

Groupe de travail Réseau
Request for Comments : 5277
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

S. Chisholm, Nortel
 H. Trevino, Cisco
 juillet 2008

Notifications d'événement NETCONF

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet sur la voie de la normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir 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 les mécanismes qui fournissent le service de livraison de notifications de message asynchrone pour le protocole de configuration de réseau (NETCONF, *Network Configuration protocol*). C'est une capacité facultative construite par dessus la définition de NETCONF de base. Le présent document définit les capacités et les opérations nécessaires pour prendre en charge ce service.

Table des matières

1. Introduction.....	1
1.1 Définition des termes.....	2
1.2 Motivation.....	2
1.3 Notifications d'événements dans NETCONF.....	3
2. Opérations relatives aux notifications.....	3
2.1 Abonnement à la réception de notifications d'événements.....	3
2.2 Envoi des notifications d'événement.....	4
2.3 Terminaison de l'abonnement.....	5
3. Prise en charge des concepts.....	5
3.1 Échange de capacités.....	5
3.2 Flux d'événements.....	5
3.3 Répétition de notification.....	8
3.4 Schéma de gestion des notifications.....	9
3.5 Données d'abonnement.....	11
3.6 Mécanismes de filtrage.....	11
3.7 Flux de messages.....	12
4. Schéma XML pour notifications d'événement.....	13
5. Exemples de filtrage.....	15
5.1 Filtrage de sous arborescence.....	17
5.2 Filtres XPATH.....	18
6. Capacité d'entrelacement.....	18
6.1 Description.....	18
6.2 Dépendances.....	18
6.3. Identifiant de capacité.....	18
6.4 Nouvelles opérations.....	19
6.5 Modifications des opérations existantes.....	19
7. Considérations sur la sécurité.....	19
8. Considérations relatives à l'IANA.....	19
9. Remerciements.....	20
10. Références normatives.....	20
Adresse des auteurs.....	21
Déclaration complète de droits de reproduction.....	21

1. Introduction

La [RFC4741] peut être partagée conceptuellement en quatre couches :

Couche	Exemple
Contenu	Données de configuration
Opérations	<get-config>, <edit-config>, <notification>
RPC	<rpc>, <rpc-reply>
Protocole de transport	BEEP, SSH, SSL, console

Figure 1

Le présent document définit les mécanismes qui fournissent un service de livraison de notifications de message asynchrones pour le protocole de la [RFC4741]. C'est une capacité facultative construite par dessus la définition de base de NETCONF. Le présent mémoire définit les capacités et opérations nécessaires pour la prise en charge de ce service.

1.1 Définition des termes

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].

élément : un élément XML.

abonnement : accord et méthode pour recevoir des notifications d'événement sur une session NETCONF. Concept relatif à la livraison de notifications (si il y en a à envoyer) impliquant la destination des notifications. Il est lié à la durée de vie d'une session.

opération : ce terme est utilisé pour se référer aux opérations du protocole NETCONF [RFC4741]. Dans le présent document, opération se réfère aux opérations du protocole NETCONF définies dans la prise en charge des notifications NETCONF.

événement : un événement est quelque chose d'intéressant qui se produit – un changement de configuration, une faute, un changement d'état, le franchissement d'un seuil, ou une entrée externe au système, par exemple. Souvent, il en résulte un message asynchrone, parfois appelé une notification ou notification d'événement, envoyé aux parties intéressées pour leur notifier que cet événement s'est produit.

répétition : capacité d'envoyer/renvoyer sur demande des notifications précédemment enregistrées. Ces notifications sont envoyées de façon asynchrone. Cette caractéristique est mise en œuvre par le serveur NETCONF et invoquée par le client NETCONF.

flux : un flux d'événements est un ensemble de notifications d'événement correspondant à des critères de transmission et disponible aux clients NETCONF par abonnement.

filtre : paramètre qui indique quel sous ensemble de tous les événements possibles sont intéressants. Un filtre est défini comme un ou plusieurs éléments de filtre [RFC4741], dont chacun identifie une portion du filtre global.

1.2 Motivation

La motivation de ce travail est de permettre l'envoi de messages asynchrones cohérents avec le modèle de données (contenu) et le modèle de sécurité utilisés dans une mise en œuvre de NETCONF.

La portée de ce travail englobe les besoins opérationnels suivants :

- o la livraison initiale devrait assurer qu'elle prend en charge les notifications dans le support des opérations de configuration ;
- o il devrait être possible d'utiliser le même modèle de données pour les notifications que pour les opérations de configuration ;
- o la solution devrait prendre en charge une limite raisonnable de taille de message (c'est-à-dire, pas trop courte) ;
- o les notifications devraient être portées sur un mécanisme de livraison en mode connexion ;
- o un mécanisme d'abonnement aux notifications devrait être fourni. Cela prend en compte qu'un serveur NETCONF n'envoie pas de notifications avant qu'on lui demande de le faire, et que c'est le client NETCONF qui initie le flux de notifications.
- o un mécanisme de filtrage de l'envoi des notifications devrait être mis en place dans le serveur NETCONF ;
- o les informations contenues dans une notification devraient être suffisantes pour qu'elle puisse être analysée indépendamment du mécanisme de transport. En d'autres termes, le contenu de données décrit pleinement une notification ; des informations du protocole ne sont pas nécessaires pour comprendre une notification ;
- o le serveur devrait avoir la capacité de répéter les notifications enregistrées en local.

1.3 Notifications d'événements dans NETCONF

Le présent mémoire définit un mécanisme par lequel le client NETCONF indique son intérêt à recevoir des notifications d'événement d'un serveur NETCONF en créant un abonnement à la réception des notifications d'événement. Le serveur NETCONF répond en indiquant si la demande d'abonnement a réussi et, si elle a réussi, il commence à envoyer des notifications d'événement au client NETCONF lorsque les événements se produisent dans le système. Ces notifications d'événement vont continuer d'être envoyées jusqu'à ce que la session NETCONF soit terminée ou que l'abonnement se termine pour une autre raison. L'abonnement à la notification d'événement permet au client NETCONF un certain nombre d'options pour spécifier quels événements l'intéressent. Elles sont spécifiées quand l'abonnement est créé. Noter qu'un abonnement ne peut pas être modifié une fois créé.

