



**FACULTAD DE INFORMÁTICA**  
UNIVERSIDAD POLITÉCNICA DE MADRID

# UNIVERSIDAD POLITÉCNICA DE MADRID

## FACULTAD DE INFORMÁTICA

TRABAJO FIN DE CARRERA

PLATE-FORME DE STREAMING POUR LES DISPOSITIFS  
MOBILES AVEC ADAPTATION DYNAMIQUE

AUTOR:  
TUTOR:

Pablo Riesgo Ferreiro  
Vasile-Marian Scuturici

priesgoferreiro@gmail.com  
marian.scuturici@insa-lyon.fr



Abstract.	v
Résumé	v
Resumen	vii
1 INTRODUCTION	11
2 STREAMING	13
2.1 MOBILE STREAMING	15
2.2 LES STANDARDS DU MOBILE STREAMING	16
2.2.1 PROTOCOLES DU STREAMING	17
2.3 LES SERVEURS DE STREAMING	19
2.4 SOLUTIONS EXISTANTES DU STREAMING	21
3 INTERNET SUR LE MOBILE	23
4 STREAMING SUR UNE PLATE-FORME D'INTERNET SUR LE MOBILE	27
4.1 SOLUTIONS EXISTANTES	28
5 PLATE-FORME DE STREAMING POUR LES DISPOSITIFS MOBILES AVEC ADAPTATION DYNAMIQUE	29
5.1 ADAPTATION AUX MARK UP	31
5.2 ADAPTATION DE CONTENU	33
5.3 ADAPTATION DES FLUX DE STREAMING	33
5.3.1 ADAPTATION AUX RÉSEAUX	33
5.3.2 ADAPTATION AUX DISPOSITIFS MOBILES	35
6 ANALYSE DE LA PERFORMANCE DE L'ENCODAGE DYNAMIQUE DES MEDIA	39
7 CONCLUSIONS ET PERSPECTIVES DE FUTUR	45
8 REMERCIEMENTS	47
9 BIBLIOGRAPHIE	49
9.1 Streaming	49
9.2 Les serveurs de streaming	49
9.3 Protocoles de streaming	50
9.4 Réseaux mobiles	50
9.5 Internet mobile	50
9.6 Mobile Streaming	51
10 ANNEXE A : EXEMPLE DE CODE DE GÉNÉRATION DYNAMIQUE DES LANGAGES DE MARK-UP	53
11 ANNEXE B : RESULTATS DE LA GÉNÉRATION DYNAMIQUE DES MARK-UP ET L'ENCODAGE DYNAMIQUE DES IMAGES	55



---

## Abstract.

This paper studies the domain of mobile internet with video streaming services. The internet mobile services are in a moment of expansion and look for the convergence with internet.

The mobile internet services' problem is the enormous heterogeneity that exists within the mobile networks, mobile terminals and mobile internet protocols. The reason for this heterogeneity are the heavy restrictions in mobile networks, specifically the limited wide band and the high time of latency, that push developers to make solutions not compatible with internet.

We propose a solution with a data base wich keeps all the capabilities of every mobile terminal. With the information from this data base and the particular environment of every user our solution makes a dynamic adaptation of the service, in order to offer the best possible service.

The adaptation of the streaming flux is done with a dynamic media encoding process. This process is a great consumer of the processor capacity. From our results we conclude that it is not possible to use intensively the media dynamic encoding.

## Résumé

Cet article traite le domaine d'Internet sur le mobile avec des services de *streaming* de vidéo. Les services d'Internet sur le mobile sont en pleine expansion et cherchent la convergence avec internet.

La problématique des services d'Internet sur le mobile est l'énorme hétérogénéité existant au niveau des réseaux mobiles, des dispositifs mobiles et des protocoles d'Internet sur le mobile. La cause de cette hétérogénéité sont les fortes contraintes des réseaux mobiles au niveau de la bande passante et du temps de latence, qui obligent à créer des solutions non compatibles avec l'Internet.

Nous proposons une solution basée sur une base des données avec les capacités de chaque dispositif. Avec les informations de ce base des données et en connaissant les caractéristiques de l'environnement particulier de chaque utilisateur nous faisons une adaptation dynamique des services, de façon à offrir le meilleur service possible.

L'adaptation des flux de *streaming* passe par l'encodage dynamique des média qui est un processus très consommateur de puissance de processeur. Selon nôtres résultats il n'est pas possible de faire un emploi intensif de l'encodage dynamique.



## Resumen

Este proyecto ha sido realizado durante una beca de intercambio *Erasmus Mundus* en la escuela francesa *INSA Lyon* en el departamento de informática. El proyecto ha sido llevado a cabo entre este departamento y una empresa privada, *3G-Factory*, proveedor de servicios de telefonía móvil. Mi tutor en la escuela *INSA Lyon* ha sido Vasile-Marian Scuturici y mi tutor en la empresa ha sido Jean-François Hugot.

La temática del proyecto se encuentra dentro del campo de las telecomunicaciones y servicios. Más concretamente trata sobre el empleo de la tecnología del *streaming* en redes de telefonía móvil.

Este es un tema de actualidad ya que los primeros servicios de *streaming* sobre el móvil empiezan a aparecer en el mercado. Sin embargo, la gran heterogeneidad de las redes móviles de datos y de los dispositivos móviles dificultan la implantación de estos servicios para todos los usuarios.

El objetivo de este proyecto es primeramente estudiar a fondo el problema, con vistas a ofrecer una posible solución para gestionar esta heterogeneidad, hasta que surja un standard ampliamente aceptado y que la convergencia entre las redes móviles e Internet sea efectiva.

La metodología en la consecución de este proyecto es la siguiente:

- Estudio sobre las tecnologías de *streaming* existentes (Capítulo 2)
- Estudio sobre las tecnologías de Internet en el móvil (Capítulo 3)
- Estudio sobre el *streaming* en un servicio de telefonía móvil (Capítulo 4)
- Solución propuesta de una plataforma de *streaming* que se adapta dinámicamente a las condiciones de la red empleada y al terminal móvil de acceso (Capítulo 5)
- Validación de la solución por medio de un análisis del desempeño de la codificación dinámica de flujos de *streaming* (Capítulo 6)

Primeramente hay que definir el *streaming* como una técnica que permite la reproducción de video y audio en tiempo real. Básicamente permite dos usos: el *streaming* a la demanda (*on-demand streaming*) y la difusión en directo (*live streaming*). Esta segunda técnica es más exigente en cuanto a tiempos de latencia.

Hay tres disciplinas que entran en juego en el proceso de *streaming* que hay que tener en cuenta en este estudio:

- La compresión de datos multimedia, formatos y *codecs*
- Los servidores de *streaming*
- Los protocolos de *streaming*

Para gestionar la problemática del *streaming* sobre redes móviles se adapta el uso del standard *3GPP PSS* definido por la *OMA (Open Mobile Alliance)* en cuanto a compresión de datos y protocolos se refiere.

Respecto al servidor de *streaming* se hace un estudio comparativo de varios servidores existentes en el mercado. En este análisis se valoran el soporte de formatos, *codecs* y protocolos del standard a utilizar, así como el desempeño y la capacidad de realizar codificaciones dinámicas.

La elección final es emplear en nuestra solución como servidor de *streaming* el *VLC Media Player 0.8.6c* de *VideoLAN*.

Respecto al estudio de las soluciones existentes para ofrecer servicios de Internet sobre le móvil existen dos opciones:

- *WAP* soportado por la *OMA*
- *iMode* desarrollado por *NTT DoCoMo* y muy presente en Asia

Además a nivel de aplicación hay varios protocolos, o lenguajes de *mark-up* en este caso, existentes:

- *WML 1.x (Wireless Mark-up Language)*
- *XHTML MP (Mobile Profile)* con varias implementaciones existentes
- *CHTML (Compact HTML)* utilizado en *iMode*

La *OMA* ha desarrollado un proyecto para gestionar esas diferencias, el *UAProf (User-Agent Profile)*. Este es una base de datos que almacena las capacidades de cada Terminal móvil. Sin embargo, no solo es una solución de pago, si no que además los resultados no son satisfactorios debido a la competencia entre las empresas que controlan la *OMA*.

Existe otro proyecto basado sobre el mismo principio pero sin embargo es un proyecto *open-source* mantenido por una comunidad de desarrolladores, este es *WURFL*.

En nuestra solución emplearemos esta segunda opción.

En cuanto al estudio del empleo de *streaming* sobre un servicio de Internet sobre el móvil, encontramos principalmente dos problemáticas:

- Heterogeneidad a nivel de redes de acceso
- Heterogeneidad a nivel de dispositivos móviles:
  - Navegador, que afecta a los lenguajes de *mark-up*
  - Dispositivos de entrada / salida, que afecta sobre todo al aspecto grafico del servicio y por tanto al *streaming* también
  - Reproductor, que afecta directamente al *streaming*

En relación con las redes consideraremos las redes de datos *GPRS (General Packet Radio Service)*, *EDGE (Enhanced Data rates for GSM of Evolution)*, *WCDMA (Wideband Code Division Multiple Access)* y *HSDPA (High Speed Downlink Packet Access)* que son la redes que funcionan principalmente en Europa, dejaremos a un lado las redes *CDMA* y *CDMA2000* que prevalecen en Norteamérica y Asia.



El problema de la heterogeneidad de los dispositivos móviles se solucionará con *WURFL* como ya hemos dicho.

Así la arquitectura de nuestra solución tendrá tres elementos principalmente:

- Base de datos *WURFL*
- Servidor *web*
- Servidor de *streaming*

Además nuestra solución se adapta dinámicamente a diversos factores cambiantes:

- Protocolo de aplicación adecuado (*mark-up*)
- Adaptación de objetos complejos que no necesiten el *streaming* (imágenes, menus, etc.)
- Adaptación de los flujos de *streaming*: a la red empleada y al terminal móvil

Los dos primeros puntos se solucionan empleando adecuadamente las páginas dinámicas de servidor y la base de datos *WURFL*.

El tercer punto es el punto crítico de este proyecto.

Cada una de las redes estudiadas tiene un ancho de banda máximo bastante limitado en comparación con las conexiones de banda ancha a que estamos acostumbrados.

Esta limitación en el ancho de banda disponible hace que la tasa de *bits* del flujo de *streaming* esté también limitada debiendo adaptar esta en función de la red empleada.

Como método para conocer la red que utiliza el usuario proponemos la interacción del usuario que elegirá su red.

La adaptación del flujo de *streaming* en función del terminal móvil será resuelta a través de la base de datos *WURFL*. Se adaptará el tamaño de las imágenes en función del tamaño de la pantalla, se adaptarán las tasas de *bits* en función de la capacidad de proceso del terminal y finalmente se definirán los *codecs* a utilizar.

De manera que proponemos una solución multidispositivo y multiprotocolo, que se adapte dinámicamente aprovechando al máximo las capacidades de cada terminal y cada red de datos móvil.

Para validar la solución propuesta se realizan unas pruebas sobre el mayor condicionante de la codificación dinámica de flujos de *streaming*, el consumo de capacidad de proceso de la CPU.

Las pruebas realizadas modifican cuatro factores en la codificación:

- Los contenedores o formato
- Los *codecs* de audio y video
- La tasa de *bits* de estos *codecs*
- El tamaño de las imágenes



Los resultados muestran que estos procesos de codificación son grandes consumidores de CPU, mas de lo esperado, restringiendo el uso intensivo de la codificación dinámica.

De manera que nuestra solución quedará restringida para la utilización en plataformas industriales de gran capacidad o para usos muy específicas que no requieran la codificación de muchos flujos al mismo tiempo, como podría ser la difusión en directo, que requeriría un numero pequeño de flujos para adaptarse a todo el espectro de terminales.

# 1 INTRODUCTION

Nous allons étudier dans ce travail les services d'Internet sur le mobile avec des fonctionnalités avancées, comme le *streaming* des flux de vidéo ou audio.

Le but de cette étude est offrir une solution qui gère tous les hétérogénéités des dispositifs mobiles, réseaux mobiles et protocoles d'Internet sur le mobile.

Premièrement dans le chapitre 2. *Streaming* nous allons définir le processus de *streaming* et étudier sa mise en place sur internet. A continuation nous allons voir les différences, contraintes et standards existants pour accomplir un service de *streaming* sur le mobile. Une analyse des serveurs de *streaming* est faite après pour faire le choix d'un serveur de *streaming* pour notre plate-forme.

Dans le chapitre 3. *Internet sur le mobile* nous analysons les services de Internet sur le mobile existants et nous allons choisir une méthode pour développer des applications indépendantes de la technique employée pour proportionner l'accès à Internet dès les réseaux mobiles.

Nous montrons l'intérêt dans le chapitre 4. *Streaming sur une plate-forme d'Internet sur le mobile* de développer des applications de *streaming* sur une plate-forme d'Internet mobile.

Dans le chapitre 5. *Plate-forme de streaming pour les dispositifs mobiles avec adaptation dynamique* nous faisons la proposition d'une solution qui offre des services d'Internet et de *streaming* sur le mobile adaptables dynamiquement.

