



institut de recherche en informatique
et systèmes aléatoires

CRIQS

GUIDE UTILISATEUR

Version: 1.0.0

Date : 12 juillet 06

Auteur(s):

Laurent Guillo



Table des matières

1. QU'EST-CE QUE "CRIQS".....	5
2. INTERFACE UTILISATEUR.....	6
3. SCRIPT DE COMMANDES RTSP	7
3.1. UTILISATION DE VARIABLES	7
3.2. DESCRIPTION D'UNE COMMANDE RTSP	7
3.3. EXTRACTION D'INFORMATION DES REPONSES DU SERVEUR STREAMING.....	8
3.3.1. <i>GetControlFlux</i>	8
3.3.2. <i>GetSessionID</i>	8
3.3.3. <i>GetReturnedValue</i>	9
3.4. CONNEXION DES FLUX MULTIMEDIA.....	9
3.5. SUSPENDRE L'INTERPRETATION DU SCRIPT.....	10
3.6. EXEMPLE DE SCRIPT COMPLET	10
3.7. VERIFICATION SYNTAXIQUE D'UN SCRIPT	10
4. INFORMATIONS DE QUALITE DE SERVICE.....	12

Table des figures

Figure 1: Interface utilisateur de « Criqs ».....	6
Figure 2: Affectation d'une constante chaîne à une variable.....	7
Figure 3: Affectation d'un retour de fonction à une variable	7
Figure 4: Exemple de commande RTSP	8
Figure 5 : Résultat de l'interprétation d'une commande RTSP	8
Figure 6 : Exemple d'appels de « GetControlFlux »	8
Figure 7: Exemple d'appel de "GetSessionID"	9
Figure 8: Exemple de chaîne retournée par le serveur streaming	9
Figure 9: Exemple d'appel de "GetReturnedValue"	9
Figure 10: Exemple d'appel à "ConnectStream"	9
Figure 11: Exemple d'appel à "Wait"	10
Figure 12: Exemple de script complet.....	10
Figure 13: Syntaxe correcte.....	11
Figure 14: Exemple d'erreurs détectées	11
Figure 15: Importation des données de QoS dans Excel.....	12

1. Qu'est-ce que "Criqs"

« Criqs »¹ est un client streaming générique sous Windows permettant :

- d'accéder à un serveur streaming supportant le protocole de commande RTSP,
- d'établir les connexions aux flux multimédias,
- de calculer des informations de qualité de service.

« Criqs » ne permet pas toutefois d'afficher la vidéo ou de restituer le son des séquences transmises par les serveurs streaming².

Les commandes RTSP et fonctions de connexions aux flux multimédias sont rassemblées au sein d'un script dont la syntaxe est détaillée dans le chapitre 3. La génération d'information de qualité de service est une option présentée dans le chapitre 4.

Auparavant, l'interface utilisateur est décrite dans le chapitre 2.

¹ CRIQS : interpréteur de Commandes Rtp avec Information de Qualité de Service.

² Criqs a été utilisé avec succès avec les serveurs Darwin, Realnetworks (quand il diffuse des .mov) et le serveur Protée de l'équipe Temics de l'INRIA/IRISA.

2. Interface utilisateur

L'interface du client streaming « Criqs » comprend trois parties principales :

- « Script RTSP », elle permet l'accès au script de commandes RTSP et la vérification de sa syntaxe,
- « Exécution », activée une fois la syntaxe du script vérifiée, elle permet de lancer son exécution,
- « Affichage », elle contient le résultat de l'analyse syntaxique du script de commandes RTSP et celui de l'exécution.