Le serveur NETCONF DOIT accepter et traiter l'opération <close-session>, même quand l'abonnement aux notifications est actif. Le serveur NETCONF PEUT accepter et traiter d'autres commandes ; autrement, elles vont être rejetées et le serveur DOIT envoyer une erreur "ressource refusée". Un serveur NETCONF annonce la prise en charge de la capacité de traiter d'autres commandes via la capacité ":interleave".

2. Opérations relatives aux notifications

2.1 Abonnement à la réception de notifications d'événements

L'abonnement à la notification d'événement est initié par le client NETCONF et le serveur NETCONF y répond. Un abonnement est lié à un seul flux pour la durée de vie de l'abonnement. Quand l'abonnement à la notification d'événement est créé, les événements intéressants sont spécifiés.

Le contenu d'un abonnement de notification d'événement peut être choisi en appliquant les filtres spécifiés par l'utilisateur.

2.1.1 <create-subscription>

Description : cette opération initie un abonnement de notification d'événement qui va envoyer des notifications d'événement asynchrones à l'initiateur de la commande jusqu'à la fin de l'abonnement.

Paramètres :

Flux : un paramètre facultatif, <stream>, qui indique quel flux d'événements est intéressant. Si il n'est pas présent, les événements dans le flux NETCONF par défaut vont être envoyés.

Filtre : un paramètre facultatif, <filter>, qui indique quel sous ensemble de tous les événements possibles est intéressant. Le format de ce paramètre est le même que celui du paramètre filtre dans les opérations du protocole NETCONF. Si il n'est pas présent, tous les événements non interdits pas d'autres paramètres vont être envoyés. Voir au paragraphe 3.6 plus d'informations sur les filtres.

Heure de début : un paramètre, <startTime>, utilisé pour déclencher la caractéristique de répétition et indique que la répétition devrait commencer à l'heure spécifiée. Si <startTime> n'est pas présent, ce n'est pas un abonnement de répétition. Il n'est pas valide de spécifier des heures de début plus tardives que l'instant actuel. Si le <startTime>

spécifié est plus tôt que ce que l'enregistrement peut supporter, la répétition va commencer à la première notification disponible. Ce paramètre est du type `dateTime` et est conforme à la [RFC3339]. Les mises en œuvre doivent prendre en charge les zones horaires.

Heure d'arrêt : un paramètre facultatif, `<stopTime>`, utilisé avec la caractéristique de répétition facultative pour indiquer les dernières notifications intéressantes. Si `<stopTime>` n'est pas présent, les notifications vont continuer jusqu'à la fin de l'abonnement. Doit être utilisé avec `<startTime>` et lui être postérieur. Les valeurs de `<stopTime>` dans le futur sont valides. Ce paramètre est du type `dateTime` et est conforme à la [RFC3339]. Les mises en œuvre doivent prendre en charge les zones horaires.

Réponse positive : si le serveur NETCONF peut satisfaire la demande, le serveur envoie un élément `<ok>`.

Réponse négative : un élément `<rpc-error>` est inclus dans la `<rpc-reply>` si la demande ne peut pas être achevée pour une raison quelconque. Les demandes d'abonnement vont échouer si un filtre avec une syntaxe invalide est fourni ou si le nom d'un flux non existant est fourni.

Si un `<stopTime>` est spécifié dans une demande sans avoir spécifié un `<startTime>`, l'erreur suivante est retournée :

Tag: `missing-element`

Error-type: `protocol`

Severity: `error`

Error-info: `<bad-element>`: `startTime`

Description: un élément attendu est manquant.

Si la caractéristique facultative de répétition est demandée mais n'est pas prise en charge par le serveur NETCONF, l'erreur suivante est retournée :

Tag: `operation-failed`

Error-type: `protocol`

Severity: `error`

Error-info: `none`

Description : la demande n'a pas pu être achevée parce que l'opération demandée a échoué pour une raison non couverte par une autre condition d'erreur.

Si un `<stopTime>` est demandé qui est plus tôt que le `<startTime>` spécifié, l'erreur suivante est retournée :

Tag: `bad-element`

Error-type: `protocol`

Severity: `error`

Error-info: `<bad-element>`: `stopTime`

Description : une valeur d'élément n'est pas correcte ; par exemple, mauvais type, hors gamme, discordance de schéma.

Si un `<startTime>` est demandé plus tard que l'instant actuel, l'erreur suivante est retournée :

Tag: `bad-element`

Error-type: `protocol`

Severity: `error`

Error-info: `<bad-element>`: `startTime`

Description : une valeur d'élément n'est pas correcte; par exemple, mauvais type, hors gamme, discordance de schéma.

2.1.1.1 Exemple d'usage

L'exemple suivant montre la création d'un simple abonnement. Des exemples plus complexes se trouvent à la Section 5.

```
<netconf:rpc message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  </create-subscription>
</netconf:rpc>
```

2.2 Envoi des notifications d'événement

Une fois l'abonnement établi, le serveur NETCONF envoie les notifications d'événement en asynchrone sur la connexion.

2.2.1 <notification>

Description : une notification d'événement est envoyée au client qui a initié une commande <create-subscription> en asynchrone quand un événement intéressant (c'est-à-dire, satisfaisant les critères de filtrage spécifiés) s'est produit. Une notification d'événement est un document XML complet et bien formé. Noter que <notification> n'est pas une méthode d'appel de procédure distante (RPC, *Remote Procedure Call*) mais plutôt l'élément de niveau supérieur qui identifie le message unidirectionnel comme une notification.

Paramètres :

eventTime : l'heure à laquelle l'événement a été généré par la source de l'événement. Ce paramètre est de type dateTime et est conforme à la [RFC3339]. Les mises en œuvre doivent prendre en charge les zones horaires. Il contient aussi un contenu étiqueté spécifique de la notification, si il en est. À l'exception de <eventTime>, le contenu de la notification sort du domaine d'application de ce document.

