

Réduction des besoins en communication des systèmes objets distribués utilisant CORBA

Peter G. Kropf, Gilbert Babin et Alexandre Hulot

Laboratoire PARADIS
Département d'informatique
Université Laval
Québec, Canada G1K 7P4
{kropf,babin}@ift.ulaval.ca

RÉSUMÉ— CORBA a été conçu en supposant que le réseau peut toujours suffire à la demande en communication. Ce n'est cependant pas toujours le cas. En particulier, on notera le cas des Systèmes de Commandement et Contrôle (SCC) utilisant CORBA pour la communication entre sous-systèmes distribués. Les SCC doivent garantir la transmission des données, même dans les situations extrêmes où les performances des liens de communication peuvent tomber jusqu'à 1 200 bps. Dans un premier temps, cet article présente les besoins et les problèmes de communication des SCC. Par la suite, un modèle indiquant comment le trafic généré par les SCC peut être réduit est décrit. Ce modèle identifie quatre types de réduction : (1) réduction pragmatique, où une connaissance du contexte d'exécution de l'application s'avère nécessaire, (2) réduction sémantique, où une connaissance de la sémantique de l'application s'avère nécessaire, (3) réduction syntaxique, où seule une connaissance du langage de communication est nécessaire et (4) réduction physique, telle la compression des données, où aucune connaissance spécifique n'est nécessaire. Finalement, l'article présente une approche d'expérimentation du modèle.

MOT-CLÉS : *CORBA, qualité de service, faible bande passante, réduction du trafic*

ABSTRACT— CORBA has been designed while supposing that the network capacity is always sufficiently large. This however, is not always the case. In particular, we note the case of Command and Control Systems (C2S) using CORBA for communication between distributed subsystems. These C2S need to guarantee the transmission of data, even in extreme situations where the communication performance may drop to as low as 1 200 bps. In this paper, we first discuss the needs and problems of communication in C2S which use CORBA. We then present a model of how to reduce data transmission in such systems. The model identifies four types of data reduction : (1) pragmatic reduction, based on the context of the application execution, (2) semantic reduction taking the application's semantics into account, (3) syntactic reduction, where only knowledge of the communication language is needed, and (4) physical reduction, such as mere data compression which does not require any specific knowledge at all. Finally, the paper presents an approach to experimentations using our model.

KEY WORDS : *CORBA, quality of service, low bandwidth, traffic reduction*

1. Introduction

Le standard *Common Object Request Broker Architecture* [Mowbray 95] (CORBA) offre un environnement de développement et d'exécution de logiciels pour des systèmes et des applications distribués. Il se base sur le paradigme des objets distribués qui apporte plus de puissance au modèle client-serveur. Les différentes parties d'une application distribuée peuvent être exécutées sur des machines différentes utilisant des systèmes d'exploitation hétérogènes et peuvent être écrites dans des langages différents. La majorité des solutions basées sur CORBA vise à construire des applications s'appuyant sur des réseaux locaux ou des réseaux étendus offrant des liens de communication fiables et à des taux de transfert élevés. Il existe cependant des environnements et des situations pour lesquelles la capacité de communication normalement supposée n'est plus garantie. Les réseaux sans fil représentent un tel environnement où la performance et la fiabilité sont imprévisibles.

1.1. Systèmes de Commandement et Contrôle

Les Systèmes de Commandement et Contrôle (SCC) sont un type d'application utilisant fréquemment plusieurs sous-systèmes de communication (réseaux locaux interconnectés) ou des réseaux étendus. Ces systèmes doivent offrir une garantie de fonctionnement dans des situations extrêmes où les performances des liens de communication peuvent tomber jusqu'à 1 200 bps, comme par exemple, lorsque des connexions par radio-onde sont utilisées.

Le modèle générique d'implantation des SCC dans un environnement distribué et orienté objet (fig. 1) peut être décrit par trois niveaux : (1) SCC distribué, (2) *middleware* (p.ex., CORBA, DCOM), (3) infrastructure (communication et calcul).

