entrée dans l'ensemble de résultats de recherche du client dont l'attribut "cn" a été modifié, l'entrée est considérée comme ayant été modifiée. La modification peut être due à une opération LDAP Modify ou par un changement de métadonnées pour l'entrée (par exemple, des attributs virtuels) qui cause un changement de la valeur des attributs spécifiés.

L'autre face de ceci est qu'une entrée NE DEVRA PAS ÊTRE considérée comme ayant changé si aucun des attributs de la liste d'attributs de la demande de recherche n'est un attribut modifié de l'entrée. Par exemple, si la demande de recherche affiche dans sa liste les attributs "cn sn uid", et qu'il y a une entrée dans l'ensemble de résultats de recherche du client dont l'attribut "foo" a été modifié, et aucun des attributs "cn" ou "sn" ou "uid" n'a été modifié, l'entrée N'EST PAS considérée comme changée.

#### 4.2.5 Définition d'une entrée qui a quitté l'ensemble de résultats

Une entrée DEVRA ÊTRE considérée comme ayant quitté l'ensemble de résultats de recherche du client si une des conditions suivantes est satisfaite :

- Durant la phase de synchronisation pour une opération de synchronisation incrémentaire, l'entrée n'est pas présente dans l'ensemble de résultats de recherche mais était présente avant ; ceci peut venir de ce que l'entrée a été supprimée via une opération Delete LDAP, ou que l'entrée a quitté l'ensemble de résultats via une opération Modify DN LDAP, ou par une modification de l'entrée qui fait qu'elle a quitté l'ensemble de résultats (par exemple, changer/supprimer une valeur d'attribut de sorte qu'elle ne correspond plus au filtre de recherche du client) ou par un changement de métadonnées qui fait que l'entrée a quitté l'ensemble de résultats (par exemple, en ajoutant un contrôle d'accès qui refuse que l'entrée soit visible par le client).
- Durant la phase persistante pour une opération de recherche persistante, l'entrée quitte l'ensemble de résultats de recherche ; ceci peut être dû au fait que l'entrée a été supprimée via une opération Delete LDAP, ou parce que l'entrée a quitté l'ensemble de résultats via une opération Modify DN LDAP, ou par une modification à l'entrée qui lui a fait quitter l'ensemble de résultats (par exemple, changer/supprimer une valeur d'attribut de sorte qu'elle ne correspond plus au filtre de recherche du client) ou par un changement de métadonnées qui fait que l'entrée a quitté l'ensemble de résultats (par exemple, en ajoutant un contrôle d'accès qui refuse que l'entrée soit visible par le client).

#### 4.2.6 Résultats pour les entrées présentes dans l'ensemble de résultats

Une entrée DEVRAIT être retournée comme présente dans les conditions suivantes :

- La demande est une demande de synchronisation initiale ou de pleine resynchronisation et l'entrée est présente dans l'ensemble de résultats de recherche du client.
- La demande est une synchronisation incrémentaire et l'entrée a changé ou est entrée dans l'ensemble de résultats depuis la dernière synchronisation.
- La recherche est dans la phase persistante et l'entrée entre dans l'ensemble de résultats ou change.

Pour un retour de SearchResultEntry, les champs de la valeur de commande Sync Update DOIVENT être réglés comme suit :
stateUpdate : DOIT être réglé à FAUX.
entryUUID : DOIT être réglé à l'UUID de l'entrée.
entryLeftSet : DOIT être réglé à FAUX.
persistPhase : DOIT être réglé à FAUX si on est durant la phase de synchronisation, ou à VRAI durant la phase persistante.
UUIDAttribute : DEVRAIT être établi seulement si c'est le premier résultat retourné ou si l'attribut a changé.
scheme : comme ci-dessus.
cookie : comme ci-dessus.

La searchResultReference retournée va paraître la même, sauf que l'UUID d'entrée n'est pas exigé. Si elle est spécifiée, elle DOIT contenir l'UUID du DSE qui contient la référence connue.

#### 4.2.7 Résultats pour les entrées qui ont quitté l'ensemble de résultats

Une entrée DEVRAIT être retournée comme ayant laissé l'ensemble de résultats sous les conditions suivantes :

- La demande est une synchronisation incrémentaire durant la phase de synchronisation et l'entrée a laissé l'ensemble de résultats.
- La recherche est dans la phase persistante et l'entrée a laissé l'ensemble de résultats.
- L'entrée a laissé l'ensemble de résultats par suite d'une opération LDAP Delete ou Modify DN contre l'entrée elle-même (c'est-à-dire, pas par suite d'une opération contre son parent ou ancêtre).

