Utilisation des Services Web dans la négociation du niveau de service

Mohamed Aymen Chalouf, Nader Mbarek, Francine Krief

Utilisation des Services Web dans la négociation du niveau de service

Mohamed Aymen CHALOUF, Nader MBAREK, Francine KRIEF

***Résumé*—Dans le monde IP, avec l'émergence de nouveaux services, la garantie d'une certaine qualité de service (QoS) de bout en bout est devenue un enjeu primordial. Pour ce faire, les responsables des domaines impliqués dans une offre de QoS doivent se mettre d'accord sur le niveau de service qu'ils peuvent garantir. C'est dans le cadre de cette négociation que s'inscrit le protocole SLNP (Service Level Negotiation Protocol). Le protocole SLNP se propose de fonctionner dans un environnement de technologies de Services Web afin de tirer profit de leur interopérabilité. Dans ce papier, nous décrivons les caractéristiques des Services Web et nous spécifions comment le protocole SLNP peut utiliser ces technologies dans la négociation du niveau de service.**

***Mots-clés*— Négociation, Niveau de service, Services Web, SLNP.**

I. INTRODUCTION

L'EVOLUTION rapide des réseaux s'est accompagnée de l'apparition de nouveaux services dont les besoins en terme de QoS (Quality of Service), mobilité et sécurité sont de plus en plus grands. Un niveau de service est ainsi caractérisé par une offre de QoS, mobilité et sécurité. Dans cet article, nous allons traiter la partie relative à la garantie de la QoS.

Quand une offre de QoS fait intervenir plusieurs domaines, les gestionnaires de ces domaines doivent établir un accord qui spécifie le niveau de service qu'ils peuvent garantir de bout en bout. Cet accord peut être établi grâce au protocole de négociation du niveau de service SLNP (Service Level Negotiation Protocol) [1]. Ce dernier a été défini dans le cadre d'une architecture de gestion autonome des réseaux afin de permettre la négociation du niveau de service. Nous proposons de le faire fonctionner dans un environnement interopérable grâce à l'utilisation des technologies de Services Web dans cette négociation.

Cet article est composé en trois sections. La première décrit l'architecture ainsi que les caractéristiques des Services Web. Dans la deuxième section, nous décrivons les paramètres de QoS négociés avec le protocole SLNP ainsi que les différents messages utilisés dans cette négociation. Dans la dernière section, nous spécifions le fonctionnement du protocole SLNP en utilisant les Services Web.

Manuscrit reçu le 10 Décembre 2006. Les auteurs appartiennent au Laboratoire Bordelais de Recherche en Informatique (LaBRI), Université de Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France (e-mail: chalouf@labri.fr; mbarek@labri.fr; krief@labri.fr)

II. LES SERVICES WEB

Les Services Web ont été conçus pour standardiser les échanges sur Internet. En effet, ils permettent à une application de trouver automatiquement sur Internet le service dont elle a besoin et d'échanger des données avec ce dernier. La principale caractéristique des Services Web est l'interopérabilité. Cette interopérabilité permet aux applications écrites dans divers langages de programmation (Java, C++, Visual Basic,…) et tournant sur diverses plateformes (UNIX, Windows,…) d'employer les Services Web pour échanger des données via Internet. Dans cette section, nous allons présenter l'architecture des Services Web ainsi que les standards sur lesquels se basent ces technologies afin de garantir une interopérabilité entre des implémentations réalisées en utilisant différentes plateformes.

A. Architecture

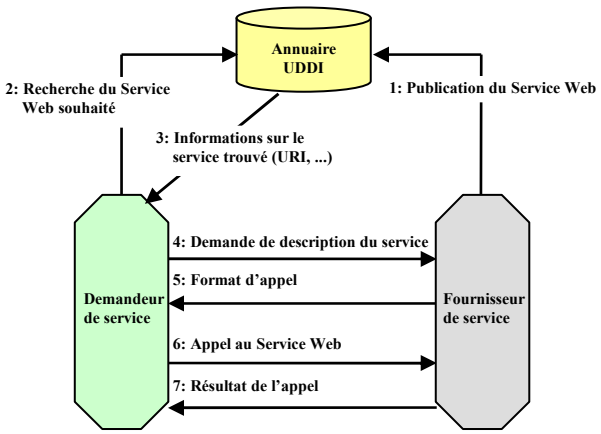

Fig. 1. Architecture des Services Web.

Dans l'architecture des Services Web (fig. 1), le fournisseur crée un service, puis publie l'adresse qui permet d'y accéder (URI : Uniform Resource Identifier) dans un annuaire de Services Web tel que l'annuaire UDDI [2]. Ce dernier rend disponible l'interface du service ainsi que ses informations d'accès, pour n'importe quel client. Ainsi, ce dernier accède à l'annuaire UDDI pour effectuer une recherche afin de trouver le service désiré en précisant ses caractéristiques. Enfin, il se lie au fournisseur du service désiré afin d'acquérir la description et le format d'appel qui vont lui permettre d'invoquer correctement ce Service Web par la suite.

