

Modélisation et transformation d'informations de QoS : une revue de la littérature

K.K. Nguyen, B. Kerhervé, B. Jaumard

Abstract—Quality of service (QoS) architectures have been required in recent years to support a wide range of distributed applications, particularly in wide-area systems. In the context of QoS information management associated with QoS activities, QoS information modeling and mapping enable QoS architectures to be developed independently of the underlying environment and/or application. This paper provides a review of existing QoS architectures on these aspects and presents our current framework.

Keywords – Quality of Service (QoS), information model, distributed systems.

Mot clés – Qualité de Service (QoS), modèle d'informations, systèmes distribués.

I. INTRODUCTION

Au cours de la dernière décennie, l'essor des réseaux à haute vitesse et des ordinateurs à grande puissance a considérablement favorisé le développement et l'utilisation d'applications multimédia réparties comme les Nouvelles-sur-Demande (News-On-Demand : NoD), la Video-sur-Demande (VoD), la Téléphonie IP, ou la vidéo conférence, des applications qui représenteront une partie importante de la future société d'information. Au niveau de leur utilisation, ces applications partagent les caractéristiques suivantes: i) architecture distribuée et hétérogène, ii) accès à distance et iii) exigences des utilisateurs concernant la qualité de service [13].

En général, la gestion de qualité de service (QoS) se compose de cinq activités [1]: la spécification, la transformation, la négociation, l'adaptation et le monitoring. La spécification a pour objectif de permettre à l'utilisateur de définir ses exigences en terme de niveau de qualité de service. La transformation réalise ensuite une traduction

automatique de ces exigences vers des contraintes compréhensibles par le système. La négociation est un processus itératif pour mettre en accord les demandes de l'utilisateur avec les offres disponibles du système. L'adaptation sert alors à maintenir aussi longtemps que possible l'accord de QoS établi dans la phase de négociation. Enfin, le monitoring est utilisé pour déterminer la qualité disponible du système et pour détecter les changements de qualité au cours d'une session de service.

En répondant aux exigences évolutives de la gestion de QoS et des services intégrés, la gestion de QoS doit nécessairement être considérée d'une manière dynamique, autrement dit, on veut pouvoir changer à tout moment la spécification et/ou la déclaration de QoS. Certaines recherches ont été menées dans cette direction avec comme but de proposer une nouvelle approche de QoS plus large que celle définie précédemment, en partant des exigences des utilisateurs et en considérant ainsi différentes catégories d'information de QoS [6]. Cette nouvelle approche, appelée *Quality Driven Delivery* (QDD), permet de considérer d'autres exigences que les exigences de performance traditionnellement supportées dans les approches existantes. De plus, la gestion de QoS traditionnelle porte notamment sur les réseaux, tandis que la QDD étudie le service entier dont le réseau n'est qu'un composant. Un des principaux objectifs de QDD est de permettre aux concepteurs, aux développeurs ainsi qu'aux utilisateurs d'ajouter des exigences différentes sans avoir besoin de refaire le système entier. Elle peut alors se voir comme une généralisation de la gestion de QoS d'un point de vue dynamique et en prenant comme point de départ les exigences exprimées par les utilisateurs.

II. MODELES D'INFORMATION DE QDS

Un des problèmes à résoudre concerne l'évolution du nombre d'informations de QoS ainsi que les relations complexes entre elles, surtout pour des systèmes hétérogènes. La modélisation d'informations de QoS est une approche visant à systématiser ces informations et également,

K.K. Nguyen, is with the Université de Montréal CP 6128 succ. Centre-ville Montréal, QC, Canada, H3C 3J7. (Tel : (1) 514-343-7478; nguyenkk@iro.umontreal.ca)

B. Kerhervé, is with the Université du Québec à Montréal C.P. 8888, succ. Centre-ville, Montréal, QC, Canada - H3C 3P8 (email: Kerherve.Brigitte@uqam.ca)

B. Jaumard is with the Université de Montréal CP 6128 succ. Centre-ville Montréal, QC, Canada, H3C 3J7. (email: jaumard@iro.umontreal.ca)

à introduire des représentations formelles. Un nombre important de modèles d'informations ont été développés dans différents domaines et plus particulièrement pour les télécommunications, la gestion de réseaux, le génie logiciel et les systèmes distribués.

