

QoS requise par une application de DIS dans un environnement réseau grande distance

°C.Chassot, °A.Lozes, °M.Diaz

°LAAS/CNRS, 7 Avenue du Colonel Roche
31077 Toulouse cedex 04. France
e-mail : chassot, alozes, diaz@laas.fr

°*L.Dairaine, *°L.Rojas

*ENSICA, 1 Place Emile Blouin
31056 Toulouse cedex. France
e-mail : dairaine, cardenas@ensica.fr

RÉSUMÉ : ces dernières années, les évolutions technologiques conjointes de l'informatique et des télécommunications ont conduit au développement de nouvelles applications distribuées, multimédias, nécessitant le traitement informatique et le transfert différé ou en temps réel de tous les types de données. Parmi ces applications, les applications de simulation interactive distribuée (DIS : Distributed Interactive Simulation) présentent des caractéristiques très contraignantes (trafic évolutif, contraintes de temps réel et de fiabilité notamment) qui rendent difficile tant la caractérisation de leur trafic que la spécification de la QoS (QoS : Quality of service) dont elles ont besoin. Cet article présente tout d'abord les caractéristiques majeures des applications DIS ; dans un second temps sont exposés les résultats d'une analyse de la QoS requise par ce type d'application dans un environnement réseau grande distance. Enfin, les conclusions de l'article présentent les principes de conception d'une architecture de bout en bout garantissant la QoS proposée dans un environnement Internet nouvelle génération.

ABSTRACT : ***

MOTS CLÉS : *Simulation Interactive Distribuée, Qualité de Service, Temps Réel, Dead Reckoning, Architecture de Bout en Bout, Internet nouvelle génération*

KEY WORDS : ***

1. Introduction

1.1. Cadre général

Ces dernières années, les évolutions technologiques conjointes de l'informatique et des télécommunications ont conduit au développement de nouvelles applications distribuées, multimédias, nécessitant le traitement informatique et le transfert différé ou en temps réel de tous les types de données. Parmi ces applications, les applications de *simulation interactive distribuée* (DIS : Distributed Interactive Simulation) présentent des caractéristiques encore très contraignantes (trafic évolutif, contraintes temporelles plus sévères notamment) qui rendent difficile tant la caractérisation de leur trafic que la spécification de la QoS dont elles ont besoin.

Dans l'optique de déployer ces nouvelles applications sur une grande échelle, les solutions réseaux à grande échelle les plus utilisées et les plus étudiées actuellement sont l'Internet d'une part et ATM de l'autre :

– en dépit de sa popularité croissante, l'Internet dans sa version actuelle (IPv4) ne permet pas de prendre en compte de façon satisfaisante les caractéristiques et les contraintes des applications multimédias ; ces dernières années, plusieurs travaux ont été initiés pour remédier à ces lacunes : en particulier, les travaux du groupe *Integrated Service* de l'IETF¹ [i] ont conduit à des propositions d'extension du modèle de service actuel de l'Internet [ii] [iii], ainsi qu'au développement de mécanismes de contrôle de trafic et de protocoles d'établissement de réservation de ressources tels que RSVP notamment [iv] [v] ;

– parallèlement, plusieurs plates-formes ATM ont été déployées en France, notamment au LAAS/CNRS², à l'INRIA³ (Sophia Antipolis) et au LIP6⁴ (Paris) ; l'interconnexion de ces plates-formes a donné naissance en 1997 à la plate-forme MIRIHADÉ (aujourd'hui SAPHIR) sur laquelle ont été effectuées un certain nombre d'expérimentations : tests d'applications multimédias, mesures de performances, etc. [vi].

L'une des activités actuelles en matière de recherche consiste à définir comment fédérer les mondes Internet et ATM dans l'optique de garantir aux applications une QoS de bout en bout indépendante du type de réseau traversé. L'étude présentée dans cet article s'insère dans ce contexte et a été réalisée dans le cadre d'un projet d' "étude et d'évaluation des mécanismes de contrôle de ressources en réseau hétérogène avec application au DIS" (cofinancement MENRT et DGA) dirigé par Dassault Electronique, et dans lequel coopèrent également le LAAS/CNRS, l'INRIA Sophia Antipolis et le LIP6. L'objectif général du projet est de caractériser et de mettre en œuvre la QoS requise par une application de DIS

¹ Internet Engineering Task Force

² Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

³ Institut National de Recherche en Informatique et Automatique

⁴ Laboratoire d'Informatique de Paris 6 (ex MASI)

dans un environnement réseau grande distance hétérogène : Internet nouvelle génération et ATM. Trois phases ont été définies dans le projet :

- l'étude et la caractérisation des besoins d'une application de DIS spécifique pour qu'elle fonctionne en environnement grande distance ;
- l'implantation de cette application dans un environnement réseau local puis grande distance hétérogène avec et sans contrôle de ressources ;
- l'évaluation des améliorations obtenues grâce au contrôle de ressources.

1.2. Contributions et plan de l'article

Les contributions exposées dans cet article sont relatives à la première des phases du projet et sont décomposables en deux parties décrites en sections 3 et 4 : la première partie (section 3) consiste en l'analyse des besoins d'une application DIS et la spécification de la QoS à requérir auprès du réseau pour satisfaire ces besoins ; la deuxième partie (section 4) consiste en une proposition de classification des composantes d'un trafic DIS en fonction de la QoS associée à leur transfert. En amont de ces deux sections sont introduites les caractéristiques majeures des applications DIS ainsi que les principaux thèmes de recherche menés dans ce domaine (section 2). Les conclusions et perspectives de nos travaux sont présentées en section 5 : ces dernières introduisent notamment les principes de l'architecture que nous avons définie pour mettre en œuvre notre proposition de QoS.

2. Les applications de simulation interactive distribuée

2.1. Généralités

La simulation interactive distribuée est issue du domaine militaire, l'objectif initialement visé étant de préparer, en simulation, des unités militaires à toute forme d'engagement (combat, déplacement, etc.) dans des environnements divers (terrestre, maritime ou aérien).

Les principes de base de ce type d'application sont les suivants :

- un *exercice DIS* regroupe un certain nombre de sites (en théorie de quelques uns à plusieurs dizaines) reliés entre eux via un réseau de communication ;
- sur chaque site s'exécutent un ou plusieurs logiciels (ou applications) de simulation, toutes les applications d'un même exercice partageant un même espace virtuel (terrestre, maritime ou aérien) ;
- chaque application simule localement le comportement (déplacement, tir, etc.) d'un certain nombre d'entités (chars, avions, projectiles, etc.) en interaction possible avec des acteurs humains participants à l'exercice : ces

derniers pilotent la simulation en commandant par exemple l'exécution d'un tir ou le changement de direction d'un véhicule ;

- la simulation étant distribuée, chaque application simule également le comportement des entités distantes sur la base d'informations fournies par les applications distantes gouvernant ces entités ; ces informations sont en majorité des informations d'états (ou *vecteurs d'état*) indiquant chacune les position, orientation, vitesse, accélération d'une entité à leur instant d'émission ; dès à présent, on peut concevoir que le transfert de ces informations présente des contraintes de temps réel liées à leur validité temporelle finie ;

- afin de rendre interopérable l'ensemble des applications d'un même exercice, un protocole de niveau applicatif, *le protocole DIS*, a été standardisé IEEE [9] ; ce protocole définit le format et les règles d'échanges des informations échangées entre applications distantes.