Réponse : pas de réponse. Non applicable.

2.3 Terminaison de l'abonnement

La clôture de l'abonnement de notification d'événement peut être faite en utilisant l'opération <close-session> à partir de la session d'abonnement ou en terminant la session NETCONF (<kill-session>) ou la session de transport sous-jacente à partir d'une autre session. Si une heure d'arrêt est fournie quand l'abonnement est créé, l'abonnement va se terminer après l'atteinte de l'heure d'arrêt. Dans ce cas, la session NETCONF va encore être une session active.

3. Prise en charge des concepts

3.1 Échange de capacités

La capacité de traiter et envoyer des notifications d'événement est annoncée durant l'échange de capacités entre le client et le serveur NETCONF.

3.1.1 Identifiant de capacité

"urn:ietf:params:netconf:capability:notification:1.0"

3.1.2 Exemple de capacité

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>
      urn:ietf:params:xml:ns:netconf:base:1.0
    </capability>
    <capability>
      urn:ietf:params:netconf:capability:startup:1.0
    </capability>
    <capability>
      urn:ietf:params:netconf:capability:notification:1.0
    </capability>
  </capabilities>
  <session-id>4</session-id>
</hello>
```


3.2.2. Format de contenu de flux d'événements

Le contenu de tous les flux d'événements rendus disponibles à un client NETCONF (c'est-à-dire, la notification envoyée par le serveur NETCONF) DOIT être codé en XML.

3.2.3 Flux d'événements par défaut

Une mise en œuvre de serveur NETCONF qui prend en charge la capacité de notification DOIT prendre en charge la notification de flux d'événements "NETCONF". Ce flux contient toutes les notifications d'événement NETCONF XML prises en charge par le serveur NETCONF. La chaîne exacte "NETCONF" est utilisée durant l'annonce de la prise en charge du flux durant l'opération <get> sur <streams> et durant l'opération <create-subscription>. La définition des notifications d'événement et de leur contenu, au delà de l'inclusion de <eventTime>, pour ce flux d'événements, sort du domaine d'application du présent document.

3.2.4 Sources de flux d'événements

À l'exception du flux d'événements par défaut (NETCONF), la spécification de sources de flux d'événements supplémentaires (par exemple, du protocole simple de gestion de réseau (SNMP), syslog) sort du domaine d'application du présent document. Les mises en œuvre de serveur NETCONF peuvent développer toute source désirée de flux d'événements dans la création des flux d'événements pris en charge.

3.2.5 Découverte de flux d'événements

Un client NETCONF restitue la liste des flux d'événements pris en charge d'un serveur NETCONF en utilisant l'opération <get>.

3.2.5.1 Restitution de nom utilisant l'opération <get>

La liste des flux d'événements disponibles est restituée en demandant la sous arborescence <streams> via une opération <get>. Les flux d'événements disponibles pour la session demandeuse sont retournés dans la réponse contenant les éléments <name> et <description>, où l'élément <name> est obligatoire, et sa valeur est unique dans la portée du serveur NETCONF. Une réponse vide est retournée si il n'y a pas de flux d'événements disponible, du fait des filtres spécifiés par l'utilisateur sur l'opération <get>.

Les informations supplémentaires disponibles sur un flux incluent si la répétition de notification est disponible et, si elle l'est, l'horodatage de la plus ancienne notification possible à répéter.

L'exemple suivant montre la restitution de la liste des flux d'événements disponibles en utilisant l'opération <get> :

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
  <filter type="subtree">
    <netconf xmlns="urn:ietf:params:xml:ns:netmod:notification">
      <streams/>
    </netconf>
  </filter>
</get>
</rpc>
```

Le serveur NETCONF retourne une liste de flux d'événements disponibles pour l'abonnement : NETCONF, SNMP, et syslog-critical dans cet exemple.

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <netconf xmlns="urn:ietf:params:xml:ns:netmod:notification">
      <streams>
```