B. Standards

Les Services Web se basent sur un ensemble de standards formant une pile protocolaire (fig. 2) et sont au nombre de quatre : XML (eXtensible Markup Language) [3], SOAP (Simple Object Access Protocol) [4], WSDL (Web Service Description Langage) [5] et UDDI (Universal Description Discovery and Integration) [2]. Dans cette pile, le protocole HTTP (Hyper Text Transfer Protocol) se trouve au niveau de la couche de transfert des messages. Au dessus de cette couche, le protocole SOAP est utilisé dans la structuration des messages à transmettre. Ensuite, le langage WSDL permet de décrire les Services Web. Enfin, un processus de découverte des Services Web tel que UDDI peut figurer dans cette pile. Ces protocoles et standards mis en jeu dans l'architecture des Services Web à savoir SOAP, WSDL et UDDI sont basés sur le langage XML.

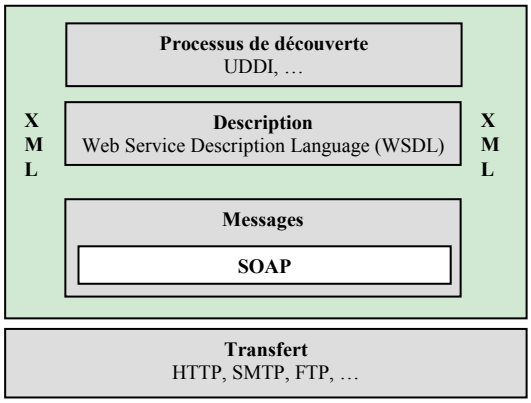

Fig. 2. Pile protocolaire des Services Web [6].

1) eXtensible Markup Language (XML)

Le langage XML permet de définir un langage à travers un vocabulaire et une grammaire associée. Les langages basés sur XML peuvent être utilisés pour décrire toutes sortes de données et de textes tels que les messages SOAP et les descriptions WSDL. Un fichier XML est un document texte qui a une structure hiérarchique en éléments. En effet, un élément peut contenir du texte ou d'autres éléments et toutes les données d'un document XML sont contenues dans un élément racine. Un document XML est dit bien formé, quand il est conforme à un certain nombre de règles. S'il est bien formé et conforme à la grammaire associée, le document XML est alors qualifié de valide. La grammaire d'un langage basé sur XML peut être définie soit en format DTD (Document Type Definition) soit en format XSD (XML Schema Definition) [7]-[8]. Dans la suite, nous utilisons le format XSD afin de définir de façon structurée le type de contenu, la syntaxe et la sémantique des messages SOAP échangés ainsi que le SLS [9] (Service Level Spécification) négocié.

2) Simple Object Access Protocol (SOAP)

Le protocole SOAP définit un ensemble de règles pour structurer des messages qui peuvent être utilisés dans de simples transmissions unidirectionnelles, mais il est aussi particulièrement utile pour exécuter des dialogues requête-réponse.

Un message SOAP est un document XML dont l'élément racine <Envelope> contient un élément optionnel <Header> et un élément obligatoire <Body> qui contient les données échangées. SOAP peut également utiliser l'élément <Fault> afin de signaler des erreurs. Ainsi, nous distinguons trois types de messages SOAP : Call, Response et Fault. Un message de type Call, permet d'invoquer une méthode du Service Web en lui passant les paramètres nécessaires à cet appel. Le résultat de cette invocation est récupéré grâce à un message de type Response. Quand un problème survient lors d'un appel, un message de type Fault peut être utilisé afin de caractériser l'erreur.

SOAP n'est pas lié à un mécanisme sous-jacent de transfert des messages mais il est souvent associé à HTTP. Il n'est pas non plus lié à un système d'exploitation ni à un langage de programmation, donc les clients et serveurs des dialogues SOAP peuvent s'exécuter sur n'importe quelle plateforme et peuvent être écrits dans n'importe quel langage.

3) Web Service Description Language (WSDL)

Le langage WSDL permet de décrire un Service Web et la manière d'y accéder, en incluant des détails tels que le protocole à utiliser, l'URI, les opérations pouvant être effectuées, les formats des messages SOAP entrants ou sortants qui sont requis pour dialoguer avec le service ainsi que les exceptions pouvant être renvoyées.

```
<definitions>
    <types>     !--abstract data types          </types>
    <message>   !--message structure            </message>
    <portType>  !--Web Service Interface        </portType>
    <binding>   !--how the service is accessed  </binding>
    <service>   !--who provides the service     </service>
</definitions>
```

Fig. 3. Structure d'un document WSDL [5].