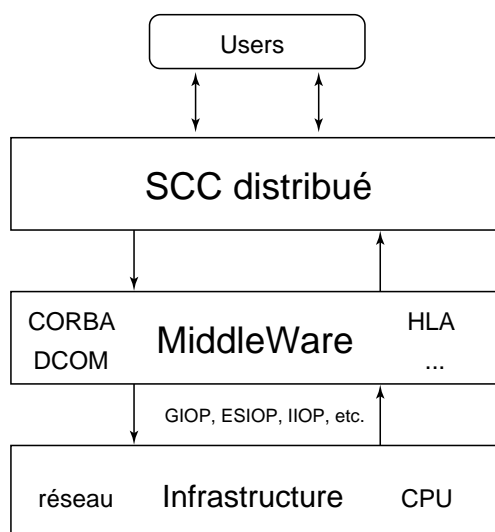


Figure 1. *Modèle générique d'implantation des SCC dans un environnement distribué et orienté objet*

Dans cet article, nous considérons en particulier les Systèmes de Commandement et Contrôle permettant la collecte, le partage et l'analyse d'informations tactiques et stratégiques pour des fins militaires. Généralement, ces SCC sont constitués de sous-systèmes localisés (1) dans les postes de commandement et (2) dans les unités déployées sur le champ de bataille. De nouveaux SCC militaires sont présentement construits en utilisant une approche orientée objet, ce qui rend l'utilisation de CORBA naturelle. En effet, les différents sous-systèmes peuvent être vus comme clients (p.ex., demande de renseignements au poste de commandement) ou serveurs (p.ex., transfert de données tactiques au poste de commandement).

1.2. Besoins et problèmes des SCC

Une caractéristique des SCC est que l'état de l'environnement change dans le temps. Ces changements peuvent découler de contraintes imposées par les procédures standards (p.ex., les communications peuvent être réduites à leur minimum pour éviter la détection par l'ennemi) ou simplement être la conséquence d'une défaillance (p.ex., une station relais micro-onde a été détruite). Malgré cela, les SCC doivent être robustes en ce sens que le service fourni doit l'être quelles que soient les conditions externes. En particulier, la transmission des données entre les différents sous-systèmes doit être préservée dans la mesure du possible. Cependant, la forme et la fréquence de ces transmissions peuvent être dégradées selon l'état de l'environnement. Ainsi, la qualité de service (QoS) offerte (c.-à-d., la garantie que la transmission des informations nécessaires au fonctionnement du SCC est effectuée) est fonction des conditions d'opération du système à un instant donné. La notion de QoS dans les SCC orientés objets est présentée plus en détail dans [Bakken 96]. Par exemple, lorsque les liens de communication le permettent, un char d'assaut peut informer le poste de commandement de ces déplacements à toutes les secondes. Cependant, cette fréquence peut être réduite lorsque les liens à haute vitesse ne sont plus disponibles. Minimale, nous considérons que l'état du réseau ne peut se détériorer de telle façon que les liens de communications soient en deça de 1 200 bps. Dans le contexte militaire, ces liens à faible bande passante utilisent les radio-ondes tel que le projet canadien *Combat Net Radio* (CNR) [Chamberlain 94]. Il est donc nécessaire de développer des moyens permettant de dynamiquement modifier les mécanismes de transmission des données tout en garantissant la QoS.

1.3. Problématique et organisation de l'article

Dans cet article, nous analysons les besoins de communication dans le contexte des SCC utilisant des mécanismes de communication entre objets basés sur CORBA et nous proposons et discutons une architecture à quatre niveaux permettant la réduction des données communiquées. Cette architecture agit aux trois niveaux d'implantation des SCC mentionnés précédemment (sect. 1.1). Plus particulièrement, nous considérons la situation où le *middleware* utilisé est CORBA.

Dans un premier temps, il est nécessaire de poser certaines hypothèses quant à l'environnement dans lequel évolue le SCC.

1. *Plusieurs liens de communication disponibles.* Les différents participants au SCC ont accès à plusieurs liens de communication. Cependant, le nombre de liens disponibles peut être plus petit que le nombre de liens existants.
2. *Fiabilité des liens de communication.* Les liens de communication disponibles sont fiables. Lorsqu'un lien de communication est disponible, celui-ci transmet les données parfaitement entre les deux points reliés. Ainsi, on dira que les liens sont fiables à 100% lorsque disponibles et fiables à 0% lorsque non disponibles.
3. *Caractéristiques des liens.* Chaque lien existant a des caractéristiques propres. Il est donc possible, selon les besoins et circonstances, de classer les liens selon ces caractéristiques pour sélectionner le meilleur lien.
4. *État de disponibilité des liens variable dans le temps.* L'approche proposée doit avoir accès à l'information sur l'état des liens pour dynamiquement modifier les mécanismes de réduction.
5. *Existence de connaissances préalables.* Il existe des connaissances préalables à l'exécution du SCC qui peuvent être dupliquées dans les différents sous-systèmes.
6. *Faisabilité de la réduction.* Il est possible de trouver une combinaison de réductions permettant de rester à l'intérieur des caractéristiques d'au moins un lien de communication disponible.
7. *Capacités relatives des ordinateurs et du réseau.* En termes relatifs, la capacité de traitement des ordinateurs est de beaucoup supérieure à la capacité de transmission du réseau.