La figure 1 illustre l'architecture classique d'une application de simulation interactive distribuée dans un environnement Internet sans réservation de ressources. L'utilisation du protocole UDP est liée à la première des caractéristiques du protocole DIS (décrites ci-après en section 2.2), en l'occurrence la diffusion d'un certain nombre de ses PDUs⁵. D'ores et déjà, notons qu'une telle architecture ne permet ni de garantir un taux de perte nul (ou quasi nul), ni une quelconque borne sur le délai de transit des PDUs.

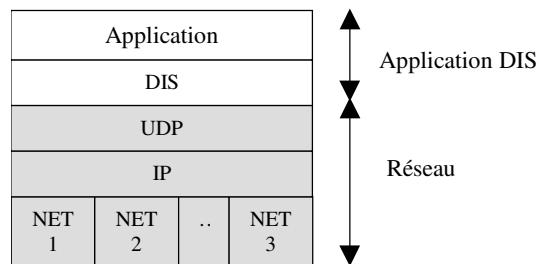


Figure 1 : Architecture de base d'une application de simulation interactive distribuée.

A partir de la section 3, nous désignerons par *application DIS* l'ensemble application / protocole DIS et par *réseau* l'ensemble des couches protocolaires dédiées à la communication. Auparavant, nous précisons les caractéristiques principales du protocole DIS.

2.2. Caractéristiques du protocole DIS

⁵ Protocol Data Unit

La plupart des PDUs émis par un logiciel⁶ DIS s'exécutant sur un site doivent être diffusés à tous les autres logiciels participants à l'exercice ; un service de transport multicast est donc requis. Tel que le stipule le standard IEEE 1278, les PDUs DIS sont actuellement au nombre de 27, regroupés en 6 familles parmi lesquelles :

- la famille des PDUs *Entity Information/Interaction*: elle comporte 2 PDUs, les PDUs *Entity State* (PDUs constituant la très grande majorité du trafic) qui fournissent des informations relatives au vecteur état de chaque entité (position, orientation, vitesse et accélération notamment), et les PDUs *Collision* qui indiquent si une entité est entrée en collision avec un obstacle (l'objectif étant notamment de permettre aux sites distants la mise à jour des dommages subis par l'entité) ;

- la famille des PDUs *Warfare* : elle comporte 2 PDUs indiquant qu'un projectile (balle, missile, ...) a été tiré (PDU *Fire*) et qu'il a éventuellement touché sa cible (PDU *Detonation*) ;

- la famille des PDUs *Simulation Management* : elle comporte 12 PDUs de gestion de la simulation permettant par exemple à une application d'entrer ou de sortir d'un exercice DIS, de créer ou de détruire des entités locales, etc. ;

- enfin, les échanges phoniques entre participants humains sont envisageables par le biais de la famille des PDUs *Radio Communications* (3 PDUs).

La mise à jour de l'état d'une entité repose sur les règles suivantes : à chaque pas de la simulation, le logiciel DIS qui s'exécute sur un site donné génère un PDU *Entity State* (ES) relatif à toute entité simulée localement dont l'état a changé depuis le pas précédent ; recevant un PDU ES, les sites distants réactualisent l'état de l'entité concernée sur la base des informations qu'elle comporte. En réalité, la procédure de mise à jour d'une entité est un peu plus complexe que celle décrite ci-dessus. Elle repose sur la mise en œuvre par tous les logiciels DIS d'un même ensemble d'algorithmes dit de *Dead Reckoning* dont l'objectif est double :

- conduire les sites émetteurs de PDUs ES à diminuer le volume du trafic correspondant à la diffusion des états des entités simulées localement ;

- permettre aux sites récepteurs de déterminer par extrapolation l'état d'une entité distante à chaque pas de la simulation.

2.2.1. Mise à jour d'une entité par *Dead Reckoning*.

La procédure de mise à jour d'une entité par application d'un algorithme de *Dead Reckoning* (DR) est initiée par le logiciel DIS qui s'exécute sur le site de simulation de l'entité concernée :

⁶ Afin d'éviter toute confusion dans l'interprétation du terme "entité", nous employons volontairement les termes "logiciel DIS" en lieu et place des termes "entité de protocole DIS" et nous utiliserons le terme "entité" dans le sens défini en section (2.1) (char, avion, etc.).

– ce dernier maintient, en plus de la représentation réelle, la représentation extrapolée de l'état de chaque entité simulée localement, telle que la détermine les sites distants. Pour une entité donnée, il génère un PDU ES dès que l'une des deux conditions suivantes est vérifiée :

– l'écart entre l'état extrapolé et l'état réel de l'entité excède l'un des seuils définis dans le standard (Th_{Pos} pour la position ou Th_{Or} pour l'orientation) ;

– l'état de l'entité n'a pas été rafraîchi depuis un temps égal au *HEART BEAT TIMER* (battement cardiaque en anglais) de la simulation, généralement cinq secondes tel que spécifié par défaut dans le standard ;

– sur tous les autres sites, la réception par DIS d'un PDU ES engendre la mise à jour de l'état de l'entité concernée ; plus précisément :

– si l'on ne dispose que de la position \mathbf{P}_i et de la vitesse \mathbf{V}_i de l'entité à l'instant t_i , il est possible d'estimer sa position $\mathbf{P}_{DR}(t)$ à tout instant $t > t_i$, grâce à une extrapolation linéaire exprimée par la relation (1) :

$$(1) \quad \mathbf{P}_{DR}(t) = \mathbf{P}_i + \mathbf{V}_i \cdot (t - t_i)$$

– si l'on dispose de la position \mathbf{P}_i , de la vitesse \mathbf{V}_i et de l'accélération \mathbf{A}_i à l'instant t_i , on peut alors estimer la position, $\mathbf{P}_{DR}(t)$, de l'entité à tout instant $t > t_i$ grâce à une extrapolation quadratique exprimée par la relation (2) :

$$(2) \quad \mathbf{P}_{DR}(t) = \mathbf{P}_i + \mathbf{V}_i \cdot (t - t_i) + \frac{1}{2} \mathbf{A}_i \cdot (t - t_i)^2$$

– enfin, il est possible de compléter les extrapolations sur la position données en (1) ou (2) par une extrapolation sur l'orientation (relation 3) ; en effet, si l'on connaît la matrice de passage \mathbf{R}_i (qui détermine l'orientation de l'entité) et le vecteur vitesse de rotation angulaire $\mathbf{\Omega}_i$ à l'instant t_i , l'extrapolation linéaire permet d'évaluer la matrice de passage $\mathbf{R}_{DR}(t)$ à tout instant $t > t_i$:

$$(3) \quad \mathbf{R}_{DR}(t) = \mathbf{R}_i \cdot e^{\left((t - t_i) \cdot \hat{\mathbf{\Omega}}_i \right)} \quad \text{où } \hat{\mathbf{\Omega}}_i \text{ représente la matrice associée au produit vectoriel par } \mathbf{\Omega}_i$$

2.3. Principaux travaux de recherche menés autour des applications DIS

Plusieurs thèmes de recherche sont à l'heure actuelle approfondis dans le domaine du DIS ; nous n'en citons que quelques uns parmi les principaux. Tel qu'il est utilisé par les applications DIS, le schéma de communication multicast (i.e. la diffusion) conduit à une utilisation massive des ressources de communication. En vue d'envisager le déploiement de ces applications à une échelle non plus locale mais grande distance (impliquant plusieurs milliers de participants), l'un des thèmes de recherche les plus approfondis concerne les problèmes du facteur d'échelle [vii] [viii]. Outre les travaux relatifs au *Dead Reckoning*, plusieurs techniques ont été proposées afin de réduire la bande passante utilisée lors d'un

exercice DIS. Celles-ci peuvent être réparties en deux groupes : les techniques d'agrégation et les techniques de filtrage. Concernant le premier groupe, [ix] propose par exemple l'agrégation des informations relatives à plusieurs entités dans un seul PDU. Une autre proposition d'agrégation consiste en rassembler plusieurs PDUs DIS dans un seul message UDP [x]. Les techniques de filtrage ont pour objectif d'assurer la délivrance des PDUs DIS selon plusieurs niveaux de fidélité [17] [xi], fonction du degré de couplage des entités émettrice et réceptrice. Par exemple, les PDUS ES relatifs à une entité située derrière une montagne ne sont pas essentielles pour les entités situées de l'autre côté de la montagne. Poursuivant le même objectif dans le domaine des jeux distribués, [xii] propose des mécanismes permettant de jouer l'application *MiMaze* sur Internet.