Une description WSDL d'un Service Web est un document XML (fig. 3) dont l'élément racine <definitions> contient cinq éléments :

<types> : contient les définitions des types de données utilisées dans les messages envoyés et reçus par le Service Web et ceci en utilisant le modèle XML Schema.

<message> : décrit le nom et le type d'un champ à transmettre et qui correspond à un paramètre d'entrée ou de sortie d'une opération du service.

<portType> : définit de manière abstraite le Service Web comme un ensemble d'opérations où chaque opération est composée d'un message d'entrée, d'un message de sortie et d'un éventuel message d'erreur. Suivant les messages qui composent une opération, WSDL permet de définir quatre types d'opérations : Request-Response (reçoit un message et retourne un message), Solicit-Response (émet un message et reçoit un message), One-Way (reçoit un message) et Notification (émet un message).

<binding> : permet de faire une liaison précise entre les descriptions, effectuées dans les éléments <message> et <portType>, et le type de protocole utilisé pour véhiculer les données.

<service> : permet de préciser les informations complémentaires nécessaires pour invoquer le service, en particulier l'URI qui permet d'y accéder.

Ainsi la description WSDL permet de spécifier le protocole à utiliser ainsi que les opérations à invoquer. Ceci n'est pas suffisant pour découvrir un Service Web qui possède une fonctionnalité dont une application a besoin. Pour savoir où trouver les Services Web, nous pouvons utiliser un mécanisme de découverte tel que l'annuaire UDDI.

4) Universal Description Discovery and Integration (UDDI)

C'est un annuaire d'informations sur les Services Web qui permet aux fournisseurs d'enregistrer leurs Services Web et aux applications de rechercher et de localiser sur le réseau les services correspondant à leurs besoins de façon normalisée. Les annuaires UDDI sont accessibles grâce au protocole SOAP et contiennent des informations détaillées, sous un format XML, concernant les services publiés et leurs emplacements. En effet, un annuaire UDDI est similaire à un répertoire téléphonique où chaque entrée se compose de trois parties principales : le fournisseur du service, les Services Web offerts et les liens vers leurs implémentations. La première partie est l'information la plus générale qui décrit le fournisseur du service. Elle inclut des noms, des adresses, des contacts et d'autres détails administratifs. La deuxième partie est la liste des Services Web disponibles. Ces services peuvent être groupés par domaines d'application, par régions géographiques, etc. La dernière partie d'une entrée UDDI est un lien vers une implémentation qui associe l'entrée de Service Web à l'URI exacte identifiant son emplacement. Elle spécifie également le protocole à utiliser pour accéder au service.

C. Plateformes

Les Services Web doivent être déployés sur des serveurs d'applications et doivent implémenter le protocole SOAP. Ainsi, nous trouvons plusieurs plateformes qui permettent de développer et déployer des Services Web telles que : Axis et le serveur Tomcat de Jakarta (deux projets open source d'Apache Software Foundation), la bibliothèque pour les développeurs de Services Web en Java (JWSDP) de Sun Microsystems et les serveurs HTTP IIS de Microsoft (avec le framework .NET).

Les Services Web forment une technologie très attrayante. En effet, le langage XML, sur lequel se basent les standards composant l'architecture des Services Web, fournit à ces derniers une grande extensibilité ainsi que la neutralité par rapport aux différentes plateformes et différents langages de programmation. Ainsi, les Services Web permettent de faire interagir des applications dans un environnement hétérogène en fournissant un moyen d'interopérabilité entre différentes applications. Dans ce qui suit, nous verrons comment le protocole de négociation SLNP peut exploiter l'interopérabilité offerte par ces technologies.

III. LE PROTOCOLE SLNP

Afin de garantir une certaine qualité de service, les différents gestionnaires impliqués dans une offre de QoS doivent établir un contrat de niveau de service SLA (Service Level Agreement). La partie technique de ce contrat appelée SLS (Service Level Specification) est négociée par l'intermédiaire du protocole SLNP.

A. Niveau de service

1) Définition

Un accord sur le niveau de service entre les gestionnaires de réseaux est spécifié dans un SLA qui sera ensuite traduit en objectifs, nommés SLO (Service Level Objectives). Chaque SLO est, à son tour, traduit en un ensemble de paramètres formant un SLS. Le SLO représente les objectifs à réaliser dans le cadre d'un SLA tandis que le SLS représente une interprétation technique du SLO et sert comme guide opérationnel afin d'aider le gestionnaire à implémenter un objectif. Un SLA peut contenir plusieurs SLO pour plusieurs objectifs tels que le SLO pour la mobilité, le SLO pour la sécurité, le SLO pour la qualité de service. De même chaque SLO peut contenir, à son tour, plusieurs SLS [10].