Dans ce contexte, nous démontrons qu'il existe quatre types distincts de réduction : pragmatique, sémantique, syntaxique et physique. Ces quatre types de réduction servent de base à un modèle intégré de réduction du trafic de communication dans les SCC (sect. 2). Les modes de fonctionnement de ces différents types de réduction sont analysés pour identifier leurs caractéristiques propres et ainsi démontrer qu'ils sont disjoints (sect. 2.1 à 2.4).

Une expérimentation devra permettre de déterminer les réductions réelles obtenues lors de l'utilisation du SCC, autant en situation normale qu'en situation d'urgence. La section 3 décrit l'approche d'expérimentation privilégiée.

2. Modèle de réduction des besoins en communication

On trouve dans la littérature plusieurs approches permettant de garantir la QoS des communications. Ces approches s'attaquent habituellement à un niveau spécifique de l'implantation des systèmes distribués (SCC distribué, *middleware*, infrastructure). Dans [Bakken 96, Loyall 98, Zinky 97], les auteurs s'attaquent au niveau SCC distribué en utilisant une approche objet et CORBA. Leur approche suggère l'utilisation d'une méthode *proxy* qui est le point d'entrée pour tout appel à un service donné. Une fois invoquée, cette méthode détermine l'état de l'infrastructure et gère le niveau de service à fournir en fonction de cet état. Ainsi, la méthode *proxy* détermine dynamiquement quelle méthode est réellement invoquée pour fournir le service à l'objet client en tenant compte de l'état de l'infrastructure. L'état de l'infrastructure est donc constamment surveillé pour que l'application s'adapte aux changements qui peuvent survenir.

L'utilisation d'une méthode *proxy* comme récepteur primaire des demandes de service se retrouve dans différentes architectures qui tiennent compte de la disponibilité et de l'état des ressources de l'infrastructure. Par exemple, Lo-DACE [Badidi 98] propose une architecture basée sur le service de médiation et la surveillance de la charge des serveurs. Les demandes de service sont adressées à une méthode *proxy* qui, après consultation des bases d'information sur la charge des serveurs et les services offerts, détermine dynamiquement le serveur le moins chargé. Cependant, cette architecture ne considère que la charge des serveurs et non pas la charge et la capacité des liens de communications.

Les travaux présentés dans [Raatikainen 97] mettent l'accent sur l'interface entre les niveaux *middleware* (CORBA) et infrastructure dans le cas des réseaux sans fil et des terminaux mobiles. L'architecture proposée se base sur le concept de ponts interopérables. Un réseau est divisé en plusieurs domaines mobiles et en réseaux fixes. Les *Object Request Broker* (ORB) des terminaux mobiles sont connectés à l'ORB fédérateur d'un réseau fixe par l'intermédiaire de deux demi-ponts. Pour la communication entre les demi-ponts, le protocole LW-IOP (*Light Weight Inter-ORB Protocol*) a été développé, permettant la réduction du trafic inter-ORB sur une infrastructure caractérisée par une bande passante limitée et variable. Ce protocole offre la même fonctionnalité que GIOP (*General Inter-ORB Protocol*) et IIOP (*Internet Inter-ORB Protocol*), conformément au standard ESIOP (*Environment Specific Inter-ORB Protocol*). Cependant, la taille des messages de LW-IOP a été minimisée par rapport à GIOP/IIOP.

Chamberlain [Chamberlain 94, Chamberlain 96] s'attarde aux niveaux SCC distribué et *middleware*. Au niveau SCC, il sépare les données distribuées des applications les utilisant. Il crée alors une *base de fait distribuée* (BFD) contenant toutes les données devant être partagées entre les systèmes distribués. En fait, il propose que seule la BFD soit distribuée. Ainsi, les communications sont limitées aux opérations de synchronisation des composantes de la BFD. De plus, cette synchronisation peut être faite en utilisant des règles préétablies. D'un point de vue militaire, on parle de procédures opérationnelles standards. Ainsi, selon l'état de l'infrastructure, la fréquence et la priorité des mises à jour peuvent varier. Au niveau *middleware*, Chamberlain préconise l'utilisation du protocole FEP (*Fact Exchange Protocol*). Ce protocole transmet des datagrammes en *multicast* et en *broadcast* à travers un CNR. De plus, il permet la concaténation de messages et l'écoute automatique (tous les messages reçus par un noeud sont préservés même s'il ne lui sont pas destinés).