La validité de cette solution est étudiée dans le chapitre 6. *Analyse de la performance de l'encodage dynamique des média*. Pour cela faire nous étudions la consommation de puissance de processeur du processus d'encodage dynamique des média

Nous finissons avec les chapitres 7. *Conclusions et perspectives de futur* et 8. *Remerciements*.

Dans le chapitre 9. *Bibliographie* se trouvent toutes les sources employées pour faire cette étude.

Sur l'Annexe A: *Exemple de code de génération dynamique des langages de mark up* et Annexe B: *Résultats de la génération dynamique des mark up et l'encodage dynamique des images* nous pouvons voir l'adaptation dynamique des applications aux différents dispositifs mobiles.



## 2 STREAMING

Le *streaming* est une technique qui permet le transfert des ressources média (vidéo, audio, etc.) en temps réel.

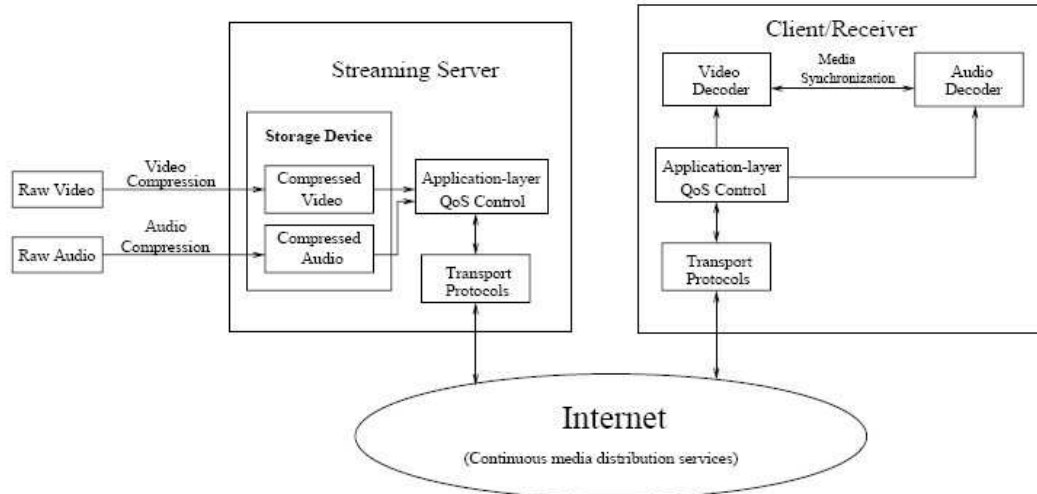
Par rapport à la technique de téléchargement il permet de réduire le temps de latence<sup>1</sup>, la taille des ressources à stocker en mémoire et de faire le *broadcast*<sup>2</sup> des événements en direct.

Le *streaming* des média est utilisé sur Internet par plusieurs applications commerciales comme l'éducation à distance, le *télécast*<sup>3</sup>, la diffusion (*broadcast*) des événements en direct, les services de vidéo à la demande, la radio sur Internet, etc. Il est aussi employé par les entreprises pour la téléconférence.

Il y a une forte croissance dans la demande des média en *streaming* [2], celle-ci est à mettre en relation avec la croissance des connexions Internet haut débit qui permettent un *streaming* de plus haute qualité et fiabilité [3].

Il faut différencier principalement deux types de flux sur Internet: le *streaming* à la demande des vidéos déjà enregistrés (*on-demand streaming*<sup>4</sup>) et le *broadcast* des événements en direct (*live streaming*<sup>5</sup>) [5, 9].

Dans l'illustration 2-1 on peut voir un diagramme du processus complet de *streaming* [10].



**Illustration2-1: Etapes dans le processus de streaming**

<sup>1</sup> Temps de latence: temps d'attente entre une requête et sa réponse.

<sup>2</sup> *Broadcast*: diffusion des données vers plusieurs clients.

<sup>3</sup> *Télécast*: broadcast des émissions de télévision.

<sup>4</sup> *On-demand streaming*: diffusion par *streaming* vers un seul client des média stockés à la demande, souvent avec fonctionnalités de contrôle VCR (*play, pause, stop*).

<sup>5</sup> *Live-streaming*: diffusion par *streaming* des événements en direct vers plusieurs clients.

Il existe plusieurs étapes dans ce processus. Chaque média est typiquement décomposé en deux flux d'audio et vidéo (éventuellement sous-titres ou autres). Il est nécessaire de compresser ces flux avec des *codecs*<sup>6</sup> *ad hoc*.

Après, ces flux comprimés sont multiplexés<sup>7</sup> dans un conteneur<sup>8</sup>, appelé couramment le format de média.

Les résultats des requêtes clients sont adaptés par la couche applicative de *QoS*<sup>9</sup> suivant l'état du réseau. Il est finalement envoyé au client avec le protocole de transport approprié.

Chez le client les différents paquets du conteneur sont reçus par la couche de transport homologue. Ceux-ci sont démultiplexés, décompressés et une synchronisation entre les différents flux est réalisée, si nécessaire.

La compression des média est critique pour accomplir un flux de *streaming* de qualité. Les deux paramètres qui affectent le plus la qualité des données est le *codec* et le débit d'encodage<sup>10</sup> employés.

Le débit peut-être statique ou dynamiquement adaptable à l'état du réseau dans le cas où il existe un service de *QoS*.

Chaque *codec* a des limites pour le débit d'encodage, en dehors desquelles la qualité du flux est dégradée.

Plusieurs éléments sont nécessaires pour mettre en place un service de *streaming* :

- La compression des média
- Le contrôle de la qualité de service (*QoS*)
- Les serveurs de *streaming*
- Les mécanismes de synchronisation des média
- Les protocoles de *streaming*

Nous allons dans un premier temps étudier le *streaming* sur un dispositif mobile (*mobile streaming*), les différents protocoles et serveurs de *streaming* existants. Puis choisir les protocoles et le serveur répondant le mieux à notre problème. Enfin, on reviendra sur des techniques de compression des média pour améliorer la performance.

Des techniques telles que les mécanismes de synchronisation sont nécessaires à la bonne reproduction du média. Elles n'ont pas de liens avec la performance de la transmission et seront donc exclues de cette étude.

---

<sup>6</sup> *Codec*: le mot-valise *codec* vient de *codeur* *décodeur* et désigne un procédé capable de compresser ou de décompresser un signal, analogique ou numérique, en un format de données.

<sup>7</sup> *Multiplexation*: procès par lequel multiples signaux (flux dans notre cas) sont combinés dans un seul signal.

<sup>8</sup> *Conteneur*: format des données que contiennent des données comprimées avec un ou plusieurs *codecs*.

<sup>9</sup> *QoS (Quality of Service)*: la qualité d'un service, comme par exemple la fiabilité.

<sup>10</sup> *Débit d'encodage*: quantité de kilobits par seconde du résultat du processus d'encodage.

## 2.1 MOBILE STREAMING

Le terme de *mobile streaming* comprend le *streaming* d'un serveur vers un dispositif mobile, portable, PDA, etc., sur un réseau sans fil de long porté (voir réseau de téléphonie mobile).

Un dispositif mobile doit avoir la capacité de se connecter à un réseau à commutation des paquets<sup>11</sup>, comme *GPRS*, *EDGE* ou *UMTS-WCDMA* et avoir le logiciel permettant la lecture d'un flux de *streaming* [27].

Le *streaming* vers les réseaux sans fil de courte portée comme *Bluetooth* ou *WLAN* (*Wireless Local Area Network*) ne seront pas considérés dans cette étude.

Les dispositifs mobiles ont les contraintes suivantes:

- Capacité faible de *CPU*
- Peu de mémoire
- Une restriction sur la consommation d'énergie.
- Différents dispositifs d'entrée/ sortie, notamment écran et clavier.

Les réseaux mobiles ont aussi leurs contraintes:

- Moins de bande passante<sup>12</sup>
- Plus de temps de latence
- Une connexion moins stable
- Une disponibilité non-prévisible
- La mobilité des terminaux

Ces contraintes posent plusieurs problèmes pour réaliser du *streaming* sur mobile.

Un des problèmes majeur de ce type de *streaming* est la limite de la bande passante du réseau. Ce problème peut être résolu avec des *codecs* ayant des algorithmes de compression plus performants.

Par contre, les appareils mobiles ont une forte limitation, à cause de la capacité limitée du processeur qui est liée à la consommation d'énergie. Ainsi, la puissance de calcul pour décoder est restreinte.

Un autre problème posé par la nature des réseaux *wireless* est la fiabilité qui n'est pas possible de garantir.

De plus les réseaux de téléphonie mobile sont des réseaux cellulaires<sup>13</sup>. Dans ces réseaux il existe un phénomène de *handover*<sup>14</sup>, qui est la transition d'une cellule à

---

<sup>11</sup> Réseau à commutation des paquets: réseau sur lequel les données sont transférées en paquets qui sont acheminées indépendamment

<sup>12</sup> Bande passante: le débit de transmission.

<sup>13</sup> Réseau cellulaire: réseau sans fil composé des émetteurs qui forment les cellules de couverture.

<sup>14</sup> *Handover*: phénomène de changement de cellule sur un réseau cellulaire.

l'autre. Dans ce cas la connexion n'est pas coupée mais il existe un fort délai. Celui-ci peut affecter le flux, avec des retards ou des pertes dans la distribution des paquets.

En plus, si nous prenons en compte un environnement avec de nombreuses petites cellules supportant un nombre important d'utilisateurs, comme dans une ville, en cas de mobilité le changement continu des cellules pourra rendre le *streaming* impossible.

Dans l'article [11] ils démontrent que la mobilité affecte le *streaming* dans les réseaux *WLAN*, donc on peut supposer qu'il en est de même pour les réseaux cellulaires. Même dans le cas idéal sans changement de cellule.

## 2.2 LES STANDARDS DU MOBILE STREAMING

Il existe principalement deux standards liés au *mobile streaming* [27, 45, 46, 47, 48, 49]:

- Le *3GPP PSS (Third Generation Partnership Project Packet Switched Streaming Service)*
- Le *MPEG-4 (Moving Picture Experts Group 4)*

Le *3GPP PSS* a été définie par le *3GPP*<sup>15</sup>. Le *3GPP PSS* a pour mission de définir les *codecs*, les protocoles et le format *3GP* recommandés pour faire du *mobile streaming*.

Le format *3GP* ainsi que tous les spécifications des *codecs* sont basés sur le travail du *MPEG-4 (Moving Picture Experts Group 4)*<sup>16</sup>.

Les protocoles du standard *3GPP PSS* seront vus ultérieurement.

Les *codecs* définies pour l'audio sont tous dans la famille *AMR (Adaptative Multi Rate)* spécifique pour le discours et dans la famille *AAC (Advanced Audio Coding)* :

- *AMR-NB (Narrow Band)*
- *AMR-WB (Wide Band)*
- *AAC LC (Low Complexity)*
- *AAC LTP (Long-Time Prediction)*

Les *codecs* définies pour la vidéo sont :

- *H.263 profile 0, level 10*
- *H.263 profile 3, level 10*
- *MPEG-4 Visual Simple Profile, level 0*

---

<sup>15</sup>*3GPP*: organisme créée en 1998 pour la standardisation des spécifications des réseaux *3G*. Maintenant il prend aussi en charge la maintenance du réseau *GSM*. Ses standards offrent une rétrocompatibilité avec la famille des réseaux *GSM*.

<sup>16</sup>*MPEG-4*: organisation qui fait des standards multimédia universaux pour la transmission des médias dans tous les domaines.



D'autres formats vidéo existent pour le mobile *streaming*, comme le *RealVideo*. Nous allons travailler avec les formats et les *codecs* du *3GPP PSS*, qui sont standardisés et les plus couramment utilisés.

Nous privilégions dans notre plate-forme l'emploi des formats et *codecs* définis par le standard *3GPP PSS*.

## 2.2.1 PROTOCOLES DU STREAMING

Dans les protocoles les plus connus nous allons d'abord voir les protocoles standardisés par le *3GPP PSS* et après les protocoles propriétaires.

Les principaux protocoles standardisés par la *IETF (Internet Engineer Task Force)*<sup>17</sup> et liés au *streaming* sont le *RTSP (Real-Time Streaming Protocol)*, le *RTCP (Real-Time Control Protocol)* et le *RTP (Real-Time Transport Protocol)*; les deux derniers travaillent sur *UDP*<sup>18</sup>; et le premier sur *TCP*<sup>19</sup>. Chacun a des fonctionnalités complémentaires, pour accomplir le *streaming*.

*RTP* est utilisé pour envoyer un fichier média en paquets, et proportionne aussi le support à la synchronisation pour la bonne reproduction des fichiers. C'est un protocole de transport standardisé du *streaming* [19,24].

Les deux autres protocoles sont des protocoles de contrôle liés à *RTP*.

*RTCP* a été créé pour compléter le protocole *RTP*. C'est un protocole de surveillance du flux *RTP* qui offre un service de *QoS*. Ce protocole va permettre de changer dynamiquement les caractéristiques du flux pour s'adapter aux conditions dynamiques du réseau [24].