Ces trois parties sont présentées dans la figure ci-dessous :

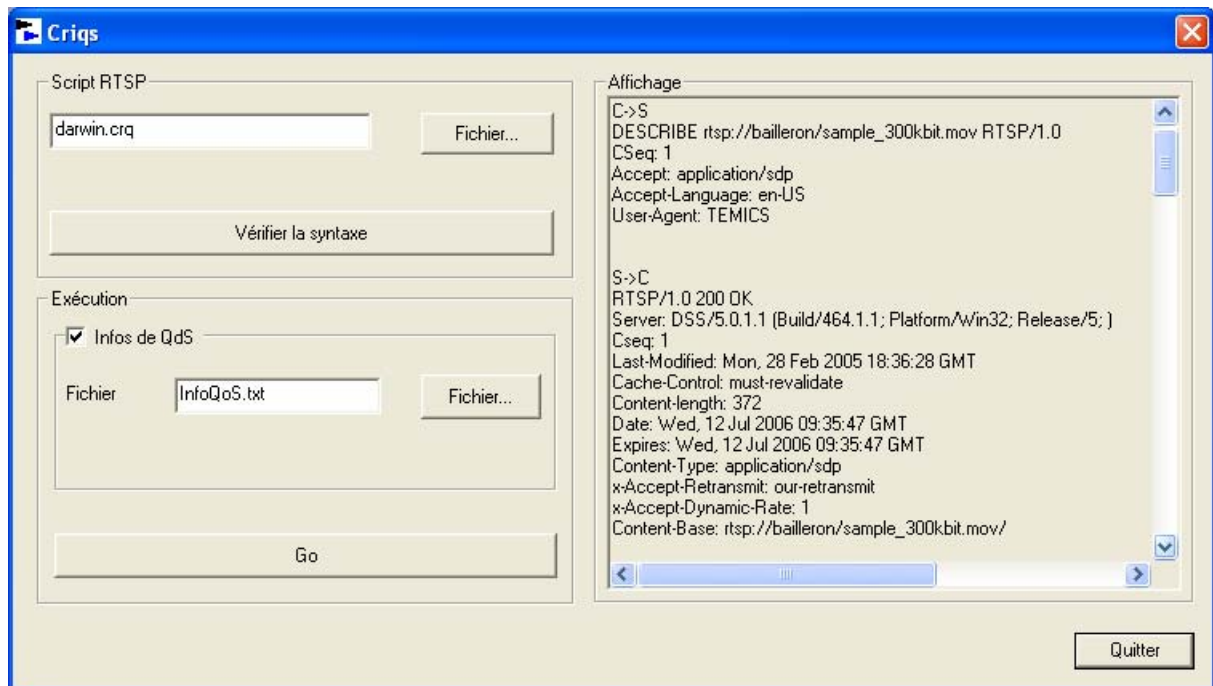


Figure 1: Interface utilisateur de « Criqs »

Au lancement de « Criqs », seule la zone « Sript RTSP » est active. L'utilisateur est alors invité à saisir le nom du fichier contenant le script RTSP ou sélectionner un fichier via le bouton « Fichier... ». Il doit ensuite vérifier que la syntaxe du script est correcte en cliquant sur le bouton « Vérifier la syntaxe ». Si c'est le cas, la zone « Exécution » devient active. L'utilisateur peut alors lancer l'exécution sans ou avec l'enregistrement des informations de qualité de service en cochant la case « Infos de QoS ». Dans ce cas, il doit aussi préciser le nom du fichier ou seront sauvegardées ces informations au format CSV³ reconnu par des outils comme Excel.

³ CSV : Comma Separated Value.

3. Script de commandes RTSP

« Criqs » s'appuie sur un langage de script simple qui permet :

- d'affecter des valeurs à des variables.
- de décrire une commande RTSP à l'aide de variables.
- d'extraire des informations des réponses du serveur streaming.
- d'appeler des fonctions de connexion aux flux multimédia.

3.1. Utilisation de variables

Les variables utilisables au sein de « criqs » sont toutes de type chaîne. Même celles contenant des numéros de port sont de ce type.

Elles sont introduites par le mot clé « var » et les valeurs qui leurs sont affectées sont soit des constantes chaînes soit des valeurs retournées par des fonctions du langage de script (cf. 3.3).