```

<stream>
  <name>NETCONF</name>
  <description>flux d'événements NETCONF par défaut
</description>
  <replaySupport>true</replaySupport>
  <replayLogCreationTime>
    2007-07-08T00:00:00Z
  </replayLogCreationTime>
</stream>
<stream>
  <name>SNMP</name>
  <description>notifications SNMP</description>
  <replaySupport>>false</replaySupport>
</stream>
<stream>
  <name>syslog-critical</name>
  <description>Critical et sévérité plus forte
</description>
  <replaySupport>true</replaySupport>
  <replayLogCreationTime>
    2007-07-01T00:00:00Z
  </replayLogCreationTime>
</stream>
</streams>
</netconf>
</data>
</rpc-reply>

```

3.2.5.2 Abonnement au flux d'événements

Un client NETCONF peut demander au serveur NETCONF la liste des flux d'événements disponibles pour cette session et produire ensuite une demande `<create-subscription>` avec le nom du flux d'événements désiré. Omettre le nom du flux d'événements de la demande `<create-subscription>` résulte en l'abonnement au flux d'événements NETCONF par défaut.

3.2.5.2.1 Filtrage des contenus de flux d'événements

L'ensemble des notifications d'événement livrées dans un flux d'événements peut être encore précisé en appliquant un filtre spécifié par l'utilisateur fourni au moment de la création de l'abonnement (`<create-subscription>`). C'est un filtre transitoire associé à l'abonnement à la notification d'événement et qui ne modifie pas la configuration du flux d'événements. L'élément filtre est appliqué au contenu de l'enveloppe `<notification>` et non à l'enveloppe elle-même. Voir des exemples à la Section 5. Une sous arborescence ou le filtrage XPATH peut être utilisé.

La prise en charge de XPATH pour la capacité de notification est annoncée au titre de l'annonce normale de capacité XPATH. Si la prise en charge de XPATH est annoncée via la capacité XPATH, alors XPATH est pris en charge pour le filtrage de notification. Si cette capacité n'est pas annoncée, XPATH n'est pas pris en charge pour le filtrage de notification.

3.3 Répétition de notification

3.3.1 Généralités

Replay est la capacité de créer un abonnement d'événement qui va renvoyer les notifications récemment générées, ou dans certains cas les envoyer pour la première fois à un client NETCONF particulier. Ces notifications sont envoyées de la même façon que les notifications normales.

Une répétition de notifications est spécifiée en incluant le paramètre facultatif `<startTime>` à la commande de l'abonnement, qui indique l'heure de début de la répétition. L'heure de fin est spécifiée en utilisant le paramètre facultatif `<stopTime>`. Si il est absent, les notifications vont continuer d'être envoyées jusqu'à la fin de l'abonnement.

Un flux de notifications qui prend en charge "replay" n'est pas supposé avoir une fourniture illimitée de notifications

sauvegardées disponibles pour satisfaire toute demande de répétition. Les clients peuvent interroger `<replayLogCreationTime>` et `<replayLogAgedTime>` pour apprendre la disponibilité des notifications à la répétition.

Le nombre réel de notifications mémorisées disponibles pour restitution à un instant donné est une affaire spécifique de la mise en œuvre du serveur NETCONF. Les paramètres de commande pour cet aspect du dispositif sortent du domaine d'application de ce document.

La répétition dépend d'un flux de notification qui prend en charge une forme d'enregistrement de notification, bien qu'il n'y ait pas de restrictions sur la taille ou la forme de l'enregistrement, ou l'endroit où il se situe dans l'appareil. Si un flux prend ou non en charge la répétition peut être découvert en faisant une opération `<get>` sur l'élément `<streams>` du schéma de gestion de notifications et en cherchant la valeur de l'objet `<replaySupport>`. Ce schéma fournit aussi l'élément `<replayLogCreationTime>` pour indiquer la plus ancienne notification enregistrée disponible.

3.3.2 Création d'un abonnement avec répétition

Cette caractéristique utilise les paramètres facultatifs de la commande `<create-subscription>` appelés `<startTime>` et `<stopTime>`. `<startTime>` identifie la date et l'heure d'intérêt la plus ancienne pour les notifications d'événement à répéter et indique aussi qu'un abonnement va fournir la répétition des notifications. Les événements générés avant cet instant ne sont pas pris en compte. `<stopTime>` spécifie la dernière date et l'heure d'intérêt pour les notifications d'événement à répéter. Si il est absent, les notifications vont alors continuer d'être envoyées jusqu'à la fin de l'abonnement.

Noter que `<startTime>` et `<stopTime>` sont associés à l'instant où un événement a été généré par la source de l'événement.

Une notification `<replayComplete>` est envoyée pour indiquer que toutes les notifications de répétition ont été envoyées et ne doit pas être envoyée pour une autre raison. Si cet abonnement a une heure de fin, alors cette session redevient une session NETCONF normale. Le serveur NETCONF va alors accepter des opérations `<rpc>` même si le serveur n'acceptait pas précédemment de telles opérations à cause de l'absence de prise en charge de l'entrelaçage. Dans le cas d'un abonnement sans heure de fin, après l'envoi de la notification `<replayComplete>`, on peut s'attendre à ce que toutes les notifications générées depuis le début de la création de l'abonnement soient envoyées, suivies par les notifications telles qu'elles surviennent naturellement dans le système.

Les notifications `<replayComplete>` et `<notificationComplete>` ne peuvent pas être filtrées. Elles vont toujours être envoyées sur un abonnement de répétition qui a spécifié respectivement un `<startTime>` et un `<stopTime>`.

3.4 Schéma de gestion des notifications

Ce schéma est utilisé pour apprendre les flux d'événements pris en charge sur le système. Il contient aussi la définition des notifications `<replayComplete>` et `<notificationComplete>`, qui sont envoyées pour indiquer, respectivement, qu'une répétition d'événement a envoyé toutes les notifications applicables, et que l'abonnement s'est terminé.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:ncEvent="urn:ietf:params:xml:ns:netconf:notification:1.0"
  xmlns:manageEvent="urn:ietf:params:xml:ns:netmod:notification"
  targetNamespace="urn:ietf:params:xml:ns:netmod:notification"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xml:lang="en" version="1.0">
  <xs:annotation>
    <xs:documentation xml:lang="fr"> Un schéma qui peut être utilisé pour apprendre les flux d'événements courants. Il
    contient aussi les notifications replayComplete et notificationComplete.
  </xs:documentation>
  </xs:annotation>

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
    schemaLocation="netconf.xsd"/>
```

```

<xs:import namespace=
  "urn:ietf:params:xml:ns:netconf:notification:1.0"
  schemaLocation="notification.xsd"/>

<xs:element name="netconf" type="manageEvent:Netconf"/>

<xs:complexType name="Netconf">
  <xs:sequence>
    <xs:element name="streams" >
      <xs:annotation>
        <xs:documentation>
          Liste des flux d'événements pris en charge par le système. Quand une interrogation est produite, l'ensemble de flux
          retourné est déterminé selon les privilèges de l'utilisateur.
        </xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence minOccurs="1" maxOccurs="unbounded">
          <xs:element name="stream">
            <xs:annotation>
              <xs:documentation>
                Nom, description, et autres informations du flux.
              </xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:sequence>
                <xs:element name="name"
                  type="ncEvent:streamNameType">
                  <xs:annotation>
                    <xs:documentation>
                      Nom du flux d'événements. Si c'est le flux NETCONF par défaut, il doit avoir la valeur "NETCONF".
                    </xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="description"
                  type="xs:string">
                  <xs:annotation>
                    <xs:documentation>
                      Description du flux d'événements, incluant des informations comme le type d'événements envoyés sur ce flux.
                    </xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="replaySupport"
                  type="xs:boolean">
                  <xs:annotation>
                    <xs:documentation>
                      Indication de si un événement de répétition est ou non disponible sur ce flux.
                    </xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="replayLogCreationTime"
                  type="xs:dateTime" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>
                      Horodatage de la création de l'enregistrement utilisé pour prendre en charge la fonction de répétition sur ce flux.
                      Noter que ce peut être plus tôt que la première notification disponible dans l'enregistrement. Cet objet est mis à jour si
                      l'enregistrement est réinitialisé pour une raison quelconque. Cet objet DOIT être présent si la répétition est prise en charge.
                    </xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="replayLogAgedTime"
                  type="xs:dateTime" minOccurs="0">

