SETIT 2007
4th International Conference: **S**ciences of **E**lectronic,
Technologies of **I**nformation and **T**elecommunications
March 25-29, 2007 – TUNISIA

Gestion prédictive de la qualité de service pour présentations multimédia au format SMIL: L'approche dynamique

BELKHIR Abdelkader* et BOUYAKOUB Fayçal M'hamed*

** USTHB, Faculté Electronique et Informatique*
Département Informatique
BP 32 El-Alia Alger ALGERIE
belkhir@wissal.dz
bouyakoub.f.m@gmail.com

Résumé: Plusieurs solutions au problème de la Qualité de Service (QoS) ont été développées, tant au niveau réseau qu'au niveau application. Notre approche, qui vise l'amélioration de la qualité de restitution des présentations SMIL, agit au niveau application, ce qui lui permet d'exploiter au mieux les caractéristiques du langage: prédiction du scénario futur et possibilité d'effectuer des préchargements. De plus, notre solution permet à la présentation de s'adapter dynamiquement aux fluctuations de la bande passante.
Mots clés: Préchargement, présentations multimédia au format SMIL, qualité de service, streaming.

INTRODUCTION

L'évolution des technologies réseaux ainsi que l'accroissement de leurs performances ont fait qu'Internet est devenu une infrastructure viable pour supporter les services multimédia. Ces derniers concernent tous les aspects de la vie moderne: la communication personne à personne, la visioconférence, les présentations de documents multimédia synchronisés, l'enseignement, la formation à distance, etc. Ces catégories d'applications sont gourmandes en ressources: capacité de stockage, vitesse d'exécution et consommation de bande passante. La disponibilité de machines ayant une grande capacité de stockage de l'ordre de giga-octets ainsi qu'une vitesse d'exécution très appréciable résout en partie la problématique des ressources. Cependant, toutes ces solutions ne peuvent pas toujours anticiper l'effet de la congestion au niveau du client.

Pour cela, la gestion de la bande passante mérite une attention particulière. Il s'agit de dépasser la stratégie *Best-effort* afin d'aboutir à une stratégie exploitant les caractéristiques de l'application: c'est la qualité de service au niveau application. Les classes d'applications à considérer sont les systèmes de présentation multimédia intégrant le son, l'image, la vidéo et le texte. A cet effet, il y a lieu de définir des solutions adaptées à la nature des applications multimédia. Elles sont caractérisées par des flux multimédia dont la rupture, même momentanée, a un impact direct sur la qualité de service.

Le format de spécification utilisé par l'environnement auteur joue un rôle important dans la réalisation d'un système de présentation multimédia, puisque la popularité du langage choisi affecte directement les possibilités de diffusion de ces présentations à grande échelle, notamment sur le Web. Ce qui peut justifier le choix du langage HTML. Cependant, l'absence de mécanismes de synchronisation fait que HTML est inadapté à la modélisation de présentations multimédia synchronisées où la composante temporelle est prédominante.

L'interface papier [BEL 05] [BEL 06] met en jeu plusieurs flux multimédia synchronisés, c'est évidemment cet aspect de synchronisation qui justifie le choix du langage SMIL [AYA 01] comme langage de spécification des présentations multimédia. De plus, le langage SMIL est un standard du Web. Il offre la possibilité de décrire l'organisation spatiale et temporelle d'une présentation grâce à un ensemble de modules. Parmi ses fonctionnalités, on retrouve le mécanisme de préchargement qui indique qu'un élément multimédia sera utilisé dans le futur. Ainsi les deux mécanismes de préchargement et de streaming constituent la pierre angulaire de notre contribution à travers cet article.

Dans la section 2, on décrit les motivations et le

choix de notre solution ainsi qu'un aperçu sur des travaux similaires. La section 3 présente notre solution de gestion prédictive de la qualité de service en utilisant un procédé statique [BEL 03a][BEL 03b]. Les résultats obtenus nous amène à développer davantage la solution en adoptant une approche dynamique. La section 4 décrit la contribution de l'approche dynamique qui est complémentaire à l'approche statique. On terminera l'article par une discussion des résultats obtenus et une conclusion.

1. Motivations

La qualité de service (QoS) se rapporte à certaines caractéristiques d'une connexion réseau (délai, gigue, bande passante, disponibilité) et s'applique sur les nœuds reliant ses deux extrémités. La garantie d'une QoS peut être obtenue par: la surabondance de bande passante, l'augmentation de la puissance des routeurs et l'adaptation des terminaux. En dépit de la disponibilité de stations et de routeurs plus puissants, il en demeure que la surabondance de la bande passante nécessite un investissement à l'échelle mondiale. Afin de corriger les défauts du service *Best-effort*, le terminal s'efforce de corriger la dégradation par un procédé de remplacement des paquets perdus ou en retard. Néanmoins, cette correction nécessite un temps de calcul et s'avère inefficace si elle dépasse quelques millisecondes. Les routeurs à leurs tours peuvent agir sur la qualité de service en adoptant les architectures *IntServ* ou *DiffServ* [BLA 98].