Pour une SearchResultEntry retournée lorsque l'entrée a laissé l'ensemble de résultats, les champs de la valeur de la commande Sync Update DOIVENT être réglés comme suit :
stateUpdate : DOIT être réglé à FAUX.
entryUUID : DOIT être réglé à l'UUID de l'entrée qui a quitté l'ensemble de résultats.
entryLeftSet : DOIT être réglé à VRAI.
persistPhase : DOIT être réglé à FAUX si on est durant la phase de synchronisation, ou à VRAI si durant la phase persistante.
UUIDAttribute : DEVRAIT n'être établi que si c'est le premier résultat retourné ou si l'attribut a changé.

scheme : comme ci-dessus.
cookie : comme ci-dessus.

La searchResultReference retournée va sembler la même, sauf que l'UUID d'entrée n'est pas exigé. Si il est spécifié, il DOIT contenir l'UUID du DSE qui a connaissance de la référence.

Certaines mises en œuvre de serveur gardent trace des entrées supprimées en utilisant une "pierre tombale" – une entrée cachée qui garde trace de l'état, mais pas de toutes les données, d'une entrée supprimée. Dans ce cas, la "pierre tombale" ne peut pas contenir les attributs originaux de l'entrée, et donc, il peut être impossible au serveur de déterminer si une entrée devrait être supprimée de l'ensemble de résultats sur la base des attributs dans la demande de recherche du client. Les serveurs DEVRAIENT conserver assez d'informations sur les attributs des entrées supprimées pour déterminer si une entrée devrait être retirée de l'ensemble de résultats. Comme ce n'est pas toujours possible, le serveur PEUT retourner une entrée comme ayant quitté l'ensemble de résultats même si elle n'est pas, ou n'a jamais été, dans l'ensemble de résultats du client. Les clients DOIVENT ignorer ces notifications.

### 4.3 Réponses qui exigent une considération particulière

Les paragraphes qui suivent décrivent le traitement particulier qui peut être exigé lors du retour des résultats.

#### 4.3.1 Retour des résultats durant la phase persistante

Durant la phase persistante, le serveur DEVRAIT retourner les entrées changées aussi vite que possible au client.

#### 4.3.2 Pas de mélange de la phase de synchronisation avec la phase persistante

Durant une phase de synchronisation, le serveur NE DOIT PAS retourner d'entrée avec le fanion persistPhase établi à VRAI, et durant la phase persistante, toutes les entrées retournées DOIVENT avoir le fanion persistPhase établi à VRAI. Le serveur NE DOIT PAS mêler et confronter des entrées de phase de synchronisation avec des entrées de phase persistantes. Si il y a des entrées de phase de synchronisation à retourner, elles DOIVENT être retournées avant qu'aucune entrée de phase persistante soit retournée.

#### 4.3.3 Retour de résultats mis à jour durant la phase de synchronisation

Il peut y avoir des mises à jour des entrées dans l'ensemble de résultats d'une recherche de phase de synchronisation durant l'opération de recherche réelle. Si l'agent système de répertoire (DSA, *Directory System Agent*) a une grosse charge de mise à jour, et si il tente d'envoyer toutes ces entrées mises à jour au client en plus des autres mises à jour qu'il prévoyait d'envoyer pour la phase de synchronisation, le serveur peut ne jamais voir la fin de la phase de synchronisation. Donc, il est laissé à la discrétion de la mise en œuvre de serveur de décider quand le client est "en synchronisation" – c'est-à-dire, quand terminer une demande syncOnly, ou quand envoyer la réponse pour information de Sync Update entre la phase de synchronisation et la phase persistante d'une demande syncAndPersist. Le serveur PEUT envoyer plusieurs fois la même entrée durant la phase de synchronisation si l'entrée change durant la phase de synchronisation.

Un comportement raisonnable est que le serveur génère un mouchard sur la base de l'état du serveur au moment où le client initie la demande LCUP, et de n'envoyer les entrées que jusqu'à ce point durant la phase de synchronisation. Les entrées mises à jour après ce point seront retournées seulement durant la phase persistante d'une demande syncAndPersist, ou seulement lors d'une synchronisation incrémentaire.

#### 4.3.4 Attributs de fonctionnement et entrées administratives

Un attribut opérationnel DEVRAIT être retourné si il est spécifié dans la liste des attributs et serait normalement retourné comme soumis aux contraintes du paragraphe 4.5 de la [RFC2251]. Si le serveur ne prend pas en charge la synchronisation des attributs opérationnels, le serveur DOIT retourner un message SearchResultDone avec un code de résultat de unwillingToPerform *(ne veut pas effectuer)*.

Les sous entrées LDAP [RFC3672] DEVRAIENT être retournées si elles auraient normalement été retournées par la demande de recherche. Si le serveur ne prend pas en charge la synchronisation des sous entrées LDAP, et si le serveur peut déterminer à partir de la demande de recherche que le client a demandée que des sous entrées LDAP soient retournées (par exemple, contrôle de recherche ou filtre de recherche) le serveur DOIT retourner un message SearchResultDone avec un code de résultat de unwillingToPerform. Autrement, le serveur PEUT simplement omettre de retourner les sous entrées LDAP.