A. Télécommunications

La QoS trouve origine dans le monde des télécommunications en particulier pour répondre aux contraintes d'utilisation des ressources pour la livraison de données continues. Dans ce contexte, les modèles d'informations de QoS se concentrent sur la définition de l'ensemble des caractéristiques de QoS les « plus importantes » du point de vue du fournisseur telles que le débit, le délai (tab. 1). Les approches existantes proposent ensuite des mécanismes spéciaux afin de maintenir la qualité du service.

Architecture Caractéristique	Internet /IETF IntServ & DiffServ	ITU/ATM Forum	OSI
Débit		PCR ¹ /SCR ²	X
Délai	X	CTD ³	X
Gigue		CDV ⁴	
Taux d'erreur		CLR ⁵	X
Disponibilité		X	X
Priorité	X		
Mode de connexion		X	
Fiabilité			X
Coût	X	X	

Tableau 1. Les informations de QoS utilisées dans les approches les plus importantes en télécommunications

B. Gestion de réseaux

L'objectif principal de la QoS dans le domaine de gestion des réseaux est l'application des politiques de contrôle des utilisateurs et/ou des ressources conformément aux contrats de QoS prédéfinis. Les informations de QoS servent à décrire : i) les services que le réseau doit fournir, et ii) les contraintes imposées sur l'utilisation de ces services. Par rapport aux informations plutôt abstraites des télécommunications, les informations de gestion de réseaux relient concrètement des équipements et des services exécutés par des différents composants : serveurs, routeurs, logiciels, systèmes d'exploitation. La QoS est alors spécifiée

en fonction des profils des composants individuels. Par contre, du point de vue de l'administrateur du réseau, ces informations doivent se représenter d'une manière systématique et convertible, ce qui aboutit à l'introduction des modèles d'informations. Le modèle d'informations le plus utilisé est CIM (*Common Information Model*), créé par DMTF (*Distributed Management Task Force*), ayant pour objectif de spécifier les informations de gestion des systèmes et des réseaux. Un modèle spécifique pour les informations de QoS est défini dans le cadre de CIM, en utilisant le langage UML pour spécifier la QoS fournie par les réseaux et servant notamment à représenter les contraintes de transfert de données du service [2].

C. Génie logiciel

Outre les télécommunications et la gestion des réseaux, le concept de QoS a été également introduit dans le domaine du génie logiciel avec deux objectifs: i) déterminer la qualité des produits logiciels, et ii) appliquer le génie logiciel dans la gestion de QoS. Le premier objectif concerne les critères pour estimer la qualité d'un produit ou un processus de développement [8]. Le deuxième objectif, en relation avec notre travail, concerne l'utilisation des activités du génie logiciel telles que la spécification, la conception et la maintenance dans la gestion de QoS. On a constaté que l'incorporation systématiquement de la QoS dans la phase de conception est essentielle et décisive pour les implémentations [5]. En général, comme une application ou un logiciel se compose de plusieurs composants qui sont dépendants les uns des autres, le choix des caractéristiques de QoS d'un composant peut influencer l'architecture globale du système. L'approche génie logiciel pour la QoS concerne les modèles d'informations décrivant les caractéristiques non-fonctionnelles des applications et des services, représentées par des langages formels. Les modèles d'informations sont alors spécifiés selon les besoins des utilisateurs et ne sont pas restreints en nombre de caractéristiques de QoS. En utilisant la sémantique des langages de spécification, une caractéristique de QoS peut être spécifiée par l'utilisateur d'une manière flexible, par exemple en utilisant QML [5].

D. Systèmes distribués

La coopération des techniques du génie logiciel dans le domaine des télécommunications a ensuite abouti aux architectures de QoS pour les systèmes distribués. La QoS est alors considérée comme la possibilité d'adapter les offres du système réparti aux besoins de l'utilisateur en tenant compte de plusieurs facteurs : les supports de communication, les applications de l'utilisateur ou encore la nature du service. La QoS utilisée dans les systèmes

¹ PCR : Peak Cell Rate

² SCR: Sustained Cell Rate

³ CTD: Cell Transfer Delay

⁴ CDV: Cell Delay Variation

⁵ CLR: Cell Loss Ratio

distribués se réfère aux architectures assurant la livraison du service de bout en bout ou de l'application à l'application.

