Groupe de travail Réseau
Request for Comments : 4743
Catégorie : Sur la voie de la normalisation

T. Goddard, ICEsoft Technologies, Inc.
décembre 2006
Traduction Claude Brière de L'Isle

Utilisation de NETCONF sur le protocole simple d'accès aux objets (SOAP)

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de 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 protocole de configuration de réseau (NETCONF) est applicable à une large gamme d'appareils dans divers environnements. Les services de la Toile sont un de ces environnements et sont présentement caractérisés par l'utilisation du protocole simple d'accès aux objets (SOAP, *Simple Object Access Protocol*). NETCONF trouve de nombreux avantages à cet environnement : de la réutilisation des normes existantes à la facilité de développement de logiciels, et à l'intégration dans les systèmes déployés. On décrit ici SOAP sur HTTP et SOAP sur les liens du protocole extensible d'échange de blocs (BEEP, *Blocks Exchange Extensible Protocol*) pour NETCONF.

Table des matières

1. Introduction

Étant donnée l'utilisation des caractéristiques du langage de balisage extensible (XML, *Extensible Markup Language*) [XML] et de l'appel de procédure à distance, il est naturel de considérer un lien des opérations de NETCONF [RFC4741] à un protocole d'application SOAP [SOAP-MF]. Le présent document propose un lien de cette forme.

En général, SOAP est un schéma naturel d'échange de messages pour NETCONF, essentiellement à cause du caractère d'appel de procédure à distance des deux. Cependant, il faut faire attention avec SOAP sur HTTP car il est par nature synchrone et piloté par le client. SOAP sur BEEP [RFC3080] est techniquement supérieur, mais n'est pas aussi largement adopté.

Quatre sujets de base sont présentés : les spécificités de SOAP intéressantes pour NETCONF, les spécificités de la mise en œuvre de NETCONF comme service de la Toile fondé sur SOAP, les considérations de sécurité, et les définitions fonctionnelles du langage de description de services de la Toile (WSDL, *Web Services Description Language*). Dans un certain sens, la partie la plus importante de ce document est le bref document WSDL présenté au paragraphe 3.7. Avec les bons outils, le WSDL combiné aux schémas XML de base de NETCONF XML fournit des descriptions lisibles par la machine, suffisantes pour le développement d'applications logicielles qui utilisent NETCONF.

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. Fondements de SOAP pour NETCONF

Pourquoi introduire SOAP comme encore une autre enveloppe autour de ce qui est déjà un message d'appel de procédure à distance ? Il y a en fait des raisons techniques et pratiques. Les raisons techniques sont peut-être moins impérieuses, mais commençons par elles.

L'utilisation de SOAP offre peu d'avantages techniques. SOAP est fondamentalement un schéma de messagerie XML (qui est capable de prendre en charge l'appel de procédure à distance) et il définit un format simple de message composé d'un "en-tête" et d'un "corps" contenus dans une "enveloppe". L'en-tête contient des méta-informations relatives au message et peut être utilisé pour indiquer des choses comme un comportement de remise différée ou des caractéristiques de transaction. De plus, SOAP spécifie un codage facultatif pour le "corps" du message. Cependant, ce codage n'est pas applicable à NETCONF car un des buts est d'avoir un XML très lisible, et le codage SOAP est plutôt optimisé pour faciliter une dé-sérialisation automatisée. Ces avantages de la structure de message SOAP sont simples, mais valent la peine parce qu'ils sont déjà normalisés.

Il y a des raisons pratiques qui font vraiment de SOAP un choix intéressant pour la gestion d'appareil. Il n'est pas difficile d'inventer un mécanisme pour échanger des messages XML sur TCP, mais ce qui est difficile est d'obtenir que ce mécanisme soit pris en charge par une grande variété d'outils et de systèmes d'exploitation et que ce mécanisme soit compris par la plupart des développeurs. SOAP sur HTTP (avec WSDL) connaît un bon succès pour cela, et cela signifie qu'un protocole de gestion d'appareils qui fait usage de ces technologies a l'avantage d'être mis en œuvre et adopté. On admet qu'il y a des problèmes d'interopérabilité avec SOAP et WSDL, mais ces problèmes font l'objet d'une grande attention et on peut s'attendre à ce qu'ils soient résolus.