#### 4.3.5 Attributs virtuels

Une entrée peut avoir des attributs dont la présence dans l'entrée, ou la présence de valeurs de l'attribut, est générée au vol, éventuellement par des mécanismes extérieurs à l'entrée, ailleurs dans la DIT. Un exemple est celui des attributs collectifs [RFC3671]. Ces attributs seront mentionnés dans le présent document comme des attributs virtuels.

LCUP traite ces attributs de la même façon que les attributs normaux, non virtuels. Un attribut virtuel DEVRAIT être retourné si il est spécifié dans la liste des attributs et serait normalement retourné comme soumis aux contraintes du paragraphe 4.5 de la [RFC2251]. Si le serveur ne prend pas en charge la synchronisation des attributs virtuels, le serveur DOIT retourner un message SearchResultDone avec un code de résultat de unwillingToPerform.

Cela a pour conséquence que si on change la définition d'un attribut virtuel de telle sorte que cela change la valeur de cet attribut dans de nombreuses entrées dans le domaine de recherche du client, cela signifie qu'un serveur peut devoir retourner de nombreuses entrées au client par suite de ce changement. On ne prévoit pas que cela arrive souvent, et le serveur a l'option de simplement forcer le client à resynchroniser si nécessaire.

Il est aussi possible qu'une commande LDAP future permette au client de demander seulement des attributs virtuels ou seulement des attributs non virtuels.

#### 4.3.6 Opérations Modify DN et Delete appliquées aux sous arborescences

Il y a un cas particulier lorsque une opération Modify DN ou Delete est appliquée à l'entrée de base d'une sous arborescence, et que l'entrée de base ou les entrées dans la sous arborescence sont dans la portée d'une demande de recherche LCUP. Dans ce cas, toutes les entrées dans la sous arborescence sont implicitement renommées ou supprimées.

Dans l'un et l'autre de ces cas, le serveur DOIT faire une des choses suivantes :
- traiter toutes ces entrées comme ayant été renommées ou supprimées et retourner chaque entrée au client comme telles,
- décider que ceci serait d'un coût prohibitif et forcer le client à resynchroniser.

Si l'objet de base de la recherche a été renommé, et si le client a reçu un noSuchObject comme résultat de la demande de recherche, le client PEUT utiliser l'UUID de l'attribut d'UUID pour localise le nouveau DN qui est le résultat de l'opération de modification du DN.

#### 4.3.7 Garanties de convergence

Si, à tout moment pendant une recherche LCUP, durant la phase de synchronisation ou la phase persistante, le serveur détermine qu'il ne peut pas garantir d'arriver à la convergence de la copie des données du client, il DEVRAIT immédiatement terminer la demande de recherche LCUP et retourner un message SearchResultDone avec un code de résultat de lcupReloadRequired *(rechargement LCUP exigé)*. Ceci peut aussi se produire au début d'une demande de synchronisation incrémentaire, si le client présente un mouchard périmé ou qu'il est par ailleurs incapable de traiter. Le client devrait alors produire une demande de synchronisation initiale.

Ceci peut se produire, par exemple, si les données sont rechargées sur le serveur, ou si il y a eu un changement des métadonnées qui rend impossible au serveur de déterminer si une entrée particulière devrait ou non faire partie de l'ensemble des résultats de recherche, ou si le changement des métadonnées rend le calcul de l'ensemble de résultats approprié trop consommateur de ressource pour le serveur.

Le serveur peut aussi retourner lcupReloadRequired si il détermine qu'il serait plus efficace pour le client d'effectuer un rechargement, par exemple, si trop d'entrées ont changé et qu'un simple rechargement serait plus rapide.

### 4.4 Terminaison de recherche LCUP

#### 4.4.1 Terminaison à l'initiative du serveur

Lorsque le serveur a réussi à terminer le traitement de la demande du client, il joint une commande Sync Done au message SearchResultDone et l'envoie au client. Cependant, si le message SearchResultDone contient un code de résultat qui n'est pas de succès ou d'annulation, la commande Sync Done PEUT être omise. Bien que le mouchard LCUP soit FACULTATIF dans la valeur de la commande Sync Done, il DOIT être réglé au code de résultat SearchResultDone si c'est un succès ou une annulation. Le serveur DEVRAIT aussi établir le mouchard si le code de résultat est lcupResourcesExhausted *(ressources LCUP épuisées)*, timeLimitExceeded *(limite de temps atteinte)*, sizeLimitExceeded *(taille limite atteinte)*, ou adminLimitExceeded *(limite administrative atteinte)*. Cela permet au client de se resynchroniser plus facilement ultérieurement. Si une erreur s'est produite, soit une erreur de recherche LDAP (par exemple, droits d'accès insuffisants) soit

une erreur LCUP (par exemple, schéma LCUP non pris en charge) le mouchard PEUT être omis. Si le mouchard est établi, le schéma DOIT être aussi établi si le format du mouchard a changé ; autrement, il PEUT être omis.