Afin d'anticiper l'effet de la congestion sur le client, il y a lieu de définir des solutions adaptées à la nature des applications multimédia. Elles sont caractérisées par des flux multimédia dont la rupture même momentanée a un impact direct sur la qualité de service. L'adoption d'une approche de niveau application est motivée par deux éléments essentiels:

- Une approche de qualité de services sur la couche application complète les approches sur les couches inférieures et profite des améliorations offertes par ces dernières.
- SMIL est un langage de niveau application. Une approche de qualité de service au niveau applicatif s'adapte avec la sémantique de ce langage.

Le choix de la technique du préchargement est fortement lié au choix du langage SMIL sur lequel se sont basés nos travaux. Le préchargement a été introduit dans SMIL 2.0 mais de manière élémentaire. Il permet de mieux exploiter les ressources disponibles sans affecter ou modifier l'architecture réseau.

Des solutions ont été élaborées afin d'assurer l'indépendance de la présentation vis-à-vis des variations de la bande passante, en réduisant les délais d'accès par chargement des médias avant la date nominale de leur présentation. Ces solutions agissent au niveau application, ce qui leur permet de s'adapter à la sémantique de SMIL. Le préchargement dans le TLSA Player [SAM 03] peut s'effectuer selon deux approches différentes. L'approche statique consiste à précharger la totalité des éléments avant le début de la présentation, ce qui risque de saturer les ressources systèmes (mémoire) et réseau (bande passante) lorsqu'on a un nombre considérable de médias. Dans l'approche dynamique, le TLSA Player se contente de précharger les éléments des *n* niveaux suivants à partir de l'état courant. Cette approche réduit la consommation des ressources, mais elle peut converger vers l'approche statique dans le cas où le niveau de préchargement est assez grand. De plus, la configuration du niveau de préchargement par des utilisateurs inexpérimentés peut causer une pénurie de ressources. La deuxième solution, présentée dans [YAN 02], tente de réduire les délais d'accès en calculant l'instant idéal pour lancer les préchargements, afin d'assurer la disponibilité des médias à temps. Cette solution présente l'avantage de ne pas saturer la mémoire, étant donné que le préchargement d'un objet se fait juste avant son temps de début. Cependant, l'échange de messages entre le client et les serveurs, augmente la charge du réseau. De plus, la nature instable de la bande passante peut influer négativement sur les temps de chargement calculés des médias, induisant le non respect des échéances temporelles.

2. Gestion prédictive de la qualité de service: L'approche statique [BEL 03a] [BEL 03b]

Le préchargement des objets multimédia dans SMIL repose sur le principe de la récupération des données par anticipation, avant leur utilisation effective dans le temps. Le préchargement doit être mis en œuvre d'une manière efficace, afin de mieux exploiter les ressources disponibles et sans affecter le scénario temporel d'exécution.

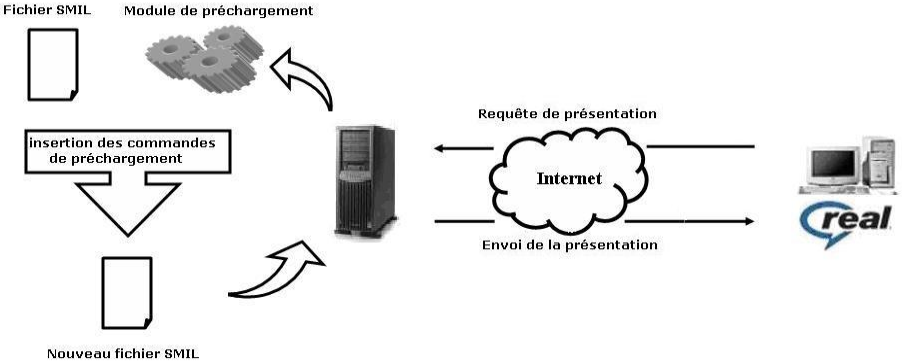

Figure 1. *Architecture du système de préchargement statique*

Lorsqu'une requête est reçue par le serveur hébergeant la présentation SMIL, le module de préchargement est automatiquement exécuté.

Ce module génère un nouveau fichier SMIL à partir du fichier de base incluant les commandes de préchargement des objets multimédia de la présentation. Côté client, l'exécution de la présentation se fera par un player supportant la version 2 de SMIL.

2.1. Modélisation du scénario temporel de la présentation: Génération de l'arbre abstrait

La production du scénario temporel d'exécution d'une présentation multimédia consiste à obtenir:

- Une modélisation des relations temporelles entre les objets multimédia de la présentation: Ces informations seront stockées dans *l'arbre abstrait* [BEL 03a] [BEL 03b] afin de décrire l'organisation logique du document en termes d'éléments composites et éléments de bases mais aussi définir l'ordre de précédence entre les différents éléments.
- Les valeurs de début et de fin effectives de chaque objet multimédia participant dans le déroulement de la présentation: Ces informations seront stockées dans une structure de données appelée la *table des médias*. Au fur et à mesure de l'analyse temporelle du document SMIL, la *table des médias* est mise à jour et on obtient à la fin les temps de début et de fin effectives de chaque objet multimédia participant dans le déroulement de la présentation.