*RTSP* est un protocole de session qui permet le contrôle et l'établissement des flux depuis le client, avec des commandes de contrôle vidéo: *play, pause, stop, record*. Typiquement *RTSP* utilise comme protocole de transport *RTP*, mais peut utiliser aussi *TCP* ou *UDP* directement [1, 4, 20].

Un protocole homologue et concurrent de *RTSP*, mais non standardisé est le *MMS (Multi Media Server)*. Il est un protocole propriétaire de *Microsoft* et la spécification n'est pas publique, par contre il existe des projets pour documenter ce protocole sans la collaboration de *Microsoft*, comme le *SDP (Streaming Download Project)*. Ce protocole travaille sur *TCP* ou *UDP* [18].

Nous devons aussi parler d'autres protocoles de session, qui complètent la gestion des sessions de *RTSP* et qui sont aussi standardisées [1].

Le *SIP (Session Initiation Protocol)* [23] est un protocole de session qui sert à contrôler différentes sessions pour plusieurs utilisateurs de mobiles. Il est utilisé pour

---

<sup>17</sup> *IETF*: groupe informel, international, qui participe à l'élaboration de standards pour Internet

<sup>18</sup> *UDP (User Datagram Protocol)*: protocole de transport en mode déconnecté, très employée sur internet.

<sup>19</sup> *TCP (Transmission Control Protocol)*: protocole de transport en mode connecté, très employée sur internet.

l'échange de données, comme le *streaming*. En plus, il travaille avec d'autres protocoles déjà existants comme le *RTP* ou le *RTSP*.

Le *SDP* (*Session Description Protocol*) [21] est aussi un protocole de session qui gère l'utilisation de *Mbone*<sup>20</sup> pour faire du *multicasting* et proportionne une description des *streams* aux utilisateurs, du *live* notamment. Comme *SIP*, il est utilisé pour plusieurs échanges de données, permettant aussi le *streaming*.

Le *SAP* (*Session Announcement Protocol*) [22] est un protocole informant les ressources des média disponibles, il retourne les sessions de *multicast* disponibles. Il travaille avec le *SDP* pour publier et décrire les sessions

En plus, il existe d'autres protocoles du *streaming* propriétaires (*RealPlayer*, par exemple a un protocole propriétaire, développé sur *RTSP*).

Dans l'illustration 2-2, nous pouvons voir la pile de protocoles du standard *3GPP PSS*.

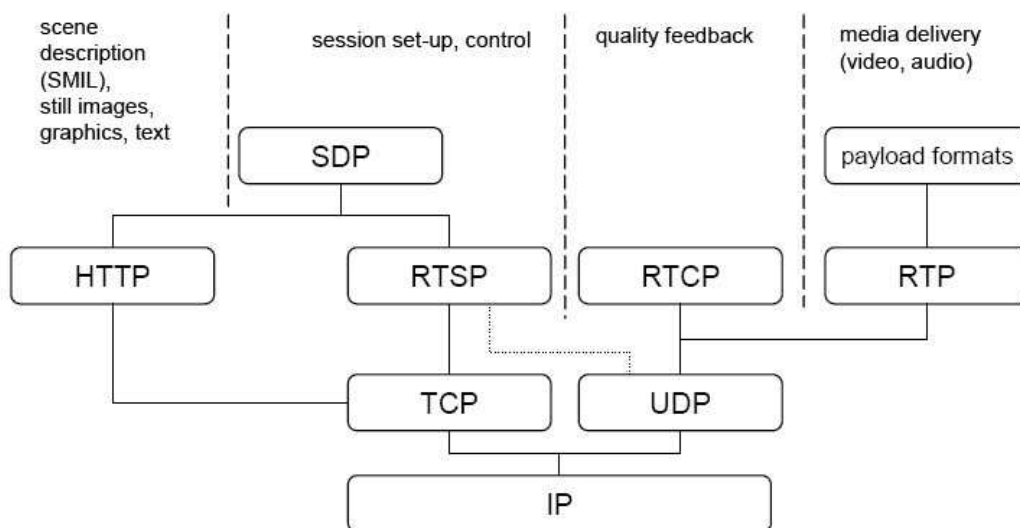


Illustration2-2: Pile de protocoles du standard *3GPP PSS*

Le *streaming* sur *TCP* et sur *HTTP* sont exclus de ce standard. Par contre ils sont employés pour la description et l'établissement de la session.

Dans ce standard ils sont recommandés les protocoles *RTSP*, *RTCP* et *SDP* pour l'établissement et le contrôle de la session. Le *RTP* est recommandé comme protocole de transport qui va utiliser toujours *UDP*. Finalement *RTCP* prends en charge les services de *QoS*.

<sup>20</sup> *Mbone*: le mot-valise *Mbone* vient de *Multicast Backbone*. Réseau qui supporte le *multicast* sur internet.

Selon une étude fait sur la période de l'année 2002 [9], le 37% du trafic de type *stream* utilisait *HTTP*, le 63% restant utilisait protocoles dédiés au *streaming*.

En plus 66% des sessions de *streaming* utilisent *TCP*, même si c'est moins performant que l'*UDP* (utilisé à 33%).

*HTTP* et *TCP* sont très couramment utilisés malgré les critiques faites à leur rencontre. *HTTP* étant le protocole courant pour Internet, il n'est pas bloqué par le pare-feux, tandis que les connexions *UDP* sont bloquées par défaut. En plus ils utilisent une infrastructure déjà implémenté, et donc stable.

Finalement, le protocole propriétaire de *streaming* le plus employé est le *MMS*, avec plus de 75% des requêtes et du trafic du média *streaming*.

Nous privilégions dans notre plate-forme l'emploi des protocoles de *streaming* dans le standard *3GPP PSS*.

## 2.3 LES SERVEURS DE STREAMING

Il existe deux façons de faire du *streaming*, en utilisant:

- Des serveurs *web*
- Des serveurs de *streaming*.

Le *streaming* avec des serveurs *web* travail sur les protocoles *HTTP* et *TCP*.

Les serveurs de *streaming* utilisent en plus des protocoles spécifiques basés sur *TCP* ou aussi *UDP*.

Le *streaming* avec des serveurs *web* a l'avantage de fonctionner sur une infrastructure *web*, mais le *streaming* avec des serveurs dédiés est beaucoup plus performant [17] :

- Le débit de la transmission est plus performant en travaillant sur *UDP*. En effet la nature du *streaming* n'impose pas la récupération des paquets perdus comme sur *TCP*. Cette récupération est inutile et dégrade la performance
- Le flux est adapté dynamiquement aux conditions du réseau grâce à des services de *QoS*
- Il offre des services avancés de contrôle de flux (*play, pause, etc.*)
- Il accepte le *multicast*<sup>21</sup> pour supporter plusieurs clients sur un seul flux. De cette façon la capacité d'une machine est beaucoup plus grande qu'avec un serveur *web* traditionnel

Plusieurs serveurs de *streaming* sont proposés sur le marché, mais faire un choix n'est pas évident. Les facteurs les plus importants à prendre en compte dans le choix d'un serveur de *streaming* sont :

- Le support des *codecs* d'audio et vidéo, notamment dans notre cas, ceux employées pour le *streaming* sur appareil mobile

---

<sup>21</sup> *Multicast*: diffusion des données vers plusieurs clients en partageant l'envoi des données pour les différents clients.

- Le support des protocoles de *streaming*
- La performance et la capacité du serveur
- La possibilité d'encoder dynamiquement des médias
- Un support multi plate-forme

Nous allons analyser les serveurs les plus connus du marché, en incluant des serveurs spécifiques du *streaming* sur mobile. Les serveurs analysés seront:

- *DSS (Darwin Streaming Server)* et *QTSS (QuickTime Streaming Servers)* [15] développés par Apple
- *Helix* [13] de *RealNetworks*
- *VLC* [16] de *VideoLan*
- *PvServer* de *PacketVideo Network Solutions* [14] qui est un serveur spécifique du *mobile streaming*

Il y a plusieurs serveurs propriétaires comme celui de *Microsoft, Windows Media Services 9 Series*. Il ne sera pas analysé ici car il ne supporte pas le standard *3GPP*.

Un autre serveur propriétaire, *Flash Media Server 2* de *Adobe* [12], propose des solutions pour le *mobile streaming*, mais cette solution est limitée par l'utilisation du *player Flash*.

Dans le tableau 2-1 nous pouvons voir les formats supportés par chaque serveur. Tous les serveurs analysés supportent le standard *3GPP*.

Dans le tableau 2-2 nous pouvons voir les autres fonctionnalités. Tous les serveurs acceptent les protocoles *RTP/RTSP*. Par contre il y a seulement deux serveurs qui peuvent faire de l'encodage et de l'enregistrement dynamique des flux. Ce sont *VLC* et *pvServer*.

Le serveur de *PacketVideo Network Solutions* est sous licence propriétaire, alors que le serveur *VLC* qui est *open-source*<sup>22</sup> est sous licence *GPL(General Public Licence)*<sup>23</sup>.

Même si le serveur *VLC* n'offre pas la fonctionnalité pour récupérer des statistiques sur les flux, nous avons décidé d'utiliser cette solution car elle est *open-source*.

Il faut remarquer qu'à cause de la licence propriétaire du *codec AMR* qui est incompatible avec la licence *GPL*. *VLC* devra être recompilé manuellement en incluant l'*AMR*.

---

<sup>22</sup> *Open-source*: s'applique aux logiciels dont le code source est disponible et la redistribution est libre.

<sup>23</sup> *GPL*: une licence des logiciels libres.

	PROTOCOLES					
	Standards			Propriétaires		
	MPEG-4	MP3	3GPP	WindowsMedia	RealMedia	QuickTime
Helix	OK	OK	OK	OK	OK	OK
QTSS	OK	X	OK	X	X	OK
DSS	OK	X	OK	X	X	OK
VLC	OK (compilation spécifique)	OK	OK	OK	X	OK
pvServer	OK	X	OK	OK	X	X

Tableau 2-1: Les serveurs de *streaming* et les formats supportés

	RTP / RTSP	Multi plate-forme	Encodage dynamique	Enregistrement dynamique
Helix	OK	Windows, Linux et Unix	X	OK (MP3 et RealMedia)
QTSS	OK	Windows, Linux et Solaris	X	X
DSS	OK	OSX	OK (modules extra)	X
VLC	OK	Windows, Linux et OSX	OK	OK
pvServer	OK	Linux	OK	OK

Tableau 2-2: Fonctionnalités des serveurs de *streaming*

## 2.4 SOLUTIONS EXISTANTES DU STREAMING

Une exemple d'application du *streaming* est *MarconiNet* [7]. Il s'agit d'un système de diffusion de radio et télé sur IP travaillant avec toute la suite de protocoles : *SAP*, *SDP*, *RTSP*, *RTP* et *RTCP*.

Un autre exemple d'application de *streaming* est la téléphonie sur Internet, aussi connue comme *VoIP*<sup>24</sup> (*Voice over IP*), [28] utilisant aussi les mêmes protocoles standardisés incluant *RTSP*.

Dans le système *freebox* de *free* [6] Internet, la télévision, la radio et la téléphonie sont inclus dans la même plateforme, grâce à la technologie du *streaming*.

Nous pouvons deviner dans ces exemples la tendance à intégrer avec le réseau *IP* les média classiques : radio, télévision et téléphonie. Cette intégration permettra l'homogénéisation des technologies employées par chaque média. Cela est possible grâce entre autres à la transmission des flux en temps réel sur Internet, à l'utilisation d'une suite de protocoles standardisés et bien modulaire mais aussi l'amélioration des réseaux.

<sup>24</sup> *VoIP*: technique qui permet de communiquer la voix sur un réseau TCP/IP comme internet.



### 3 INTERNET SUR LE MOBILE

Pour accomplir l'accès à l'Internet sur le mobile c'est nécessaire faire une liaison entre le réseau *IP* et les réseaux cellulaires mobiles. Internet est un réseau à commutation des paquets tandis que les réseaux cellulaires sont des réseaux à commutation des circuits<sup>1</sup>. Sur l'infrastructure des réseaux cellulaires il existe des réseaux à commutation des paquets mise en place pour proportionner un accès performant à l'Internet.

Le *WAP* (*Wireless Application Protocol*) [41] est une spécification pour faire d'Internet sur les mobiles fait par le *WAP Forum* [39], qui maintenant appartient au *OMA* (*Open Mobile Alliance*)<sup>2</sup> [34]. La dernière version de la spécification est le *WAP 2.0*.

Pour faire du *WAP* depuis un dispositif mobile il est nécessaire se connecter à un réseau à commutation de paquets. Ultérieurement nous allons étudier les différents réseaux mobiles à commutation des paquets.

Les objectifs de la spécification *WAP* sont:

Permettre l'accès à l'Internet et à des services avancés aux dispositifs mobiles.

Définir une suite de protocoles complètement indépendants du réseau et des dispositifs d'accès.

Compatibilité avec d'autres standards déjà existants.

Le *WAP* doit adapter les protocoles Internet aux contraintes dues aux réseaux et aux dispositifs mobiles, notamment la bande passante limitée des réseaux et les dispositifs d'entrée/sortie particuliers des dispositifs mobiles.

Sur la illustration 3-1 nous pouvons voir l'architecture originale et la pile de protocoles du *WAP*.