```

```

    <xs:annotation>
      <xs:documentation>
Horodatage de la dernière notification périmée dans l'enregistrement. Cet objet DOIT être présent si la répétition est prise
en charge et si des notifications sont périmées dans l'enregistrement.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="ReplayCompleteNotificationType">
  <xs:complexContent>
    <xs:extension base="ncEvent:NotificationContentType"/>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:element name="replayComplete"
  type="manageEvent:ReplayCompleteNotificationType"
  substitutionGroup="ncEvent:notificationContent">
  <xs:annotation>
    <xs:documentation>
      Cette notification est envoyée pour signaler la fin d'une portion de répétition d'un abonnement.
    </xs:documentation>
  </xs:annotation>
</xs:element>

```

```

<xs:complexType name="NotificationCompleteNotificationType">
  <xs:complexContent>
    <xs:extension base="ncEvent:NotificationContentType"/>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:element name="notificationComplete"
  type="manageEvent:NotificationCompleteNotificationType"
  substitutionGroup="ncEvent:notificationContent">
  <xs:annotation>
    <xs:documentation>
      Cette notification est envoyée pour signaler la fin d'un abonnement de notification. Elle est envoyée dans le cas où
      stopTime était spécifié durant la création de l'abonnement.
    </xs:documentation>
  </xs:annotation>
</xs:element>

```

```

</xs:schema>

```

3.5 Données d'abonnement

Les abonnements sont des informations d'état non persistantes, et leur durée de vie est définie par leur session ou par le paramètre <stopTime>.

3.6 Mécanismes de filtrage

Si un élément de filtre est spécifié pour chercher des données d'une valeur particulière, et si l'élément de données n'est pas

présent dans une notification d'événement particulière pour la valeur recherchée, la notification va être filtrée. Par exemple, si on cherche la valeur 'severity=critical' dans une configuration de notification d'événement où ce champ n'est pas pris en charge, la notification va être filtrée.

Pour le filtrage de sous arborescence, un ensemble de nœuds non vide signifie que le filtre correspond. Pour le filtrage XPath, les mécanismes définis dans [XPATH] devraient être utilisés pour convertir la valeur retournée en booléen.

3.6.1 Filtrage

Le filtrage est explicitement déclaré quand est créé l'abonnement à la notification d'événement. Cela est spécifié via le paramètre 'filter'. Un filtre n'existe que comme paramètre à l'abonnement.

3.7 Flux de messages

La figure suivante décrit le flux de messages entre un client NETCONF (C) et le serveur NETCONF (S) afin de créer un abonnement et commencer le flux de notifications. Cet abonnement spécifie un <startTime>, de sorte que le serveur commence par répéter les notifications enregistrées. Il est possible que de nombreuses séquences rpc/rpc-reply se produisent avant que l'abonnement soit créé, mais ce n'est pas décrit dans la figure.

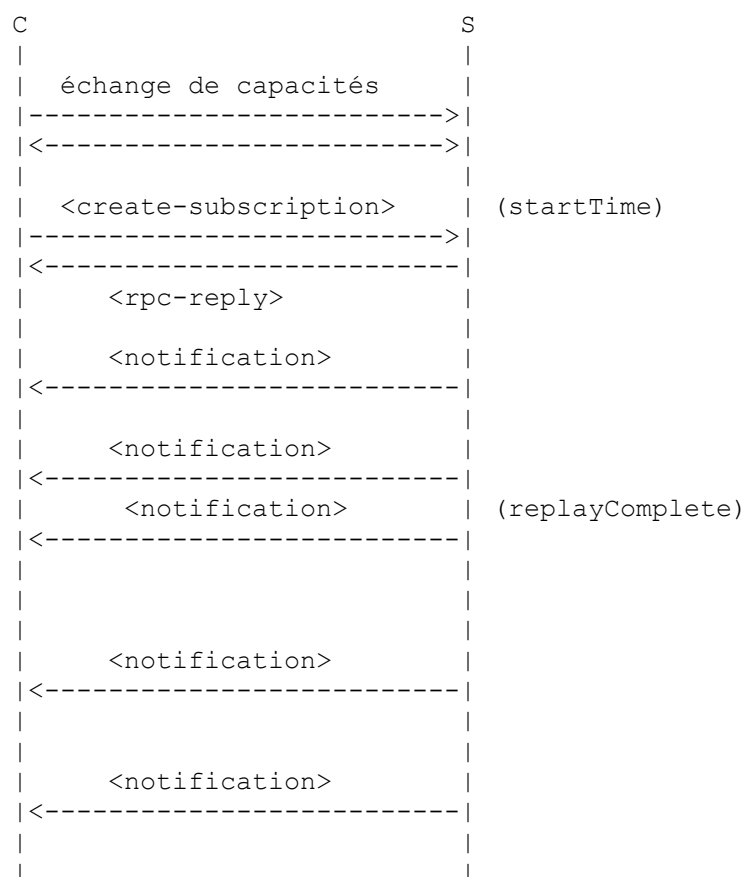


Figure 3

La figure suivante décrit le flux de messages entre un client NETCONF (C) et un serveur NETCONF (S) afin de créer un abonnement et commencer le flux de notifications. Cet abonnement a spécifié un <startTime> et un <stopTime> de sorte qu'il commence par répéter les notifications enregistrées et ensuite retourne à une session NETCONF normale de commande-réponse après que les notifications <replayComplete> et <notificationComplete> ont été envoyées et qu'il est disponible pour traiter les demandes <rpc>. Il est possible que de nombreuses séquences rpc/rpc-reply se produisent avant que l'abonnement soit créé, mais ce n'est pas décrit dans la figure.