2.1 Utilisation et mémorisation de WSDL et XSD

Un des avantages de l'utilisation de formats lisibles par la machine (comme le langage de description de services de la Toile (WSDL, *Web Services Description Language*) [WSDL] et les schémas XML [XML-Struct]) est qu'ils peuvent être utilisés automatiquement dans le processus de développement du logiciel. Avec les outils appropriés, WSDL et XSD peuvent être utilisés pour générer des classes qui agissent comme des interfaces distantes ou des structures de données spécifiques d'application. D'autres utilisations, comme de génération de document et localisation de services, sont aussi courantes. Une grande innovation qui se trouve dans de nombreux langages de définition fondés sur XML est l'utilisation d'hyperliens pour se référer aux documents qui contiennent les définitions de support.

```
<import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
      location="http://www.iana.org/assignments/xml-registre/schema/netconf.xsd" />
```

Par exemple, en WSDL, la déclaration d'importation ci-dessus importe les définitions des types et éléments XML depuis le schéma de base NETCONF. Idéalement, le fichier contenant ce schéma est hébergé sur un serveur de la Toile sous l'autorité de l'organisme de normalisation qui a défini le schéma. De cette façon, des normes dérivées peuvent être construites à temps, et toutes sont accessibles aux outils logiciels automatisés qui assurer la conformité aux normes. Le registre de l'IANA à cette fin est décrit dans "Registre XML de l'IETF" [RFC3688].

Noter que les déclarations WSDL pour les liens SOAP sur BEEP ne sont pas encore normalisées.

2.2 SOAP sur HTTP

Bien que SOAP se concentre sur les messages et puisse être lié à différents protocoles sous-jacents comme HTTP, SMTP, ou BEEP, la plupart des mises en œuvre SOAP existantes prennent en charge seulement HTTP ou HTTP/TLS.

Il y a un certain nombre d'avantages à considérer SOAP sur des protocoles autres que HTTP, car HTTP alloue des rôles de client et serveur très distincts à l'initiation de la connexion. Cela cause des difficultés pour prendre en charge des notifications asynchrones et peut être arrangé souvent en remplaçant HTTP par BEEP.

2.3 Inconvénients de HTTP

HTTP n'est pas le transport idéal pour la messagerie, mais il est adéquat pour l'interprétation la plus basique d'un "appel de procédure à distance". HTTP se fonde sur un schéma de communication dans lequel le client (qui initie la connexion TCP) fait une "demande" au serveur. Le serveur retourne une "réponse", et ce processus se poursuit (éventuellement sur une connexion persistante, comme décrit ci-dessous). Ceci correspond à l'idée de base d'un appel de procédure à distance où l'appelant invoque une procédure sur un serveur distant et attend la valeur retournée.

Les critiques potentielles sur HTTP pourrait inclure :
- o le flux de données initié par le serveur est difficile à fournir ;
- o les en-têtes sont diffus et fondés sur le texte ;
- o les connexions inactives peuvent être closes par des mandataires intermédiaires ;
- o l'encapsulation des données doit respecter les extensions multi objets de messagerie Internet (MIME) [RFC2045] ;
- o le transfert en vrac s'appuie sur un ordre fondé sur le flux.

De nombreuses façons, ces critiques sont dirigées sur des compromis particuliers de la conception de HTTP. À ce titre, il est important de les prendre en compte, mais il n'est pas clair qu'elles résultent en des inconvénients fatals pour un protocole de gestion d'appareil.

2.4. BCP 56 : de l'utilisation de HTTP comme substrat

Les bonnes pratiques courantes 56 [RFC3205] présentent un certain nombre de considérations importantes sur l'utilisation de HTTP dans les protocoles d'application. En particulier, il soulève les problèmes suivants :
- o HTTP peut être plus complexe que nécessaire pour l'application.
- o L'utilisation de HTTP peut masquer l'application à certains pare-feu.
- o Un service substantiellement nouveau ne devrait pas réutiliser l'accès 80 alloué à HTTP.
- o La mise en mémoire tampon de HTTP peut masquer l'état de la connexion.