Le résultat final sera sous forme de deux tables:

- La table du scénario ou "*MediaTable*" qui contient toutes les informations relatives aux objets multimédia.
- La table des composites ou "*TagTable*" qui contient les informations relatives aux objets composites. Chaque ligne de notre table correspond à un nœud de l'arbre.

La combinaison de ces deux tables, ainsi que le chaînage entre les éléments de ces dernières forme l'arbre abstrait.

2.2. Insertion des commandes de préchargement

Dans notre approche dite statique [BEL 03a] [BEL 03b], les commandes de préchargement sont insérées dans le document source avant l'exécution de la présentation. Pour cela nous distinguons deux cas:

- Le premier concerne les objets en séquence (<seq>...</seq>).
- Le deuxième concerne les objets en parallèle (<par>...</par>).

Les règles définies précédemment sont traduites par des algorithmes [BEL 03a] [BEL 03b].

2.2.1. *Affectation de la bande passante*

L'autre étape consiste à déterminer la quantité de bande passante à affecter à chaque commande de préchargement insérée. Ce paramètre dépend d'une part, du type de la connexion du client, et d'autre part des besoins du système et des objets en cours d'exécution.

La bande passante disponible au niveau du client est déterminée en utilisant l'attribut SMIL "*systemBitrate*". Cet attribut, combiné avec la balise <*switch*>, permet de tester si la bande passante disponible est supérieure ou égale à la valeur spécifiée par l'attribut "*systemBitrate*". Il suffit donc de dupliquer le code SMIL pour chaque classe de bande passante. L'ordre de classification doit être décroissant. Au niveau du client, le player SMIL sélectionne le bloc qui correspond à la bande passante disponible et exécute le code correspondant.

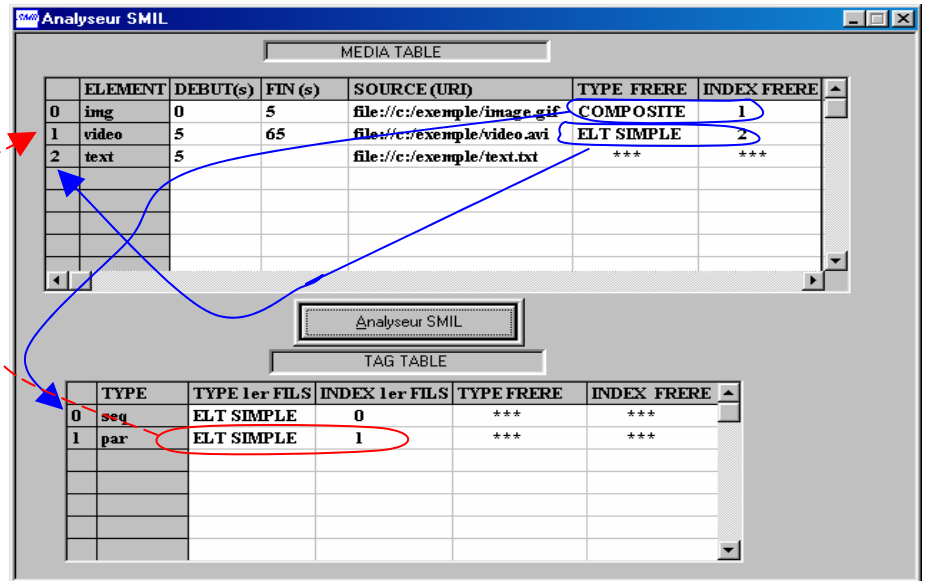

Figure 2. *Interface de l'arbre abstrait*

Des études statistiques [REA 03], ont montré qu'environ 30% de la bande passante est réservée pour le trafic d'en-tête et la correction des erreurs. La bande passante exploitable est donc 70% de la bande passante totale. Pour déterminer la bande passante consommée par les objets multimédia, nous les classons en deux catégories:

- **Les objets hébergés par un serveur Web:** Ces objets utilisent le protocole HTTP. Ils doivent être téléchargés entièrement avant d'être joués. HTTP étant un protocole sans états, la notion de flux est inexistante, il n'y a donc aucune règle permettant de définir les besoins de ces objets en termes de bande passante. Nous pouvons contourner ce problème en attribuant une quantité nécessaire de bande passante à ces objets pendant les premières secondes de leur exécution. Lorsque l'objet est entièrement téléchargé, il libère la bande passante ce qui nous permettra de l'allouer au préchargement [REA 03].
- **Les objets hébergés par un serveur de streaming:** Ces objets sont généralement des médias continus tels que l'audio et la vidéo. L'objet multimédia est récupéré et diffusé en temps réel à partir d'un serveur de streaming. Dans ce cas, il est nécessaire de garantir une bande passante égale au format de compression [REA 03]. Par conséquent, la bande passante à allouer au préchargement d'un objet donné est obtenue par la formule suivante:

$$Bp\,\mathrm{Pr}\,efetch(objet) = 70\%\,BpTotale - \sum BpObjetsRt\,sp - \sum BpObjetsHt\,tp \quad (1)$$

Où:

BpPrefetch(objet): Bande passante à attribuer au préchargement de "*objet*". Dans le cas d'un bloc "seq", cette quantité sera entièrement allouée au préchargement de "*objet*". Quant au cas d'un bloc "par", elle sera répartie sur les préchargements des éléments ayant un décalage temporel par rapport au début du bloc.