Les exemples ci-dessous illustrent les deux cas.

```
var    StreamingServer = "bailleron",
       FichierPresentation = "sample_300kbit.mov",
       UserAgent = "TEMICS";
```

Figure 2: Affectation d'une constante chaîne à une variable

```
/ les variables suivantes ont leur valeur extraite de la réponse du serveur
var    ControlFlux = GetControlFlux(1),
       ControlFlux2 = GetControlFlux(2);
```

Figure 3: Affectation d'un retour de fonction à une variable

3.2. Description d'une commande RTSP

« Criqs » permet d'émettre des commandes RTSP à partir de leur description fournie dans le script.

Ces commandes RTSP sont comprises entre les caractères « {« et « } » avec nécessairement un saut de ligne avant le « } ». Leur contenu peut être calculé dynamiquement à l'exécution du script en fonction des variables qu'elles contiennent. Les noms de ces variables sont encadrés par les caractères « < » et « > ». L'interpréteur de commande émet la commande lorsqu'il rencontre le caractère « } ».

L'extrait de script suivant décrit une commande RTSP DESCRIBE qui fait référence aux variables « StreamingServer » et « FichierPresentation » :

```

/ description de la commande DESCRIBE
/ la commande est:
/           - complétée en fonction des valeurs des variables déclarées ci dessus
/           - émise lorsque que le caractère '}' est rencontré
{
DESCRIBE rtsp://<StreamingServer>/<FichierPresentation> RTSP/1.0
CSeq: 1
Accept: application/sdp
Accept-Language: en-US
User-Agent: <UserAgent>
}

```

Figure 4: Exemple de commande RTSP

Une fois interprétée cette commande et en prenant en compte les valeurs des variables données en exemple au chapitre 3.1 se traduira par l'envoi de la commande :

```

DESCRIBE rtsp://bailleron/sample_300kbit.mov RTSP/1.0
CSeq: 1
Accept: application/sdp
Accept-Language: en-US
User-Agent: TEMICS

```

Figure 5 : Résultat de l'interprétation d'une commande RTSP

3.3. Extraction d'information des réponses du serveur streaming

Les requêtes RTSP émises par « Criqs » amènent le serveur streaming à répondre. Ces réponses contiennent des informations nécessaires au bon déroulement de l'interprétation du script, comme par exemple, les identifiants de flux, les numéros de port RTP/RTCP du serveur, ...

Plusieurs fonctions permettent d'extraire des informations. Elles sont décrites ci-après.

3.3.1. GetControlFlux

La fonction « GetControlFlux » permet d'extraire du SDP retourné par la commande RTSP DESCRIBE les identifiants de flux, i.e. la valeur de l'attribut « a=control : ». Une séquence pouvant comprendre plusieurs flux, la fonction « GetControlFlux » prend en paramètre un entier pour indiquer le rang du flux dont on cherche le nom.

```

/ les variables suivantes ont leur valeur extraite de la réponse du serveur
var      ControlFlux = GetControlFlux(1),
        ControlFlux2 = GetControlFlux(2);

```

Figure 6 : Exemple d'appels de « GetControlFlux »

3.3.2. GetSessionID

L'identifiant de session est fourni par le serveur dans la réponse à la première requête RTSP DESCRIBE. Une fonction dédiée permet d'extraire sa valeur. Il s'agit de « GetSessionID ». Un exemple d'appel est donné ci-après.

```
var SessionID = GetSessionID();
```

Figure 7: Exemple d'appel de "GetSessionID"

3.3.3. GetReturnedValue

Plusieurs valeurs ne peuvent pas être extraites aussi facilement que les identifiants de session ou ceux de flux. Aussi, une fonction plus générique d'extraction est disponible. Il s'agit de « GetReturnedValue ». Cette fonction extrait de la réponse du serveur une chaîne comprise entre une chaîne donnée et un caractère donné (non compris dans la valeur extraite).