À l'instar de Chamberlain, nous croyons nécessaire de séparer les données des applications. De cette manière, seuls quelques services simples doivent être partagés entre les différents sites : création d'un objet, suppression d'un objet, mise à jour d'un objet, consultation d'un objet, abonnement à une classe. Les quatre premiers services sont des fonctions classiques des bases de données. Le service d'abonnement correspond à la situation où un client

reçoit automatiquement des informations sur les changements survenus à des objets d'une classe (*pushed information*), plutôt que de faire une demande active d'information en provenance de ces objets (*pulled information*). Selon Chamberlain [Chamberlain 94], l'approche par abonnement permet de mieux contrôler le trafic sur le réseau.

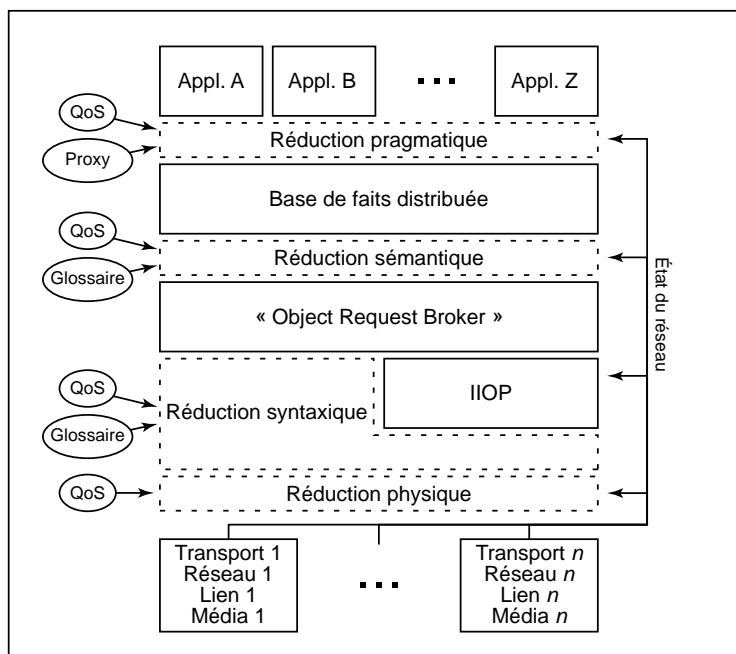


Figure 2. Vue d'ensemble du modèle de réduction

Le modèle de réduction que nous proposons (fig. 2) généralise les résultats précités. Dans ce modèle, le niveau SCC distribué est subdivisé en un niveau *Application* et un niveau *Base de faits distribuée* (BFD), le niveau *middleware* est subdivisé en un niveau ORB et un niveau inter-ORB (GIOP/IIOP), et le niveau infrastructure correspond aux couches de protocole des réseaux de communication utilisés (les couches *Transport*, *Réseau*, *Lien*, et *Média* selon le type de réseau). Les réductions possibles sont de quatre types :

Pragmatique : survient lorsque l'application fait appel aux services de la BFD (sect. 2.1);

Sémantique : permet l'encodage des données transmises via l'ORB, et possiblement via GIOP/IIOP (sect. 2.2);

Syntaxique : permet la réduction des messages transmis à l'infrastructure, en fonction de l'état de l'infrastructure (sect. 2.3).

Physique : permet la compression du message avant son transfert via l'infrastructure (sect. 2.4).

Les réductions dépendent du niveau de qualité de service (QoS) demandé et par le fait même, de l'état du réseau influençant cette QoS. L'état du réseau est en fait l'information indiquant qu'un lien est disponible ou non. En effet, comme précisé à la section 1.3, les liens sont soit disponibles à 100%, soit non disponibles. À partir de l'état du réseau, il est donc possible de déterminer le meilleur lien pouvant être utilisé à un instant donné. En particulier, le seul lien disponible peut n'être qu'un lien à basse vitesse, tel CNR, auquel cas, tous les types de réductions sont appliqués de façon maximale. Notons enfin que les types de réduction sémantique et syntaxique font usage d'un glossaire permettant l'encodage de l'information transmise.

La nomenclature des types de réduction suit la logique suivante. On suppose que l'objectif global est de réduire la quantité et la longueur des messages transmis sur le réseau sans perte d'information. Les messages sont codés en

utilisant un langage de communication, défini par sa syntaxe et son lexique (p.ex. IIOP), et basé sur un alphabet (p.ex. ASCII). La réduction physique agit uniquement au niveau de l'alphabet, sans tenir compte du lexique ou de la syntaxe. La réduction syntaxique se sert des connaissances de la syntaxe et du lexique du langage. La réduction sémantique se sert des connaissances sur la signification du message alors que la réduction pragmatique se sert du contexte de fonctionnement du système. Il ressort donc que les quatre types de réduction agissent sur la quantité et la longueur des messages à quatre niveaux d'abstraction distincts, relativement à la notion de langage de communication.