Dans les systèmes distribués, la QdS est prise en considération dans un cadre plus large que ceux du génie logiciel et des télécommunications. Il s'agit de la coopération des composants du système, y compris le support de communication, l'application et les logiciels de gestion dans le but de fournir un service de qualité aux utilisateurs. Elle est par conséquent étudiée dans différents aspects : la gestion des ressources [9], la gestion de base de données [15], et la gestion coopérative [4]. L'objectif commun des architectures de QdS pour les systèmes distribués est de fournir une communication de qualité de bout-en-bout d'une manière transparente à l'utilisateur. La partition du service et l'allocation des ressources ainsi que l'optimisation sont réalisées à l'intérieur du système. Il n'existe pas en réalité une approche commune pour la QdS des systèmes distribués mais plusieurs tentatives ont été développées en utilisant des architectures en couches [1][3][10] ou orientées objet [12][14].

III. REVUE DES ARCHITECTURES DE QDS

Chronologiquement, les architectures de QdS pour les systèmes distribués ont été introduites au début des années 90s [1]. Cependant, ces premiers travaux ne peuvent pas encore être considérés comme des produits officiels car ils ne sont pas complétés au niveau de l'implémentation et de l'utilisation. Dans cette section, nous présentons les caractéristiques des architectures les plus remarquées :

- $2K^Q$ [10] est une architecture de QdS pour les applications temps réelles sur les supports à haute qualité (i.e. ATM). Son modèle d'informations se voit comme un ensemble de caractéristiques concernant les exigences de l'utilisateur, le fonctionnement de l'application et les ressources du système. $2K^Q$ réalise la transformation des exigences de l'utilisateur, considérées comme des informations génériques, vers les paramètres spécifiques des ressources, par l'intermédiaire des informations composites du système.
- QoSME [3] est un environnement de gestion de QdS qui se compose : i) d'un modèle et d'un langage de spécification de QdS (QuAL), ii) d'un mécanisme de monitoring automatique des informations de QdS, et iii) d'une approche d'intégration de la gestion de QdS avec le protocole de gestion des réseaux SNMP existant. Les activités de QdS se basent sur une base d'informations MIB géré par SNMP. Les informations contenues dans la MIB caractérisent l'application et les connexions et par définition se catégorisent en 4 groupes : i) application, ii) entrée, iii) sortie, et iv) définition des métriques. Les deux groupes

entrée et sortie contiennent les informations exprimant respectivement les exigences de l'application et l'offre des ressources. La transformation concerne principalement les informations entrées et sorties. L'architecture est attachée à SNMP et la base d'information n'est pas extensible.

- L'architecture de gestion de QdS pour les applications multimédia distribuées présentée par Hafid et Bochmann [7] se porte particulièrement sur les activités négociation et re-négociation. Elle prend en considération les informations statiques, qui identifie le service, et les informations dynamiques, qui se caractérisent la session de communication. Son modèle d'information regroupe les profils de l'utilisateur, de l'application et des ressources. La transformation met en correspondance les profils pour réaliser la négociation.

- QuO [14] est une architecture de QdS orientée objet. Elle définit trois langages de spécification : CDL (*Contrat Definition Language*), RDL (*RessourceDefinition Language*) et SDL (*Structure Definition Language*) dans le but de pouvoir traduire toute exigence de QdS de l'utilisateur en langage compréhensible par les ressources. Les contrats de QdS sont alors définis directement par les programmeurs et sur les objets spécifiés. QuO divise les informations de QdS en deux régions : i) la région négociée s'exprimant l'attente de QdS de l'utilisateur ou du système, et ii) la région réelle déclarant la QdS disponible ou réellement exigée. La transformation des informations d'un objet vers la plateforme est réalisée par CORBA.