Si le serveur manque de ressources, il peut terminer une ou plusieurs opérations de recherche en envoyant un message SearchResultDone aux clients avec un code de résultat de lcupResourcesExhausted. Le serveur DEVRAIT joindre une commande Sync Done avec le mouchard établi. Une politique côté serveur est utilisée pour décider quelles recherches terminer. Ceci peut aussi être utilisé comme mécanisme de sécurité pour déconnecter les clients qui sont soupçonnés d'actions malveillantes, mais si le serveur peut déduire que le client est malveillant, il DEVRAIT plutôt retourner lcupSecurityViolation.

### 4.4.2 Terminaison à l'initiative du client

Si le client a besoin de terminer le processus de synchronisation et si il souhaite obtenir le mouchard qui représente l'état actuel de ses données, il produit une opération LDAP Cancel [RFC3909]. Le serveur répond immédiatement par une réponse LDAP Cancel [RFC3909]. Le serveur PEUT envoyer toutes les PDU SearchResultEntry ou SearchResultReference en instance si le serveur ne peut pas facilement interrompre ou supprimer ces résultats de recherche de sa file d'attente de sortie. Le serveur DEVRAIT envoyer aussi peu que possible de ces messages restants. Finalement, le serveur envoie le message SearchResultDone avec la commande Sync Done jointe. Si la recherche a réussi jusqu'à ce point, le champ resultCode du message SearchResultDone DOIT être annulé [RFC3909], et le mouchard DOIT être établi dans la commande Sync Done. Si il y a une condition d'erreur, le serveur PEUT retourner comme décrit au paragraphe 4.4.1, ou PEUT retourner comme décrit dans la [RFC3909].

Si le client n'est pas intéressé par les informations d'état, il peut simplement abandonner l'opération de recherche ou se déconnecter du serveur.

## 4.5 Taille et limites de temps

Le serveur DEVRA prendre en charge les limites de taille et de délai spécifiées à la Section 5 de la [RFC2251]. Le serveur DEVRAIT s'assurer que si l'opération est terminée à cause de ces conditions, le mouchard est renvoyé au client.

## 4.6 Fonctionnement sur la même connexion

Il est permis au client de produire d'autres opérations LDAP sur la connexion utilisée par le protocole. Comme chaque demande/réponse LDAP porte un identifiant de message, il n'y aura pas d'ambiguïté sur à quelle opération la PDU appartient. En partageant la connexion entre plusieurs opérations, le serveur sera capable de préserver ses ressources.

## 4.7 Interactions avec d'autres commandes

LCUP ne définit ni restriction ni garantie sur la capacité d'utiliser les commandes définies dans le présent document en conjonction avec d'autres commandes LDAP, sauf les suivantes : un serveur PEUT ignorer les commandes non critiques fournies avec la commande LCUP. Un serveur PEUT ignorer une commande définie par LCUP si elle n'est pas critique et est fournie avec d'autres commandes critiques. Si un serveur reçoit une commande critique LCUP avec une autre commande critique, et si le serveur ne prend pas en charge les commandes en même temps, le serveur DEVRAIT retourner unavailableCriticalExtension *(extension critique indisponible)*.

Il appartient à la mise en œuvre de serveur de déterminer si le serveur accepte des commandes telles que Sort ou VLV ou des commandes similaires qui changent l'ordre des entrées envoyées au client. Mais noter qu'il peut être difficile ou impossible à un serveur d'effectuer une synchronisation incrémentaire en présence de telles commandes, car le mouchard va normalement se fonder sur un numéro de changement, ou un numéro de séquence de changement (CSN, *Change Sequence Number*), ou un horodatage, ou un autre critère que l'ordre alphabétique.

## 4.8 Considérations de reproduction

L'utilisation d'un mouchard LCUP avec plusieurs DSA dans un environnement de duplication n'est pas défini par LCUP. Une mise en œuvre de LCUP peut accepter la continuation d'une session LCUP avec un autre DSA détenant une réplique du contexte LCUP. Les clients PEUVENT soumettre des mouchards retournés par un DSA à un DSA différent ; il appartient au serveur de déterminer si un mouchard est un de ceux qu'il reconnaît ou non et sinon de retourner un code de résultat approprié.

# 5. Considérations sur le côté client

## 5.1 Utilisation de mouchards avec différents critères de recherche

Le mouchard reçu du serveur après une session de synchronisation DEVRAIT être utilisé seulement avec la même spécification de recherche que celle qui a généré le mouchard. Certains serveurs PEUVENT permettre au mouchard d'être utilisé avec une spécification de recherche plus restrictive que celle qui a généré le mouchard. Si le serveur n'accepte pas le mouchard, il DOIT retourner lcupInvalidCookie. Ceci parce que autrement le client peut finir avec un lot de données incomplètes. Une spécification de recherche plus restrictive est celle qui génèrerait un sous ensemble des données produites par la spécification de recherche originale.