2.1. Réduction pragmatique

L'objectif de la réduction pragmatique est d'utiliser les connaissances qu'ont les usagers de la nature du SCC pour réduire le trafic selon l'état de l'infrastructure. On cherche en fait à réduire la redondance dans l'information transmise entre les différents systèmes. La réduction pragmatique est appliquée lorsque l'application doit faire appel à un service de la BFD. Pour chaque service, plusieurs implantations sont disponibles afin de fournir ce service selon les niveaux de QoS requis. Cependant, une seule méthode est accédée par l'application (méthode *proxy*), comme proposé dans [Bakken 96, Loyall 98, Zinky 97]. Cette méthode *proxy* utilise l'état de l'infrastructure pour déterminer quelle implantation devra être utilisée à un instant donné.

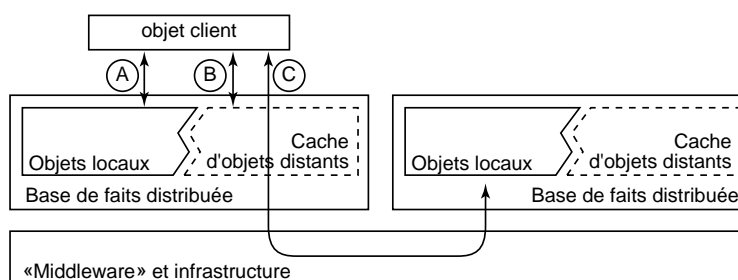


Figure 3. Fonctionnement de la base de faits distribuée

La réduction pragmatique dépend directement de l'approche d'implantation de la BFD. Sur chaque site, la BFD est composée d'un ensemble d'objets locaux et d'une cache d'objets distants (fig. 3). La cache permet de limiter, sous certaines conditions, les accès aux objets distants. Ainsi, lorsqu'un objet client fait une demande de service à la BFD, trois situations peuvent survenir, correspondant respectivement aux lettres A, B et C dans la figure 3 :

- A. La requête est transmise à un objet local. Dans ce cas, aucun message n'est transmis sur le lien de communication.
- B. La requête est transmise à une copie d'un objet distant localisé dans la cache. C'est le cas lorsqu'une application locale s'est abonnée au service de la classe en question et que la requête en est une de consultation. Ici encore, aucun message n'est transmis sur le lien de communication lors de la requête. Par contre, des messages de mise à jour de la cache sont nécessaires à une fréquence qui dépend de la QoS et de l'état du réseau.
- C. La requête est transmise à l'objet distant. On aura alors un message correspondant à la requête et un message correspondant à la réponse, lorsque nécessaire.

Il est clair que ce genre de système de gestion d'objets distribués peut faire partie intégrante du SCC distribué. Cependant, ces fonctionnalités seraient certainement plus efficaces si elles étaient disponibles au niveau de l'ORB.

La figure 4 montre les interactions possibles des objets des applications (OA_i) avec un objet OD de la BFD par l'intermédiaire de la méthode *proxy*; la QoS ainsi que l'état du réseau sont utilisés par la méthode *proxy* pour déterminer le degré et l'importance de la réduction pragmatique devant avoir lieu, et ainsi déterminer le service de l'objet OD à utiliser. Par exemple, l'objet OA_1 veut modifier l'objet OD . La méthode *proxy* déterminera si le changement est propagé immédiatement ou plus tard.

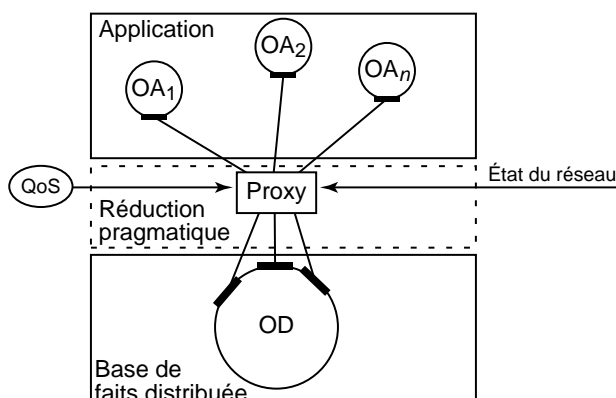


Figure 4. Réduction pragmatique

2.2. Réduction sémantique

La réduction sémantique tire avantage des connaissances préalables partagées par toutes les applications du SCC. En effet, dans le contexte des SCC, surtout militaire, une certaine discipline est imposée quant à l'utilisation de termes spécialisés. Cette connaissance préalable peut être utilisée pour coder ces termes communs ayant un sens précis et ainsi réduire le volume d'information à transmettre via l'ORB.