BpTotale: Bande passante totale disponible suivant le type de connexion du client.

BpObjetsRtsp: Bande passante consommée par l'objet diffusé en streaming en cours d'exécution.

BpObjectsHttp: Bande passante allouée à l'objet téléchargé en HTTP en cours d'exécution.

2.2.2. *Contraintes à respecter*

La commande de préchargement est un outil puissant à utiliser avec modération. Une utilisation exagérée de cette commande conduit à une dégradation de la présentation à cause de l'épuisement des ressources système (mémoire) et réseau (bande passante).

- **Bande passante:** La bande passante allouée au préchargement doit tenir compte des besoins du système d'une part et des objets en cours d'exécution d'autre part. Lorsque la bande passante à allouer au préchargement est trop faible, le gain est insignifiant. Nous fixons un seuil de 2 Kbps à garantir pour effectuer les préchargements.
- **Mémoire:** La technique de préchargement maintient les données en mémoire jusqu'à leur exécution. Le téléchargement de fichiers volumineux tels que l'audio et la vidéo peut rapidement saturer la mémoire. Pour éviter une telle situation, nous nous limitons au préchargement du *preroll* estimé par [REA 03] à 15 secondes.
- **Durée du préchargement:** Le préchargement n'a aucun intérêt si sa durée est trop petite. Nous fixons alors un seuil limite minimal de 5 secondes.
- **Respect des échéances temporelles:** L'insertion des commandes de préchargement ne doit pas affecter le scénario temporel de la présentation, les échéances temporelles des objets doivent rester inchangées.

2.3. Tests et analyse des résultats

Les documents SMIL et les objets multimédia ont été hébergés sur un serveur de streaming exécutant *Real System Server* [REA 03]. On dispose pour chaque présentation de deux versions: l'une sans commandes *prefetch* et l'autre avec des commandes *prefetch*. Pour exécuter les présentations, nous avons utilisé le logiciel *RealOne Player* [REA 03] qui offre la possibilité de visualiser un diagramme du flux de données entrant. Ces diagrammes ont été interprétés et nous avons obtenu le diagramme des flux moyens de données.

La courbe en pointillés représente le flux de données moyen des présentations initiales, et la courbe en trait le flux moyen de données des présentations après l'insertion des commandes de préchargement.

L'interprétation des résultats montre que les présentations sans *prefetch* sont soumises aux variations de la bande passante (cas de congestion), ce qui provoque des *rebufférisations* continues. Par contre, pour les présentations avec préchargement, la qualité de service reste satisfaisante du fait que le player ne se contente pas de charger l'objet courant, mais tire profit de la disponibilité de la bande passante pour anticiper le chargement des objets selon les commandes *prefetch* insérées dans le document SMIL. Nous remarquons que le flux moyen de données reste supérieur de 30% en moyenne au flux des présentations initiales (sans préchargement).

Lorsqu'il y a congestion, nous avons constaté que ses effets tardent à se faire sentir dans le cas de la présentation avec préchargement. Ceci est dû à l'exploitation de la réserve obtenue à travers le préchargement. Ce qui permet de résister à la chute de bande passante en attendant qu'elle retrouve un taux acceptable.

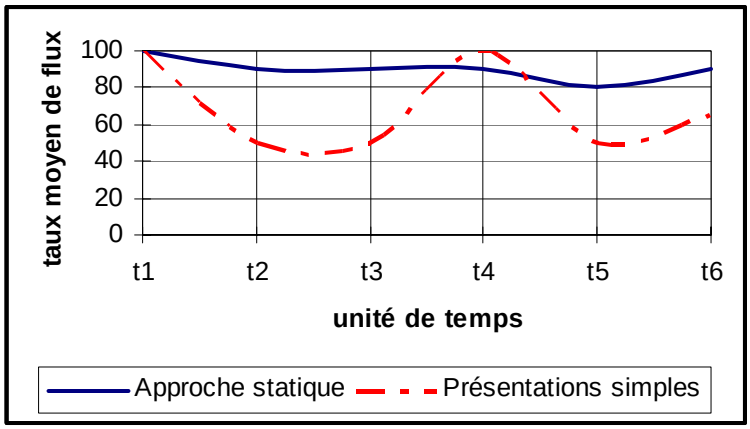

Figure 3. *Flux moyen de données*

En cas de congestion prolongée, la réserve s'épuise et on se retrouve dans le cas d'une présentation sans préchargement. De plus, la bande passante n'est pas fixe, sa valeur fluctue dans le temps, or cette méthode de préchargement insère les commandes de préchargement dans le code du document SMIL *statiquement* avant le début de la présentation. Par conséquent, une surabondance de la bande passante en un moment donné de la présentation ne pourra être pleinement exploitée pour effectuer des préchargements car cette situation ne peut être connue à priori qu'au moment de la présentation. Les résultats satisfaisant obtenus par l'approche statique nous ont encouragés à étendre cette solution en passant à l'approche dynamique qui devrait améliorer davantage la qualité de service. La solution consiste à effectuer les préchargements de manière dynamique au moment de la présentation, les valeurs de bande passante étant plus précises, ceci permettrait une meilleure gestion des préchargements.