Par exemple, si le serveur retourne la chaîne suivante :

```
RTSP/1.0 200 OK
Server: DSS/5.0.1.1 (Build/464.1.1; Platform/Win32; Release/5; )
Cseq: 2
Last-Modified: Mon, 28 Feb 2005 18:36:28 GMT
Cache-Control: must-revalidate
Session: 103281078593414
Date: Wed, 12 Jul 2006 09:35:47 GMT
Expires: Wed, 12 Jul 2006 09:35:47 GMT
Transport: RTP/AVP;unicast;client_port=7970-7971;source=131.254.10.40;server_port=6970-6971;ssrc=00005EE1
```

Figure 8: Exemple de chaîne retournée par le serveur streaming

Alors l'appel à « GetReturnedValue » suivant :

```
var PortServeur = GetReturnedValue("server_port=", "-");
```

Figure 9: Exemple d'appel de "GetReturnedValue"

Permettra d'affecter à la variable « Portserveur » la valeur « 6970 ».

3.4. Connexion des flux multimedia

Les connexions RTP/RTCP sont possibles après une commande RTSP SETUP valide. Dans cas, il est possible d'établir ces connexions en faisant appel à la fonction « ConnectStream ». Elle prend en paramètre :

- le nom ou l'adresse IP du serveur,
- le port client RTP
- le port serveur RTP.

Ces paramètres sont les valeurs de variables mais, à la différence de celles utilisées dans les commandes RTSP elles ne sont pas encadrées par des « < » et des « > » :

```
ConnectStream(StreamingServer, PortRTPClientVideo, PortServeur)
```

Figure 10: Exemple d'appel à "ConnectStream"

3.5. Suspending l'interprétation du script

Il est possible de suspendre l'interprétation du script de commandes RTSP. Cela est particulièrement utile une fois la commande RTSP PLAY exécutée pour laisser la vidéo se diffuser. Il est ainsi possible grâce à la commande « Wait » de préciser le nombre de secondes d'attente avant la reprise de l'interprétation. Pendant ce temps, les paquets RTP/RTCP sont sûrs reçus et les informations de qualité de service calculées si elles étaient demandées.

```
Wait(10)
```

Figure 11: Exemple d'appel à "Wait"

3.6. Exemple de script complet

L'exemple ci-dessous correspond à un script complet permettant d'établir la connexion RTSP et les connexions RTP/RTCP pour chacun des flux d'une séquence.

```
/ ceci est un fichier test de l'interpreteur de commande RTSP

var      StreamingServer = "bailleron",
         FichierPresentation = "sample_300kbit.mov",
         UserAgent = "TEMICS";

var      PortRTCPClientAudio = "7970",
         PortRTCPClientAudio = "7971",
         PortRTCPClientVideo = "7972",
         PortRTCPClientVideo = "7973";

/ description de la commande DESCRIBE
/ la commande est:
/          - complétée en fonction des valeurs des variables
déclarées ci dessus
/          - émise lorsque que le caractère ']' est rencontré
{
DESCRIBE rtsp://<StreamingServer>/<FichierPresentation> RTSP/1.0
CSeq: 1
Accept: application/sdp
Accept-Language: en-US
User-Agent: <UserAgent>
}

/ les variables suivantes ont leur valeur extraite de la réponse du serveur
var      ControlFlux = GetControlFlux(1),
         ControlFlux2 = GetControlFlux(2);

/ la commande SETUP peut alors etre construite
{
SETUP rtsp://<StreamingServer>/<FichierPresentation>/<ControlFlux2> RTSP/1.0
CSeq: 2
Transport:          RTP/AVP;unicast;client_port=<PortRTCPClientAudio>-
<PortRTCPClientAudio>
User-Agent: <UserAgent>
}

var PortServeur = GetReturnedValue("server_port=", "-");
ConnectStream(StreamingServer, PortRTCPClientAudio, PortServeur)

/ lecture des champs retournes par le serveur

var      SessionID = GetSessionID();
{
SETUP rtsp://<StreamingServer>/<FichierPresentation>/<ControlFlux> RTSP/1.0
CSeq: 3
Session: <SessionID>
Transport:          RTP/AVP;unicast;client_port=<PortRTCPClientVideo>-
<PortRTCPClientVideo>
User-Agent: <UserAgent>
}

var PortServeur = GetReturnedValue("server_port=", "-");
ConnectStream(StreamingServer, PortRTCPClientVideo, PortServeur)

/ emission de la commande PLAY
{
PLAY rtsp://<StreamingServer>/<FichierPresentation> RTSP/1.0
CSeq: 4
Range: npt=-
Session: <SessionID>
User-Agent: <UserAgent>
}

Wait(10)

/ emission de la commande PAUSE
{
PAUSE rtsp://<StreamingServer>/<FichierPresentation> RTSP/1.0
CSeq: 5
Session: <SessionID>
User-Agent: <UserAgent>
}

/ emission de la commande TEARDOWN
{
TEARDOWN rtsp://<StreamingServer>/<FichierPresentation> RTSP/1.0
CSeq: 6
Session: <SessionID>
User-Agent: <UserAgent>
}
```