L'approche que nous privilégions (fig. 5) utilise un glossaire pour emmagasiner les tables permettant le codage des termes communs entre les applications. Le nombre de termes encodés varie selon le niveau de QoS exigé et l'état du réseau.

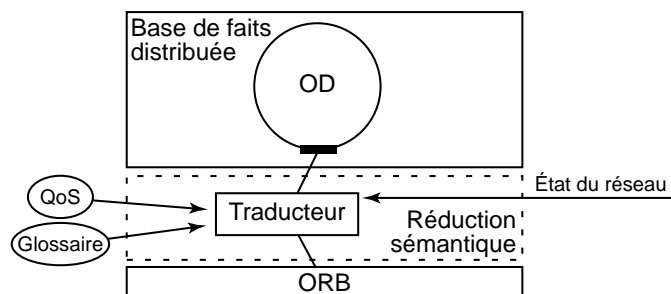


Figure 5. Réduction sémantique

2.3. Réduction syntaxique

La réduction syntaxique agit sur le langage de communication utilisé pour transmettre les messages. Ce type de réduction est en fait la composition des réductions pragmatique et sémantique au niveau du *middleware*, plutôt qu'au niveau du SCC distribué lui-même. La composante de réduction pragmatique réduit le nombre et la taille des messages nécessaires au lien de communication; ce type de réduction est celui utilisé dans [Raatikainen 97]. La composante de réduction sémantique se limite à encoder l'information de contrôle utilisé dans les messages.

De plus, on constate que les SCC forment normalement des domaines d'interopérabilité fermés (c.-à-d., tous les objets participants au SCC sont connus). Cette caractéristique rend possible l'implantation transparente d'un ESIOP entre les objets formant le SCC. Un tel protocole permet de réaliser la réduction syntaxique, tout en restant compatible

avec CORBA.

Le modèle générique de réduction syntaxique est illustré à la figure 6. Le *traducteur* réalise deux fonctions :

- une composante de réduction pragmatique, implantée sur la base de ESIOP;
- une composante de réduction sémantique, utilisant un glossaire pour encoder les informations de contrôle utilisées par GIOP/IIOP ou une instance d'un ESIOP.

Dans les deux cas, le niveau de QoS et l'état du réseau serviront à choisir dynamiquement le protocole inter-ORB utilisé et le niveau d'encodage des informations de contrôle.

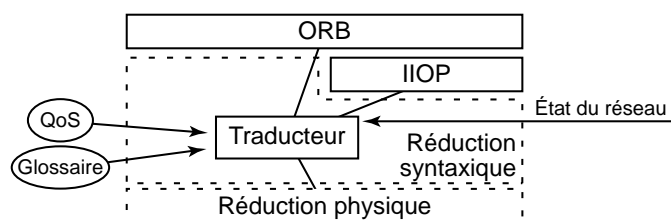


Figure 6. Réduction syntaxique

2.4. Réduction physique

La réduction physique consiste simplement en la compression des données transmises sur le lien de communication. On veut éliminer toute redondance dans l'information transmise tout en s'assurant que l'information originale est préservée. Ainsi, le *compresseur* (fig. 7) ne fait qu'appliquer une méthode de compression de données permettant d'atteindre le niveau de QoS requis en fonction de l'état du réseau.

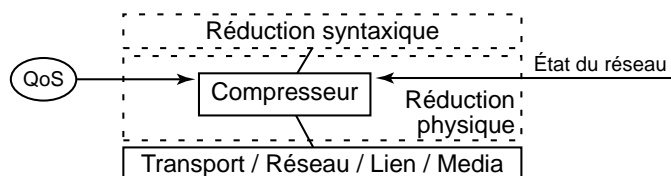


Figure 7. Réduction physique

3. Approche d'expérimentation

L'expérimentation doit nous permettre de mesurer :

- La *réduction effective du trafic de données*. Cette réduction correspond au ratio

$$\frac{\sum_{i=1}^m L_i - L'_i}{\sum_{i=1}^m L_i},$$

où m est le nombre de messages transmis, L_i est la longueur du message i avant la réduction et L'_i est la longueur du message i après la réduction.

- *Le délai moyen avant transmission des messages.* Ce délai est calculé en fonction du lien de communication utilisé avec la formule

$$\frac{\sum_{i=1}^m T'_i - T_i}{m},$$

où T_i est l'heure d'arrivée du message i au niveau infrastructure et T'_i est l'heure de transmission du message par l'infrastructure.