- TAO [12] est un autre modèle de QdS orienté objet, basé sur l'architecture *middleware* distribuée CORBA et se focalisant sur les informations concernant la performance d'exécution (en terme du temps) et l'ordonnement des événements. Les améliorations de TAO par rapport au CORBA standard sont : i) la spécification et la validation de la QdS, ii) les caractéristiques temps réel, et iii) l'optimisation de la performance. Les informations de QdS sont spécifiées par des interfaces IDLs prédéfinies. La transformation se base généralement sur la transformation traditionnelle de CORBA vers les plates-formes spécifiques mais ajoute les méthodes de transformation particulières pour les propriétés de QdS.

Le tableau 2 présente une comparaison de ces architectures de QdS pour les systèmes distribués. Les aspects que nous prenons en compte sont: l'architecture de l'implémentation, les niveaux de spécification, les informations de QdS, le support de transmission, la transformation des informations de QdS et les méthodes de monitoring utilisées.

IV. ANALYSE ET PERSPECTIVE

Une tendance indispensable que l'on constate est que le développement dans les domaines étudiés (télécommunications, gestion de réseaux, génie logiciel, systèmes distribués) va certainement introduire de nouvelles informations concernant la QdS. Il devient alors indispensable d'avoir une approche systématique de la gestion de ces informations à travers une approche basée sur la modélisation. Néanmoins, dans les architectures de QdS les plus récentes et présentées précédemment, la gestion d'informations est réalisée de manière ad-hoc et intégrée directement dans le code des systèmes. Ceci amène plusieurs inconvénients

- la modification d'un des paramètres de qualité, s'il avait lieu, obligerait à modifier l'implémentation, voire l'architecture des modules du système,
- l'extension du système deviendrait difficile, car la définition d'un nouveau paramètre demande de changer l'implémentation et la conception des modules,
- l'impossibilité de modifier les aspects concernant la QdS lorsque l'application est en marche,
- la maintenance des aspects concernant le traitement de qualité du système coûte cher.

De l'autre côté, la transformation devient aussi indispensable pour la gestion de QdS, ceci étant justifié par l'hétérogénéité des informations utilisées et des standards existants. Étant donné que les descriptions quantitatives de QdS du système sont disponibles grâce à des outils de monitoring, la transformation des informations de QdS de l'application vers système est exigée et décisive pour réserver les ressources appropriées ou pour réaliser les négociations. Elle est demandée aussi bien lors de la phase d'initialisation du service que lors de la phase de re-négociation s'il y a des changements dans l'offre du système.

Nous travaillons présentement sur une approche qui vise à séparer la gestion d'information de QdS des modules du système de gestion de QdS [11] car notre revue de la littérature fait apparaître clairement que la modélisation et la transformation des informations de QdS permettront de gérer plus efficacement la QdS. Dans cette perspective, la gestion d'information de QdS est réalisée par un gestionnaire d'informations de QdS qui rend disponible des informations nécessaires à toutes les activités concernant la QdS. Il réalise alors la gestion d'une base d'information de QdS et la transformation de ces informations en fonctionnant indépendamment des autres modules du système. Cette approche est basée sur la gestion et la transformation de modèles d'informations de QdS. Dans un tel système, la

transformation se fait de telle sorte que les modèles d'informations des différents composants soient interchangeables, ce qui rend transparentes les activités de QdS du point de vue de l'application. Notre modèle prend en considération également la coopération au niveau des informations provenant d'architectures existantes.

V. CONCLUSION

Nous avons présenté dans cet article une revue synthétique des modèles d'informations de QdS ainsi que des mécanismes de transformation utilisés dans différentes architectures. Les comparaisons et analyses ont mis en évidence la nécessité d'avoir une approche modélisée pour la gestion des informations de QdS, qui constitue un des objectifs de notre recherche.