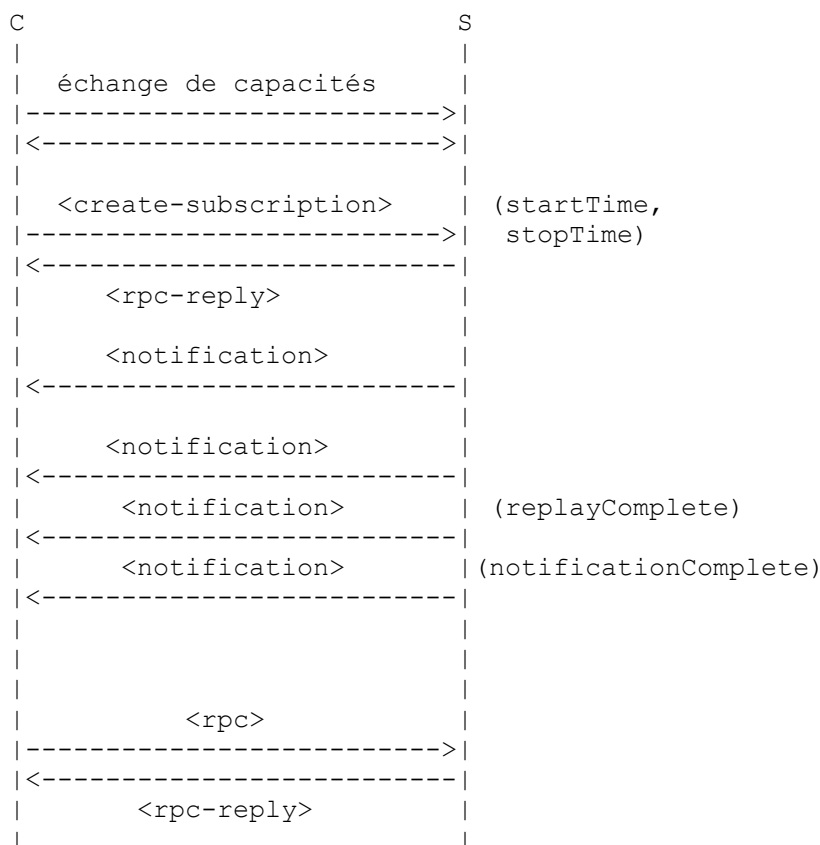


Figure 4

4. Schéma XML pour notifications d'événement

Le schéma XML suivant définit les notifications d'événements NETCONF.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"
  targetNamespace="urn:ietf:params:xml:ns:netconf:notification:1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xml:lang="en">

<!-- importation des définitions XML standard -->

<xs:import namespace="http://www.w3.org/XML/1998/namespace"
  schemaLocation="http://www.w3.org/2001/xml.xsd">
  <xs:annotation>
    <xs:documentation>
      Cette importation accède aux groupes d'attributs xml: pour le xml:lang comme déclaré dans l'élément error-message.
    </xs:documentation>
  </xs:annotation>
</xs:import>

<!-- importation des définitions netconf de base -->
<xs:import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
  schemaLocation="netconf.xsd"/>

```

```

<!-- ***** Opérations symétriques *****-->

<!-- opération <create-subscription> -->

<xs:complexType name="createSubscriptionType">
  <xs:complexContent>
    <xs:extension base="netconf:rpcOperationType">
      <xs:sequence>
        <xs:element name="stream"
          type="streamNameType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              Paramètre facultatif qui indique quel flux d'événements est intéressant. Si il n'est pas présent, les événements
              dans le flux NETCONF par défaut vont alors être envoyés.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="filter"
          type="netconf:filterInlineType"
          minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              Paramètre facultatif qui indique quel sous ensemble de tous les événements possibles est intéressant. Le
              format de ce paramètre est le même que celui du paramètre filtre dans les opérations du protocole NETCONF. Si il n'est
              pas présent, tous les événements non interdits par d'autres paramètres vont être envoyés.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="startTime" type="xs:dateTime"
          minOccurs="0" >
          <xs:annotation>
            <xs:documentation>
              Paramètre utilisé pour déclencher le dispositif de répétition qui indique que la répétition devrait
              commencer à l'instant spécifié. Si l'heure de début n'est pas présente, ce n'est pas un abonnement de répétition.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="stopTime" type="xs:dateTime"
          minOccurs="0" >
          <xs:annotation>
            <xs:documentation>
              Paramètre facultatif utilisé avec le dispositif facultatif de répétition pour indiquer les plus récentes
              notifications intéressantes. Si l'heure d'arrêt n'est pas présente, les notifications vont continuer jusqu'à la fin de
              l'abonnement. Doit être utilisé avec startTime.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="streamNameType">
  <xs:annotation>
    <xs:documentation>
      Nom d'un flux d'événements.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<xs:element name="create-subscription"

```

```

type="createSubscriptionType"
substitutionGroup="netconf:rpcOperation">
<xs:annotation>
  <xs:documentation>
    Commande pour créer un abonnement de notifications. Elle prend comme argument le nom du flux et du filtre de
    notification. Ces deux options limitent le contenu de l'abonnement. De plus, il y a deux paramètres relatifs au temps,
    startTime et stopTime, qui peuvent être utilisés pour choisir l'intervalle de temps intéressant pour le dispositif de répétition
    de notification.
  </xs:documentation>
</xs:annotation>
</xs:element>

```

```
<!-- ***** Opérations unidirectionnelles *****-->
```

```
<!-- Opération <Notification> -->
  <xs:complexType name="NotificationContentType"/>
```

```
<xs:element name="notificationContent"
  type="NotificationContentType" abstract="true"/>
```

```
<xs:complexType name="NotificationType">
  <xs:sequence>
    <xs:element name="eventTime" type="xs:dateTime">
      <xs:annotation>
        <xs:documentation>
          Heure à laquelle l'événement a été généré par la source de l'événement.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="notificationContent"/>
  </xs:sequence>
</xs:complexType>
```