2) Paramètres d'un SLS de QoS

Les paramètres d'un SLS qui peuvent être négociés via SLNP sont une extension du format de SLS défini dans [9] :

- *Identification du trafic* : adresses IP, ports, identification du protocole, etc.
- *Scope* : point d'entrée et point de sortie d'un domaine.
- *Temps de service* : le temps pendant lequel la QoS négociée doit être garanti.
- *Garantie de performance* : délai, gigue, taux de perte, bande passante.
- *Conformité du trafic* : taille des paquets, débit crête, etc.
- *Traitement d'excès* : le traitement que le réseau applique aux paquets hors profil.
- *Mode de négociation* : SLS prédéfini par le gestionnaire ou non.
- *Intervalle de renégociation* : le temps pendant lequel un SLS ne peut être renégocié.
- *Fiabilité* : temps d'inaccessibilité et temps d'établissement suite à une panne.

Tous ces paramètres peuvent ne pas figurer en même temps lors de la spécification d'un SLS négocié avec SLNP. Mais, le schéma XML de cet SLS, que nous proposons (fig. 8), doit être générique et comprendre tous les paramètres que nous avons cité.

B. Architecture de SLNP

1) Terminologie

Nous généralisons la terminologie adoptée par SLNP dans le cadre de la gestion autonome [11] :

- *SE (Service level negotiation Entity)* : une entité réseau qui supporte le protocole SLNP.
- *SI (Service level negotiation Initiator)* : une SE qui initie la procédure de négociation.
- *SR (Service level negotiation Responder)* : une SE qui répond aux messages SLNP.

- *SF (Service level negotiation Forwarder)* : une SE qui se situe entre SI et SR sur le chemin du flux de la négociation.

Une SE peut être considérée comme une SI, SR ou SF suivant le rôle qu'elle joue dans un scénario de négociation (fig. 4).

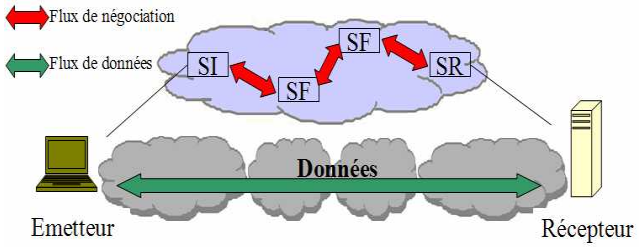

Fig. 4. Entités de négociation avec SLNP.

2) Modèle de fonctionnement

Durant une négociation de SLS entre une SI et une SR, des messages SLNP, qui seront détaillés par la suite, sont échangés et donc véhiculés dans les deux sens (SI SR et SR SI). Ces messages sont généralement générés dans les extrémités de la négociation (SI et SR), mais ils sont traités et modifiés, si nécessaire, dans les SF.

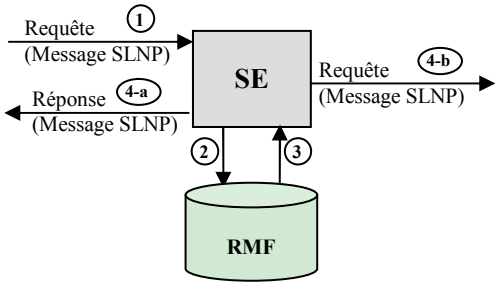

Fig. 5. Fonctionnement au niveau d'une SE.

Le fonctionnement de SLNP nécessite un routage entre les différents domaines qui permet à chaque SE de trouver l'adresse de la SE suivante sur le chemin de la négociation à partir du domaine de destination des flux de données, et ce afin de lui transmettre les messages SLNP. Le traitement des messages de négociation par une SE est influencé par une interaction avec le RMF (Resource Management Function) (fig. 5) qui représente d'une façon simplifiée les fonctionnalités du composant LAM (Low-level Autonomic Manager) de l'architecture de gestion autonome définie dans [11]. En effet, cette interaction permet de prendre une décision qui peut être une réponse (acceptation, rejet ou alternative) ou une transmission de la requête, modifié si nécessaire, à la SE suivante. Cette décision dépend de la disponibilité des ressources ainsi que des autorisations de la demande. Ensuite, si les SE arrivent à un accord alors elles procèdent à l'enregistrement du SLS négocié et le SLS est alors établi. Un SLS déjà établi peut être modifié suite à la demande de la SI. Quand cette demande est acceptée par toutes les SE concernées, le SLS est mis à jour. La résiliation d'un SLS établi ne se fait que suite à la demande de la SI. Cette demande entraîne la suppression du SLS désigné au niveau de toutes les SE concernées.