Un autre thème d'étude concerne la définition d'une architecture (HLA/RTI⁷) destinée (notamment) à faciliter l'interopérabilité des simulateurs et à permettre la mise en œuvre de simulations autres que militaires (extension (extension de la liste des PDUs, ...) [xiii] ; en outre, cette architecture vise à répondre aux limites de l'architecture actuelle dans un environnement réseau grande distance.

Enfin, un dernier thème de recherche concerne la mesure du trafic DIS. Si plusieurs campagnes de mesures ont été publiées ces dernières années, [17] [xiv] [xv] notamment, elles ne permettent cependant pas de caractériser un trafic DIS en s'appuyant sur un modèle formel.

3. Caractérisation de la QoS requise par une application DIS

Ces dernières années, de nombreux travaux de recherche ont porté sur l'analyse des besoins des applications multimédias (manipulant en particulier l'audio et la vidéo) et ont conduit à plusieurs propositions de QoS [xvi] [xvii] [xviii] [xix] ; cependant, aucune de ces propositions ne vise explicitement les applications DIS. L'objectif de cette section est de caractériser la QoS qu'une application DIS doit requérir auprès du réseau pour s'exécuter "correctement" au regard de critères de fonctionnement que nous explicitons ci-après.

3.1. Caractérisation de la QoS requise par une application DIS

Par définition, une application DIS présente un certain nombre de contraintes liées à sa nature distribuée et interactive ; par exemple, la mise à jour de la position d'une entité sur son site de simulation doit être propagée aux autres sites en un temps suffisamment court pour que sur chacun des sites, les acteurs humains (ou participants) aient une même vue de l'entité. Nous analysons maintenant plus en détails ces différentes contraintes.

De façon générale, le trafic généré par une application de DIS est décomposable en trois parties :

⁷ High Level Architecture / Real Time Infrastructure

- le trafic correspondant à la diffusion : des états des entités simulées (PDUs Entity State notamment) et des événements (PDUs Tir, Detonation, Collision, etc.) ;
- le trafic correspondant à la communication entre participants et aux échanges entre capteurs : communications radio (PDUs *Signal*), interactions électromagnétiques (PDUs *EM-emission*) ;
- le trafic correspondant à la gestion de la simulation distribuée : initialisation d'une phase de simulation, synchronisation des participants, bilan (PDUs *Start, Resume, etc.*).

En conséquence, les contraintes requises auprès du réseau sont les suivantes :

- le réseau doit d'une part être capable de supporter l'ensemble du trafic, c'est à dire d'écouler le débit correspondant en tenant compte de ses variations ;
- d'autre part, il doit également assurer un acheminement des PDUs DIS *suffisamment rapide* et *suffisamment fiable* pour que les participants aient une vision *cohérente* de la simulation.

Les aspects *suffisamment rapide* et *suffisamment fiable* des garanties que doit assurer le réseau font référence de façon volontairement informelle au délai de transit et à la fiabilité du transfert des PDUs DIS. Afin d'en donner la sémantique exacte et de quantifier la valeur de ces deux paramètres, nous précisons tout d'abord la notion de *cohérence* introduite ci-dessus.

3.1.1. *Notion de cohérence*

Par définition, une application DIS est distribuée entre différents sites possédant chacun à tout instant une représentation *cohérente* : (1) de l'état de toute entité simulée : position, orientation, etc. et (2) des événements intervenus : collision, détonation, etc. Cette cohérence peut être considérée sous deux aspects que nous détaillons ci-après : (1) un aspect *spatial* relatif aux états des entités ; (2) un aspect *temporel* relatif aux événements.

Notion de cohérence temporelle

Du point de vue d'un site, avoir une vue *temporellement cohérente* du système consiste à connaître l'occurrence de tout événement intervenu sur un autre site avec un retard borné. Par exemple, le standard DIS permet de déduire que l'occurrence d'un tir sur un site donné doit être connue par tous les autres sites directement concernés avec un retard d'au plus 100 millisecondes (ms) : cette garantie est nécessaire pour assurer une vue temporellement cohérente du système par tous les sites impliqués dans l'exercice en cours.

Notion de cohérence spatiale

Du point de vue d'un site, avoir une vue *spatialement cohérente* du système consiste à connaître à chaque instant l'état de toutes les entités du système "à une certaine erreur près". Plus concrètement, considérons l'exemple d'une entité A représentant un avion, simulée sur un site S_1 . A un instant donné, un site S_2 a une vue spatialement cohérente de l'état de A si l'écart entre la connaissance qu'il a de l'état de A et l'état réel de A (i.e. celui simulé sur le site S_1) n'excède pas un certain seuil, défini dans le standard DIS comme étant l'erreur maximale acceptable sur la position, Th_{Pos} , et sur l'orientation, Th_{Or} .

La figure 2 illustre sur un même plan les positions et orientations de l'entité A sur son site de simulation S_1 (avion de gauche) et sur le site S_2 (avion de droite). Tant que E_p et E_o n'excèdent pas leur seuil respectif, S_1 et S_2 ont une vue spatialement cohérente de l'état de A. Au delà de l'un de ces seuils, il est nécessaire de réactualiser sur S_2 l'état de A à partir d'une information d'état (typiquement un PDU ES) fournie par S_1 .

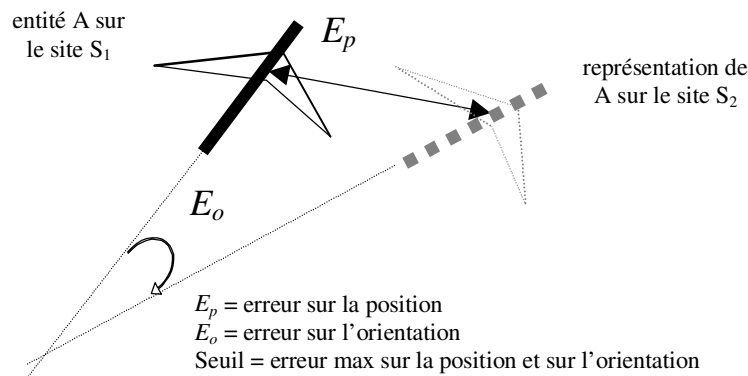


Figure 2 : Position et orientation d'une entité.