## 5.2 Changement de dénomination de l'objet de base

Comme un client LCUP spécifie la zone de l'arborescence avec laquelle il souhaite se synchronise au moyen de la spécification standard de recherche LDAP, le client peut se voir retourner une erreur noSuchObject si la racine de la zone de synchronisation a été renommée entre les sessions de synchronisation ou durant une session de synchronisation. Si cette condition se produit, le client peut tenter de localiser la racine en utilisant l'UUID de la racine sauvegardé dans les mémoires de données locales du client. Il peut alors répéter la demande de synchronisation en utilisant la nouvelle base de recherche. En général, un client peut détecter qu'une entrée a été renommée et appliquer les changements reçus de la bonne entrée en utilisant l'UUID plutôt que l'adressage fondé sur le DN.

## 5.3 Utilisation de recherches persistantes par rapport aux ressources

Chaque opération persistante active exige qu'une connexion TCP reste ouverte entre un client LDAP et un serveur LDAP qui ne pourrait pas rester ouverte autrement. Donc, les mises en œuvre de client sont invitées à éviter d'utiliser des opérations persistantes pour des tâches non essentielles et à clore les connexions LDAP inactives aussitôt que c'est faisable. Le serveur peut clore les connexions si ses ressources deviennent insuffisantes.

## 5.4 Continuation de références sur d'autres contextes LCUP

Le client PEUT recevoir une référence de continuation (SearchResultReference au paragraphe 4.5.3 de la [RFC2251]) si la demande de recherche s'étend sur plusieurs parties de la DIT, dont certaines peuvent exiger un mouchard LCUP différent, et certaines peuvent même ne pas être gérées par LCUP. Le client DEVRAIT tenir une antémémoire des URL LDAP retournés dans les références de continuation et les mouchards qui leur sont associés. Le client est responsable de la réalisation d'une autre recherche LCUP pour suivre les références, et DEVRAIT utiliser le mouchard correspondant à l'URL LDAP pour cette référence (si elle a un mouchard).

## 5.5 Traitement des références

Le client peut recevoir une référence (Referral au paragraphe 4.1.11 de la [RFC2251])lorsque la base de recherche est une référence subordonnée, et cela va terminer l'opération.

## 5.6 Copies multiples de la même entrée durant la phase de synchronisation

Le serveur PEUT envoyer plusieurs fois la même entrée durant une phase de synchronisation si l'entrée change durant la phase de synchronisation. Le client DEVRAIT utiliser la dernière copie envoyée de l'entrée comme étant celle en cours.

## 5.7 Épuisement des ressources du serveur traitant

Si le client reçoit un code de résultat lcupResourcesExhausted ou lcupSecurityViolation, il DEVRAIT attendre au moins 5 secondes avant de tenter une autre opération. Il est RECOMMANDÉ que le client utilise une stratégie d'attente à croissance exponentielle, mais des clients différents peuvent vouloir utiliser des stratégies d'attente différentes.

# 6. Considérations de mise en œuvre de serveur

## 6.1 Prise en charge des UUID par le serveur

Les serveurs DOIVENT prendre en charge les UUID. Les UUID sont exigés dans la commande Sync Update. De plus, les

mises en œuvre de serveur DEVRAIENT rendre disponibles les valeurs d'UUID pour les entrées comme un attribut de l'entrée, et fournir un mécanisme d'indexation ou autre pour permettre aux clients de chercher une entrée en utilisant l'attribut UUID dans le filtre de recherche. La commande syncUpdate fournit un champ UUIDAttribute pour permettre au serveur de faire savoir au client le nom ou l'OID de l'attribut à utiliser pour rechercher une entrée par son UUID.

### 6.2 Exemple d'utilisation d'un RUV comme valeur de mouchard

Par conception, le protocole accepte plusieurs schémas de mouchard. C'est pour donner à des mises en œuvre différentes la souplesse de mémorisation de toutes les informations applicables à leur environnement. Une mise en œuvre raisonnable de serveur conforme à LDUP serait d'utiliser le vecteur de mise à jour de réplique (RUV, *Replica Update Vector*). Pour chaque maître, le RUV contient le plus grand CSN vu depuis ce maître. De plus, le RUV mis en œuvre par des serveurs de répertoires (pas encore dans LDUP) contient une génération de répliques – une chaîne opaque qui identifie le magasin de données de la réplique. La valeur de génération de réplique change chaque fois que les données de la réplique sont rechargées. La génération de réplique est destinée à signaler aux homologues de réplication/synchronisation que les données de la réplique ont été rechargées et que toutes les autres répliques doivent être réinitialisées. Le RUV satisfait aux trois plus importantes propriétés du mouchard : (1) il identifie de façon univoque l'état des données du client, (2) il peut être utilisé pour synchroniser plusieurs serveurs, et (3) il peut être utilisé pour détecter que les données du serveur ont été rechargées. Si le RUV est utilisé comme mouchard, les entrées modifiées en dernier par un maître particulier doivent être envoyées au client dans l'ordre des dernières modifications de leur CSN. Cet ordre garantit que le RUV peut être mis à jour après l'envoi de chaque entrée.