Fondamentalement, ces problèmes reposent directement sur l'usage courant de SOAP sur HTTP, plutôt que sur l'application de SOAP sur HTTP à NETCONF. Comme l'indique le BCP 56, on peut contester que HTTP soit un protocole approprié pour SOAP, et il est probable que BEEP serait un protocole supérieur pour la plupart des applications SOAP. Malheureusement, SOAP sur HTTP est d'usage courant et doit être pris en charge si les avantages pratiques de SOAP sont réalisés. Noter que la nature verbeuse de SOAP le rend en fait plus directement traité par les pare-feu, bien qu'ils soient conçus pour traiter les messages SOAP.

Les antémémoires HTTP NE DEVRAIT PAS être insérées entre des gestionnaires et des agents NETCONF car l'état de session NETCONF est lié à l'état de la connexion de transport sous-jacente. Trois actions défensives peuvent être prises :
- o La mise en mémoire tampon DOIT être prohibée par l'utilisation des en-têtes HTTP Cache-Control et Pragma: no-cache.
- o Des mandataires HTTP NE DEVRAIENT PAS être déployés dans le réseau de gestion.
- o L'utilisation de HTTPS.

Il est aussi possible de répondre au problème de la réutilisation de l'accès 80. Tout service SOAP NETCONF DOIT toujours être pris en charge sur le nouvel accès standard pour NETCONF sur SOAP, et toutes les mises en œuvre conformes DOIVENT par défaut tenter les connexions sur ce nouvel accès standard pour NETCONF. Un accès standard pour NETCONF sur SOAP (sur HTTP) a été alloué dans les considérations relatives à l'IANA du présent document.

2.5 Caractéristiques importantes de HTTP 1.1

HTTP 1.1 [RFC2616] comporte deux importantes caractéristiques qui assurent un transport relativement efficace des messages SOAP. Ces caractéristiques sont les "connexions persistantes" et "le codage de transfert de tronçons".

Les connexions persistantes permettent à une seule connexion TCP d'être utilisée sur plusieurs demandes HTTP. Cela permet que plusieurs paires de messages de demande/réponse SOAP soient échangées sans les frais généraux de création d'une nouvelle connexion TCP pour chaque demande. Étant donné qu'un seul flux est utilisé pour les demandes et les réponses, il est clair qu'une forme de tramage est nécessaire. Pour les messages dont la longueur est connue à l'avance, ceci est traité par l'en-tête HTTP "Longueur de contenu". Pour les messages de longueur dynamique, le "tronquage" est requis.

Le "tronquage" HTTP ou "codage de transfert tronqué" permet à l'envoyeur d'envoyer une quantité indéfinie de données binaires. Ceci est réalisé en informant le receveur de la taille de chaque "tronçon" (sous chaîne des données) avant la transmission du tronçon. Le dernier tronçon est indiqué par un tronçon de longueur zéro. Le tronçonnage peut être efficacement utilisé pour transférer un grand document XML où le document est généré en ligne à partir d'une forme non XML en mémoire.

En termes d'application aux échanges de messages SOAP, les connexions persistantes sont clairement importantes pour des raisons de performances et sont particulièrement importantes quand la persistance de connexions authentifiées est en cause. Quand on considère que les messages de longueur dynamique sont la règle plutôt que l'exception pour les messages SOAP, il est aussi clair que le tronçonnage est très utile. Dans certains cas, il est possible de mettre en mémoire tampon une réponse SOAP et de déterminer sa longueur avant l'envoi, mais les exigences de mémorisation pour cela sont prohibitives pour de nombreux appareils. Ensemble, ces deux caractéristiques fournissent de bonnes fondations pour la gestion d'appareils qui utilise SOAP sur HTTP. Le tronçonnage HTTP et les connexions persistantes [RFC2616] DEVRAIENT être utilisés.