La notion de cohérence ayant été explicitée, nous analysons à présent la manière dont le standard DIS définit la QoS à requérir auprès du réseau pour satisfaire les besoins des applications de simulation interactives distribuées. Au travers de cette analyse, nous montrons en quoi cette expression ne permet pas de satisfaire les besoins en termes de cohérence spatiale d'une application DIS, dès lors que l'on se place dans un environnement réseau grande distance. Nous formulons dans un second temps une proposition de QoS remédiant à ce problème.

3.1.2. Expression et limites de la QoS définie dans le standard DIS

Dans le standard DIS [7], [8], [9], les besoins en communication d'une application DIS sont exprimés de la façon suivante :

- l'application doit disposer d'un service de transport multicast permettant à chaque site de joindre tous les autres sites impliqués dans l'exercice en cours ;

- la QoS du transport des PDU DIS est exprimée au moyen de trois paramètres :

- la fiabilité, définie comme étant le pourcentage maximum de pertes admissibles de PDUs ;

- le délai maximum d'acheminement de ces PDUs ;

- la variation maximale de ce délai par rapport à une valeur moyenne (en d'autres termes, la gigue) pour l'ensemble des PDUs relatifs aux communications radio.

Les valeurs des paramètres du second item dépendent du *couplage* des entités auxquelles sont associés les PDUs, couplage dont nous rappelons à présent le principe.

3.1.2.1. Notion de couplage

La notion de couplage est spécifique au standard DIS qui distingue deux situations, le *couplage faible* et le *couplage fort*, illustrés comme suit. Deux chars distants de 5 kilomètres en terrain découvert constituent un exemple d'entités *faiblement couplées*. Plusieurs chars en formation évoluant rapidement lors d'une manœuvre compliquée constituent un exemple d'entités *fortement couplées*.

A partir de cette formulation, le standard définit la QoS à associer au transfert des PDU ES de façon binaire ; les PDU ES relatifs à une entité *E* simulée sur un site donné doivent être transportés avec :

- un délai inférieur ou égal à 100 millisecondes (ms) et un taux de perte inférieur ou égal à 2 % si *E* est fortement couplée avec au moins une autre entité ;

- un délai inférieur ou égal à 300 ms et un taux de perte inférieur ou égal à 5% si *E* est faiblement couplée avec au moins une autre entité.

Nous analysons maintenant les limites de la notion de couplage.

3.1.2.2. Limites de la notion de couplage

Telle qu'elle est définie dans le standard, la notion de couplage présente au moins trois limites.

(L1) De part son imprécision, la notion de couplage est ambiguë et laisse libre cours à des interprétations différentes au niveau de chaque site⁸.

(L2) Sa prise en compte apparaît difficile pour deux raisons majeures :

– elle implique que chaque site évalue, pour toute entité simulée localement, si celle-ci est en fort ou en faible couplage avec chacune des entités distantes. En cela, la notion de couplage (1) apparaît coûteuse à mettre en œuvre en terme de temps de calculs et (2) ne permet pas de déterminer la QoS au seul regard de la nature des PDU ES, puisqu'elle dépend des couples (site émetteur, site récepteur) ;

– la nature dynamique d'un exercice DIS engendre l'évolution en temps réel du couplage entre deux entités : en d'autres termes, deux entités faiblement couplées à un instant donné peuvent devenir fortement couplées un instant plus tard ; en conséquence, on ne peut déterminer statiquement l'ensemble des relations liant les différentes entités d'un exercice DIS, ce qui accentue le coût en temps de calcul évoqué précédemment.

(L3) La QoS associée à la notion de couplage ne permet pas de prendre en compte l'influence du délai de transit sur l'erreur de position / orientation : en cela, elle ne répond pas aux besoins applicatifs tels que nous les avons exprimés en terme de cohérence spatiale. Nous illustrons maintenant plus en détails cette troisième limite.

Illustration de la limite (L3)

Considérons l'exemple suivant. : soient deux sites S_e et S_r appartenant à un même exercice DIS. On s'intéresse à une entité A simulée sur S_e et dont l'état est mis à jour sur S_r par application d'un algorithme de *Dead Reckoning* (DR) selon la procédure décrite en section (2.2.1). Pour simplifier la présentation qui suit, l'état de l'entité A sera réduit à la seule position de A et l'on supposera en outre que son mouvement est rectiligne (l'état est alors représenté avec une seule dimension).

La figure 3 illustre sur un même schéma l'évolution de l'erreur d'extrapolation E faite sur la position de A en fonction du temps, respectivement sur le site S_e (partie supérieure de la figure) et sur le site S_r (partie inférieure de la figure). On suppose que les deux sites ont une même référence de temps ; enfin, on rappelle qu'ils utilisent le même algorithme de DR et que seul le site émetteur S_e est en mesure de calculer effectivement E .

Les pastilles noires de la partie supérieure de la figure 3 correspondent aux dates d'émission des PDUs ES rafraîchissant l'état de A . Sur cette figure, on

⁸ Si deux sites A et B simulent respectivement deux entités E_A et E_B , A peut considérer que E_A et E_B sont fortement couplées et B qu'elles ne le sont que faiblement. A requerra donc une QoS plus exigeante pour informer B de l'état de E_A , alors que B considère cette information comme peu critique. Inversement B requerra une QoS moindre pour informer A de l'état de E_B alors que A considère sa mise à jour comme urgente.

suppose que l'origine des temps correspond à une date de rafraîchissement (T_{e0}).

Les pastilles grises de la partie inférieure de la figure 3 correspondent aux dates de réception des PDU ES rafraîchissant l'état de A. Sur la figure, les dates T_{r0} , T_{r1} , T_{r2} et T_{r3} correspondant aux dates de réception des PDU ES émises aux dates T_{e0} , T_{e1} , T_{e2} et T_{e3} décrites précédemment.

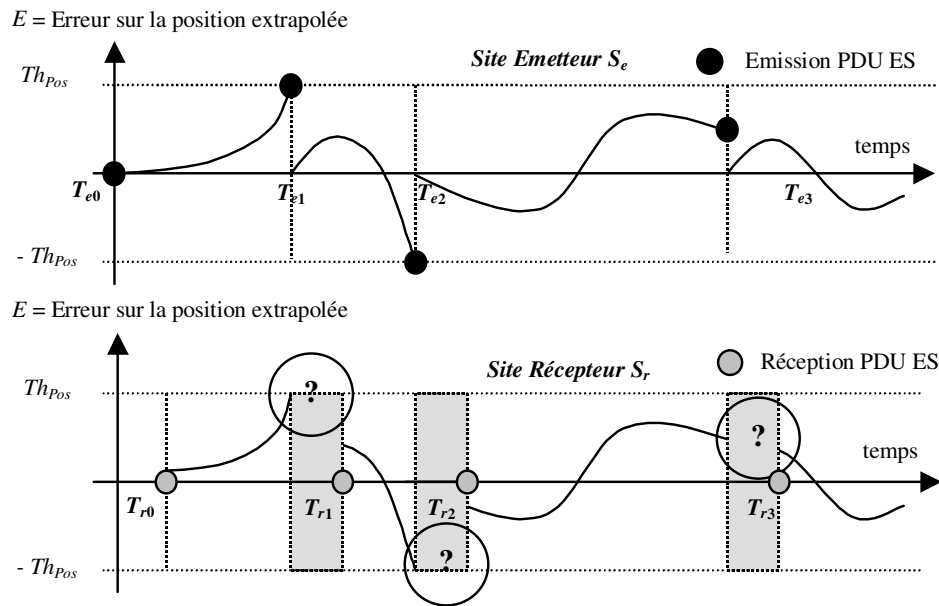


Figure 3 : Evolution de l'erreur d'extrapolation sur la position d'une entité sur l'émetteur sur le récepteur