3. Gestion prédictive de la qualité de service: L'approche dynamique

L'approche dynamique permet de compléter l'approche statique en effectuant des préchargements à la volée, i.e. pendant l'exécution de la présentation. Pour cela, on a intégré notre solution au sein du player SMIL. Notre solution passe par deux étapes:

- Insertion des balises <ref /> et création des fichiers de préchargement
- Remplissage des fichiers de préchargement à la volée

Cette solution se base, tout comme l'approche statique, sur l'arbre abstrait afin de récupérer les informations de synchronisation de la présentation SMIL. L'insertion des balises de référence se fait automatiquement au début de la présentation aux endroits où les opérations de préchargement sont possibles. Cette opération est suivie par la création d'un fichier de préchargement pour l'objet à précharger.

3.1. Insertion des balises de référencements

Cette balise permet de jouer dans une même instance de *RealOne Player* la présentation principale et un autre élément tel qu'une image, une vidéo, une page Web, une animation ou même une deuxième présentation SMIL. C'est d'ailleurs cette dernière option qui sera la base de notre approche. Nous distinguons deux cas d'insertion des balises de référencement:

3.1.1. *Cas d'un groupe séquentiel d'éléments*

Dans le cas des éléments en séquentiel, on insère la balise de référence en parallèle avec l'élément en cours. Cette balise fait référence à un fichier SMIL appelé *PrefetchFile.smi* et généré à l'insertion de la balise <ref/>. Ce fichier contiendra une balise <prefetch/> de l'élément qui suit l'élément courant.

3.1.2. *Cas d'un groupe parallèle d'éléments*

Dans ce deuxième cas de figure, on a deux possibilités:

- **Cas des décalages à l'intérieur du groupe:** Dans cette situation, on profitera de ces décalages pour insérer une balise <ref/> à l'intérieur du groupe "par". Cette balise fait référence à un fichier SMIL qui contient le préchargement de l'objet ou des objets décalé(s) par rapport au début du père.
- **Cas d'un élément multimédia frère du groupe "par":** C'est le cas où juste avant le "par", on a un élément appartenant à un "seq". Dans ce cas, si on dispose d'une quantité suffisante de bande passante pour effectuer des préchargements sur un ou plusieurs objets du "par", nous insérons en parallèle à l'élément courant une balise <ref/> qui fait référence à un fichier de préchargement contenant un groupe parallèle de commandes de préchargement des objets commençant au début du "par".

Voici des exemples de chaque cas.

- **Groupe séquentiel d'éléments**

Code initial

```
<seq>
  <text src="../text1.txt" id="text1" dur="20s"/>
  <video src="../video.rm" id="video" dur="1:00"/>
  <text src="../text2.txt" id="text2" dur="20s" />
</seq>
```

Le code obtenu après insertion des balises de référencements

```
<seq>

    <par>
      <text src="../text1.txt" id="text1" dur="20s"/>
      <ref src="../PrefetchFile1.smi" begin="10s"/>
    </par>

    <par>
      <video src="../video.rm" id="video" dur="1:00"/>
      <ref src="../PrefetchFile2.smi"  begin="10s"/>
    </par>
     <text src="../text2.txt" id="text2" dur="20s" />

</seq>
```

```
<smil>
 <body>
   <prefetch  src="../video.rm" … />
 </body>
</smil>
```

PrefetchFile1.smi

```
<smil>
 <body>
   <prefetch  src="../text2.txt"  … />
 </body>
</smil>
```

PrefetchFile2.smi

- **Groupe parallèle d'éléments**
  - **Cas (a)**

Code initial

```
<par>
 <img begin= "20s" src= "../image.jpg"/>
 < video dur= "1:00src= "../video.rm"/>
</par>
```

Le code SMIL obtenu après l'insertion des balises de référencement est le suivant:

```
<par>

    <ref src="../PrefetchFile1.smi"/>
    <img src="../image.jpg" begin= "20s"/>
    <video src= "../video.rm" dur= "1:00" />

</par>
```

```
<smil>
 <body>
   <prefetch  src="../image.jpg" dur="20s" … />
 </body>
</smil>
```

PrefetchFile1.smi

  - **Cas (b)**

Code SMIL initial

```
<img src="../image1.jpg" dur="1:00"/>
<par>
 <img src= "../image.jpg" begin= "20s"/>
 <video src= "../video.rm" dur= "1min" />
</par>
```

Le code SMIL obtenu est le suivant

```
<par>
  <img src="../image1.jpg" dur="1:00"/>
  <ref src="../PrefetchFile1.smi"/>
</par>
<par>
    <img  src= "../image.jpg"/>
    <video dur= "1min" src= "../video.rm"/>
</par>
```

```
<smil>
 <body>
   <par dur="1:00" >
     <prefetch  src="../image.jpg" … />
     <prefetch  src="../video.rm"   … />
   </par>
 </body>
</smil>
```

PrefetchFile1.smi

Dans le cas d'éléments *streamés*, le décalage permet au player de précharger le preroll de cet objet s'il n'a pas été préchargé auparavant. Par la suite, la bande passante allouée au préchargement sera la bande passante restante puisque l'objet *streamé* consommera de la bande passante pendant son exécution selon son format de codage.