D'autres configurations sont possibles. Pour plus d'informations sur les protocoles et ses possibles configurations toutes les spécifications sont sur le [39].

Dans cette architecture il y a un routeur *WAP* qui relie le réseau mobile et l'Internet. Nous remarquons que dans ce routeur il est fait une conversion des protocoles de la pile du *WAP* aux protocoles standard d'Internet.

Malgré l'effort des organismes comme l'*OMA*, le monde d'Internet mobile est très hétérogène. Déjà nous pouvons voir qu'ils existent plusieurs implémentations des spécifications du *WAP*. Les implémentations du *WAP 1.1* utilisent le langage du *markup WML 1.x* (*Wireless Mark-up Language*) [40], tandis que les implémentations du

---

<sup>1</sup> Réseau à commutation des circuits: réseau sur lequel les données sont transférées sur circuits physiques ou logiques réservés pendant toute la durée de la communication.

<sup>2</sup> *OMA*: organisation pour la standardisation des services sur les réseaux mobiles, il est formé par plusieurs entreprises dans le secteur; et son but le plus important est l'interopérabilité.

WAP 2.0 emploie le *XHTML MP* (*eXtensible HTML Mobile Profile*). Par contre, il existe au moins trois implémentations du *XHTML MP* sur les navigateurs [44].

En plus il existe d'autres implémentations propriétaires pour la communication dans des réseaux mobiles à commutation de paquets. La plus connue est *iMode* [36] qui appartient à *NTT DoCoMo*<sup>3</sup>, et qui est surtout utilisé en Asie, mais il démarre aussi ailleurs.

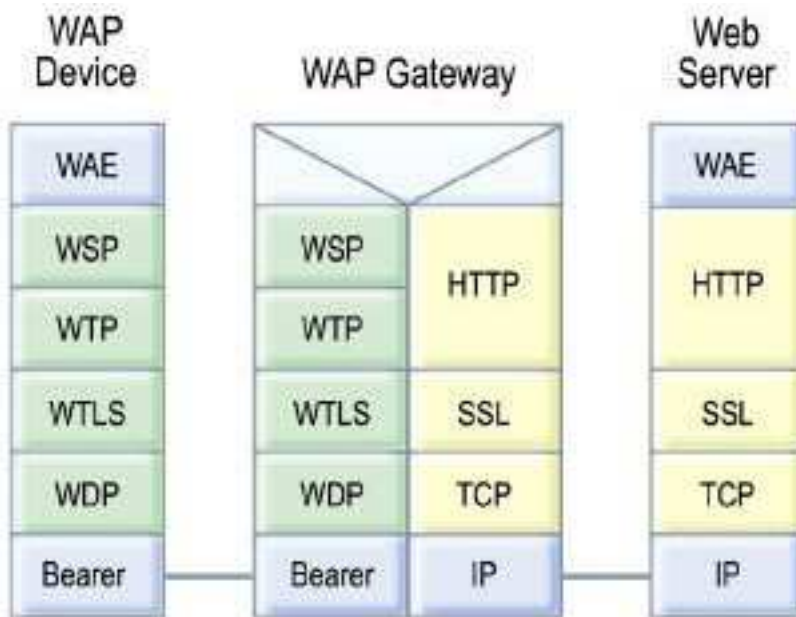


Illustration3-1: Pile de protocoles original du WAP

*iMode* utilise autres protocoles propriétaires et au niveau d'application il utilise encore une autre langage de *mark up*, le *CHTML* (*Compact HTML*) [30], compatible avec le *HTML*<sup>4</sup>.

Le projet de l'*OMA* pour assurer l'interopérabilité entre dispositifs mobiles est le *UAProf* (*User Agent Profile*) [38]. C'est basé sur une base de données avec une description des capacités de chaque dispositif. Par contre, plusieurs problèmes font que cette solution n'atteint pas son objectif.

- Les dispositifs ne sont pas tous présents dans la base
- 20% des liens ne sont pas mis à jour ou n'existent plus

<sup>3</sup> NTT DoCoMo: opérateur mobile numéro un au Japon.

<sup>4</sup> HTML (Hyper-Text Mark-up Language): le langage de mark up employé sur Internet pour les pages web.



- Il y a des erreurs dans les descriptions
- Il n'y a pas un standard accepté par tous les fabricants pour les descriptions, et vu que les descriptions sont faites par les fabricants il existe des problèmes de cohérence
- L'utilisation n'est pas gratuite [37]. Il est possible de faire une consultation gratuite en ligne mais c'est trop lent pour l'utilisation dynamique dans une application

*WURFL* [43] est un projet parallèle qui utilise le même principe que l'*UAProf*. Par contre la base de données de description des capacités est faite par une communauté de développeurs. Les sources sont ouvertes donc il est possible de consulter la base des données dynamiquement en local, et les corriger en cas d'erreur. Il propose un mécanisme d'héritage des capacités des dispositifs de la même famille, qui permet de compresser et de bien structurer tout la base des données dans un fichier *XML*<sup>5</sup>.

Avec la solution de *WURFL* nous pouvons créer le bon *mark up* après avoir interrogé le dispositif et avoir consulté la base de données de *WURFL*. Cela veut dire que nous allons créer des sites *WAP 1.1*, *WAP 2.0* et *iMode* dynamiquement.

Nous privilégions l'emploi de *WURFL* pour assurer l'interopérabilité dans les différents dispositifs, soit *WAP*, soit *iMode*.

---

<sup>5</sup> *XML (eXtensible Mark-up Language)*: langage de mark up générique avec le but de gérer l'hétérogénéité.



---

## 4 STREAMING SUR UNE PLATE-FORME D'INTERNET SUR LE MOBILE

L'intérêt de faire une plateforme de *streaming* sur le mobile est offrir des services de vidéo à la demande, de télécast, de radio sur *IP*, de téléconférence, etc.

Il existe deux façons de implémenter un service de *streaming* sur le mobile:

- Avec un service de vidéo appel<sup>1</sup>
- Avec un service d'Internet sur le mobile

La première solution est contraignante à la troisième génération des réseaux mobiles et travaille sur un réseau à commutation des circuits, donc c'est incompatible avec internet. Par contre cette solution offre des services de *QoS* comme réservation de la bande passante propres aux réseaux à commutation des circuits.

La deuxième solution est celui qui nous concerne. Cette solution fonctionne sur des réseaux mobiles de la deuxième et troisième génération et offre une convergence vers l'Internet.

Due à l'indépendance de la couche réseau des différentes spécifications d'*Internet mobile*, les réseaux de deuxième et troisième génération sont employés pour l'Internet sur le mobile. Donc, notre solution ne travaille pas sur un réseau précis, sinon sur un réseau mobile à commutation de paquets n'importe lequel.

Les réseaux mobiles ont des caractéristiques différentes au niveau de la bande passante. Le flux de *streaming* adéquat pour chaque réseau a des caractéristiques très différentes en relation avec la bande passante disponible.

Dans le dispositif mobile il existe trois éléments particuliers à chaque dispositif qui nous concernent:

- Le navigateur
- Les dispositifs d'entrée / sortie
- Le *player* du *streaming*

Le navigateur et les dispositifs d'entrée/sortie vont poser des problèmes de compatibilité au niveau de la navigation. Tandis que le *player* de *streaming* va poser des problèmes au niveau des flux de *streaming*.

Sur notre plateforme de *streaming* sur le mobile, Il y a trois problématiques à résoudre:

- La gestion des hétérogénéités des dispositifs mobiles au niveau de la navigation, par rapport au navigateur et aux dispositifs d'entrée/sortie
- La gestion des hétérogénéités des *players* de *streaming*
- La gestion des différentes caractéristiques des réseaux mobiles

---

<sup>1</sup> Vidéo-appel: une appel téléphonique avec de l'audio et du vidéo.

La solution passe par offrir une adaptation dynamique au réseau d'accès et aux différents dispositifs mobiles.

## 4.1 SOLUTIONS EXISTANTES

Il existe plusieurs implémentations propriétaires de *mobile streaming* sur une plateforme d'*Internet mobile*.

Quelques solutions existantes sont *Mobile YouTube* [31] et *MobiBase* [32].

Dans les deux cas, nous avons trouvé des problèmes de compatibilité selon le dispositif d'accès et le *player* présente dans le dispositif.

Le premier exemple de *YouTube* travaille sur *RTSP*, donc il fonctionne sur plusieurs *players*; par contre la solution de *MobiBase* utilise la solution propriétaire de *Flash Media Player 2*, donc il faut avoir installé avant le *Flash Player* qui utilise un protocole propriétaire.

*Mobile YouTube* est une sélection des vidéos présentes sur le site *web*. Nous supposons que la raison est que l'encodage en format mobile n'est pas dynamique, et donc celui-ci est fait statiquement seulement pour une sélection des vidéos.

Par contre, on ne connaît pas les détails de l'implémentation d'aucune de ces services.

Les implémentations existantes que nous avons étudiées sont loin d'être accessibles à tous les dispositifs mobiles au même niveau que dans le *web*.

Il existe des études sur la performance du *mobile streaming* comme les études faites dans le *MediaLab Teliasonera* [50, 51]. Dans ces études un *software* client / serveur dédiée est employé, notamment, le *software* du *player* qui a été développé spécifiquement pour récupérer des statistiques. Avec ces statistiques ils étudient la performance des protocoles de *streaming* et analysent les phénomènes de *handover* et la mobilité.

En plus, il existe des études de performance du *streaming* dans des réseaux mobiles de court porté, comme le *WIFI* [8].

Nous proposons dans cette étude une solution standardisée du *mobile streaming* sur une plate-forme d'Internet sur le mobile, avec capacité d'adaptation dynamique à chaque dispositif mobile et à chaque réseau mobile. De ce façon le même service sera accessible depuis n'importe quelle réseau ni quelle dispositif mobile.

## 5 PLATE-FORME DE STREAMING POUR LES DISPOSITIFS MOBILES AVEC ADAPTATION DYNAMIQUE

Notre objectif est d'offrir un service de *streaming* sur mobile qui soit adaptable dynamiquement à la plupart des dispositifs mobiles et aux différents réseaux.

Le service nécessite des informations concernant le dispositif mobile de l'utilisateur et le réseau d'accès, et s'adapte aux ces caractéristiques.

L'architecture de notre plate-forme est présentée dans l'illustration 5.1.

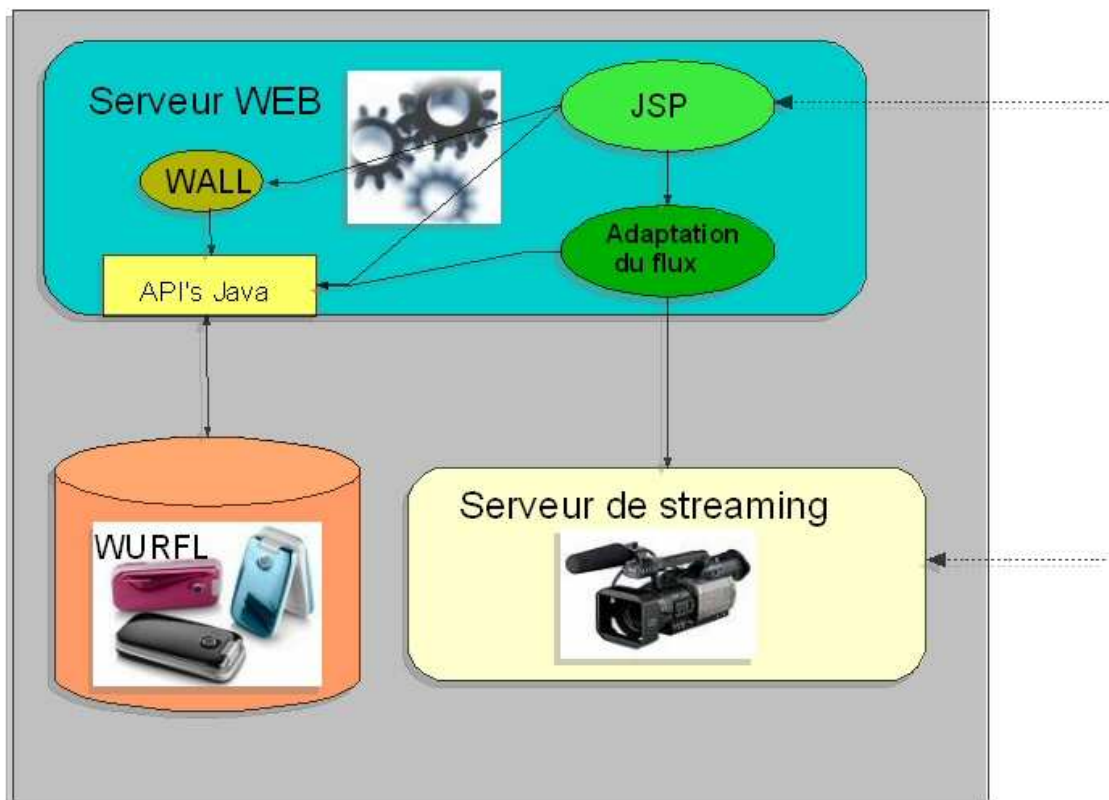


Illustration 5-1: Architecture de la plateforme de *streaming* pour les dispositifs mobiles avec adaptation dynamique

Les composantes de l'architecture sont les suivantes:

- La base des données *wurfl*
- Le serveur *web*
- Le serveur de *streaming*

La base des données *wurfl* (voir section d'Internet sur le mobile) contient les caractéristiques des différents dispositifs mobiles. Nous disposons d'une *API*<sup>1</sup> *Java* pour accéder à cette base de données, permettant d'obtenir les différentes caractéristiques d'un dispositif identifié par l'identificateur d'*user-agent*.