Analisons les dates T_{e1} , T_{e2} et T_{e3} :

- à partir de T_{e0} , l'erreur faite entre la position extrapolée de A et sa position réelle augmente et atteint la valeur maximale admissible (Th_{Pos} , correspondant au seuil de DR sur la position) à la date T_{e1} : l'application du mécanisme de DR engendre alors l'émission d'un PDU ES indiquant (entre autre) la position réelle de A ; l'erreur E redevient nulle ;

- à partir de T_{e1} , l'erreur faite entre la position extrapolée de A et sa position réelle augmente en valeur absolue et atteint le seuil de DR ($-Th_{Pos}$) à la date T_{e2} : l'application du mécanisme de DR engendre alors l'émission d'un PDU ES indiquant la position réelle de A ; l'erreur E redevient nulle ;

- à partir de T_{e2} , l'erreur faite entre la position extrapolée de A et sa position réelle oscille entre $-Th_{Pos}$ et Th_{Pos} sans jamais dépasser l'une ou l'autre de ces valeurs ; au bout du HEART BEAT TIMER (cf. section 2.2.1), soit à $T_{e3} = T_{e2} + 5$ secondes (par défaut), l'application du mécanisme de DR engendre automatiquement l'émission d'un PDU ES indiquant la position réelle de A ; l'erreur E redevient nulle.

Analysons à présent l'évolution de l'erreur E faite sur le site S_r :

- à partir de la date origine T_{e0} , l'erreur E faite entre la position extrapolée de A et sa position réelle est identique à celle faite sur S_e ; en particulier, elle atteint la valeur maximale admissible Th_{Pos} à T_{e1} , date d'émission par S_e du PDU ES rafraîchissant A . Cependant, la mise à jour de la position de A n'est effectuée qu'à la date T_{r1} , l'intervalle de temps séparant T_{e1} et T_{r1} correspondant au délai de transit du PDU ES au travers du réseau. En conséquence, il apparaît une indétermination quant à la valeur de E entre les dates T_{e1} et T_{r1} , période durant laquelle E peut notamment excéder la valeur Th_{Pos} , engendrant par la même une violation de la cohérence spatiale, telle que définie en section (3.1.1) ;

- de façon analogue, on met en évidence une absence de maîtrise de E entre les dates T_{e2} et T_{r2} et entre les dates T_{e3} et T_{r3} , matérialisée sur la figure par les zones grisées.

De cette analyse, il résulte donc que la définition actuelle de la QoS (au travers de la notion de couplage) garantit le respect de la cohérence spatiale sauf éventuellement dans tous les intervalles de temps $[T_e, T_r]$ qui séparent la date d'émission d'un PDU ES de sa date de réception : durant ces intervalles, l'erreur E peut excéder en valeur absolue et de façon transitoire la valeur maximale admissible. La figure 4 illustre un tel cas de figure.

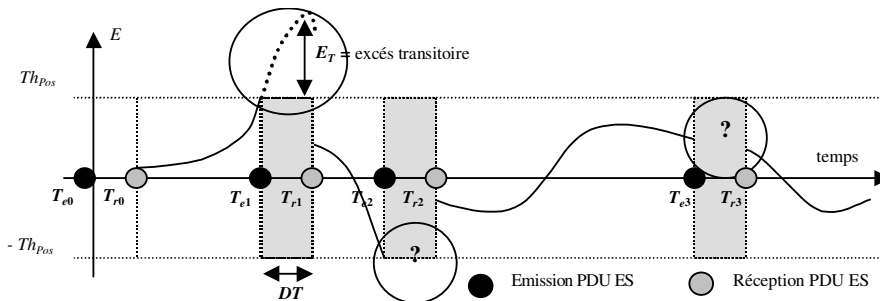


Figure 4 : Illustration de l'excès transitoire

Deux questions se posent alors : (1) l'absence de garantie de cohérence spatiale est-elle préjudiciable au déroulement correct d'un exercice DIS ? (2) si oui, est-il envisageable de remédier à ce problème ; autrement dit, est-il possible de maîtriser l'excès transitoire E_T illustrée sur la figure 4 ?

Concernant la première question, l'excès transitoire potentiellement observable devient préjudiciable dès lors que le délai de transit (DT sur la figure 4) est non négligeable au regard de l'intervalle de temps séparant deux réceptions consécutives de PDU ES relatifs à la même entité. Si tel est effectivement le cas, il en résulte que le (les) site(s) récepteur(s) a (ont) une vue spatialement incohérente de l'état de l'entité durant des périodes de temps non négligeables, au pire

“ presque tout le temps ” si la période de rafraîchissement d'une entité est négligeable devant le délai de transit des PDU ES qui lui sont associés ; naturellement, ce risque, inexistant en environnement réseau local, s'accroît lorsque l'on considère la mise en œuvre d'un exercice DIS dans un environnement réseau grande distance, tel que celui du contexte de notre étude. Concernant la deuxième question et compte tenu de la réponse précédente, nous formulons maintenant une proposition de QoS visant à maîtriser l'excès transitoire ; nous présentons ensuite le moyen de mettre en œuvre cette maîtrise.

3.1.3. Proposition de QoS

La proposition de QoS que nous formulons repose sur les points suivants :

- l'application doit tout d'abord disposer d'un service de transport multicast ;
- la QoS du transport des PDU DIS est exprimée au moyen des trois mêmes paramètres que ceux définis dans le standard :
 - la fiabilité, définie par le pourcentage maximum de pertes admissibles de PDUs (une fiabilité totale signifiant 0 % de perte tolérable ; une fiabilité nulle signifiant 100 % de perte tolérable) ;
 - le délai maximum d'acheminement de ces PDUs (éventuellement infini) ;
 - la variation maximale de ce délai (gigue) pour l'ensemble des PDUs relatifs aux communications radio ;

A la différence de la proposition du standard, la QoS ici proposée implique :

- l'abandon de la notion de couplage ;
- la détermination de la valeur des paramètres précédents :
 - à partir des seules informations contenues dans chaque PDU : en cela, cette proposition remédie à la limite (L2) ;
 - de façon à garantir à tout instant une erreur maximale, en particulier durant les périodes transitoires précédemment décrites : en cela, cette proposition remédie à la limite (L3).

3.1.4. Mise en œuvre de la garantie d'une erreur maximale

Tel que nous l'avons démontré dans la section précédente, l'erreur entre la position (resp. l'orientation) effective et la position (resp. l'orientation) extrapolée sur un site récepteur n'est inférieure au seuil Th_{Pos} (resp. Th_{Or}) que dans les intervalles de temps $[t+DT, t_{i+1}]$ ⁹. Dans les intervalles $[t_i, t+DT]$, l'erreur peut de ma-

⁹ Dans toute cette section, les instants t_i représentent les instants successifs d'émission des PDUs ES et les réceptions ont lieu aux dates successives $t+DT$ (avec DT supposé constant). Sur la figure 3 par exemple, les instants t_i correspondent donc aux dates successives $T_{e0}, T_{e1}, T_{e2}, T_{e3}, \dots$ et les instants $t+DT$ aux dates successives $T_{r0}, T_{r1}, T_{r2}, T_{r3}, \dots$

nière transitoire dépasser le seuil Th_{Pos} (resp. Th_{Or}) et conduire à une incohérence.