3) Messages SLNP

Les messages SLNP [11] permettent de satisfaire les besoins de négociation (établissement, modification et résiliation d'un SLS) et sont au nombre de six :

- *Negotiate* : généré par la SI à destination de la SR, ce message permet de spécifier les valeurs et les paramètres du SLS qui vont être négociés. Il est transmis à travers les entités intermédiaires qui peuvent le modifier jusqu'à l'arrivée à la SR.
- *Revision* : émis par la SR en direction de la SI afin de proposer une alternative au SLS demandé.
- *Modify* : généré par la SI à destination de la SR, ce message sert à demander la modification d'un SLS suite à un changement des besoins de la SI ou à la réception d'un message Notify de la part d'une autre SE.
- *Notify* : émis par n'importe quelle SF ainsi que par la SR directement vers la SI pour lui demander d'améliorer ou de dégrader l'ancien niveau de service suite à des problèmes dans le réseau tel que la congestion. Cette dernière peut répondre positivement à la notification reçue et elle peut donc procéder à la modification du SLS.
- *Release* : émis par la SI en direction de la SR, ce message sert à résilier un SLS déjà établi.
- *Response* : peut être émis par la SR en direction de la SI afin de répondre à une requête de type Negotiate, Modify ou Release. Il peut être également émis par la SI en direction de la SR afin de répondre à une requête de type Revision. Dans ces deux cas, ce message passe par toutes les SF afin qu'elles puissent effectuer le traitement nécessaire (enregistrement, mise à jour ou suppression du SLS concerné). Ce message peut aussi être émis par la SI directement vers la SE (SF ou SR) afin de répondre à une requête de type Notify.

4) Scénario de négociation via SLNP

Nous présentons un scénario de négociation en utilisant les messages SLNP afin d'établir un SLS. Notons que les messages SLNP sont généralement plus ou moins modifiés en fonction des résultats des interactions des SE avec leurs RMF. Ces modifications concernent surtout les éléments du SLS qui influencent le niveau de service : délai, gigue, bande passante, taux de perte, etc.

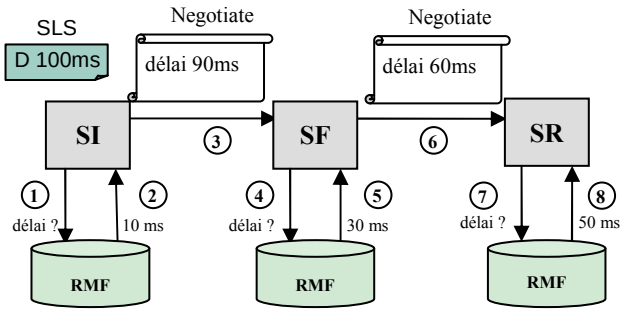

Fig. 6. Scénario de négociation d'un SLS.

Dans le scénario de la fig. 6, nous considérons le paramètre délai. La SI négocie un SLS avec un délai de bout en bout de 100 ms. Après avoir consulté leurs RMF respectifs, la SF et la SR peuvent répondre positivement à la requête de la SI, s'il n'y a pas de problème avec les autres paramètres négociés, et procéder ainsi à l'enregistrement du SLS.

IV. Utilisation des Services Web par SLNP

Afin de fonctionner dans un environnement de technologies des Services Web, le protocole de négociation SLNP se base alors sur des échanges de messages SOAP à base de XML. Ainsi les paramètres de SLS sur lesquels portera la négociation via SLNP formeront une partie de ce document XML. D'abord nous avons établi le schéma XML du SLS négocié. Ensuite, les schémas XML des messages SLNP ont été définis. Enfin, nous avons spécifié la composition d'une entité SLNP qui permet le fonctionnement correct du protocole SLNP.

A. Schémas XML

Les symboles utilisés dans les représentations graphiques des schémas XML, que nous spécifions dans cette section, sont résumés dans le tableau (tab. 1).

Graphique	Schéma XML
sls / SlsType	Elément
[	Tous [séquence]
{	Choix
?	Optionnel
+	Répétable
*	Optionnel et répétable

Tab. 1. Modèle graphique du schéma XML.

1) SLS

Nous proposons un schéma XML d'un SLS générique qui comprend tous les paramètres sur lesquels peut porter la négociation avec SLNP. Ce schéma XML précise la structure d'un élément 'sls', c'est-à-dire tous les éléments et les attributs qu'il peut contenir. En effet, l'élément 'sls' (fig. 8) a une structure hiérarchique puisqu'il est composé d'autres éléments tels que le 'flowId' qui servira à identifier le flux concerné par la négociation, ou encore le 'serviceSchedule' qui permet de préciser le temps de service.

Les paramètres que comprend un SLS sont fortement liés à l'aspect qualité de service mais SLNP reste ouvert et assez générique pour échanger d'autres paramètres tels que les paramètres de sécurité et ce grâce à l'extensibilité des schémas XML. Il suffit donc, d'étendre les attributs du SLS à des aspects liés à la sécurité ou encore à la mobilité, en modifiant son schéma XML, afin de rendre ces attributs compréhensibles par les entités de négociation.

