



institut de recherche en informatique
et systèmes aléatoires

SERVEUR STREAMING PROTEE

GUIDE UTILISATEUR

Version: 1.0

Date : 21 Juillet 2006.

Auteur(s):

Laurent Guillo



Table des matières

1. PRESENTATION DE PROTEE	7
2. ARCHITECTURE DU SERVEUR PROTEE.....	8
3. MISE EN ŒUVRE ET COMMANDES UTILISATEURS.....	10
3.1. MISE EN ŒUVRE.....	10
3.2. COMMANDES UTILISATEURS	11
3.2.1. <i>Start</i>	11
3.2.2. <i>Stop</i>	11
3.2.3. <i>Quit</i>	11
3.2.4. <i>Status</i>	11

Table des figures

Figure 1 : Architecture générale du serveur Protée.....	8
Figure 2 : Architecture détaillée.....	9

Glossaire

ADU : Application Data Unit.

IP : Internet Protocol.

NDU : Network Data Unit.

RTP : Real time Transport Protocol.

RTCP : Real-Time Control Protocol.

RTSP : Real Time Streaming Protocol.

SDP : Session Description Protocol.

TCP: Transmission Control Protocol.

UDP : User Datagram Protoco

1. Présentation de Protée

Ce document présente le serveur streaming « Protée » réalisé par l'équipe Temics de l'Irisa dans le cadre du projet RNRT COSINUS pour Windows XP. Ce serveur diffuse des flux vidéo scalables (spatialement, temporellement et en qualité) encodés par le codeur basé ondelette Wavix. Ces vidéos encodées peuvent éventuellement comporter des informations redondantes qui seront transmises sur des flux séparés.

La scalabilité des flux permet de s'adapter aux conditions du réseau en ajustant dynamiquement le débit d'émission et en réorganisant les flux encodés selon les résolutions disponibles. Une demande explicite du client peut déclencher aussi une réorganisation des flux.

L'évaluation des conditions du réseau s'effectue en fonction d'informations provenant des clients (qualité perçue) et de la couche liaison (approche « cross layer »).

Ce document présente tout d'abord l'architecture du serveur et les modules le composant. Sa mise en œuvre les commandes associées sont ensuite décrites.

2. Architecture du serveur Protée

Le serveur Protée comprend quatre modules principaux :

- un gestionnaire de client RTSP, chargé de gérer les connexions RTSP avec le client streaming vidéo,
- un fournisseur d'ADU, qui extrait les données du fichier encodé,
- un émetteur d'ADU qui découpe les ADU en autant de paquets réseau que nécessaire avant de les émettre en respectant les contraintes de débits,
- un module de régulation de débit qui en fonction des conditions du réseau tire parti de la scalabilité des flux Wavix pour adapter le débit d'émission.

Ces quatre modules sont présentés dans le schéma ci-dessous.