La base de données *wurfl* est dupliquée sur chaque serveur, nous permettant de faire des requêtes locales. Cette base de données est mise à jour par une communauté des développeurs, garantissant une réactivité importante pour chaque nouveau dispositif mobile apparu sur le marché.

Le serveur *web* va travailler avec la base des données *wurfl* pour proportionner des services d'Internet mobile adaptés à chaque dispositif.

La sous-couche d'adaptation de flux va employer la base des données pour faire une requête au serveur de *streaming* adapté aux caractéristiques du dispositif et du réseau.

Le serveur de *streaming* recevra la bonne requête du serveur *web* et fera le transcodage des média en conséquence, envoyant un flux personnalisé vers l'utilisateur.

La plate-forme décrite ci-dessous travaille dans le environnement montré dans l'illustration 5-2.

Plusieurs étapes sont nécessaires pour la personnalisation d'un service :

- Génération du **langage de *mark-up*** adéquat au dispositif, pour le bon affichage des pages *Web*. L'idée est d'exploiter au maximum les capacités d'un dispositif. Les dispositifs puissants recevront un contenu riche, mais en même temps les dispositifs limités recevront un contenu basique
- **Adaptation des objets complexes ne nécessitant pas le *streaming* (images, menus, ...)** aux dispositifs. Par exemple, le format et la taille des images affichées ou la mise en page par rapport à la taille de l'écran
- Adaptation du flux de *streaming* en fonction du dispositif mobile de l'utilisateur. Le flux doit utiliser des **codecs d'audio et vidéo** adéquates; le **débit** doit être aussi dans la capacité du dispositif mobile
- Adaptation au **réseau mobile** d'accès, en terme de débit selon la bande passante du réseau particulier

Nous nous intéressons dans ce mémoire aux deux dernières étapes.

---

<sup>1</sup> *API* (*Application Programming Interface*): interface de programmation.

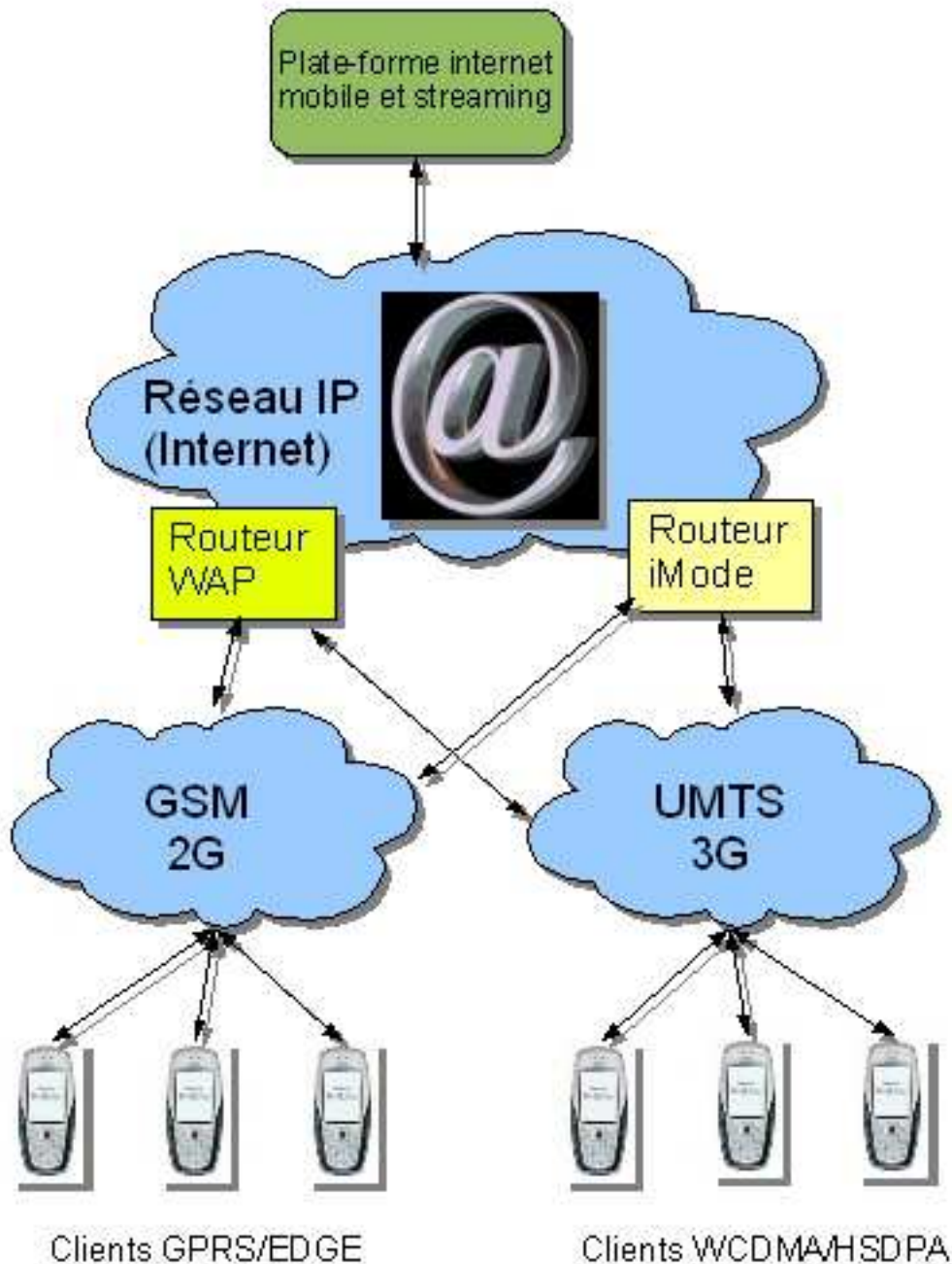


Illustration 5-2: La plateforme et son environnement

## 5.1 ADAPTATION AUX MARK UP

Comme présenté dans l'état de l'art, il existe plusieurs langages de *mark-up* qui ne sont pas compatibles entre eux. Les navigateurs existants sur les dispositifs mobiles sont loin d'être standardisés par rapport à ces langages, et chacun accepte un ou plusieurs *mark-up* particuliers. Comme *mark-up* nous mentionnons le *XHTML MP* (*Extensible*

HTML Mobile Profile), le WML 1.x (Wireless Mark-up Language) et le CHTML (Compact HTML), qui est le langage de l'iMode, et en plus il y a plusieurs implémentations différentes de XHTML MP dans les navigateurs.

La solution à ce problème passe par l'interrogation du navigateur pour générer le bon *mark-up*. Il est possible d'utiliser directement la base de données *wurfl* pour obtenir les *mark-up* supportés. Par contre, avec cette méthode il faudra coder un site WAP dédié pour chaque *mark-up*, avec les problèmes dérivés de maintenance de chacun des sites.

La solution que nous privilégions est d'utiliser *Wall*, une autre couche sur l'API Java d'accès au *wurfl*. *Wall* est une librairie de *tags JSP* (Java Server Pages). Les *tags wall* permettent de coder une page *web* dynamique qui génère le *mark-up* adéquat selon le dispositif. De ce façon il y aura qu'un seul site qui génère alternativement du code WML 1.x, XHTML MP ou CHTML.

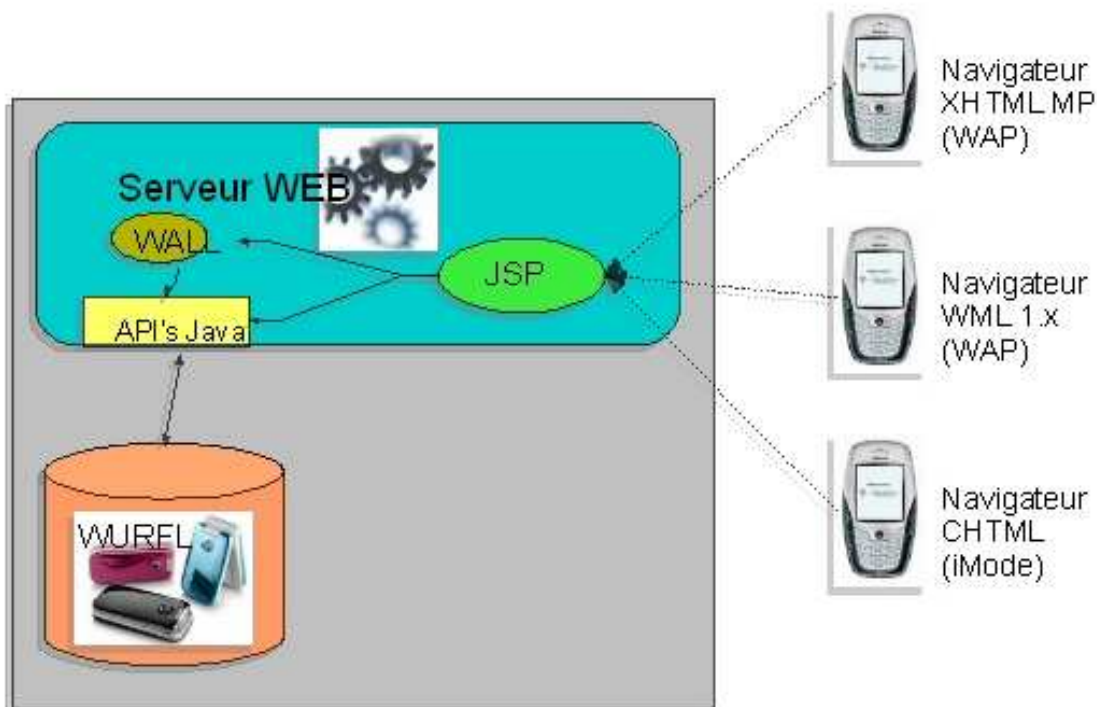


Illustration 5-3: Génération dynamique des *mark-up*.

La liste de code 1 sur l'annexe A correspond a une page WAP simple avec des *tags Wall* en combinaison avec des *tags JSP*. Il est composé d'un bandeau et un menu, même si on essaye de maîtriser au mieux le développement des sites WAP, les écrans des portables sont tellement petits qu'il faut rester dans la simplicité, sinon le site devient inutilisable.



## 5.2 ADAPTATION DE CONTENU

Un site WAP a des contenus, à part le *streaming*, à personnaliser. En utilisant les *API's Java* de *wurfl*, le service est adaptable à chaque dispositif. Par exemple, on peut personnaliser le format et la taille des images affichées ou la disposition des menus selon la taille d'écran du dispositif.

Pour personnaliser les images nous allons employer une solution basé sur les *API's Java* de *wurfl*, le *Image Server* [42]. L'*Image Server* fonctionne comme un filtre de façon à adapter dynamiquement le format et la taille des images selon l'*user-agent* du dispositif.

Les capacités sur la base des données de *wurfl* vont permettre adapter autres contenus et la mise en page.

## 5.3 ADAPTATION DES FLUX DE STREAMING

Dans un premier temps il faut remarquer que la bande passante nécessaire pour accomplir un flux de *streaming* vidéo est entre 32 et 384 kbps. Mais par contre avec une bande passante minimum de 32 kbps le *streaming* serait contraignante aux vidéos sans motion.

Pour un flux d'audio il est nécessaire un bande passante à partir de 5kbps.

La navigation sur pages simples comme celles qu'on propose demande seulement entre 8 et 32 kbps [25, 26].

Avec ces contraintes sur la bande passante premièrement nous allons considérer le *streaming* et la navigation sur le mobile dans les différents réseaux mobiles à commutation de paquets.

Après nous allons étudier l'adaptation dynamique pour chaque dispositif.

### 5.3.1 ADAPTATION AUX RÉSEAUX

Selon le réseau d'accès la bande passante disponible pour l'utilisateur change fortement.

Pour adapter nôtres flux aux différents réseaux mobiles, on propose une adaptation avec l'interaction de l'utilisateur. La solution est trop simple, mais aussi efficace. L'utilisateur va choisir sur un menu son réseau d'accès, de façon à déterminer la bande passante disponible pour sa session. Cette solution avec l'interaction de l'utilisateur pour déterminer sa bande passante est aussi employée sur internet.

Par contre, avec cette solution l'*inter-system handover (ISHO)*<sup>2</sup> n'est pas gérée. Le flux de *streaming* ne sera pas adapté au réseau après ce phénomène.

---

<sup>2</sup> *Inter-system Handover*: changement de réseau d'accès sans perte dans la communication, mais avec un certain délai et un changement de la bande passante.

Nous allons faire une analyse sur les débits et latences des différents réseaux mobiles qui sont les plus employés en Europe.