Les longueurs des messages à transmettre sur le réseau sont recueillies avant et après l'exécution de chacun des modules de réduction (fig. 8). De cette manière, on peut déterminer la réduction effective à l'intérieur d'un seul (p.ex., réduction syntaxique) ou de plusieurs modules de réduction. De plus, on recueille l'heure d'arrivée d'un message au niveau de l'infrastructure. De cette manière, on peut calculer le délai en fonction de la longueur du message, de l'heure d'arrivée et de la vitesse et capacité du lien de communication.

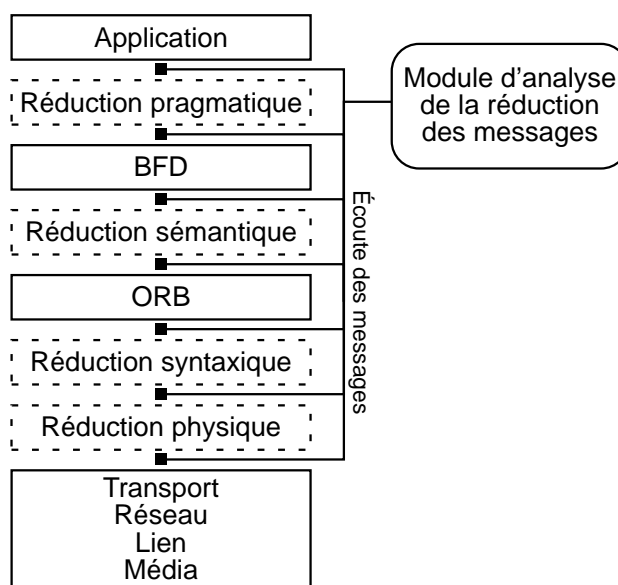


Figure 8. Approche d'expérimentation

Les valeurs de la réduction effective et du délai moyen seront calculées de façon analytique, en étudiant les mécanismes de réduction proposés, et de façon empirique, en utilisant un prototype d'application. Il sera alors possible de comparer les résultats analytiques et empiriques.

Dans un premier temps, les mesures empiriques seront effectuées avec un prototype de SCC distribué simple. Ce premier prototype mettra l'accent sur le niveau SCC distribué et implémentera les modules de réduction pragmatique et sémantique. En particulier, le niveau *middleware* se servira de IIOP pour les communications inter-ORB.

Un deuxième prototype permettra d'étudier plus à fond les réductions syntaxique et physique par l'implantation d'une instance d'un ESIOP. Ce prototype permettra d'évaluer la performance de l'architecture de réduction du trafic de données dans un environnement à faible bande passante.

4. Conclusion

Nous avons décrit dans cet article la problématique des systèmes distribués objet dans le cas d'une infrastructure de communication ayant des capacités de transmission limitées ou restreintes. Ceci représente une nouvelle problématique puisque les environnements de développement et d'exécution de logiciels distribués, tel que COBRA, se basent

toujours sur des moyens de communication à des vitesses de transmission élevées. Les systèmes de commandement et de contrôle (SCC) exigent de plus le fonctionnement dans des situations extrêmes où les taux de transmission peuvent tomber à des niveaux très bas. Ces taux peuvent même être beaucoup plus bas que dans le cas des réseaux sans fil pour lesquelles l'utilisation de CORBA a été traitée, par exemple dans [Raatikainen 97].

Nous avons présenté une analyse des besoins de communication dans le contexte des SCC utilisant une approche orienté objet (CORBA). En prenant une position d'intégration de systèmes, un ensemble d'hypothèses portant sur l'environnement de communication dans lequel les SCC sont mis en oeuvre a été établi. Nous présentons ainsi une architecture de réduction du trafic de données à quatre types de réduction. Ces types de réduction (pragmatique, sémantique, syntaxique et physique) interviennent au niveau des applications, du *middleware* et de la communication des données. Notre approche de réduction offre une grande flexibilité pour répondre à la qualité de service requise en fonction de l'état du réseau de communication. Il est ainsi possible d'effectuer une réduction orientée application, utilisant des bases de faits distribuées (concept introduit par Chamberlain [Chamberlain 94]), tout autant qu'une réduction indépendante de l'application au niveau du *middleware* et de l'infrastructure de communication. L'architecture présentée permet de traiter et de réaliser ces différentes réductions dans le cadre d'une approche unifiée.

Au moment d'écrire cet article, nous sommes en train d'implanter un environnement d'expérimentation qui utilise l'ORB de Visigenic. Cette expérimentation inclut en particulier l'implantation d'une base de faits distribuée type. Le travail futur portera sur l'implantation, l'analyse et la validation de notre architecture dans le contexte des SCC et d'autres applications ou infrastructures qui sont exposées au problème d'une bande passante limitée.