```

  <xs:element name="notification" type="NotificationType"/>
</xs:schema>

```

5. Exemples de filtrage

Cette Section donne des exemples pour illustrer les diverses méthodes de filtrage de contenu sur un abonnement de notification d'événement.

Afin d'illustrer l'utilisation des expressions de filtre, il est nécessaire de supposer des contenus de notification d'événement. Les exemples ci-dessous supposent que la définition de schéma de notification d'événement a un élément <event> au niveau supérieur consistant en la classe d'événements (par exemple, faute, état, configuration) une entité qui fait rapport, et un état de sévérité ou de fonctionnement.

Les exemples dans cette section sont générés à partir du schéma fictif suivant :

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://example.com/event/1.0"
  xmlns="http://example.com/event/1.0"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ncEvent="urn:ietf:params:xml:ns:netconf:notification:1.0">

  <xs:import namespace=
    "urn:ietf:params:xml:ns:netconf:notification:1.0"
    schemaLocation="notification.xsd"/>

```

```

<xs:complexType name="eventType">
  <xs:complexContent>
    <xs:extension base="ncEvent:NotificationContentType">
      <xs:sequence>
        <xs:element name="eventClass" />
        <xs:element name="reportingEntity">
          <xs:complexType>
            <xs:sequence>
              <xs:any namespace="##any"
                processContents="lax"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:choice>
          <xs:element name="severity"/>
          <xs:element name="operState"/>
        </xs:choice>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="event"
  type="eventType"
  substitutionGroup="ncEvent:notificationContent"/>
</xs:schema>

```

La définition de notification fictive ci-dessus pourrait résulter en la liste d'échantillons de notification suivante, qui est utilisée dans les exemples de cette section.

```

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:01:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>
      <card>Ethernet0</card>
    </reportingEntity>
    <severity>major</severity>
  </event>
</notification>

```

```

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:02:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>
      <card>Ethernet2</card>
    </reportingEntity>
    <severity>critical</severity>
  </event>
</notification>

```

```

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:04:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>

```



```

    <card>ATM1</card>
  </reportingEntity>
  <severity>minor</severity>
</event>
</notification>

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:10:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>state</eventClass>
    <reportingEntity>
      <card>Ethernet0</card>
    </reportingEntity>
    <operState>enabled</operState>
  </event>
</notification>

```

5.1 Filtrage de sous arborescence

Le filtrage de sous arborescence XML ne convient pas bien pour la création de définitions élaborées de filtre étant donné qu'il prend en charge seulement les comparaisons d'égalité et l'application de l'opérateur logique OU (par exemple, dans une sous arborescence d'événements, "donnez moi toutes les notifications d'événement qui ont severity=critical, severity=major, ou severity=minor"). Néanmoins, il peut être utilisé pour définir de simples filtres de transmission de notification d'événement comme on le montre ci-dessous.

L'exemple suivant illustre comment choisir des événements de fautes qui ont des sévérités de critical, major, ou minor. L'évaluation des critères de filtrage est comme suit :

((fault & severity=critical) | (fault & severity=major) | (fault & severity=minor))

```

<netconf:rpc netconf:message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="subtree">
      <event xmlns="http://example.com/event/1.0">
        <eventClass>fault</eventClass>
        <severity>critical</severity>
      </event>
      <event xmlns="http://example.com/event/1.0">
        <eventClass>fault</eventClass>
        <severity>major</severity>
      </event>
      <event xmlns="http://example.com/event/1.0">
        <eventClass>fault</eventClass>
        <severity>minor</severity>
      </event>
    </filter>
  </create-subscription>
</netconf:rpc>

```

L'exemple suivant illustre comment choisir l'état ou la configuration EventClasses ou des événements de faute qui se rapportent à une carte Ethernet0. L'évaluation des critères de filtrage est comme suit :

(state | config | (fault & (card=Ethernet0)))

```

<netconf:rpc netconf:message-id="101" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="subtree">

```

```

<event xmlns="http://example.com/event/1.0">
  <eventClass>state</eventClass>
</event>
<event xmlns="http://example.com/event/1.0">
  <eventClass>config</eventClass>
</event>
<event xmlns="http://example.com/event/1.0">
  <eventClass>fault</eventClass>
  <reportingEntity>
    <card>Ethernet0</card>
  </reportingEntity>
</event>
</filter>
</create-subscription>
</netconf:rpc>

```

5.2 Filtres XPATH

L'exemple XPATH suivant illustre comment choisir des notifications EventClass de fautes qui ont des sévérités de critical, major, ou minor. L'évaluation des critères de filtrage est comme suit :

```
((fault) & ((severity=critical) | (severity=major) | (severity = minor)))
```

```

<netconf:rpc netconf:message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="xpath"
      xmlns:ex="http://example.com/event/1.0"
      select="/ex:event[ex:eventClass='fault' et
        (ex:severity='minor' ou ex:severity='major' ou ex:severity='critical')]">
    </create-subscription>
  </netconf:rpc>

```

L'exemple suivant illustre comment choisir l'état et la configuration EventClasses ou événement de faute de toutes sévérités qui viennent de card Ethernet0. L'évaluation des critères de filtrage est comme suit :

```
( state | config | (fault & card=Ethernet0))
```

```

<netconf:rpc message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription
    xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <filter netconf:type="xpath"
      xmlns:ex="http://example.com/event/1.0"
      select="/ex:event[
        (ex:eventClass='state' ou ex:eventClass='config') ou ((ex:eventClass='fault' et ex:card='Ethernet0'))]">
    </create-subscription>
  </netconf:rpc>

```

6. Capacité d'entrelacement

6.1 Description

La capacité :interleave indique que l'homologue NETCONF prend en charge la capacité d'entrelacer d'autres opérations NETCONF au sein d'un abonnement de notifications. Cela signifie que le serveur NETCONF DOIT recevoir, traiter, et répondre aux demandes NETCONF sur une session avec un abonnement de notification actif. Cette capacité aide à l'adaptabilité en réduisant le nombre total de sessions NETCONF requises par un opérateur ou application de gestion.

6.2 Dépendances

Cette capacité dépend de la capacité de notification prise en charge.

6.3 Identifiant de capacité

La capacité :interleave est identifiée par la chaîne de capacités suivante :

```
urn:ietf:params:netconf:capability:interleave:1.0
```

6.4 Nouvelles opérations

Aucune.