Figure 12: Exemple de script complet

3.7. Vérification syntaxique d'un script

Une fois sélectionné, le fichier script doit être syntaxiquement vérifié en cliquant sur le bouton « Vérifier la syntaxe ». Une syntaxe correcte conduit à l'affichage suivant :

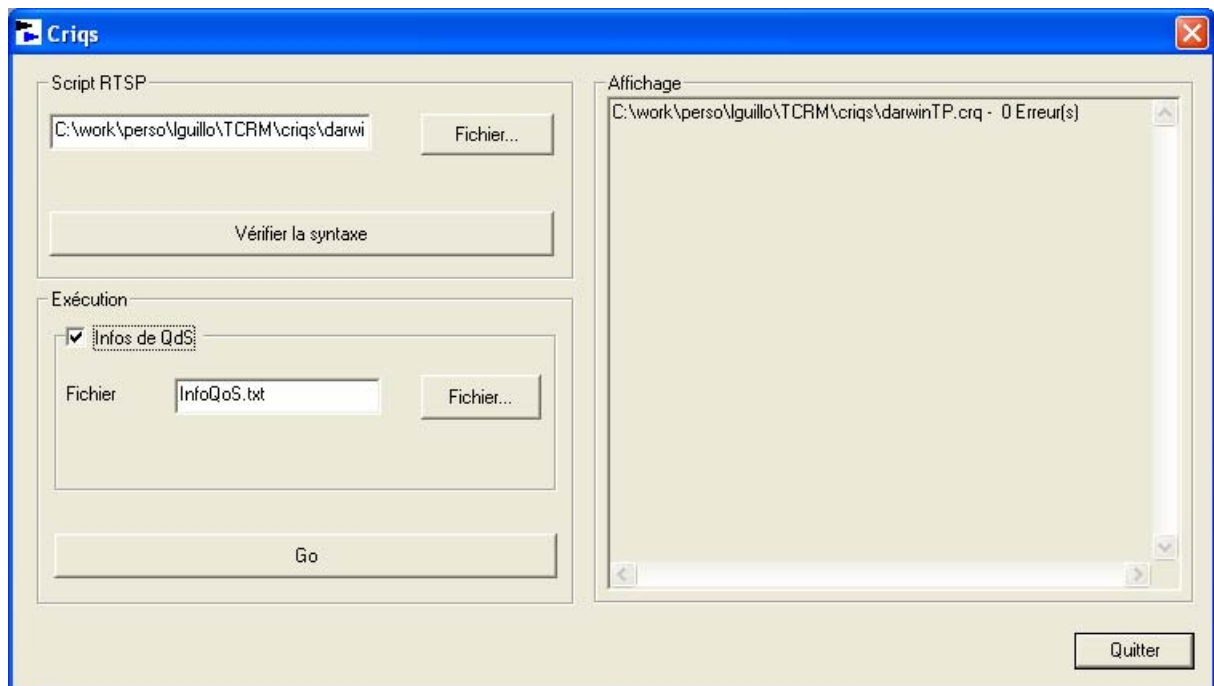


Figure 13: Syntaxe correcte

En cas d'erreurs (mots clés inconnus, « ; » oubliés, ...), la liste est affichée dans la zone « Affichage » avec les numéros de lignes associés :