Architecture de QoS	Implémentation	Support de transmission	Niveaux de spécification	Informations de QoS	Transformation	Méthode de monitoring
2K ^Q	en couches, modulé	Haute qualité (réseau ATM)	Utilisateur Système Ressource	- génériques - composites - spécifiques	Compilateur transformant les informations génériques vers les informations spécifiques	les méthodes prédéfinies
Architecture de Hafid et Bochmann	en couches, modulé	Best effort (Internet), mais souvent les réseaux multimédia à haute qualité	N/A (quelques interfaces simples de l'utilisateur)	2 types : - Statique - Dynamique Représentées par des profiles	Transformation des informations de l'utilisateur aux configurations de l'application	outils disponibles des infrastructures et les protocoles de réservation de ressources
QoSME	en couches, modulé	Indépendant des protocoles de transport et des OSs	Utilisateur (utilisant le langage QuAL)	Une base d'informations MIB, mise à jour par le système	Mise en correspondance directement les valeurs entrées et sorties de la MIB	un mécanisme privé utilisant le protocole de transport et le OS disponibles
QuO	orienté objet	Défini par CORBA	Utilisateur Système (objet) Ressource (par les langages QDLs)	Deux régions d'informations : négociées et réelles	N/A (utilise CORBA)	chargée par l'infrastructure au-dessous ou implémentée dans CORBA
TAO	orienté objet	Défini par CORBA	IDLs prédéfinies	Informations principalement concernent le temps d'exécution	Extension de la transformation du CORBA vers les plates-formes spécifiques	chargée par l'infrastructure au-dessous ou implémentée dans CORBA

Tableau 2. Comparaison des architectures de QoS

REFERENCES

- [1] Campbell, A.T., « A Quality of Service Architecture,» Ph.D thesis, p18-94, Lancaster University, England, 1996.
- [2] DMTF Inc. «CIM QoS Device Sub-Model Specification». 2000.
- [3] Patricia Gomes Soares Florissi, « QoSME: QoS Management Environment,» Ph.D. Thesis, Columbia University, USA, 1996.
- [4] Fischer, S., Hafid, A., de Meer, H. « Cooperative QoS Management for Multimedia Applications,» IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS'97), Ottawa, Canada, 1997.
- [5] Frolund, Svend; Koisten, Jari , « QML: A Language for Quality of Service Specification,» HP technical report, 1998.
- [6] Gerbé, O., Kerhervé, B., Srinivasan, U. «Model Operations for Quality-Driven Multimedia Delivery,» in the Proceeding of ICCS 2003, International Conference on Conceptual Structures (ICCS2003), July 21-25, 2003 in Dresden, Germany.
- [7] Hafid, A., Bochmann, G., «An Approach to QoS Management in Distributed Multimedia Applications: Design and an Implementation,» Multimedia Tools and Applications, vol. 9, no. 2, 1999.
- [8] International Standard ISO/IEC 9126, « Information technology - Software product evaluation - Quality characteristics and guidelines for their use, International Organization for Standardization,» International Electrotechnical Commission, Geneva., 1991
- [9] Nahrstedt, K., Steinmetz, R., « Resource Management in Networked Multimedia Systems,» IEEE Computer 28(5), p52-63, 1995
- [10] Nahrstedt, K., Wichadakul, D., Xu, D., « Distributed QoS Compilation and Runtime Instantiation,» IEEE/IFIP International Workshop on Quality of Service, pp. 198-207, June 2000.
- [11] Nguyen, K., Fetjah, L., Kerhervé, B., « Quality of service information base (QoSIB) Manager for Electronic Commerce

Application,» Poster presented at CITR Annual Conference, Ottawa, Canada, 2001.

- [12] Schmidt, D., Levine, D., Cleeland, C., « Architectures and Patterns for High- performance, Real-time CORBA Object Request Brokers,» In *Advances in Computers*, Marvin Zelkowitz, Ed., Academic Press, 1999.
- [13] Steinmetz R., Nahrstedt K., « *Multimédia: Computing, Communications & Application*». Prentice Hall, 1995.
- [14] Zinky, J., Bakken, D., Schantz, R., « Architecture Support for Quality of Service for CORBA Objects,» *Theory and Practice of Object Systems*, vol. 3, no.1, Jan. 1997.
- [15] Ye H., Kerhervé B., Bochmann G., Oria, V. «Pushing Quality of ServiceInformation and Requirements into Global Query Optimization,» in the *Proceeding of the International Database Engineering and Applications Symposium (IDEAS)*, Hong Kong, July 2003. Published by IEEE.