La deuxième génération des réseaux mobiles utilise le réseau à commutation des circuits *GSM (Global System for Mobile communications)*<sup>3</sup>.

La troisième génération emploie le réseau *UMTS (Universal Mobile Telecommunications System)*<sup>4</sup>.

Nous n'allons pas étudier les réseaux *CDMA* et *CDMA2000* qui s'utilisent plus en Amérique et en Asie [26].

Le *GPRS (General Packet Radio Service)*<sup>5</sup> était développé pour permettre la transmission des données sur un réseau à commutation de paquets basé sur *IP* en utilisant l'infrastructure du *GSM*. Le débit moyen de ce réseau est entre 30 et 40 kbps. Ce débit est à la limite pour faire un *streaming* de très basse qualité. La latence est d'environ 700 ms.

Le réseau *EDGE (Enhanced Data rates for GSM Evolution)*<sup>6</sup> était développé après le *GPRS* pour améliorer la bande passante en profitant mieux du partage des canaux. Le débit moyen est entre 100 et 130 kbps, et c'est avec ce réseau qu'on peut pour la première fois faire *streaming* sur un dispositif mobile de qualité. La latence est d'environ 350 ms.

Le *WCDMA (Wideband-Code Division Multiple Acces)*<sup>7</sup> due à sa basse latence permet une échange des données en temps réel et donc, la mise en place des applications de *VoIP (Voice over IP)* et de téléconférence. Le débit moyen est entre 220 et 320 kbps. La latence est d'environ 300 ms.

Le *HSDPA (HighSpeed Downlink Packet Acces)*<sup>8</sup>, est plus performant que le *WCDMA*. Le débit moyen est entre 550 et 1100 kbps. La latence est d'environ 150 ms.

Le réseau *GPRS* ne supporte pas la plupart des flux de *streaming*, il supporte que des vidéos sans mouvement (motion). L'adaptation à faire sur ce réseau est sur le contenu, donc un répertoire de contenu dédié sera nécessaire.

Le réseau *EDGE* accompli le flux de *streaming* n'importe la motion de la vidéo. Par contre la latence importante ne permet pas l'utilisation des applications en temps réel comme la téléconférence ou le *VoIP*.

---

<sup>3</sup> *GSM*: réseau mobile à commutation des circuits correspondant à la deuxième génération qui utilise la technique d'accès de *TDMA (Time Division Multiple Acces)*

<sup>4</sup> *UMTS*: un réseau mobile à commutation des circuits correspondant à la troisième génération qui utilise la technique d'accès de *CDMA (Code Division Multiple Acces)*

<sup>5</sup> *GPRS*: réseau à commutation des paquets considéré comme la génération 2.5

<sup>6</sup> *EDGE*: réseau à commutation de paquets plus performant que le *GPRS* considéré comme une technologie troisième génération, même si il est basé sur des technologies de la deuxième génération.

<sup>7</sup> *WCDMA*: réseau à commutation des paquets basé sur la technologie de troisième génération.

<sup>8</sup> *HSDPA*: réseau à commutation des paquets basé sur la technologie de troisième génération plus performant que le *WCDMA*.

Pour les réseaux basés sur le *UMTS* il n'y a pas des restrictions sur les applications de *streaming*, en permettant des applications en temps réel.

La navigation sera bien réalisée dans tous ces réseaux.

Dans cette étape d'adaptation les limites pour le débit du flux de *streaming* sont définies.

### 5.3.2 ADAPTATION AUX DISPOSITIFS MOBILES

La solution pour adapter un flux à chaque dispositif est basée sur la base des données de *wurfl*. Les capacités sur la base de données relatives au *streaming* sont les suivantes.

Premièrement si le téléphone supporte le *streaming*:

- `streaming_video`

Capacités relatives aux formats de vidéo supportés:

- `streaming_real_media_8`
- `streaming_real_media_9`
- `streaming_real_media_10`
- `streaming_3gpp`
- `streaming_mp4`
- `streaming_wmv`
- `streaming_mov`

Les formats standardisées et plus utilisées sont le *3GPP* et le *mp4* en profil *SVP niveau 0* (voir *Mobile Streaming Standards*).

Les capacités relatives aux tailles des vidéos sont les suivants:

- `streaming_video_qcif`
- `streaming_video_qcif_max_width`
- `streaming_video_qcif_max_height`
- `streaming_video_sqcif`
- `streaming_video_sqcif_max_width`
- `streaming_video_sqcif_max_height`

Les suivantes sont les capacités relatives à l'encodage de vidéo et audio:

- `streaming_video_max_bit_rate`
- `streaming_video_max_video_bit_rate`
- `streaming_video_min_video_bit_rate`

- `streaming_video_max_audio_bit_rate`
- `streaming_video_max_frame_rate`
- `streaming_video_size_limit`

Ces dernières valeurs sont dépendantes de la capacité du processeur de chaque dispositif. La capacité des processeurs des dispositifs mobiles est un des limitations les plus importantes.

Les dernières sont les capacités référents aux *codecs* d'audio et vidéo supportés.

- `streaming_video_vcodec_h263_0`
- `streaming_video_vcodec_h263_3`
- `streaming_video_vcodec_mpeg4`
- `streaming_video_acodec_amr`
- `streaming_video_acodec_awb`
- `streaming_video_acodec_aac`
- `streaming_video_acodec_aac_ltp`

Nous allons encoder dynamiquement pour chaque dispositif les paramètres suivants du flux:

- La taille des images, soit en *QCIF*<sup>9</sup> ou en *SQCIF*<sup>10</sup>.
- Le débit d'encodage d'audio et vidéo.
- Les *codecs* d'audio et vidéo.

La taille des vidéos au départ est *QCIF*, si nécessaire ils seront encodés dynamiquement vers *SQCIF*.

La politique pour le choix du débit d'encodage est offrir toujours le maximum possible, de telle façon qu'il soit bien supportée par le dispositif mobile et le réseau d'accès.

Les *codecs* audio et vidéo proportionnent des résultats avec des qualités différentes selon la bande passante qu'ils consomment.

Les *codecs* consomment plus de bande passante dans l'ordre suivant:

- Pour l'audio *AMR narrow band*, *AMR wide band*, *AAC* et *AAC LTP*.
- Pour le vidéo *H.263* niveau 0, *H.263* niveau 3 et *mpeg4*.

---

<sup>9</sup> *QCIF* (*Quarter Common Interchange Format*): format d'image numérique défini par l'*UIT* (*Union International de Telecommunications*) de taille 176 x 144

<sup>10</sup> *SQCIF* (*Sub-Quarter Common Interchange Format*): format d'image numérique défini par l'*UIT* (*Union International de Telecommunications*) de taille 128 x 96

Il faut remarquer que le *AMR* est un *codec* spécifique pour le discours, tandis que le *AAC* est plus généraliste et offre une meilleure qualité.

Les valeurs d'encodage pour les *codecs* d'audio sont les suivants:

- *AMR narrow band*, entre 1.80 et 12.20 kbps.
- *AMR wide band*, entre 6.60 et 23.85 kbps.
- *AAC* a des valeurs fixes; 18, 24, 32 et 64 kbps

Les *codecs* de vidéo ne sont pas limités comme ceux d'audio.

Nous privilégions l'emploi du *codec* le plus performant possible selon les capacités du dispositif et la bande passante disponible sur le réseau.

La solution proposée emploie l'encodage dynamique des média sur le serveur de *streaming* pour adapter les flux à chaque dispositif.

L'encodage des média est un processus très coûteux en termes de calcul. Dans la section suivante nous allons étudier la charge sur le *CPU* de plusieurs processus d'encodage.



## 6 ANALYSE DE LA PERFORMANCE DE L'ENCODAGE DYNAMIQUE DES MEDIA

L'encodage dynamique des médias est un processus très consommateur des ressources du processeur. Ce processus passe par la démultiplexation des flux (audio et vidéo), le décodage des flux, l'encodage de ces flux et la multiplexation.

Il existe plusieurs paramètres d'entrée et de sortie qui vont définir la performance du processus:

- Les conteneurs.
- Les *codecs* d'audio et de vidéo.
- Les débits des *codecs*.
- La taille des images.

Nous allons étudier les ressources nécessaires du processeur pour lancer plusieurs processus d'encodage dynamique. De telle façon que nous pourrions établir la capacité d'encodage du serveur de *streaming*.

Le serveur de *streaming* mis en place pour faire des tests est le *VLC Media Player 0.8.6c* sur une machine avec double processeur *Intel Pentium 4* à 3 Ghz et une mémoire *RAM* de 2 GB.

Nous avons employée comme source pour le processus d'encodage dynamique trois encodages différents de la même vidéo:

- Source 1: Vidéo en format *mov*, avec le *codec* vidéo *mjpg* à 30 fps, le *codec* audio *araw* à 63 kbps et une taille d'image 320 x 240.
- Source 2: Vidéo en format *3gp*, avec le *codec* vidéo *H.263* à 300 kbps et 15 fps, le *codec* audio *AAC-LC* à 56 kbps et une taille d'image *QCIF*.
- Source 3: Vidéo en format *3gp*, avec le *codec* vidéo *H.264* à 300 kbps et 15 fps, le *codec* audio *AAC-LC* à 56 kbps et une taille d'image 320 x 240.

Nous avons fait trois tests d'encodage dynamique pour les différentes sources:

- Test 1: *Codec* vidéo de sortie *H.263* à 42 kbps, *codec* audio de sortie *AAC* à 8 kbps et une taille d'image *QCIF*.
- Test 2: *Codec* vidéo de sortie *H.263* à 100 kbps, *codec* audio de sortie *AAC* à 20 kbps et une taille d'image *QCIF*.
- Test 3: *Codec* vidéo de sortie *H.263* à 300 kbps, *codec* audio de sortie *AAC* à 45 kbps et une taille d'image *QCIF*.

Ces processus d'encodage dynamique proposés emploient des paramètres de sortie en concordance avec le standard *3GPP PSS* (voir section *Mobile Streaming Standards*) et les flux de sortie sont adaptés à la bande passante des différents réseaux mobiles.

Dans le tableau 6-1 nous observons les résultats de consommation de *CPU* pour un, deux et trois processus d'encodage dynamique simultanés.

Nous pouvons constater que l'encodage le moins consommateur est avec la source 2, résultat logique vu que les *codecs* de vidéo d'entrée et de sortie sont pareils.

Par rapport au débit d'encodage, le moins le débit, le moins consommateur de *CPU* le processus. Résultat celle-ci aussi prévisible.

CPU %	Test 1			Test 2			Test 3		
Source 1	17,20%	33,60%	51,10%	18,50%	36,20%	53,90%	19,90%	36,20%	56,90%
Source 2	10,80%	23,20%	33,10%	12,50%	24,20%	39,30%	14,70%	29,30%	45,50%
Source 3	19,40%	40,20%	61,80%	20,70%	41,80%	61,20%	22,10%	47,90%	70,20%
Nb processus d'encodage	1	2	3	1	2	3	1	2	3

Tableau 6-1: Résultats de consommation de *CPU* de l'encodage dynamique

Par contre, ce qui n'était pas si prévisible est l'ordre de consommation de *CPU*. Dans le cas le plus favorable, source d'entrée numéro 2 et test 1, nous remarquons que un processus d'encodage consomme 10.8% des ressources de *CPU* de la machine et 33.1% pour trois processus simultanées.

Selon ces résultats nous pouvons établir que les encodages qui utilisent des *codecs* d'entrée et de sortie différents sont beaucoup plus consommateurs de *CPU*.

Pour le cas qui nous concerne il est intéressant d'établir le nombre maximum de processus d'encodage dynamique qui peuvent être exécutés sur cette machine. Ces résultats sont obtenus avec une interpolation linéaire sur les valeurs obtenues lors des tests, donc les résultats de l'interpolation ne sont que des approximations.

Les graphiques 6-1, 6-2 et 6-3 montrent les résultats de l'interpolation pour les sources 1, 2 et 3.

Les résultats de l'interpolation montrent que pour le plus favorable des cas, il sera possible d'encoder dynamiquement sur la machine de test jusqu'à huit flux de *streaming* dynamiquement. Par contre avec le *codec* d'entrée *H.264* la machine pourra encoder que quatre flux.