2.6 SOAP sur BEEP

Bien que non largement adopté par la communauté des services de la Toile, BEEP est un excellent substrat pour SOAP [RFC4227]. En particulier, il assure les échanges de messages demande/réponse initiés par l'un ou l'autre des homologues BEEP et permet un nombre arbitraire de messages de réponse (y compris zéro). Le profil BEEP pour SOAP utilise simplement un seul canal BEEP pour échanger les messages SOAP et tire parti de la force inhérente de BEEP pour l'échange de message sur une seule connexion de transport.

2.7 Considérations de mise en œuvre de SOAP

Le but du présent document n'est pas de couvrir en détails la spécification de SOAP [SOAP-MF].On donne plutôt quelques commentaires qui peuvent intéresser une mise en œuvre de NETCONF sur SOAP.

2.7.1 Exploitation de caractéristique de SOAP

NETCONF sur SOAP ne fait pas une utilisation extensive des caractéristiques de SOAP. Par exemple, les opérations NETCONF ne sont pas coupées en parties de message SOAP, et l'en-tête SOAP n'est pas utilisé pour convoyer les méta données <rpc>. C'est une décision de conception délibérée car cela permet à la mise en œuvre de fournir facilement NETCONF sur plusieurs substrats tout en traitant les messages sur ces différents substrats d'une manière commune.

2.7.2 En-têtes SOAP

Les mises en œuvre de NETCONF sur SOAP devrait être conscientes des caractéristiques suivantes des en-têtes SOAP : un en-tête SOAP peut avoir l'attribut "mustUnderstand" *(doit comprendre)*, et, si il l'a, le receveur doit traiter le bloc d'en-tête ou ne pas traiter du tout le message SOAP, et alors générer une faute. Un en-tête "mustUnderstand" ne doit pas être éliminé en silence.

En général, cependant, les en-têtes SOAP sont destinés à des usages spécifiques de l'application. Le lien NETCONF SOAP n'utilise pas les en-têtes SOAP.

2.7.3 Fautes SOAP

Une faute SOAP est retournée dans le cas d'une <rpc-error> NETCONF. Elle est construite essentiellement comme une enveloppe pour la <rpc-error>, mais elle permet au processeur SOAP de propager la <rpc-error> au code d'application en utilisant un mécanisme d'exception approprié au langage.

Une faute SOAP est construite à partir d'une <rpc-error> comme suit : la valeur de code de faute SOAP est "Receiver" dans l'espace de noms d'enveloppe SOAP, le texte de raison de la faute SOAP est le contenu de l'étiquette d'erreur "error-tag" de la <rpc-error> NETCONF, et le détail de la faute SOAP est la structure <rpc-error> originale.

Par exemple, avec la <rpc-error> suivante,

```
rpc-error xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <error-type>rpc</error-type>
  <error-tag>MISSING_ATTRIBUTE</error-tag>
  <error-severity>error</error-severity>
  <error-info>
    <bad-attribute>message-id</bad-attribute>
    <bad-element>rpc</bad-element>
  </error-info>
</rpc-error>
```

Le message de faute SOAP associé est :

```
<soapenv:Envelope
   xmlns:soapenv=
     "http://www.w3.org/2003/05/soap-envelope"
   xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <soapenv:Body>
    <soapenv:Fault>
      <soapenv:Code>
        <soapenv:Value>env:Receiver</soapenv:Value>
      </soapenv:Code>
      <soapenv:Reason>
        <soapenv:Text
           xml:lang="en">MISSING_ATTRIBUTE</soapenv:Text>
      </soapenv:Reason>
      <detail>
        <rpc-error xmlns=
           "urn:ietf:params:xml:ns:netconf:base:1.0">
          <error-type>rpc</error-type>
          <error-tag>MISSING_ATTRIBUTE</error-tag>
          <error-severity>error</error-severity>
          <error-info>
            <bad-attribute>message-id</bad-attribute>
            <bad-element>rpc</bad-element>
          </error-info>
        </rpc-error>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

3. Service SOAP pour NETCONF

3.1 Cas d'usage fondamental

Le cas d'utilisation fondamental pour NETCONF sur SOAP est une console de gestion (rôle de "gestionnaire") qui gère un ou plusieurs appareils qui font fonctionner des agents NETCONF (rôle "d'agent"). Le gestionnaire initie une conexion HTTP ou BEEP avec un agent et pilote la session NETCONF via une séquence de messages SOAP. Quand le gestionnaire clôt la connexion, la session NETCONF est aussi close.

3.2 Établissement de session NETCONF

Une session NETCONF sur SOAP est établie par l'échange initial de messages sur le substrat sous-jacent. Pour HTTP, une session NETCONF est établie une fois qu'un message SOAP est adressé à l'URI d'application de la Toile de NETCONF. Pour BEEP, une session NETCONF est établie quand le profil BEEP pour la prise de contact SOAP a établi le canal SOAP.

3.3. Échange de capacités NETCONF

L'échange de capacités et l'établissement d'identifiant de session sont effectués par l'échange des messages <hello>. Dans le cas de SOAP sur HTTP, le client HTTP DOIT envoyer le premier message <hello>. Le cas de SOAP sur BEEP n'impose pas de contrainte d'ordre. Par exemple, on montre ci-dessous l'échange des messages <hello> et l'établissement d'une valeur d'identifiant de session de 4. On observe que le client de gestion initie l'échange et que l'agent de serveur alloue l'identifiant de session.