3.1.3. *Les attributs utilisés dans la balise de référence*

La balise de référencement contient les attributs suivants:

- "***src***": Recevra l'url du fichier de préchargement contenant les commandes de préchargement.
- "***begin***": Cet attribut sert à effectuer un décalage par rapport à l'objet courant. Si cet élément est un élément Http, alors ce décalage va servir pour laisser le temps au téléchargement de cet objet. Dans ce cas, la bande passante allouée au préchargement sera la totalité de la bande passante disponible. Dans le cas d'éléments *streamés*, le décalage permet au player de précharger le *preroll* de cet objet s'il n'a pas été préchargé auparavant. Par la suite, la bande passante allouée au préchargement sera la bande passante restante puisque l'objet streamé consommera de la bande passante pendant son exécution selon son format de codage.
- *"**dur***": Sert à arrêter le fichier de préchargement (arrêt de la présentation *PrefetchFile.smil*) dans le cas où l'objet courant se termine avant la fin du préchargement afin de respecter le scénario temporel de la présentation.

3.2. Remplissage des fichiers de préchargement

L'approche dynamique constitue une contribution à l'approche statique. En effet, l'approche statique ne permettait pas de connaître la valeur instantanée de la bande passante disponible au niveau du client. Elle procédait par tranche de type de connexion (entre 28 et 56, entre 56 et 128 Kbps…). De plus, l'insertion des commandes de préchargement se faisait à la création de la présentation, ce qui nous donnait des résultats approximatifs pour les paramètres de préchargement.

Dans l'approche dynamique, le calcul de la bande passante à affecter au préchargement est donné par la formule suivante:

$$BpDisponible = 70\% * RealOne.GetConnectionBandwidth() - RealOne.GetCurrentBandwidth() \quad (2)$$

Où

BpDisponible: Bande passante à affecter aux préchargements

RealOne.GetConnectionBandwidth(): Bande passante de la connexion

RealOne.GetCurrentBandwidth(): Bande passante courante

3.3. Fonctionnement

Cette solution est une amélioration de l'approche statique. Les procédures de création de l'arbre abstrait ainsi que les procédures de calculs des temps de début et de fin, de chaque objet multimédia, utilisées dans l'approche statique seront réutilisées. Cependant, de nouvelles fonctions ont été ajoutées: Nous citerons la procédure d'insertion des balises de référencement ainsi que les procédures de création et de remplissage des fichiers de préchargement. L'utilisation d'un timer s'avère nécessaire dans notre puzzle. En effet, le timer nous permettra de synchroniser notre application avec l'état d'avancement de la présentation afin de signaler l'arrivée de l'instant de remplissage du fichier de préchargement. Le timer est armé après le remplissage de chaque fichier de préchargement afin de signaler le prochain instant de remplissage. Cette valeur est calculée suivant la formule suivante:

$$PrefetchingInstant = ObjectBeginningTime - RealOne.GetPosition() - \Delta t \quad (3)$$

Où:

PrefetchingInstant: Est l'instant de remplissage du fichier de préchargement

ObjectBeginningTime: Est le temps de début du média à précharger

RealOne.GetPosition(): Est le temps courant de la présentation

Δt: Intervalle de temps nécessaire pour préparer le fichier de préchargement avant le lancement de l'objet préchargé. Cette valeur varie selon la vitesse de calcul de la machine.

Remarque

Selon les statistiques de [REA 03], seul 85% de la bande passante disponible sera affecté au préchargement. Nous réservons une marge de 15% afin d'éviter la saturation de la bande passante en situation de congestion.

3.4. Maintien de la synchronisation

Afin de maintenir la synchronisation entre l'application et la présentation, nous avons utilisé les *Callbacks* (messages envoyés suite à un événement) de RealOne. En effet, RealOne offre une série de *Callbacks* qui nous permettent de suivre l'état du player. On s'est intéressé plus particulièrement à *OnPlayStateChange(NewState)*. Ce message est envoyé par le player suite à un changement d'état dans la présentation. Le nouvel état est exprimé par le paramètre *NewState*. Ce paramètre admet plusieurs valeurs. Les deux états qui nous intéressent sont: *buffering* et *playing*. Quand le player est dans un état de *rebufférisation*, l'action à effectuer est l'arrêt du timer de l'application. Dés le changement d'état du timer (à la reprise de la diffusion de la présentation), la player envoie un *Callback* avec comme paramètre *playing*. Dans ce cas, il suffit de réarmer le timer avec une nouvelle valeur.