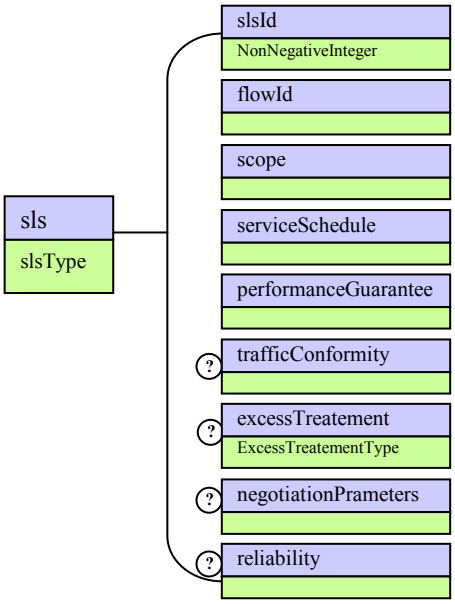

Fig. 8. Structure XML du SLS.

2) Messages SLNP

Les schémas XML des messages (fig. 9 et fig. 10) servent à spécifier les éléments qui les composent et qui assurent le fonctionnement correct du protocole de négociation.

Chaque message contient un élément 'negotiationId' ainsi qu'un élément 'sls'. L'élément 'negotiationId' sert à associer une requête aux réponses relatives à celle-ci et à identifier cet ensemble d'une manière unique alors que le rôle de l'élément 'sls' varie d'un message à l'autre. Par exemple, dans un message de type Negotiate, l'élément 'sls' permet à la SI de spécifier le niveau de service souhaité en précisant les attributs et leurs valeurs.

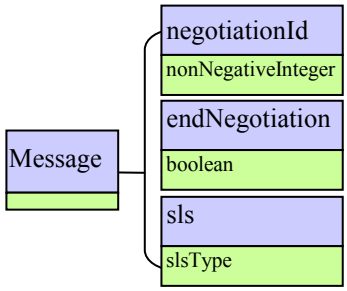

Fig. 9. Structure d'une requête.

Les messages de type Negotiate, Revision, Modify, Notify et Release contiennent un élément de type booléen 'endNegotiation'. Quand il est égal à 'true', cet élément permet d'empêcher le vis-à-vis de proposer une alternative et l'obliger à répondre avec un message Response afin de mettre fin à la négociation. D'où, la grande utilité de ce champ dans

la gestion du temps de la négociation.

Le message de type Response contient un champ 'resp' qui désigne la nature de la réponse (Ack ou Nack) ainsi q'un champ 'reason' qui peut être utilisé dans le cas ou la réponse est négative (Nack) et ce afin d'expliquer la raison pour laquelle on ne peut pas répondre positivement à la requête.

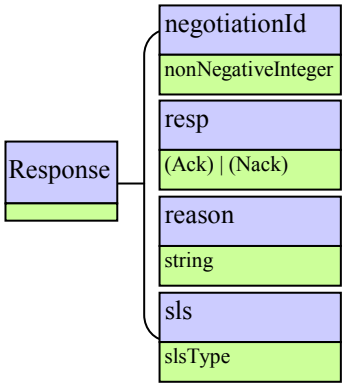

Fig. 10. Structure d'une réponse.

B. *Entité SLNP*

Nous savons qu'une SE peut être une SI et initier une négociation d'un SLS. Elle peut être également une SR ou SF sollicitée par une autre SE dans une autre négociation. Quand une SE initie une négociation, elle a besoin d'une application cliente qui va invoquer le Service Web de la SE adjacente sur le chemin de la négociation. D'autre part, quand elle joue le rôle d'une SF ou quand elle constitue la destination finale d'une négociation, elle a besoin d'un Service Web qui va contenir des opérations qui vont recevoir les messages requêtes, les traiter et retourner les messages réponses appropriés. Ainsi une entité SLNP doit être composée au moins d'un Service Web et d'une application cliente.

SI (initiale)	*SF (intermédiaire)*	*SR (finale)*
--------- / Negotiate	Negotiate / Revision / Response	Negotiate / Revision / Response
--------- / Response	Response / ---------	Response / ---------
--------- / Modify	Modify / Revision / Response	Modify / Revision / Response
Notify / Response	--------- / Notify	--------- / Notify
--------- / Release	Release / Response	Release / Response

Fig. 11. Messages échangés lors d'une négociation.

1) *Service Web de négociation*

Spécifier le Service Web de négociation revient à spécifier les différentes opérations que va contenir ce dernier, les différents messages d'entrée et de sortie de chaque opération, les protocoles utilisés, ainsi que les adresses qui permettent d'y accéder. Toutes ces caractéristiques peuvent figurer dans la description WSDL du Service Web (fig. 12).