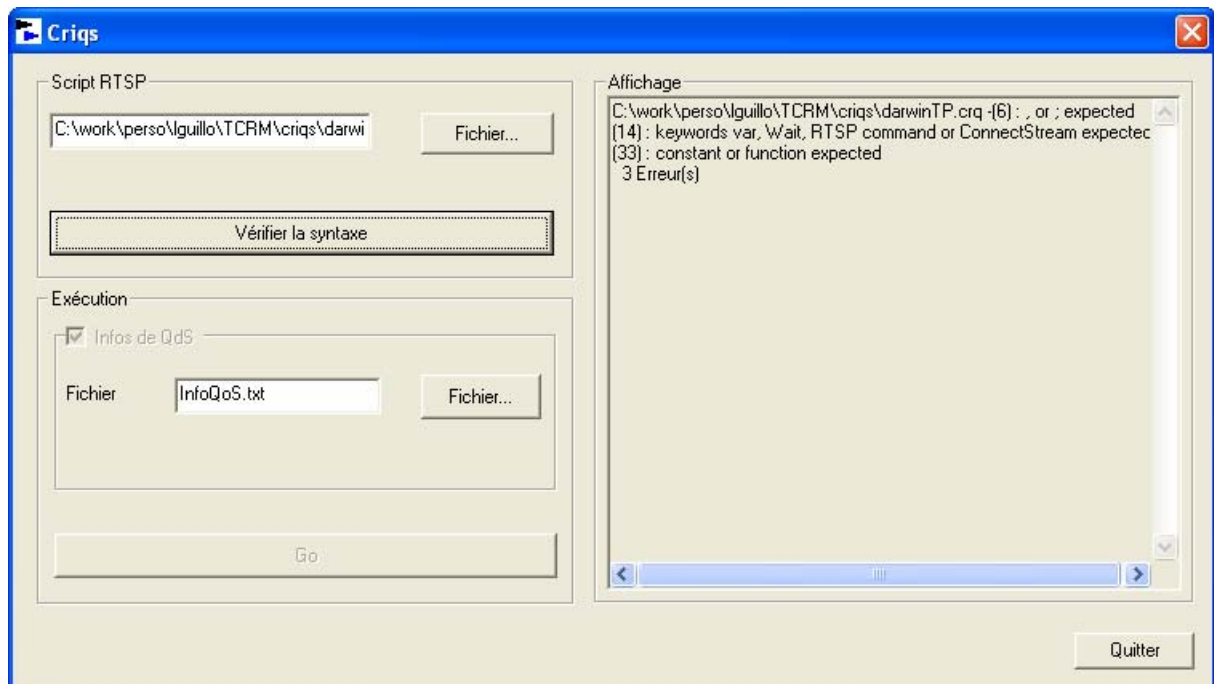


Figure 14: Exemple d'erreurs détectées

4. Informations de qualité de service

Lorsque la case « Infos de QoS » est cochée, l'utilisateur fournit un nom de fichier dans lequel les informations de qualité de service sont calculées au fur et à mesure de la réception des paquets RTP et cela pour chacun des flux.