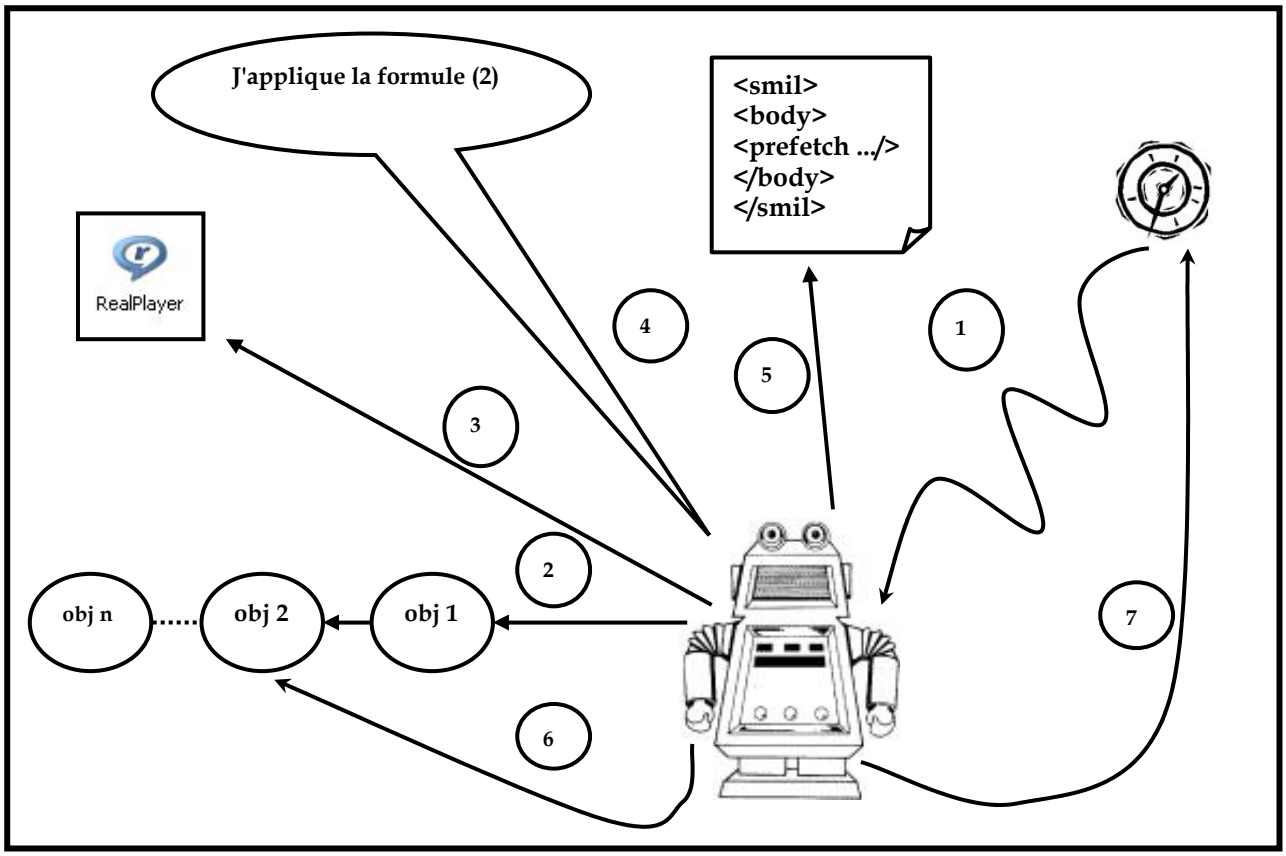

Figure 4. *Fonctionnement de l'approche dynamique*

1: Signal d'horloge indiquant l'arrivée du temps de remplissage du fichier de préchargement

2: Récupération des informations nécessaires concernant l'objet à précharger à partir d'une liste contenant des informations sur les objets à précharger

3: Récupération à partir des méthodes de RealOne Player de la bande passante disponible

4: Application de la formule de calcul ***(2)***

5: Remplissage du fichier de préchargement

6: Récupération à partir de la liste des médias à précharger le temps de début du prochain élément

7: Armement du timer avec la valeur calculée par la formule ***(3)***

3.5. Tests

Les tests ont été effectués sur le réseau du CERIST[1]. Les documents SMIL, ainsi que les objets multimédia ont été hébergés sur un serveur supportant le streaming. Pour jouer les présentations, nous avons utilisé le logiciel RealOne Player [REA 03] qui offre la possibilité de visualiser le flux de données entrant. Pour simuler l'environnement d'Internet, nous avons introduit une charge supplémentaire sur le réseau.

Chaque document SMIL a été soumis au module d'insertion des commandes de préchargement (approche statique et dynamique). Les présentations ont été jouées dans des conditions semblables pour pouvoir les comparer. Nous avons suivi la même procédure que celle réalisée dans la première partie où nous avons interprété les diagrammes de RealOnePlayer sous forme d'un diagramme des flux de données moyens.

4. Discussion

L'approche statique permet d'améliorer la disponibilité des médias en exécutant les commandes de préchargement insérées avant le lancement de la présentation. Cependant, cette solution présente quelques limitations. En effet, les commandes de préchargement étant insérées statiquement avant le début de la présentation, une surabondance de la bande passante en un moment donné de la présentation ne pourra être pleinement exploitée pour effectuer des préchargements, car cette situation ne peut être connue à priori qu'au moment de la présentation. Cette situation n'est pas prévisible dans le cas d'une approche statique. La solution de préchargement dynamique dispose d'un mécanisme de gestion de la bande passante en temps réel, lui permettant une exploitation optimale de cette ressource. De manière générale, les tests effectués ont montré que le mécanisme de préchargement dynamique améliore davantage (environ 35%) la qualité des présentations SMIL en