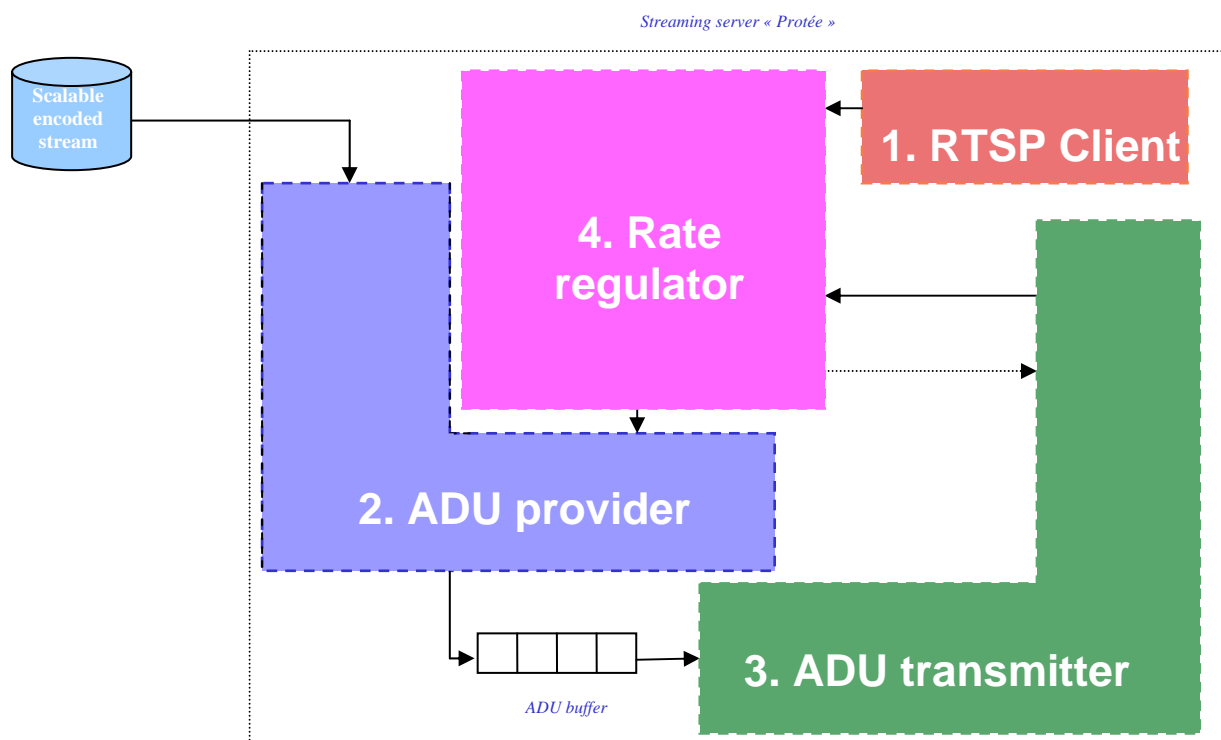


Figure 1 : Architecture générale du serveur Protée

Les trois premiers modules sont traditionnellement présents dans les serveurs streaming du marché. Le quatrième module est l'élément innovant qui permet d'exploiter le plus finement la scalabilité des flux encodés.

Le schéma ci-dessous présente l'architecture détaillée du serveur. Les quatre modules principaux sont éclatés en sous-modules de même couleur.

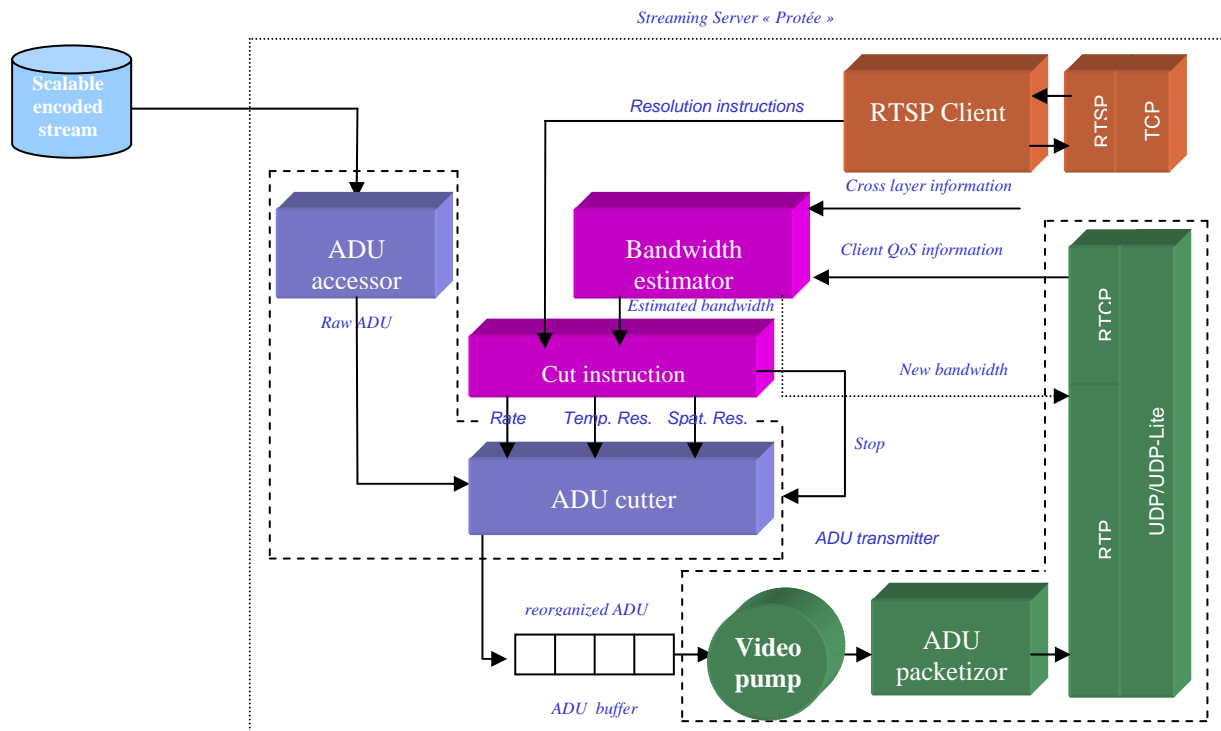


Figure 2 : Architecture détaillée

La fonctionnalité principale du serveur, le régulateur de débit, se décompose en deux sous-modules :

- l'estimateur de bande passante,
- le générateur d'instructions de coupes.