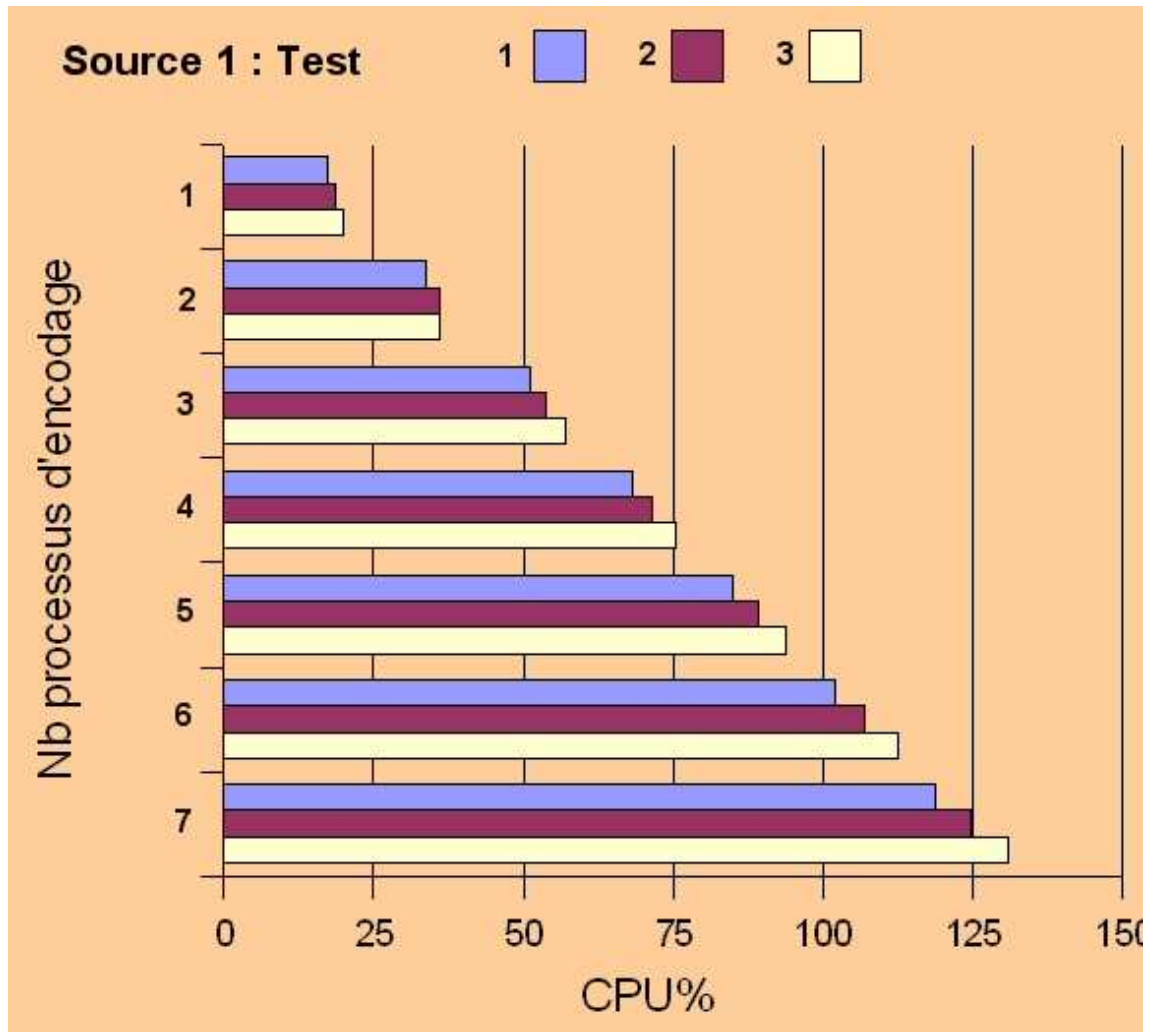
Ce résultat n'est pas très favorable pour une application de vidéo à la demande, dans laquelle il sera nécessaire un processus d'encodage pour chaque requête.

Par contre, pour une application de broadcast du live, comme le *télécast* ou la radio sur *IP*, il sera faisable de lancer plusieurs processus d'encodage dynamique, vu qu'un seul flux est employé pour plusieurs clients. De cette façon le service va offrir le même flux avec différentes caractéristiques d'encodage pour atteindre la plupart des clients.

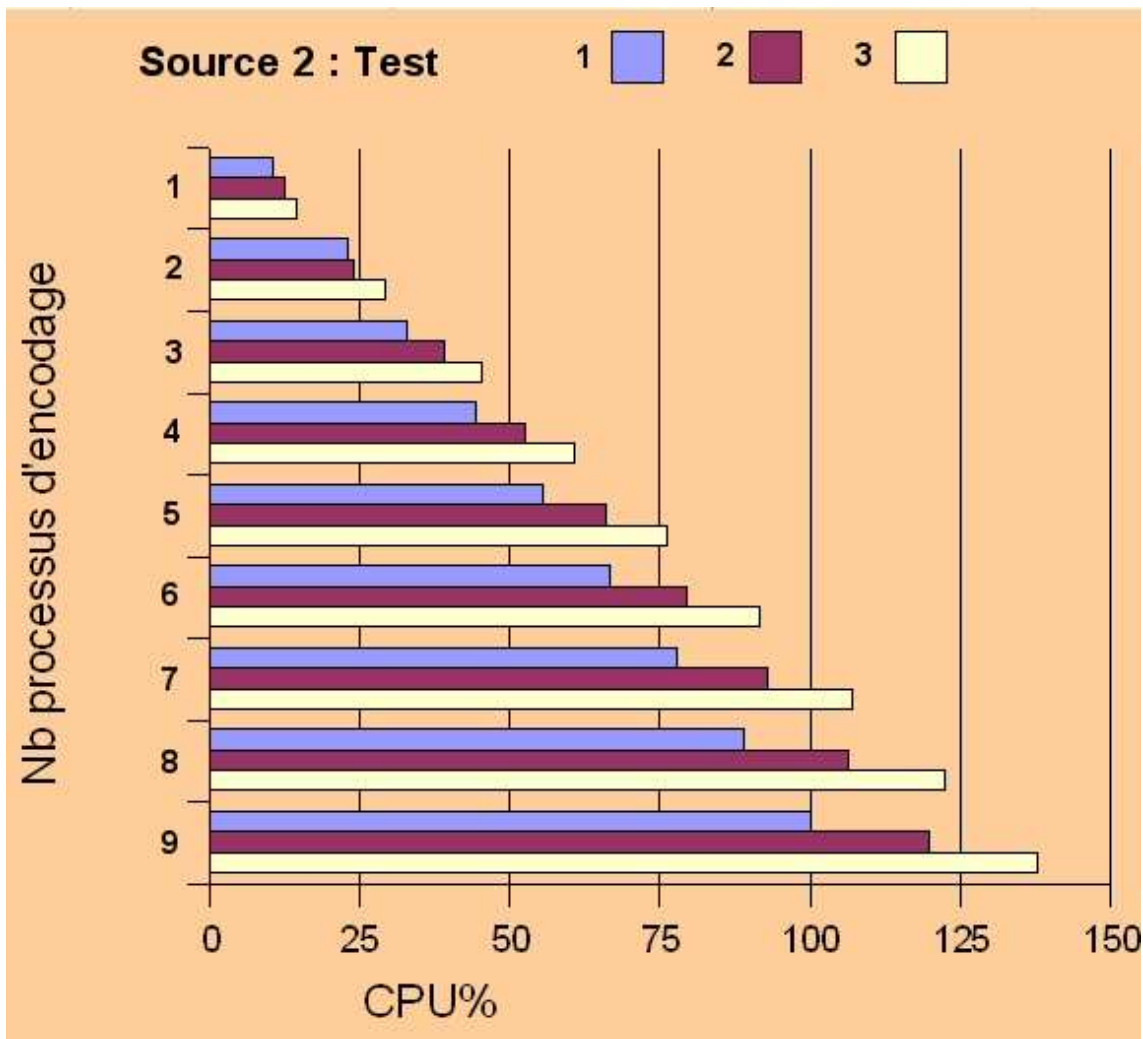


Nous remarquons comme conclusion que nous privilégions l'emploi de l'encodage dynamique en modifiant le débit et la taille des images, mais sans modifier les *codecs* ni les conteneurs, de façon que le processus soit moins consommateur des ressources de *CPU*.

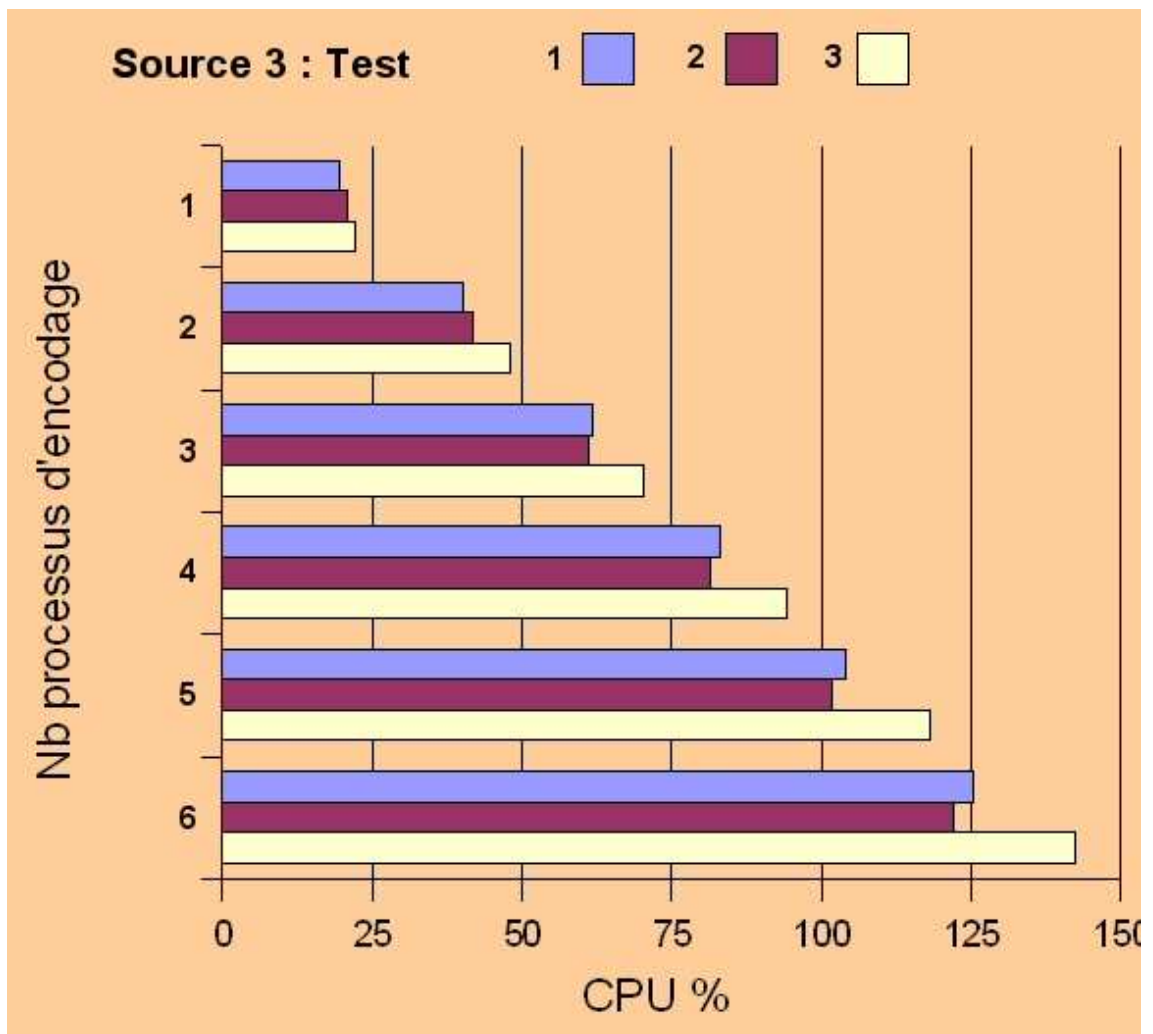
L'utilisation d'une machine plus puissante en terme de processeur est recommandée pour héberger un plus grand nombre des encodages dynamiques. L'emploi d'une plateforme de *streaming* industrielle peut minimiser grandement la consommation de *CPU*.



Graphique 6-1: Consommation de CPU par la source 1



Graphique 6-2: Consommation de CPU par la source 2



Graphique 6-3: Consommation de CPU par la source 3



## 7 CONCLUSIONS ET PERSPECTIVES DE FUTUR

Nous avons étudié l'énorme hétérogénéité qui règne dans le domaine des dispositifs mobiles, principalement sur trois domaines:

- Les protocoles d'Internet mobile
- Les dispositifs mobiles
- Les réseaux mobiles

Ils sont tous des domaines commerciaux très actifs avec des changements rapides et continus.

Il existe des organismes qui travaillent pour la standardisation des services mobiles comme l'*OMA* ou le *3GPP*. Par contre sont des organismes gérés par des fabricants des mobiles et fournisseurs des services de téléphonie mobile et donc, la concurrence entre eux est forte et chacun défend ses positions dans le marché.

Dans cette étude nous proposons une gestion de cette hétérogénéité. Particulièrement dans les domaines:

- De l'Internet mobile, avec ses plusieurs formes sur les dispositifs mobiles, voir *WAP* ou *iMode*
- Du *streaming* sur mobile sur
  - Les différents réseaux mobiles à commutation des paquets
  - Les différents dispositifs mobiles

Pour l'Internet mobile il est nécessaire gérer l'existence des différents standards sur les réseaux mobiles, comme le *WAP 1.0*, *WAP 2.0* et le *iMode*.

Pour les dispositifs mobiles il existe énormément des marques, des modèles et des nouvelles fonctionnalités qui sortent dans les nouveaux dispositifs. Le but de notre solution est de développer des applications compatibles avec tous les dispositifs, mais aussi en bien exploitant les possibilités de chaque dispositif.