- *Les opérations*

La spécification WSDL définit quatre modèles d'interactions entre le nœud appelant et le nœud destinataire de l'appel : one-way, request-response, solicit-response, notification. Parmi ces quatre modèles, seuls les deux premiers sont implémentés. Ainsi, en tenant compte de ce fait et en essayant de résumer tous les scénarios possibles dans un tableau récapitulatif (fig. 11), nous distinguons cinq différentes opérations dont quatre utilisent le mode d'interaction request-response alors que le mode d'interaction de la dernière est de nature one-way :

- *negotiationOperation* : invoquée chez un Service Web d'une SR ou d'une SF lors d'une négociation d'un SLS. Elle est de type request-response dont le message d'entrée est un Negotiate alors que le message de sortie peut être un Revision ou un Response. Le message de sortie de cette opération est créé par l'entité elle-même dans le cas d'une SR. Dans le cas d'une SF, ce message n'est autre que le résultat de l'appel à la negotiationOperation du Service Web de la SF suivante sur le chemin de la négociation.
- *modificationOperation* : invoquée chez un Service Web d'une SE autre que la SI lors de la modification d'un SLS. Elle est de type request-response dont le message d'entrée est un Modify alors que le message de sortie peut être un Revision ou un Response. Ce message est crée par l'entité elle-même dans le cas d'une SR, ou d'une SF quand le SLS à modifier n'existe pas. Il peut être le résultat de l'appel à la modificationOperation du Service Web de l'entité adjacente dans le cas d'une SF et quand le SLS à modifier existe déjà.
- *notificationOperation* : ne peut être invoquée que chez un Service Web d'une SI et elle sert à traiter les notifications qui peuvent venir de la SR ainsi que des SF. Elle est de type request-response dont le message d'entrée est un Notify alors que le message de sortie est un Response.
- *releaseOperation* : peut être invoquée chez un Service Web d'une SE autre que la SI lors de la résiliation d'un SLS. Elle est de type request-response dont le message d'entrée est un Release alors que le message de sortie est un Response. Le message Response peut être crée par l'entité elle-même (SR ou SF) dans le cas où le SLS à résilier n'existe pas, ou encore le résultat de l'appel à la releaseOperation du Service Web de la SE adjacente (cas d'une SF).
- *responseOperation* : peut être invoquée chez un Service Web d'une SR ou d'une SF lors de l'établissement ou la modification d'un SLS. En effet, elle est appelée quand la SI répond à une alternative proposée par la SR via un message Revision. C'est la seule opération qui est de type one-way dont le message d'entrée est un Response. Dans le cas d'une SF, cette opération ne fait que forwarder le message Response reçu.

- *Le portType*

Dans la description WSDL d'un Service Web (fig. 12), le portType sert à définir d'une manière abstraite un ensemble d'opérations ayant quelques caractéristiques en commun

telles que l'utilisation du même protocole. Dans le cas d'un Service Web de négociation, les six opérations sont regroupées dans le même portType (NegotiationPortType) parce qu'elles utilisent le protocole SOAP sur HTTP.

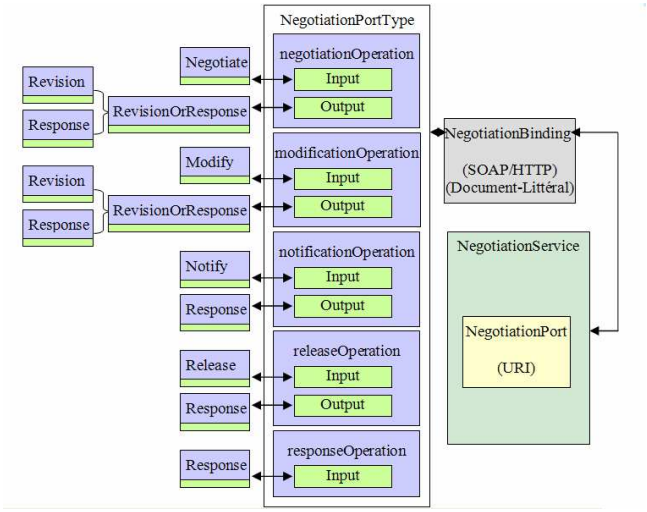

Fig. 12. Schéma de la description WSDL du Service Web de négociation.

- *Le binding*

Dans le binding nous précisons le format des messages véhiculés ainsi que le protocole que nous utilisons. En effet nous utilisons le protocole SOAP sur HTTP, le type de format Document et le style de mise en forme Litteral. Nous rappelons que la spécification SOAP définit trois combinaisons de format et style : RPC-Encoded, RPC-Litteral et Document-Litteral. Lors de la définition du binding, nous avons opté pour le format Document-Litteral, parce que c'est un format qui fournit une grande flexibilité dans les échanges des messages et que les données véhiculées sont déjà au format XML.

- *Le service*

Le service de négociation est constitué d'un seul port qui sert à définir l'URI du point d'accès au Service Web à travers l'association du binding à une adresse réelle. Ce port est le seul point d'accès au service de négociation, et il est caractérisé par le mode de communication SOAP/HTTP spécifié dans le binding.