6.5 Modifications des opérations existantes

Quand un <create-subscription> est envoyé alors qu'un autre abonnement est actif sur cette session, l'erreur suivante va être retournée :

Tag : operation-failed

Error-type: protocol

Severity: error

Error-info: none

Description: la demande ne peut pas être satisfaite parce que l'opération demandée a échoué pour une raison non couverte par une autre condition d'erreur.

7. Considérations sur la sécurité

Les considérations sur la sécurité du document [RFC4741] de base s'appliquent aussi à la capacité Notification.

Le cadre de contrôle d'accès et le choix du transport vont avoir un impact majeur sur la sécurité de la solution.

Les éléments <notification> ne sont jamais envoyés avant que la couche de transport et la couche NETCONF, incluant l'échange de capacités, aient été établies et que le gestionnaire ait été identifié et authentifié.

Il est recommandé de prendre soin de sécuriser l'exécution :

- o de l'invocation de <create-subscription>

- o de <get> sur les modèles de données en lecture seule

- o du contenu de <notification>.

Exécution sécurisée signifie de s'assurer qu'un transport sûr est utilisé ainsi que de s'assurer que l'utilisateur a une autorisation suffisante pour effectuer la fonction qui lui est demandée à l'égard du sous ensemble spécifique de contenu NETCONF impliqué. Quand un <get> est reçu qui se réfère au contenu défini dans le présent mémoire, les clients devraient être seulement capables de voir le contenu pour lequel ils ont des privilèges suffisants. Une opération <create-subscription> peut être considérée comme un <get> différé, et le contenu auquel les différents utilisateurs peuvent accéder peut varier. Cet accès différé est reflété dans la <notification> à laquelle les différents utilisateurs sont capables de s'abonner.

Un problème potentiel de sécurité est le transport de données provenant de flux non NETCONF, comme syslog et SNMP. Ces données peuvent être plus vulnérables (ou moins vulnérables) quand elles sont transportées sur NETCONF que quand elles sont transportées en utilisant le protocole normalement utilisé pour les transporter, selon les accreditifs de sécurité des deux sous systèmes. Le serveur NETCONF est chargé d'appliquer le contrôle d'accès au contenu du flux.

Le contenu des notifications, ainsi que les noms de flux d'événements, peuvent contenir des informations sensibles et il faut veiller à s'assurer qu'ils ne sont vus que par des utilisateurs autorisés. Le serveur NETCONF NE DOIT PAS inclure de contenu dans une notification que l'utilisateur n'est pas autorisé à voir.

Si un abonnement est créé avec un <stopTime>, la session NETCONF va revenir à une session NETCONF normale de commande-réponse quand la répétition est terminée. Il est de la responsabilité du client NETCONF de fermer cette session

quand elle n'est plus utilisée.

Si un client NETCONF malveillant ou fautif envoie un certain nombre de demandes <create-subscription>, ces abonnements s'accumulent et peuvent épuiser les ressources du système. Dans cette situation, les abonnements peuvent être terminés en mettant fin aux sessions NETCONF sous-jacentes suspectes en utilisant l'opération <kill-session>.

8. Considérations relatives à l'IANA

Le présent document enregistre trois URI pour l'espace de noms NETCONF XML dans le registre XML de l'IETF [RFC3688].

Suivant le format de la RFC 3688, l'IANA a fait l'enregistrement suivant. Noter que les URN de capacités sont aussi conformes au paragraphe 10.3 de la [RFC4741].

Indice	Identifiant de capacité
notification	urn:ietf:params:netconf:capability:notification:1.0
interleave	urn:ietf:params:netconf:capability:interleave:1.0

URI: urn:ietf:params:xml:ns:netmod:notification

URI: urn:ietf:params:xml:ns:netconf:notification:1.0

Contact d'enregistrement : IESG.

XML : N/A, l'URI demandé est un espace de noms XML.

De plus, l'IANA a enregistré le schéma XML défini dans la Section 4.

9. Remerciements

Merci à Gilbert Gagnon, Greg Wilbur, et Kim Curran de leur apports aux premiers travaux sur ce document. De plus, les éditeurs tiennent à remercier les apports de la session d'édition de Vancouver de la part des personnes suivantes : Orly Nicklass, James Balestriere, Yoshifumi Atarashi, Glenn Waters, Alexander Clemm, Dave Harrington, Dave Partain, Ray Atarashi, David Perkins, et les personnes supplémentaires suivantes de la session d'édition de Montreal : Balazs Lengyel, Phil Shafer, Rob Enns, Andy Bierman, Dan Romascanu, Bert Wijnen, Simon Leinen, Juergen Schoenwaelder, Hideki Okita, Vincent Cridlig, Martin Bjorklund, Olivier Festor, Radu State, Brian Trammell, et William Chow. Nous tenons aussi à remercier Li Yan de ses nombreuses relectures, ainsi que Suresh Krishnan de sa révision générale du document.

10. Références normatives

- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [RFC3339] G. Klyne, C. Newman, "[La date et l'heure sur l'Internet](#) : horodatages", juillet 2002. (P.S.)
- [RFC3688] M. Mealling, "[Registre XML de l'IETF](#)", BCP 81, janvier 2004.
- [RFC4741] R. Enns, éd., "[Protocole de configuration NETCONF](#)", décembre 2006. (P.S.)
- [XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, February 1998, <<http://www.w3.org/TR/1998/REC-xml-19980210>>.
- [XMLSchema] Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", W3C <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/structures.html>, octobre 2004.

[XPATH] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", W3C <http://www.w3.org/TR/1999/REC-xpath-19991116>, novembre 1999.

Adresse des auteurs

Sharon Chisholm
Nortel
3500 Carling Ave
Nepean, Ontario K2H 8E9
Canada
mél : schishol@nortel.com

Hector Trevino
Cisco
Suite 400
9155 E. Nichols Ave
Englewood, CO 80112
USA
mél : htrevino@cisco.com

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2008)

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournis sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY, le IETF TRUST et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les 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 fourni par l'activité de soutien administratif de l'IETF (IASA).