```
C : POST /netconf HTTP/1.1
C : Host: netconfdevice
C : Content-Type: text/xml; charset=utf-8
C : Accept: application/soap+xml, text/*
C : Cache-Control: no-cache
C : Pragma: no-cache
C : Content-Length: 376
C :
C : <?xml version="1.0" encoding="UTF-8"?>
C : <soapenv:Envelope
C :   xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
C :   <soapenv:Body>
C :     <hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
C :       <capabilities>
C :         <capability>
C :           urn:ietf:params:netconf:base:1.0
C :         </capability>
C :       </capabilities>
C :     </hello>
C :   </soapenv:Body>
C : </soapenv:Envelope>
S : HTTP/1.1 200 OK
S : Content-Type: application/soap+xml; charset=utf-8
S : Content-Length: 600
S :
S : <?xml version="1.0" encoding="UTF-8"?>
S : <soapenv:Envelope
S :   xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
S :   <soapenv:Body>
S :     <hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
S :       <capabilities>
S :         <capability>
S :           urn:ietf:params:netconf:base:1.0
S :         </capability>
S :         <capability>
S :           urn:ietf:params:netconf:capability:startup:1.0
```

```
S :       </capability>
S :       <capability>
S :         http:/exemple.net/router/2.3/macaractéristique
S :       </capability>
S :     </capabilities>
S :     <session-id>4</session-id>
S :   </hello>
S :  </soapenv:Body>
S : </soapenv:Envelope>
```

3.4 Utilisation de la session NETCONF

Les sessions NETCONF sont persistantes pour des raisons de performances et de sémantique. L'état de session NETCONF contient ce qui suit :

1. Informations d'authentification
2. Informations de capacité
3. Verrous
4. Opérations en cours
5. Numéros de séquence d'opération

L'authentification doit être conservée tout au long d'une session parce qu'elle est coûteuse à établir. Les informations de capacité sont conservées afin que les opérations appropriées puissent être appliquées durant une session. Les verrous sont libérés à la fin d'une session car cela rend le protocole plus robuste. Les opérations en cours vont et viennent à l'existence durant le cours normal des opérations d'appel de procédure à distance (RPC). Les numéros de séquence d'opération fournissent les petites mais nécessaires informations d'état pour se référer aux opérations durant la session.

Dans le cas de SOAP sur HTTP, une session NETCONF est prise en charge par une connexion HTTP avec un utilisateur authentifié. Pour SOAP sur BEEP, une session NETCONF est prise en charge par un canal BEEP fonctionnant en accord avec le profil BEEP pour SOAP [RFC4227].

3.5 Suppression de session NETCONF

Pour permettre un nettoyage automatique, la suppression de session NETCONF sur SOAP a lieu quand la connexion (dans le cas de HTTP) ou le canal (dans le cas de BEEP) sous-jacent est clos. Noter que la cause de base de cette supression peut être la clôture de la connexion TCP sous HTTP ou BEEP selon le cas. Les gestionnaires et agents NETCONF doivent être capable de clore programmatiquement les connexions de transport associées aux sessions NETCONF, comme en réponse à une opération <close-session> ; donc, la mise en œuvre de substrat HTTP ou BEEP doit exposer cela de façon appropriée.

3.6 Exemple de NETCONF sur SOAP

Comme le WSDL proposé (au paragraphe 3.7) utilise le codage document/literal, l'utilisation d'un en-tête et corps SOAP a peu d'impact sur la représentation d'une opération NETCONF. Cet exemple montre HTTP/1.1 pour rester simple. Un exemple pour BEEP serait similaire.