Etudions à présent comment majorer l'excès transitoire E_T de la figure 4. La seule façon de garantir une borne supérieure E_{Tmax} à l'excès transitoire E_T est de disposer d'une garantie de la part du réseau que le délai de transit DT des PDU ES n'excédera pas une valeur DT_{max} (figure 5).

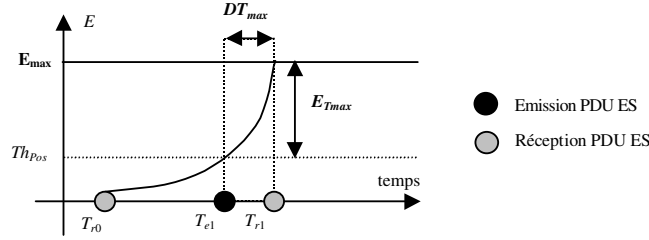


Figure 5 : Garantie d'un excès transitoire maximal

Intuitivement, on conçoit que l'excès transitoire E_T dépend de la dynamique (accélération) de l'entité simulée et qu'en conséquence, le délai de transit maximal permettant de garantir une valeur E_{Tmax} pour E_T dépend également de la dynamique de l'entité. Analysons ceci de manière plus formelle. L'influence du délai de transit DT sur la connaissance de l'état des entités par chaque site est liée à l'expression de l'écart en valeur absolue $e_P(t)$, entre la position effective $\mathbf{P}_a(t)$ et la position extrapolée $\mathbf{P}_{DR}(t)$ pour une entité quelconque (relation 4) :

$$(4) \begin{cases} \mathbf{P}_a(t) = \int_{t_i}^t du \int_{t_i}^u \mathbf{A}_a(\tau) \cdot d\tau \\ e_P(t) = \|\mathbf{P}_a(t) - \mathbf{P}_{DR}(t)\| = \left\| \int_{t_i}^t du \int_{t_i}^u [\mathbf{A}_a(\tau) - \mathbf{A}_i] \cdot d\tau \right\| \end{cases}$$

où $\mathbf{A}_a(t)$ représente l'accélération effective de l'entité à l'instant t et \mathbf{A}_i la valeur de l'accélération utilisée par l'algorithme de DR (valeur éventuellement nulle pour la relation (1)).

Sous l'hypothèse que l'entité possède une accélération bornée, $\|\mathbf{A}_a(t)\| \leq A_{max}$, $e_P(t)$ possède alors une borne supérieure dans l'intervalle de temps $[t_{i+1}, t_{i+1}+DT]$.

Suivant le principe du DR, $e_P(t)$ est inférieur ou égal au seuil Th_{Pos} pour tout $t \leq t_{i+1}$; on peut donc majorer $e_P(t_{i+1}+DT)$ par la somme de trois termes (relation 5) :

$$(5) \quad e_P(t_{i+1} + DT) \leq \left\| \int_{t_i}^{t_{i+1}} du \int_{t_i}^u [\mathbf{A}_a(\tau) - \mathbf{A}_i] \cdot d\tau \right\| + \left\| \int_{t_{i+1}}^{t_{i+1}+DT} du \int_{t_i}^{t_{i+1}} [\mathbf{A}_a(\tau) - \mathbf{A}_i] \cdot d\tau \right\| + \left\| \int_{t_{i+1}}^{t_{i+1}+DT} du \int_{t_{i+1}}^u [\mathbf{A}_a(\tau) - \mathbf{A}_i] \cdot d\tau \right\|$$

– le premier terme est évidemment majoré par le seuil Th_{Pos} ;

– le troisième terme est majoré par $(\frac{1}{2}) A_{max} \cdot DT^2$ pour une extrapolation linéaire (relation (1)) et par $A_{max} \cdot DT^2$ pour une extrapolation quadratique (relation(2)) ;

– il reste donc à majorer le second terme, en fait égal au produit de DT par une constante qui représente la différence entre la vitesse effective et la vitesse extrapolée à l'instant t_{i+1} où est émis le PDU ES (relation 6) :

$$(6) \quad DT \cdot \int_{t_i}^{t_{i+1}} [A_a(\tau) - A_i] \cdot d\tau = [(V_{i+1} - V_i) - A_i \cdot (t_{i+1} - t_i)] \cdot DT = (V_a(t_{i+1}) - V_{DR}(t_{i+1})) \cdot DT$$

Cette différence est maximale (en valeur absolue) dans le cas où la différence entre l'accélération effective et l'accélération extrapolée est elle-même maximale et constante sur tout l'intervalle $[t_i, t_{i+1}]$. Dans ces conditions, le second terme de la relation (5) est majoré par $\sqrt{2A_{max} \cdot Th_{Pos}} \cdot DT$ pour une extrapolation linéaire (relation (1)) et par $2\sqrt{A_{max} \cdot Th_{Pos}} \cdot DT$ pour une extrapolation quadratique (relation(2)).

En conclusion, on obtient l'expression de la borne supérieure de l'erreur à tout instant par la relation (7) :

$$(7) \quad e_p(t) \leq \begin{cases} \left(\sqrt{Th_{Pos}} + DT \cdot \sqrt{\frac{1}{2} A_{max}} \right)^2 & \text{pour une extrapolation linéaire} \\ \left(\sqrt{Th_{Pos}} + DT \cdot \sqrt{A_{max}} \right)^2 & \text{pour une extrapolation quadratique} \end{cases}$$

Si l'on souhaite que l'erreur transitoire E_T soit inférieure à une valeur maximale E_{Tmax} , on doit alors garantir pour chaque PDU ES un délai de transit de bout en bout DT_{max} dont la valeur est fonction de l'accélération maximale de l'entité associée (relation (8)) :

$$(8) \quad DT_{max} \leq \begin{cases} \frac{\sqrt{Th_{Pos} + E_{Tmax}} - \sqrt{Th_{Pos}}}{\sqrt{0,5 \cdot A_{max}}} & \text{pour une extrapolation linéaire} \\ \frac{\sqrt{Th_{Pos} + E_{Tmax}} - \sqrt{Th_{Pos}}}{\sqrt{A_{max}}} & \text{pour une extrapolation quadratique} \end{cases}$$

Il est ainsi possible d'assurer la cohérence spatiale en garantissant la position de toute entité à l'intérieur d'une boule d'incertitude de rayon $Th_{Pos} + E_{Tmax}$, à condition de garantir un délai de transit des PDUs ES inférieur ou égal à DT_{max} . Remarquons enfin que pour une même précision souhaitée, $Th_{Pos} + E_{Tmax}$, on peut faire varier la valeur de DT_{max} :

– si l'on diminue le seuil Th_{Pos} , DT_{max} augmente et les contraintes temporelles sont plus faciles à satisfaire ; en contre partie, la mise à jour des PDUs ES risque d'être plus fréquente, ce qui augmente le débit ;

– si l'on utilise pour le DR une interpolation non plus linéaire mais quadratique, les contraintes sur DT_{max} sont accrues (DT_{max} diminue de 30% environ), mais la fréquence moyenne de mise à jour des PDUs devrait diminuer, l'extrapolation étant améliorée par la prise en compte de l'accélération.

Sur les bases de la proposition de QoS formulée dans cette section, nous proposons maintenant une répartition des PDUs DIS en classes de QoS, en précisant la sémantique et la valeur des paramètres de chacune de ces classes.