### 6.3 Questions de prise en charge de mouchard

#### 6.3.1 Prise en charge de plusieurs schémas de mouchard

Un serveur peut prendre en charge un ou plusieurs schémas de mouchard LCUP. On s'attend à ce que les schémas soient publiés avec leurs OID comme des RFC. La DIT du serveur peut être partagée en différentes sections qui peuvent avoir des mouchards associés différents. Par exemple, certains serveurs peuvent utiliser une certaine sorte de mécanisme de réplication pour prendre en charge LCUP. S'il en est ainsi, la DIT peut être partitionnée en plusieurs répliques. Un client peut envoyer une demande de recherche LCUP qui s'étend sur plusieurs répliques. Certaines parties de la DIT couvertes par la portée de la demande de recherche peuvent prendre en charge LCUP et d'autres non. Le serveur DOIT envoyer une SearchResultReference *(référence de résultat de recherche)* (voir au paragraphe 4.5.3 de la [RFC2251]) lorsque change le contexte LCUP pour une entrée retournée. Le serveur DEVRAIT envoyer d'abord toutes les références aux autres contextes LCUP dans la portée de recherche, afin de permettre aux clients de traiter ces recherches en parallèle. Les URL LDAP retournés DOIVENT contenir les DN de la base d'une autre section de la DIT (bien que la mise en œuvre de serveur ait partitionné la DIT). Le client va alors produire une autre recherche LCUP en utilisant l'URL LDAP retourné. Chaque section de la DIT PEUT exiger une valeur de mouchard différente, de sorte que le client DEVRAIT tenir une antémémoire, transposant les différentes valeurs d'URL LDAP pour les différents mouchards. Si le mouchard change, le schéma peut changer aussi, mais le schéma de mouchard DOIT être le même au sein d'un certain contexte LCUP.

#### 6.3.2 Informations contenues dans le mouchard

Le mouchard doit contenir assez d'informations pour permettre au serveur de déterminer si le mouchard peut être utilisé en toute sécurité avec la spécification de recherche à laquelle il est rattaché. Comme exposé plus tôt dans ce document, le mouchard DEVRAIT n'être utilisé qu'avec la spécification de recherche qui est égale à celle pour laquelle le mouchard a été généré, mais certains serveurs PEUVENT accepter d'utiliser un mouchard avec une spécification de recherche plus restrictive que celle utilisée pour générer le mouchard.

### 6.4 Temps de réponse de la phase persistante

La spécification ne donne aucune garantie sur le moment où un serveur devrait envoyer la notification d'un changement d'entrée au client durant la phase persistante. Ceci est intentionnel car tout délai maximum spécifique serait impossible à tenir dans une mise en œuvre répartie de service de répertoire. Les mises en œuvre de serveur sont invitées à minimiser le délai avant l'envoi des notifications pour s'assurer de la satisfaction des besoins des clients en matière de notification en temps utile des changements.

### 6.5 Considérations d'échelle

Les mises en œuvre de serveurs qui prennent en charge le mécanisme décrit dans le présent document devraient s'assurer que leur mise en œuvre s'adapte bien lorsque augmente le nombre d'opérations persistantes actives et le nombre de changements faits au répertoire. Les mises en œuvre de serveurs sont aussi invitées à accepter un grand nombre de connexions de clients si

elles ont besoin de prendre en charge de grands nombres d'opérations persistantes.

### 6.6 Déréférencement d'alias

La conception de LCUP ne prend pas en compte les questions associées au déréférencement d'alias dans les recherches. Les clients DOIVENT spécifier derefAliases soit comme neverDerefAliases, soit comme derefFindingBaseObj. Les serveurs vont retourner protocolError si le client spécifie derefInSearching ou derefAlways.

## 7. Synchronisation de mémorisations de données hétérogènes

Les clients, comme un moteur joint de méta répertoire, qui synchronisent plusieurs magasins de données inscriptibles, ne fonctionneront correctement que si chaque élément d'information vient d'une seule source de données d'autorité. Dans un environnement de répliques, un contexte LCUP devraient employer le même schéma de résolution de conflit pour toutes ses répliques parce que des systèmes différents ont des notions différentes du temps et des procédures différentes de résolution de mise à jour. Par suite, un changement appliqué dans un système peut être supprimé par l'autre, empêchant donc la convergence des magasins de données.