```
C : POST /netconf HTTP/1.1
C : Host: netconfdevice
C : Content-Type: text/xml; charset=utf-8
C : Accept: application/soap+xml, text/*
C : Cache-Control: no-cache
C : Pragma: no-cache
C : Content-Length: 465
C :
C : <?xml version="1.0" encoding="UTF-8"?>
C : <soapenv:Envelope
C :   xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
C :   <soapenv:Body>
C :     <rpc message-id="101"
C :         xmlns="xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
C :       <get-config>
```

```
C :       <filter type="subtree">
C :         <top xmlns="http://exemple.com/schema/1.2/config">
C :           <users/>
C :         </top>
C :       </filter>
C :     </get-config>
C :   </rpc>
C :  </soapenv:Body>
C : </soapenv:Envelope>
```

La réponse HTTP/1.1 est aussi directe :

```
S : HTTP/1.1 200 OK
S : Content-Type: application/soap+xml; charset=utf-8
S : Content-Length: 917
S :
S : <?xml version="1.0" encoding="UTF-8"?>
S : <soapenv:Envelope
S :   xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
S :   <soapenv:Body>
S :     <rpc-reply message-id="101"
S :              xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
S :       <data>
S :         <top xmlns="http://exemple.com/schema/1.2/config">
S :           <users>
S :             <user>
S :               <name>root</name>
S :               <type>superuser</type>
S :               <full-name>Charlie Root</full-name>
S :                 <dept>1</dept>
S :                 <id>1</id>
S :               </company-info>
S :             </user>
S :             <user>
S :               <name>fred</name>
S :               <type>admin</type>
S :               <full-name>Fred Flintstone</full-name>
S :                 <dept>2</dept>
S :                 <id>2</id>
S :               </company-info>
S :             </user>
S :           </users>
S :         </top>
S :       </data>
S :     </rpc-reply>
S :   </soapenv:Body>
S : </soapenv:Envelope>
```

3.7 WSDL SOAP NETCONF

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="urn:ietf:params:xml:ns:netconf:soap:1.0"
xmlns:netb="urn:ietf:params:xml:ns:netconf:base:1.0"
targetNamespace="urn:ietf:params:xml:ns:netconf:soap:1.0"
name="netconf-soap_1.0.wsdl">

    <import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
```

```
        location="http://www.iana.org/assignments/xml-registre/schema/netconf.xsd" />

  <message name="helloRequest">
    <part name="in" element="netb:hello"/>
  </message>
  <message name="helloResponse">
    <part name="out" element="netb:hello"/>
  </message>

  <message name="rpcRequest">
    <part name="in" element="netb:rpc"/>
  </message>
  <message name="rpcResponse">
    <part name="out" element="netb:rpc-reply"/>
  </message>

  <portType name="netconfPortType">
    <operation name="rpc">
      <input message="tns:rpcRequest"/>
      <output message="tns:rpcResponse"/>
    </operation>
    <operation name="hello">
      <input message="tns:helloRequest"/>
      <output message="tns:helloResponse"/>
    </operation>
  </portType>

  <binding name="netconfBinding" type="tns:netconfPortType">
    <SOAP:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="hello">
      <SOAP:operation/>
      <input>
        <SOAP:body use="literal"
            namespace="urn:ietf:params:xml:ns:netconf:soap:1.0"/>
      </input>
      <output>
        <SOAP:body use="literal"
            namespace="urn:ietf:params:xml:ns:netconf:soap:1.0"/>
      </output>
    </operation>
    <operation name="rpc">
      <SOAP:operation/>
      <input>
        <SOAP:body use="literal"
            namespace="urn:ietf:params:xml:ns:netconf:base:1.0"/>
      </input>
      <output>
        <SOAP:body use="literal"
            namespace="urn:ietf:params:xml:ns:netconf:base:1.0"/>
      </output>
    </operation>
  </binding>

</definitions>
```