L'estimateur de bande passante permet de déterminer la bande passante disponible en fonction d'informations sur la qualité perçue par le client comme, par exemple, le taux d'événement de pertes et le calcul de paramètres comme le RTT. Cette estimation basée modèle se doit d'être TCP-Friendly afin de limiter les impacts sur les flux TCP concurrents transitant via le même réseau.

Une fois la bande passante estimée, sa valeur est transmise au générateur d'instructions de coupes. Ces instructions de coupes permettant d'exploiter la scalabilité des flux Wavix en précisant pour les ADU à émettre:

- la résolution spatiale (CIF ou QCIF),
- la résolution temporelle (30, 15 ou 7,5 hz),
- le débit.

Ces trois types de consignes ou instructions de coupes sont exploités par le « cutter » ou « réorganisateur » de bitstream afin d'extraire du flux scalable les résolutions demandées.

Ainsi, le serveur peut adapter son débit suivant les conditions du réseau et/ou les demandes explicites du client.

3. Mise en œuvre et commandes utilisateurs

3.1. Mise en œuvre

Le serveur streaming Protée se présente sous la forme d'une exécutable Windows « protee.exe ». Le serveur « Protée » se configure en prenant en compte les paramètres suivants:

- le nombre maximum de clients supportés à un même instant,
- le nombre maximum de connexions clientes en attente ou « backlog »¹,
- le nom et le chemin complet du répertoire où se trouvent les séquences vidéo,
- le numéro du port RTSP,
- la durée du timeout au bout duquel une session cliente est terminée si aucun message (RTSP ou RTCP) du client n'est arrivé au serveur,
- la taille des buffers de chaque session cliente contenant les ADU (plus petit ensemble de données décodable par un player),
- la capacité du serveur à supporter le protocole de transport UDP-Lite (1 le serveur supporte le protocole UDP-Lite, 0 non),
- le temps en milliseconde pendant lequel un thread prend le CPU.

Les valeurs de ces paramètres sont contenues dans la base de registre à partir de la clé suivante : « HKEY_LOCAL_MACHINE\SOFTWARE\Irisa\Protee ».

Sous cette clé, se trouvent les valeurs par défaut associées aux paramètres :

Nom	Type	Données
MaxBacklog	REG_DWORD	5
MaxNbOfClients	REG_DWORD	16
MediaDirectory	REG_SZ	C:\Protee\videos
RTSPPort	REG_DWORD	554
KeepAliveTimeout	REG_DWORD	200
ADUBufferSize	REG_DWORD	6
UDPLiteOn	REG_DWORD	1
TimeSlice	REG_DWORD	15

Si ces valeurs n'existent pas dans la base de registre, les paramètres sont créés dans la base de registre avec ces valeurs par défaut.

¹ Le « backlog » est un des paramètres de la fonction « listen ».

3.2. Commandes utilisateurs

Après avoir lancé le serveur en mode console (pour cela il faut entrer « `protee -c` »), l'utilisateur est invité à saisir une des commandes suivantes :

- `start`,
- `stop`,
- `quit`,
- `status`.

Ces commandes sont décrites ci-après.

3.2.1. Start

La commande « `start` » lance l'initialisation du serveur et le met en attente d'une connexion cliente.

En cas de succès, le message « Serveur en attente de connexion... » est affiché. Dans le cas contraire, le message « Serveur non lancé » est affiché éventuellement suivi d'un libellé d'erreur².

3.2.2. Stop

La commande « `stop` » arrête le serveur mais pas l'application Protée : le gestionnaire de commande est toujours en fonctionnement. Une commande « `stop` » ne peut être réalisée que si, au préalable, la commande « `start` » a été exécutée.

En cas de succès, le message « Serveur arrêté » est affiché. Dans le cas contraire, le message « Arrêt du serveur impossible » est affiché éventuellement suivi d'un libellé d'erreur.

3.2.3. Quit

La commande « `quit` » arrête définitivement le serveur et, contrairement à la commande « `stop` », le gestionnaire de commande est lui aussi arrêté. La commande « `quit` » ne peut être réalisée si le serveur est en attente d'une connexion client, il faudra d'abord exécuter la commande « `stop` ».

En cas de succès, l'application Protée est fermée et l'utilisateur se retrouve au niveau de l'invite système. Dans le cas contraire, le message « Sortie de l'application impossible » est affiché et éventuellement suivi d'un libellé d'erreur.

3.2.4. Status

La commande « `status` » permet de récupérer des informations sur le serveur. Les informations qui peuvent être obtenues sont :

- L'état du serveur (« Serveur arrêté », « en cours d'exécution »),
- Le nombre de clients connectés est affiché ainsi que leur adresse IP.

² Par exemple une erreur peut se produire si le port d'écoute RTSP n'est pas disponible.