5. Remerciements

Ce travail est supporté par le Centre de Recherches pour la défense Valcartier (CRDV, contrat no. W7701-7-1840/001/XSK).

6. BIBLIOGRAPHIE

- [Badidi 98] E. Badidi, R.K. Keller, P.G. Kropf, et V. Van Dongen. Dynamic server selection in distributed object computing systems. Dans *DCW'98*, Germany, June 1998. University of Rostock.
- [Bakken 96] David E. Bakken. Object-oriented qos for c2 adaptivity and evolvability (position paper). Dans *DARPA Workshop on Security Technology for Next-Generation C2 Systems*, Institute for Defense Analyses, Alexandria, VA, July 1996. BBN Systems and Technologies. article disponible, <http://www.dist-systems.bbn.com/papers/1996/paper-96-01.html>, visité 03/98.
- [Chamberlain 94] Samuel C. Chamberlain. Automated information distribution in bandwidth-constrained environments. Dans *1994 IEEE/AFCEA Military Communications Conference, MILCOM 94*, vol. 2, pp. 537–541. U.S. Army Research Laboratory, October 1994. article disponible, <http://homepage.arl.mil/wildman/PAPERS/papers.html>, visité 03/98.
- [Chamberlain 96] Samuel C. Chamberlain. Resilient data replication mechanisms for battle command systems. Dans *1996 SYMPOSIUM ON C2 RESEARCH & TECHNOLOGY*. U.S. Army Research Laboratory, June 1996. article disponible, <http://homepage.arl.mil/wildman/PAPERS/papers.html>, visité 03/98.
- [Loyall 98] Joseph P. Loyall, Richard E. Schantz, John A. Zinky, et David E. Bakken. Specifying and measuring quality of service in distributed object systems. Dans IEEE, éditeur, *Proceedings of ISORC'98*, Kyoto, April 1998. BBN Technologies. article disponible, <http://www.dist-systems.bbn.com/papers/1998/ISORC/isorcweb.html>, visité 03/98.
- [Mowbray 95] T.J. Mowbray et R. Zahavi. *The Essential CORBA: Systems Integration Using Distributed Objects*. John Wiley & Sons, New York, NY, USA, 1995.

- [Raatikainen 97] K. Raatikainen. Bridging and wireless access for terminal mobility in CORBA. DOLMEN/LK-OMG01 AC036, EC/ACTS, November 1997. DOLMEN Consortium.
- [Zinky 97] John A. Zinky, David E. Bakken, et Richard E. Schantz. Architectural support for quality of service for corba objects, April 1997. BBN Systems and Technologies, article disponible, <http://www.dist-systems.bbn.com/papers/1997/TAPOS/TAPOS.html>, visité 03/98.

Biographies

Peter G. Kropf a reçu une maîtrise en mathématique (1984) et un doctorat en informatique (1991) de l'Université de Berne, Suisse. Depuis 1994, il est professeur adjoint au Département d'informatique de l'Université Laval, Québec. Ses intérêts de recherche actuels portent sur les systèmes parallèles et distribués, le calcul scientifique et l'Internet computing. M. Kropf est cofondateur du Laboratoire PARADIS, laboratoire de recherche sur les systèmes parallèles et distribués.

Gilbert Babin a obtenu son baccalauréat et sa maîtrise de l'Université de Montréal en 1986 et 1989, respectivement. Il a par la suite poursuivi des études de doctorat jusqu'en 1993 au Rensselaer Polytechnic Institute (Troy, New York, États-Unis), où il a étudié les techniques d'intégration de systèmes hétérogènes et distribués. Depuis juin 1998, il est professeur agrégé au Département d'informatique de l'Université Laval, où il a débuté comme professeur adjoint en 1993. Ces intérêts de recherche gravitent autour de la modélisation des systèmes d'information, de l'utilisation de ces modèles pour améliorer l'efficacité des systèmes d'information et de l'utilisation du potentiel du Web. M. Babin est cofondateur du Laboratoire PARADIS, laboratoire de recherche sur les systèmes parallèles et distribués.

Alexandre Hulot est étudiant à l'école d'ingénieur CPE Lyon (France) et obtiendra le diplôme d'ingénieur en juillet 1998. Après avoir effectué à Lyon ses deux premières années en tant qu'élève ingénieur, il a travaillé pendant un an chez Philips au sein des laboratoires de recherche en systèmes médicaux (Hambourg, Allemagne). Il est actuellement étudiant d'échange à l'Université Laval et effectue son stage de fin d'études au sein du Laboratoire PARADIS. Il travaille présentement sur l'intégration de CORBA dans des réseaux à faible bande passante. Ses intérêts de recherche portent sur les langages orientés objets et sur les objets distribués.