3.8 Exemple de WSDL de définition de service

Le document WSDL suivant suppose une situation locale pour les définitions de WSDL NETCONF sur SOAP. Un déploiement normal d'un appareil gérable via NETCONF sur SOAP fournirait une définition de service similaire à celle qui

suit pour identifier l'adresse de l'appareil.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:nets="urn:ietf:params:xml:ns:netconf:soap:1.0"
  targetNamespace="urn:myNetconfService"
  name="myNetconfService.wsdl">

  <import namespace="urn:ietf:params:xml:ns:netconf:soap:1.0"
        location="http://localhost:8080/netconf/schema/netconf-soap_1.0.wsdl"/>

  <nom de service="netconf">
    <port name="netconfPort" binding="nets:netconfBinding">
      <SOAP:address location="http://localhost:8080/netconf"/>
    </port>
  </service>

</definitions>
```

4. Considérations sur la sécurité

NETCONF est utilisé pour accéder et modifier les informations de configuration, de sorte que la capacité d'accéder à ce protocole devrait être limitée aux utilisateurs et systèmes qui sont autorisés à voir ou modifier les données de configuration de l'agent.

Parce que les informations de configuration sont envoyées dans les deux directions, il n'est pas sufficient que juste le client ou l'utilisateur soit authentifié auprès du serveur. L'identité du serveur devrait aussi être authentifiée auprès du client.

Les données de configuration peuvent inclure des informations sensibles, comme les noms d'utilisateur ou les clés de sécurité. Donc, NETCONF devrait seulement être utilisé sur des canaux de communications qui fournissent un chiffrement fort pour la confidentialité des données.

Si le serveur NETCONF fournit un accès distant par des protocoles non sûrs, comme HTTP, on devrait faire attention à empêcher l'exécution du programme NETCONF quand une authentification forte de l'utilisateur ou la confidentialité des données n'est pas disponuble.

L'accès alloué par l'IANA DEVRAIT être utilisé, car cela donne le moyen d'un filtrage efficace par un pare-feu durant une possible attaque de déni de service.

4.1 Intégrité, confidentialité, et authentification

Le lien NETCONF SOAP s'appuie sur un transport sûr sous-jacent pour l'intégrité et la confidentialité. De tels transports sont supposés inclure TLS [RFC4346] (qui, lorsque combiné à HTTP, est appelé HTTPS) et IPsec. Il y a un certain nombre d'options pour l'authentification (dont certaines sont spécifiques du déploiement) :

- o au sein du transport (comme avec les certificats de client TLS)
- o au sein de HTTP (comme l'authentification d'accès par résumé [RFC2617])
- o au sein de SOAP (comme avec une signature numérique dans l'en-tête [SOAP-Sec])

L'authentification au niveau HTTP, BEEP, et SOAP peut être intégrée au service d'authentification à distance de l'utilisateur appelant (RADIUS) [RFC2865] pour prendre en charge les bases de données d'authentification à distance.

Au miniumum, toutes les mises en œuvre conformes de NETCONF sur SOAP DOIVENT prendre en charge TLS. Précisément, NETCONF sur SOAP sur HTTP DOIT prendre en charge NETCONF sur SOAP sur HTTPS, et NETCONF sur SOAP sur BEEP DOIT prendre en charge NETCONF sur SOAP sur BEEP sur TLS.

4.2 Vulnérabilités

Les protocoles ci-dessus peuvent avoir diverses vulnérabilités, et elles peuvent être héritées par NETCONF sur SOAP.

NETCONF lui-même peut avoir des vulnérabilités parce qu'un modèle d'autorisation n'est pas spécifié actuellement.

Il est important que les capacités et autorisations d'appareil restent constantes pour la durée de toute session NETCONF en cours. Dans le cas de NETCONF, il est important de considérer que la gestion d'appareil peut avoir lieu sur de multiples substrats (en plus de SOAP) et il est important que les différents substrats aient un modèle d'authentification commun.