4. Classification des PDUs DIS

Rappelons que la classification qui suit a pour but de permettre la conception d'une architecture de communication reposant sur la mise en œuvre de canaux de communication à QoS spécifique, répondant chacun aux besoins en terme de transfert d'une classe de PDUs, indépendamment du couplage des entités représentées par ces PDUs.

La répartition que nous proposons comporte quatre classes majeures, la quatrième de ces classes étant raffinées en sous-classes :

- la classe des PDUs “ événementiels ” ;
- la classe des PDUs de gestion de la simulation ;
- la classe des PDUs *Signal* ;
- la classe des PDUs “ continus ”.

Classe des PDUs “ événementiels ” : Cette classe regroupe les PDUs *Collision*, *Fire* et *Détonation* pour lesquels sont requis :

– une fiabilité totale ; cette qualité de service est jugée nécessaire du fait de l'impossibilité de récupérer l'information attachée aux PDUs concernés en cas de perte par le réseau ; en effet, le caractère événementiel des PDUs *Collision*, *Fire* et *Détonation* permet difficilement d'en extrapoler les occurrences ;

– un délai maximum lié au degré d'interactivité souhaitée ; 100 ms correspondant à une forte interaction, c'est cette valeur que nous avons gardée compte tenu de la sémantique des PDUs concernés.

Classe des PDUs de gestion de la simulation : Cette classe regroupe les PDUs de gestion de la simulation (*Start*, *Stop*, ...) pour lesquelles sont requises :

– une fiabilité totale ; en effet, les règles d'échange de ces PDUs définies dans le standard DIS ne précisent aucunement l'action à entreprendre par les logiciels DIS en cas de perte d'une au moins des PDUs de cette classe : implicitement, ces règles s'appuient sur l'hypothèse d'un réseau fiable et sans déséquilibrage ;

– un délai maximum non borné, résultant du fait que les échanges de PDUs de cette classe ne présentent pas de contrainte temporelle spécifique.

Classe du PDU “ Signal ” : Cette classe ne contient qu'un seul PDU : le PDU *Signal* pour lequel sont requis :

- une fiabilité “ élevée ” ; ce paramètre, quantifié de façon volontairement “ ouverte ”, traduit le fait qu'un transfert audio peut présenter diverses contraintes de fiabilité selon le codage utilisé ;
- un délai maximum de 250 ms, correspondant au délai jugé admissible pour une application d'audioconférence distribuée en temps réel ;
- une gigue de 50 ms, correspondant à la gigue jugée admissible pour une application d'audioconférence.

Classe des PDUs “ continus ” : Cette classe regroupe principalement les PDUs *Entity State*, mais également les PDUs *Transmitter* et *Designator* pour lesquels sont requis :

- une fiabilité totale ;
- un délai maximum fonction de l'accélération de l'entité correspondante ; en conséquence sont définies plusieurs sous-classes de cette classe que nous présentons dans le tableau de la figure 5.

Domaine de A_{max}	DT_{max} garanti
$[0, A_1]$	DT_1
$[A_1, A_2]$	DT_2
...	...
$[A_{n-1}, A_n]$	DT_n

Figure 5 : Répartition des entités en sous-classes de QoS.

A chaque plage d'accélération maximale $[A_{i-1}, A_i]$ est associée un délai de transit DT_i dont la valeur correspond au délai maximal déduit de la formule (8) qui garantit le respect une erreur transitoire maximale E_{Tmax} pour les entités d'accélération maximale comprise entre A_{i-1} et A_i .

5. Conclusions et perspectives

Les travaux qui ont été présentés dans cet articles sont relatifs à un projet d'étude et d'évaluation de mécanismes de contrôle de ressources en réseau hétérogène avec application au DIS. Plus précisément, les contributions ici fournies portent sur la caractérisation de la QoS nécessaire au fonctionnement des applications de simulation interactive distribuées en environnement réseau grande distance.

Dans un premier temps (section 2), nous avons introduit les principes de base des applications DIS en soulignant la difficulté d'en caractériser le trafic et la QoS.

Ensuite (section 3), nous avons exposé la première partie de nos travaux : la caractérisation de la QoS requise par une application DIS dans un environnement réseau grande distance. Ce travail nous a conduit à démontrer les limites de la définition actuelle de la QoS, telle qu'elle est décrite dans le standard DIS. Trois limites ont été mises en évidence dont les deux principales sont les suivantes :

- telle qu'elle est définie dans le standard, la QoS ne permet pas d'envisager la conception d'une architecture de transport reposant sur une approche du type " un canal à QoS spécifique par flux utilisateur, fonction des caractéristiques et des contraintes du flux véhiculé " ;

- la deuxième limite concerne l'absence de garantie à tout instant d'une borne supérieure de l'erreur existant entre l'état extrapolé d'une entité et son état réel ; nous avons souligné le caractère problématique de cette limite en environnement réseau grande distance en montrant que l'état extrapolé d'une entité pouvait différer (de façon inadmissible au regard du standard) de son état réel de façon quasi permanente.

Sur les bases de cette analyse, nous avons formulé une proposition de QoS visant à remédier aux limites précédentes. A la différence de celle définie dans le standard DIS, cette proposition préconise l'abandon de la notion de couplage entre deux entités et permet de déterminer la valeur des paramètres de QoS (fiabilité, délai et gigue) :

- à partir des seules informations contenues dans chaque PDU ;
- de façon à garantir à tout instant une erreur maximale existant entre l'état extrapolé d'une entité et son état réel.

La deuxième partie de notre étude (exposée en section 4) a consisté à proposer une répartition des PDUs DIS en classes de QoS, correspondant chacune à un besoin particulier en communication. Cette classification nous a conduit à distinguer quatre classes précisant toutes la sémantique et la valeur des paramètres de QoS identifiés dans notre premier travail. En particulier, la quatrième de ces classes (regroupant notamment les PDUs *Entity State*) a été décomposée en sous-classes, fonction chacune d'un intervalle d'accélération maximale.

Les perspectives de ces travaux sont de deux types :

- le premier est lié à la mise en œuvre des classes de QoS exposées dans cet article ; cette perspective correspond aux travaux que nous avons ou que nous sommes en train de réaliser dans le cadre du projet, et dont nous introduirons ci-après les éléments principaux.

- le second consiste à quantifier l'impact pour l'application, d'une fiabilité non plus totale mais partielle du réseau, et d'évaluer ses bénéfices en termes d'utilisation des ressources du réseau ; en effet, la classification que nous avons proposée impose la mise en œuvre d'une architecture de communication garantissant une fiabilité totale, l'objectif visé étant de garantir le respect d'une borne supérieure sur l'erreur transitoire définie en section 3.

Sans trop entrer dans les détails (faute de place), nous présentons maintenant les éléments essentiels liés à la mise en œuvre des classes de QoS (premier des points précédents). Pour mettre en œuvre les classes définies et la QoS correspondante, trois besoins ont été identifiés :

- la disponibilité d'un réseau à réservation de ressources ;
- la caractérisation du trafic DIS généré par un site afin de dimensionner les ressources à réserver ;
- la spécification d'une architecture de bout en bout.

Concernant le premier de ces besoins, les choix effectués dans le cadre du projet nous ont conduit à considérer l'Internet nouvelle génération au travers des modèles de service définis par l'IntServ (*le Guaranteed Service* et *le Controlled Load*) et du protocole RSVP.