[1] CEntre de Recherche sur l'Information Scientifique et Technique: www.cerist.dz

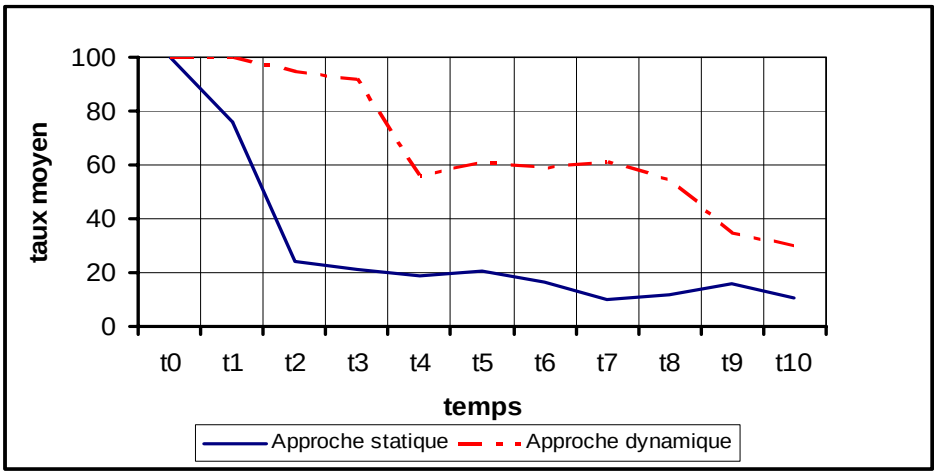

Figure 5. *Flux moyen de données*

permettant d'obtenir des présentations plus fiables et plus résistantes aux perturbations de la bande passante, grâce à un mécanisme de gestion en temps réel.

En cas de congestion prolongée, l'impact sur la présentation est retardé, mais une congestion prolongée induit une chute du niveau de flux de données, menant à l'arrêt de la présentation.

La solution proposée est basée sur une approche prédictive, elle tire profit de la possibilité d'avoir le scénario temporel de la présentation à l'avance pour anticiper le chargement des médias (en utilisant les commandes prefetch de SMIL 2.0) lorsque la bande passante est abondante, et disposer ainsi d'une réserve locale en prévision d'une situation de congestion.

5. Conclusion

Nous avons vu, à travers les expériences menées, que les commandes de préchargement permettent de maintenir un niveau de qualité relativement stable. Nous remarquons aussi, qu'en moyenne, le flux de données reste supérieur au flux des présentations de l'approche statique.

Les tests effectués prouvent que l'approche dynamique, par son mécanisme de gestion de la bande passante en temps réel, permet de surmonter les limitations de l'approche statique. En effet, une utilisation optimale de la bande passante dans l'approche dynamique permet de retarder l'effet des situations de congestions sur les présentations par rapport à l'approche statique. De plus, les valeurs de bande passante étant plus précises et en temps réel, ceci permet une meilleure gestion de la bande passante, en prévision d'une situation de congestion.

De plus, lorsqu'une congestion se produit, nous avons vu que ses effets tardent à se faire sentir, ceci est dû à l'exploitation de la réserve locale emmagasinée grâce aux préchargements effectués. Cette réserve permet à la présentation de résister aux chutes de bande passante en attendant qu'elle retrouve un taux acceptable.

REFERENCES

[AYA 01] Ayars J.: "*Synchronized Multimedia Integration Language (SMIL 2.0)*", recommandation du W3C, http://www.w3.org/TR/smil20/, 2001.

[BEL 03a] Belkhir A., Bouyakoub F.M. et Smail S.: "Gestion prédictive de la qualité de service pour les présentations multimédia au format SMIL", *VI th International Symposium on Programming and Systems, ISPS 2003*, Algérie, Mai 2003.

[BEL 03b] Belkhir A., Bouyakoub F.M. et Smail S.: "Predictive QoS management for SMIL presentations", *4th international conference on Information Technology based Higher Education and Training, ITHET03*, Maroc, Juillet 2003.

[BEL 05] Belkhir A., Bouyakoub F.M. et Smail S.: "Interface papier pour des présentations multimédia", *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, SETIT 2005*, Tunisie, Mars 2005.

[BEL 06] Belkhir A., Bouyakoub F.M. et Bouyakoub S.: "Toward a search engine of multimedia presentations ", *Web Information Systems and Technologies, WEBIST 2006*, Portugal, Avril 2006.

[BLA 98] Blake S., Black D., Carlson M., Davies E., Wang Z. et Wies W.: "An Architecture for Differentiated services", RFC 2475, 1998.

[REA 03] RealNetworks, en ligne à: http://www.realnetworks.com, 2003.

[SAM 03] Sampaio M.P.N: "Conception formelle de documents multimédia interactifs: une approche s'appuyant sur RT-LOTOS", thèse de Doctorat, université Paul Sabatier de Toulouse, Avril 2003.

[YAN 02] Yang C.C. et Yang Y.Z.: "Design of the data-retrieving engine for distributed multimedia presentations", Multimedia and Communications Laboratory, Department of Computer Science and Information Engineering, National Chi-Nan University, Taiwan, 2002.