4.3 Spécificités environnementales

Certains déploiements de NETCONF sur SOAP peuvent choisir d'utiliser des transports sans chiffrement. Cela présente des vulnérabilités mais peut être choisi pour des déploiements qui impliquent des réseaux clos ou des scénarios de débogage.

Un appareil géré par NETCONF peut interagir (sur des protocoles en dehors de NETCONF) avec des appareils gérés par d'autres protocoles, tous de sécurité différente. Chaque point d'entrée amène avec lui une vulnérabilité potentielle.

5. Considérations relatives à l'IANA

L'IANA a alloué l'accès TCP (833) pour NETCONF sur SOAP sur BEEP, et l'accès TCP (832) pour NETCONF sur SOAP sur HTTPS.

L'IANA permettra l'allocation d'un espace de noms XML au sein de l'espace de noms NETCONF "urn:ietf:params:xml:ns:netconf" pour les définitions WSDL de NETCONF sur SOAP. Suivant les politiques mentionnées dans la [RFC2434], les valeurs allouées dans cet espace de noms subordonné le sont selon la politique de "spécification exigée".

URI : urn:ietf:params:xml:ns:netconf:soap

6. Références

6.1 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)

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Rendue obsolète par la* RFC5226)

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par* 2817, 6585)

[RFC2617] J. Franks et autres, "Authentification HTTP : Authentification d'accès de base et par résumé", juin 1999. *(DS.)*

[RFC2865] C. Rigney et autres, "Service d'authentification à distance de l'utilisateur appelant (RADIUS)", juin 2000. (*MàJ par* RFC2868, RFC3575, RFC5080, RFC8044) *(D.S.)*

[RFC3080] M. Rose, "Cœur du protocole extensible d'échange de blocs (BEEP)", mars 2001. *(P.S.)*

[RFC3205] K. Moore, "Sur l'utilisation de HTTP comme sous strate", février 2002. (BCP0056)

[RFC3688] M. Mealling, "Registre XML de l'IETF", BCP 81, janvier 2004.

[RFC4227] E. O'Tuathail, M. Rose, "Utilisation du protocole simple d'accès aux objets (SOAP) dans le protocole extensible d'échange de blocs (BEEP)", janvier 2006. (*Remplace* RFC3288) *(P.S.)*

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006.

(Remplace RFC2246 *; Remplacée par* RFC5246 *; MàJ par* 5746, 6176, 7465, 7507, 7919)

[RFC4741] R. Enns, éd., "Protocole de configuration NETCONF", décembre 2006. *(P.S.)*

[SOAP-MF] Gudgin, M., Hadley, M., Moreau, JJ., and H. Nielsen, "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation REC-soap12-part1-20030624, juin 2002, <http://www.w3.org/TR/soap12-part1/ >.

[XML] Bray, T., Paoli, J., Sperberg-McQueen, C., and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C REC REC-xml-20001006, octobre 2000, <http://www.w3.org/TR/2000/REC-xml-20001006 >.

[XML-Struct] Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures", W3C REC-xmlschema-1-20010502, mai 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>.

6.2 Références pour information

[RFC2045] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. *(D. S., MàJ par* 2184, 2231, 5335.)

[SOAP-Sec] Brown, A., Fox, B., Hada, S., LaMacchia, B., and H. Maruyama, "SOAP Security Extensions: Digital Signature", Note W3C NOTE-SOAP-dsig-20010206, février 2001, <http://www.w3.org/TR/SOAP-dsig/ >.

[WSDL] Christensen, E., Curbera, F., Meredith, G., and S. Weerawarana, "Web Services Description Language (WSDL) 1.1", Note W3C NOTE-wsdl-20010315, mars 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315 >.

Adresse de l'auteur

Ted Goddard
ICEsoft Technologies Inc.
Suite 300, 1717 10th St. NW
Calgary, AB T2M 4S2
Canada

téléphone : (403) 663-3322
mél : ted.goddard@icesoft.com
URI : http://www.icesoft.com

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est fourni par l'activité de soutien administratif (IASA) de l'IETF.