Sur ces bases, le second besoin nous a conduit à définir et à réaliser une méthodologie permettant de caractériser un trafic DIS suivant le modèle du *Token Bucket* (utilisé par RSVP via les paramètres du *Tspec*). Cette méthodologie permet d'associer des profils (débit moyen et débit crête) aux différentes *morphologies*¹⁰ identifiées dans un exercice DIS, les entités de même morphologie étant supposées générer un trafic similaire¹¹. De ce résultat, il est alors possible de caractériser, en les termes du *Tspec* de RSVP, le trafic généré par un site sur la base d'informations simples de niveau applicatif : morphologie et nombre des entités simulées localement. Notons que cette méthodologie est applicable à tout exercice DIS ainsi qu'à de nouvelles entités.

Enfin, le troisième besoin nous a conduit à spécifier une architecture de bout en bout (figure 6) reposant sur les principes suivants :

- établissement d'un canal multicast à QoS spécifique par classe de QoS ;
- choix du *Fixed Filter* comme style de réservation RSVP ;
- processus de réservation en deux temps (suivant le modèle RSVP) :
 - (1) spécification du profil de trafic émis par chaque émetteur ;
 - (2) réservation par les récepteurs.

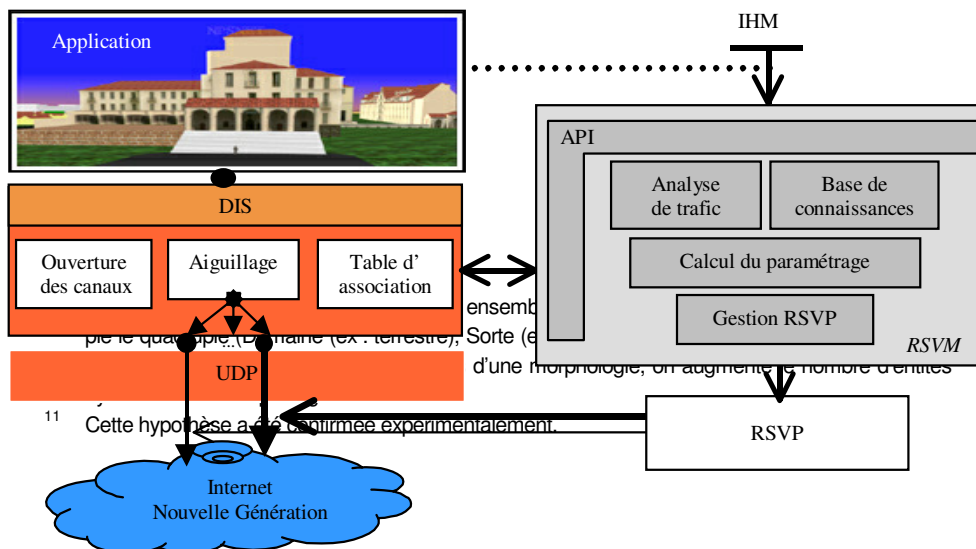


Figure 6 : Architecture fonctionnelle

Deux types de problèmes ont été identifiés :

- la gestion des canaux : ouverture et aiguillage des paquets DIS dans les canaux adéquats (via une table d'association) ;
- le contrôle de la QoS des canaux (rôle dédié au gestionnaire de réservation de ressources - RSVM : Resource reSerVation Manager) : établissement des sessions RSVP sur les canaux (gestion RSVP), lien entre paramètres applicatifs et paramètres RSVP (calcul du paramétrage), spécification d'une interface de programmation (API) permettant à l'application de caractériser les composantes de sa simulation non pas directement en terme de trafic, mais en termes d'entités (morphologie et nombre).

Cette architecture, entièrement spécifiée, est à l'heure actuelle en cours d'implantation ; sa première version a été utilisée lors d'une démonstration contractuelle du projet.

Ces travaux ont été réalisés dans le cadre de la convention n° 97/73-291, cofinancée par le Ministère de l'Education Nationale, de la Recherche et de la Technologie et la Direction Générale de l'Armement.

6. Bibliographie

- [i] IETF : Integrated Service Working Group. <http://www.ietf.org/html.charters/intserv-charter.html>.
- [ii] S. Shenker, C. Partridge, R. Guerin. "Specification of Guaranteed Quality of Service", RFC 2212, IETF Network Working Group, 1997.
- [iii] J. Wroclawski. "Specification of the Controlled Load Network Element Service", RFC 2211, IETF Network Working Group, 1997.
- [iv] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin. "Resource reSerVation Protocol : Functional Specification", RFC 2205, IETF Network Working Group, 1997.
- [v] J. Wroclawski. "The use of RSVP with Integrated Services", RFC 2210, IETF Network Working Group, 1997.
- [vi] P. Owezarski, V. Roca, P. Capière, S. Fdida, M. Diaz, C. Diot. "Protocoles multimédias haut débit et expérimentations sur MIRIHADÉ", Rapport du projet CNRS MIRIHADÉ, mars 98.
- [vii] J. M. Pullen, Fellow, D. C. Wood, Networking Technology and DIS, Proceedings of the IEEE, vol 83, no. 8, August 1995.
- [viii] J. M. Pullen, M. Myjak, C. Bouwens, Limitations of the Internet Protocol Suite for distributed Simulation in the Large Multicast Environment, Simulation Interoperability Workshop (SIW), Spring 1997.
- [ix] S. A. Schricket, R. W. Franceschini, M. D. Petty, Implementation Experiences with the DIS Aggregate Protocol, Spring Simulation Interoperability Workshop, U.S.A, March 1998.

- [x] K. L. Russo, Effectiveness of Various New Bandwidth Reduction Techniques in ModSAF, 13th DIS Workshop, Sept. 1995.
- [xi] S. Aggarwal, B. L. Kelly, Hierarchical Structuring for Distributed Interactive Simulation, 13th DIS Workshop, Sept. 1995.
- [xii] L. Gauthier, C. Diot. "Design and Evaluation of MIMaze, a Multi-player Game on the Internet", IEEE multimedia Systems Conference, Austin, USA, June 28 – July 1, 1998.
- [xiii] R. Ginder, et al., DIS to HLA Integration, a Comparative Evaluation, 15th DIS Workshop, Sept. 1996.
- [xiv] S. Pratt and D. Pratt, Implementation of the IsGroupOf PDU for Network Bandwidth Reduction, 15th DIS Workshop, Nov. 1996.
- [xv] S. G. Purdy Jr., R. D. Wuerfel, A comparison of HLA and DIS Real-Time Performance, Spring Simulation Interoperability Workshop, U.S.A, March 1998.
- [xvi] M. Woo, N.U. Qazi, A. Ghafoor. "A Synchronisation Framework for Communication of Pre Orchestrated Multimedia Information", IEEE Networks, pp. 52-61, January/February 1995.
- [xvii] M. Diaz, P. Sénac, P. De Saqui-Sannes. "Towards a formal specification of multimedia synchronisation", Annales des Télécommunications, Mai/Juin 1994.
- [xviii] M.Diaz, K.Drira, A. Lozes , C. Chassot. "Definition and representation of the quality of service for multimedia systems ", 6th international conference on high speed networking, HPN'95, Palma de Majorca (Balearic Islands), Spain, Sept. 11-15, 1995.
- [xix] M.Fournier, C.Chassot, A.Lozes, M.Diaz. "Service Definition of a multimedia partial order connection"., 2nd COST 237 workshop on Teleservices and Multimedia Communications, LNCS 1052, Copenhagen, Denmark, Nov. 1995.