## 8. Considérations relatives à l'IANA

Le présent document donne la liste de plusieurs valeurs qui ont été enregistrées par l'IANA. Les codes de résultat LDAP suivants ont été alloués par l'IANA comme décrit au paragraphe 3.6 de la [RFC3383] :
lcupResourcesExhausted : 113
lcupSecurityViolation : 114
lcupInvalidData : 115
lcupUnsupportedScheme : 116
lcupReloadRequired : 117

Les trois commandes définies dans le présent document ont été enregistrées comme mécanisme de protocole LDAP comme décrit au paragraphe 3.2 de la [RFC3383]. Un OID, 1.3.6.1.1.7, a été alloué par l'IANA comme décrit au paragraphe 3.1 de la [RFC3383]. Les OID pour les commandes définies dans le présent document sont déduites comme suit de celui alloué par l'IANA :
LCUP Sync Request Control : 1.3.6.1.1.7.1
LCUP Sync Update Control : 1.3.6.1.1.7.2
LCUP Sync Done Control : 1.3.6.1.1.7.3

## 9. Considérations sur la sécurité

Dans certaines situations, il peut être important d'empêcher une exposition générale des informations sur les changements qui surviennent dans un serveur LDAP. Donc, les serveurs qui mettent en œuvre le mécanisme décrit dans le présent document DEVRAIENT fournir un moyen pour appliquer un contrôle d'accès sur les entrées retournées et PEUVENT aussi fournir des mécanismes de contrôle d'accès spécifiques pour contrôler l'utilisation des commandes et des opérations étendues définies dans le présent document.

Comme avec les demandes de recherche LDAP normales, un client malveillant peut initier un grand nombre de demandes de recherche persistantes pour tenter de consommer toutes les ressources de serveur disponibles et dénier le service aux clients légitimes. Le protocole donne le moyen d'arrêter les clients malveillants en les déconnectant du serveur. Les serveurs qui mettent en œuvre le mécanisme DEVRAIENT fournir le moyen de détecter les clients malveillants. De plus, les serveurs DEVRAIENT donner le moyen de limiter le nombre de ressources qui peuvent être consommées par un seul client.

## 10. Références

### 10.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2251] M. Wahl, T. Howes et S. Kille, "Protocole léger d'accès à un répertoire (v3)", décembre 1997.

[RFC3383] K. Zeilenga, "Autorité d'allocation des numéros de l'Internet (IANA) : Considérations sur le protocole léger d'accès à un répertoire (LDAP)", septembre 2002. (*Obsolète, voir* RFC4520)

[RFC3909] K. Zeilenga, "Opération Cancel du protocole léger d'accès à un répertoire (LDAP)", octobre 2004. *(P.S.)*

[X.680] Recommandation UIT-T X.680, "Notation de syntaxe abstraite numéro un (ASN.1) - Spécification de la notation de base", Genève, 1994.

[X.690] Recommandation UIT-T X.690, "Spécification des règles de codage ASN.1 : règles de codage de base, canoniques, et distinctives", Genève, 1994.

[UUID] Organisation Internationale de Normalisation (ISO), "Technologies de l'information - Interconnexion des systèmes ouverts - Procédure d'appel à distance", ISO/CEI 11578 : 1996.

### 10.2 Références pour information

[RFC3384] E. Stokes et autres, "Protocole léger d'accès à un répertoire (version 3) : Exigences de duplication", octobre 2002. *(Info.)*

[RFC3671] K. Zeilenga, "Attributs collectifs dans le protocole léger d'accès à un répertoire (LDAP)", décembre 2003. *(P.S.)*

[RFC3672] K. Zeilenga, "Sous-entrées dans le protocole léger d'accès à un répertoire (LDAP)", décembre 2003. *(P.S.)*

## 11. Remerciements

Le protocole LCUP se fonde en partie sur le mécanisme de notification de changement de recherche persistante défini par Mark Smith, Gordon Good, Tim Howes, et Rob Weltman, sur la commande de recherche LDAPv3 déclanchée définie par Mark Wahl, et la commande LDAP de synchronisation de répertoire définie par Michael Armijo. Les membres du groupe de travail LDUP de l'IETF ont fait des contributions significatives au présent document.

## Appendice – Dispositifs laissés en dehors de LCUP

Il y a plusieurs dispositifs qui sont présents dans d'autres protocoles ou qui sont considérés comme utiles par les clients qui ne sont actuellement pas inclus dans le protocole principalement parce qu'ils sont difficiles à mettre en œuvre sur le serveur. Ces dispositifs sont brièvement exposés dans cette section.

Type de changement de recherche déclanché.
Ce dispositif est présent dans la spécification de recherche déclanchée. Un fanion est attaché à chaque entrée retournée au client qui indique la raison pour laquelle cette entrée est retournée. Les raisons possibles données dans le document sont :
- notChange : l'entrée existait dans le répertoire et correspondait à la recherche au moment où l'opération a été effectuée,
- enteredSet : l'entrée rentre dans le résultat,
- leftSet : l'entrée a quitté le résultat,
- modified : l'entrée faisait partie de l'ensemble de résultats, elle a été modifiée ou renommée, et est encore dans l'ensemble de résultats.