2) *Application cliente de négociation*

Elle doit être capable de procéder à :

- *L'établissement d'un SLS* : elle peut négocier l'établissement d'un SLS en prenant en arguments les caractéristiques du SLS à négocier telles que l'identification du flux, les paramètres de la garantie de performance etc.
- *La modification d'un SLS* : elle peut également négocier la modification d'un SLS déjà établi au cas où elle voit ses besoins changer ou procéder à une modification lorsque l'entité SI reçoit une notification d'une autre entité impliquée dans l'établissement d'un SLS en cours de validité et y réponds positivement.
- *La résiliation d'un SLS* : quand le service prend fin ou lorsque le demandeur de service n'est plus satisfait du niveau de service offert, l'application cliente peut procéder à la résiliation du SLS.

Notre implémentation du protocole SLNP a été réalisée en utilisant le langage Java. Dans cette implémentation nous avons choisi d'utiliser le serveur d'applications Tomcat d'Apache ainsi que l'implémentation Axis du protocole SOAP. Ce choix ne veut pas dire que toute entité qui implémente le protocole SLNP doit utiliser la même plateforme. Au contraire, nous avons utilisé les Services Web dans l'implémentation de SLNP afin de permettre l'interopérabilité entre des entités hétérogènes susceptibles de participer à la négociation d'un niveau de service via SLNP. Un test de fonctionnement de notre implémentation a été réalisé localement avec succès et nous sommes maintenant entrain d'élaborer d'autres scénarios de test afin d'évaluer différents paramètres de performances du protocole SLNP.

V. CONCLUSION ET PERSPECTIVES

Dans ce papier, nous avons tout d'abord présenté les technologies des Services Web (architecture, fonctionnement et standards). Ensuite, nous avons décrit le fonctionnement du protocole de négociation SLNP ainsi que les différents messages échangés lors d'une procédure de négociation de SLS via ce protocole. Ceci nous a permis de spécifier comment SLNP peut utiliser les Services Web afin d'exploiter l'interopérabilité de ces technologies dans la négociation du niveau de service. En effet, nous avons défini le schéma XML du SLS négocié ainsi que ceux des messages échangés lors d'une négociation du SLS via le protocole SLNP. Nous avons également proposé une description des interfaces des Services Web de négociation grâce au langage WSDL.

La négociation du niveau de service en utilisant les Services Web, que nous avons spécifié, ne porte que sur des paramètres de QoS. En plus de la QoS, le niveau de service fait intervenir des paramètres de sécurité et de mobilité. Ainsi, dans l'avenir, nous allons prendre en compte ces paramètres dans la négociation du niveau de service via SLNP tout en essayant d'étudier l'impact de la sécurité et de la mobilité sur la QoS.

REFERENCES

[1] N. Mbarek et F. Krief, " La Négociation Du Niveau De Service Dans Une Architecture De Gestion Autonome," 7 ème Colloque Francophone de Gestion de REseaux et de Services, pp. 207-222, Bordeaux, France, Mai 2006.

[2] T. Bellwood, L. Clément, D. Ehnebuske, et A. Hately, "Universal Description, Discovery and Integration (UDDI) specification," *Rapport technique*, Comité OASIS, Juillet 2002.

[3] T. Bray, et al., "Extensible Markup Language (XML) 1.0 (Third Edition)," *Recommendation W3C*, Fevrier 2004. Disponible : http://www.w3.org/TR/REC-xml

[4] D. Box, et al., "Simple Object Access Protocol (SOAP) 1.1," *Note W3C*, Mai 2000. Disponible : http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[5] E. Christensen, et al., "Web services Description Language (WSDL) 1.1," *Note W3C*, Mars 2001. Disponible: http://www.w3.org/TR/wsdl

[6] D. Booth, et al., “Web Services Architecture,” *Note W3C*, Fevrier 2004. Disponible: http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[7] H. S. Thompson, et al., “XML Schema Part 1: Structures,” *Recommandation W3C*, Mai 2001. Disponible: http://www.w3.org/TR/xmlschema-1/

[8] P. V. Biron, et al., “XML Schema Part 2: Datatypes,” *Recommandation W3C*, Mai 2001. Disponible: http://www.w3.org/TR/xmlschema-2/

[9] “Attributes of a service level spécification (SLS) template,” Internet Draft, Network Working Group, *draft-tequila-sls-03*, Octobre 2003.

[10] B. Benammar, N. Thi Mai Trang, G. Pujolle et V. Yilmaz, “Définition d’un SLA / SLS,” Livrable IPSig, Décembre 2003.

[11] N. Mbarek and F. Krief, “Service Level Negotiation in Autonomous Systems Management,” The International Conference on Autonomic and Autonomous Systems (ICAS'06), publié par IEEE Computer Society Press, pp. 35-41,Silicon Valley, USA, Juillet 2006.