Nous proposons une solution pour mettre en place des services d'Internet sur mobile avec *streaming* multi dispositif et multi protocole.

Cette solution fait une adaptation dynamique du service aux différents protocoles d'Internet mobile, différents réseaux mobiles et différents dispositifs mobiles. Elle est basée sur une base des données avec toutes les descriptions des dispositifs.

Pour offrir un service de *streaming* il est nécessaire d'adapter le flux dynamiquement. L'adaptation dynamique des flux passe par l'encodage dynamique des média.

Dans une dernière étape nous avons fait une étude sur la consommation de processeur du processus d'encodage.

La consommation de processeur est plus grande que prévu, de façon qui il restreint l'emploi de l'encodage dynamique. Dans le cas le plus favorable un seul processus d'encodage consommait plus de 10% de la capacité de processeur de la machine de test.

Donc, il n'est pas possible d'employer l'encodage dynamique d'une façon intensif. Il faudra employer des serveurs de *streaming* dédiés avec une grande capacité de processeur et peut-être une plate-forme de *streaming* industrielle.

Par rapport aux réseaux mobiles, il existe plusieurs réseaux mobiles actifs, vu que l'adoption de une nouvelle technologie a un temps de transition et que le monde des réseaux mobiles avance à une vitesse énorme. Dans les réseaux actifs aujourd'hui il y a des très grosses différences de bande passante et de latence, de façon qu'il faille adapter les services pour offrir les bons services à chaque utilisateur selon le réseau d'accès.

L'adaptation proposée est une solution simple mais effective, qui demande l'interaction de l'utilisateur pour choisir les capacités de son réseau d'accès. Ce solution était employé sur Internet dans une étape antérieure de développement des connections Internet homologues à l'actuelle sur les mobiles.

Ce n'est pas possible offrir un flux de *streaming* dans tous les réseaux à commutation des paquets mobiles. Sur le réseau *GPRS* le flux de *streaming* est limité à vidéos sans motion et avec une qualité pas très grande.

Par contre, sur le reste de réseaux avec une bande passante plus grande c'est possible faire des flux de *streaming* sans problèmes en adaptant les *codecs*, *bit rate* et *frame rate* des vidéos pour optimiser les possibilités de chaque réseau.

En plus, sur les réseaux à commutation des paquets de troisième génération, la basse latence permet de faire des flux de *streaming* pour des applications de *streaming* en temps réel comme le *VoIP* et la téléconférence.

Notre solution pour gérer les différents protocoles d'application et les dispositifs mobiles dépend de la bonne maintenance de la base des données de *WURFL*, donc même si cela fonctionne maintenant il n'est pas une solution à long terme.

Par rapport aux réseaux mobiles, l'identification du réseau par l'utilisateur peut poser des problèmes. D'abord l'utilisateur peut être qu'il ne connaisse pas le réseau qu'il utilise. Et aussi, peut-être que en recevant un flux de *streaming* l'utilisateur change de réseau, en changeant tous les limitations particuliers au réseau de début, et éventuellement arrivant à un mauvais *streaming*.

Nous avons vu que l'Internet sur les dispositifs mobiles avec les services les plus avancées d'Internet comme le *streaming* est un domaine en plein expansion, par contre son développement dépend grandement des influences du marché, et donc par le moment il n'y a pas un standard commun à tous les fournisseurs de services et fabricants des dispositifs mobiles. La solution proposée est simplement une façon temporelle de gérer toutes ces hétérogénéités.

Par contre, selon les prévisions les mondes des télécommunications et d'Internet vont converger, de façon qu'ils aillent utiliser tous le même réseau, voir *IP* ou un autre. La performance des réseaux mobiles physiques ne permet pas cela par le moment. Et en attendant il faudra trouver des solutions facilement déployables pour sauver les incompatibilités.

---

## 8 REMERCIEMENTS

Nous voulons remercier l'aide prêté par toute l'équipe de 3G-Factory, spécialement Jean François Hugot, Thierry Barnier et Dante Tota. Aussi mentionner un élève de l'INSA Lyon Axel Haussant qui nous a aidé avec des problèmes techniques. Finalement remercier au tuteur Vasile-Marian Scuturici son aide et compréhension.





## 9 BIBLIOGRAPHIE

### 9.1 Streaming

- [1] A Streaming Architecture for Next Generation Internet. A. Dutta, H. Schulzrinne. 2001
- [2] Barclay Dutson, Claudia Dutson, and Stephen Drayson. New opportunities in streaming media report. In *Vision Consultancy Group*, September 2000.
- [3] Broadband Content Splits. Bruce Kasrel, Josh Bernoff, and Meredith Gerson. In Forrester Research, October 2000.
- [4] Design Considerations for an RTSP-Based Prefix-Caching Proxy for Multimedia Streams. S. Gruber, J. Rexford, A. Basso. Septembre 1999.
- [5] Enterprise Streaming: different challenges from internet streaming. John Apostolopoulos, Mitchell Trott, Ton Kalker, and Wai-Tian Tan. Streaming Media Systems Group, Hewlett-Packard Labs. May 2005
- [6] FreeBox. Free. <http://adsl.free.fr/>
- [7] MarconiNet - An Architecture for Internet Radio and TV Networks. A. Dutta, H. Schulzrinne, Y. Yemini. 1999.
- [8] Performance live video streaming in 802.11b WLAN environnement under different load conditions. Yevgeni Koucheryavy, Dmitri Moltchanov, Jarmo Harju. Institut of communication engineering. Tampere University of Technology. November 2003
- [9] Streaming VideoTraffic: Characterization and Network Impact. J. Merwe, S. Sen, C. Kalmanek. AT&T Labs Research. August 2002
- [10] Streaming Video over the Internet: Approaches and Directions. Dapeng Wu, Yowei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, Jon M. Peha. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 11, NO. 3, MARCH 2001
- [11] The Effects of Mobility on Wireless Media Streaming Performance. Guangwei Bai, CareyWilliamson. Department of Computer Science. University of Calgary. 2004

### 9.2 Les serveurs de streaming

- [12] Flash Media Server 2. <http://www.adobe.com/products/flashmediaserver/>
- [13] Helix Universal Server Administration Guide. RealNetworks 2002.
- [14] pvServer 5.0 Product Description. PacketVideo Network Solutions. Mars 2007.
- [15] QuickTime Streaming Server 5.5 Administration. 2005
- [16] VideoLAN Streaming HowTo. A. Lattre, J. Bilien, A. Daoud, C. Stenac, A. Cellier, J. Saman. VideoLAN Project 2005.

- [17] Web Server Vs. Streaming Server. MicroSoft Windows Media.  
<http://www.microsoft.com/windows/windowsmedia/compare/WebServVStreamServ.aspx>

### **9.3 Protocoles de streaming**

- [18] MMS Streaming Protocol. SDP(Streaming Download Project) Team. Decembre 2003.
- [19] RFC 1889 RTP: A Transport Protocol for Real-Time Applications. H. Schulzrinne. Janvier 1996
- [20] RFC 2326 Real Time Streaming Protocol. Schulzrinne, et. al. April 1998
- [21] RFC 2327 SDP : Session Description Protocol. M. Handley, V. Jacobson. Avril 1998.
- [22] RFC 2974 Session Announcement Protocol. M.Handley, C. Perkins, E. Whelan. Octobre 2000.
- [23] RFC 3261 SIP : Session Initiation Protocol. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R.Sparks, M. Handley, E. Schooler. Juin 2002.
- [24] RFC 3550 RTP: A Transport Protocol for Real-Time Applications. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. Juillet 2003.

### **9.4 Réseaux mobiles**

- [25] Data Capabilities: GPRS to HSDPA and beyond. Peter Rysavy. Septembre 2005
- [26] Delivering voice and data: comparing CDMA2000 and GSM/GPRS/EDGE/UMTS. CDMA Development Group. Decembre 2005
- [27] Packet Switched Streaming Service. MediaLab Teliasonera. Novembre 2003.
- [28] Internet telephony: architecture and protocols – An IETF Perspective. H. Schulzrinne, J. Rosenberg. 1999

### **9.5 Internet mobile**

- [29] Ch@rlie Th@i's i-browser. <http://www.charlietai.com/imode/>
- [30] Compact HTML for Small Information Appliances. Tomihisa Kamada W3C. Fevrier 1998.
- [31] Mobile Youtube <http://m.youtube.com/>
- [32] Mobibase <http://www.mobibase.com/>
- [33] Nokia Mobile Browser 4.0 SDK User's guide. Avril 2003
- [34] OMA : Open Mobile Alliance. <http://www.openmobilealliance.org>
- [35] OpenWave Phone Simulator V7.

- [http://developer.openwave.com/dvl/tools\\_and\\_sdk/phone\\_simulator/](http://developer.openwave.com/dvl/tools_and_sdk/phone_simulator/)
- [36] Site officiel d'imode de NTT DoCoMo  
<http://www.nttdocomo.com/services/imode/index.html>.
- [37] UAProf site. <http://www.uaprofile.com>
- [38] WAG UAProf. WAP Forum. Octobre 2001.
- [39] WAPForum. Open Mobile Alliance. <http://www.wapforum.org>
- [40] WAP Wireless *Mark-up* Language Version 1.3. WAP Forum. Fevrier 2000
- [41] Wireless Application Protocol Architecture Specification. Version 12. WAP Forum. Julliet 2001.
- [42] Wurfl Image Server. <http://wurfl.sourceforge.net/utilities/imageserver.php>
- [43] WURFL site. <http://wurfl.sourceforge.net>
- [44] XHTML Mobile Profile. WAP Forum. Octobre 2001.

## **9.6 Mobile Streaming**

- [45] 3GPP : Third Generation Partnership Project. <http://www.3gpp.org>
- [46] Moving Picture Experts Group. <http://www.chiariglione.org/mpeg>
- [47] MPEG-4. MediaLab Teliasonera. Aout 2004
- [48] Status of the 3GPP PSS Standard. Dr. Markus Kampmann Ericsson Research, Aachen. Juin 2004
- [49] Streaming in mobile networks. MediaLab TeliaSonera Finland. Augoust 2004
- [50] Streaming Tester Software for Mobile Networks. S. Tiainen. Tampere University of Technology, MediaLab Teliasonera. Juin 2004.
- [51] Testing Streaming in Mobile Networks. MediaLab Teliasonera. Juillet 2004.



## 10 ANNEXE A : EXEMPLE DE CODE DE GÉNÉRATION DYNAMIQUE DES LANGAGES DE *MARK-UP*

```

<%@ taglib uri="/WEB-INF/tld/wall.tld" prefix="wall" %><%@ taglib
prefix="c" uri="http://java.sun.com/jsp/jstl/core"
%><wall:document><wall:xmlpidtd />

<!--
loads the capabilities of WURFL
-->
<wall:load_capabilities/>
<wall:head>
<wall:title >Exemple Menu</wall:title>
<!-- Sets the good gridsize concordant with the resolution -->
<c:set var="gridsize" value="2" />
<wall:cool_menu_css colnum="${gridsize}"/>
</wall:head>
<wall:body>

<wall:block>
<wall:img alt="" src=" bandeau.jpg"/>
<wall:br/>
</wall:block>

<wall:cool_menu colnum="${gridsize}">
<wall:cell>
<wall:img alt="" src="film.jpg"/>
<wall:a href="/" title="movies">Movies</wall:a>
</wall:cell>
<wall:cell>
<wall:img alt="" src="micro.jpg"/>
<wall:a href="/" title="videoclips">Videoclips</wall:a>
</wall:cell>
<wall:cell>
<wall:img alt="" src="music.jpg"/>
<wall:a href="/" title="music">Music</wall:a>
</wall:cell>

```

```
<wall:cell>  
<wall:img alt="" src="TV.jpg"/>  
<wall:a href="/" title="TV">TV</wall:a>  
</wall:cell>  
</wall:cool_menu>  
</wall:body>  
</wall:document>
```

## 11 ANNEXE B : RESULTATS DE LA GÉNÉRATION DYNAMIQUE DES *MARK-UP* ET L'ENCODAGE DYNAMIQUE DES IMAGES



Illustration 11-1: Pages WAP avec le simulateur *Openwave V7*



Illustration 11-2: Page WAP avec le simulateur *Nokia Mobile Browser*



Illustration 11-3: Page *i-Mode* avec le simulateur *i-Browser*



Dans les illustrations 11-1, 11-2 et 11-3 on peut voir, respectivement, le résultat des requêtes au serveur pour le code montré dans l'annexe A pour les simulateurs mobiles suivants:

- *Openwave V7 Simulator* [35], un simulateur WAP.
- *Nokia Mobile Browser 4.0* [33], un autre simulateur WAP.
- *Ch@rlie Th@i's i-browser* [29], un simulateur iMode.

A gauche se trouve la page *XHTML MP* généré pour le simulateur d'OpenWave, au centre se trouve la page *WML* crée pour le simulateur de Nokia et finalement à droite se trouve la page généré pour le *Ch@rlie Th@i's i-Browser* en *mark-up CHTML*.

Avec cet exemple nous montrons les possibilités de personnalisation des sites web pour chaque dispositif mobile au niveau de *mark-up* et au niveau des contenus complexes.