La caractéristique leftSet est particulièrement utile parce qu'elle indique au client qu'une entrée n'est plus dans la spécification de recherche du client et que le client peut supprimer les données associées dans son magasin de données. Ironiquement, ce dispositif est le plus difficile à mettre en œuvre sur le serveur parce que le serveur ne garde pas trace de l'état du client et n'a pas de moyen facile pour dire quelles entrées sont sorties de la portée entre les sessions de synchronisation avec le client. Un compromis pourrait être trouvé en ne fournissant ce dispositif que pour les opérations qui surviennent lorsque le client est connecté au serveur. Ceci est plus facile à réaliser parce que la décision sur le type de changement peut être prise sur la seule base du changement sans qu'il soit besoin d'aucune information d'historique. Ceci ajouterait cependant de la complexité au protocole.

Type de changement de recherche persistant.
Ce dispositif est présent dans la spécification de recherche persistante. La recherche persistante a la notion de changeTypes. Le client spécifie quel type de mise à jour va causer le retour des entrées, et facultativement, si le serveur étiquette chaque entrée retournée avec le type de changement qui a causé le retour de cette entrée. Pour LCUP, l'intention est la synchronisation

complète, et non partielle. Chaque entrée retournée par une recherche LCUP va avoir des changements associés qui peuvent concerner le client. Le client peut devoir avoir un indice local des entrées par DN ou UUID pour déterminer si l'entrée a été ajoutée ou juste modifiée. Il est facile aux clients de déterminer si l'entrée a été supprimée parce que la valeur de entryLeftSet de la commande Sync Update sera VRAI.

Envoi des changements
Des protocoles antérieurs de synchronisation n'envoyaient aux clients que les attributs modifiés de l'entrée plutôt que l'entrée entière. Bien que cette approche puisse significativement réduire la quantité de données retournée au client, elle présente plusieurs inconvénients. D'abord, sauf si un mécanisme séparé (comme le type de changement décrit ci-dessus) est utilisé pour notifier au client les entrées qui bougent dans la portée de recherche, n'envoyer que les changements peut avoir pour résultat que le client n'a qu'une version incomplète des données. Examinons un exemple. Un attribut d'une entrée est modifié. Par suite du changement, l'entrée entre dans la portée de la recherche du client. Si seuls les changements sont envoyés, le client ne verrait jamais les données initiales de l'entrée. Ensuite, ce dispositif est difficile à mettre en œuvre car le serveur peut ne pas contenir d'informations suffisantes pour construire les changements sur la seule base de l'état du serveur et du mouchard du client. D'un autre côté, ce dispositif peut être facilement mis en œuvre par le client en supposant que le client a la version précédente des données et peut effectuer des comparaisons valeur par valeur.

Limites de taille des données
Des protocoles antérieurs de synchronisation permettaient aux clients de contrôler la quantité de données qui leur sont envoyées dans la réponse à la recherche. Ce dispositif était destiné à permettre aux clients ayant des ressources limitées de traiter par lots les données de synchronisation. Cependant, une opération de recherche LDAP donne déjà le moyen au client de spécifier la limite de taille en réglant le champ sizeLimit dans la demande de recherche au nombre maximum d'entrées que le client veut recevoir. Bien que la granularité ne soit pas la même, l'hypothèse est que les clients LDAP réguliers qui peuvent traiter les limitations du protocole LDAP vont mettre en œuvre LCUP.

Ordre des données
Des protocoles antérieurs de synchronisation permettaient à un client de spécifier que les entrées parentes devraient être envoyées avant les enfants pour les opérations d'ajout et les entrées filles devaient être envoyées avant leurs parents durant les opérations de suppression. Cet ordre aide les clients à conserver une vue hiérarchique des données dans leur magasin de données. Bien qu'elle puisse être utile, cette disposition est relativement difficile à mettre en œuvre et est coûteuse à effectuer.

## Adresse des auteurs

Rich Megginson
Netscape Communications Corp
360 W. Caribbean Drive
Sunnyvale, CA 94089
USA
téléphone : +1 505 797-7762
mél : rmegginson0224@aol.com

Olga Natkovich
Yahoo, Inc.
701 First Ave.
Sunnyvale, CA 94089
USA
tél. : +1 408 349-6153
olgan@yahoo-inc.com

Mark Smith
Pearl Crescent, LLC
447 Marlpool Drive
Saline, MI 48176
USA
tél. : +1 734 944-2856
mcs@pearlcrescent.com

Jeff Parham
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
USA
tél. : +1 425 882-8080
jeffparh@microsoft.com

## Déclaration complète de droits de reproduction

documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr .

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf- ipr@ietf.org .

**Remerciement**
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society