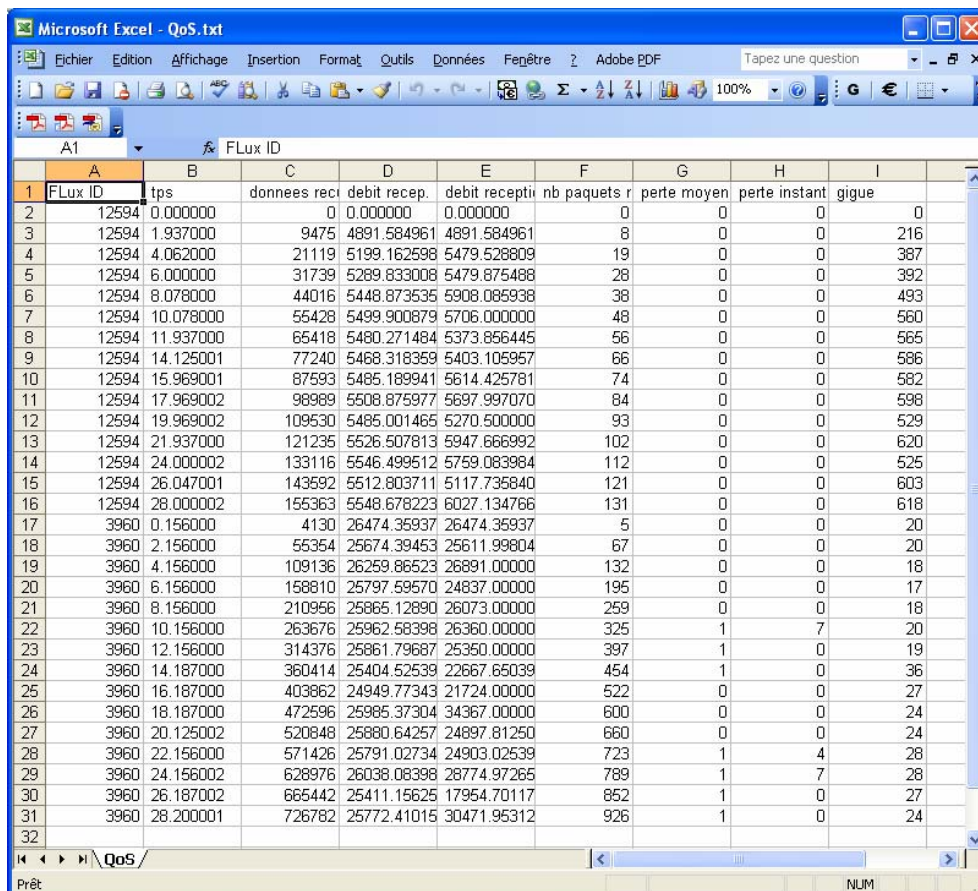
Ces informations sont les suivantes :

- le nombre d'octets reçus,
- le débit en réception moyen,
- le débit en réception instantané,
- le nombre de paquets reçus,
- le pourcentage de pertes moyen,
- le pourcentage de pertes instantané,
- la gigue d'inter-arrivée des paquets.

A ces informations, sont ajoutés :

- l'identifiant du flux concerné,
- l'instant de la prise de ces mesures.

Ces données sauvegardées au format CSV et utilisables par des outils comme Excel se présentent sous la forme :



	A	B	C	D	E	F	G	H	I
1	FLux ID	tps	donneses reci	debit recep.	debit recepti	nb paquets r	perte moyen	perte instant	gigue
2	12594	0.000000	0	0.000000	0.000000	0	0	0	0
3	12594	1.937000	9475	4891.584961	4891.584961	8	0	0	216
4	12594	4.062000	21119	5199.162598	5479.528809	19	0	0	387
5	12594	6.000000	31739	5289.833008	5479.875488	28	0	0	392
6	12594	8.078000	44016	5448.873535	5908.085938	38	0	0	493
7	12594	10.078000	55428	5499.900879	5706.000000	48	0	0	560
8	12594	11.937000	65418	5480.271484	5373.856445	56	0	0	565
9	12594	14.125001	77240	5468.318359	5403.105957	66	0	0	586
10	12594	15.969001	87593	5485.189941	5614.425781	74	0	0	582
11	12594	17.969002	98989	5508.875977	5697.997070	84	0	0	598
12	12594	19.969002	109530	5485.001465	5270.500000	93	0	0	529
13	12594	21.937000	121235	5526.507813	5947.666992	102	0	0	620
14	12594	24.000002	133116	5546.499612	5759.083984	112	0	0	525
15	12594	26.047001	143692	5512.803711	5117.735840	121	0	0	603
16	12594	28.000002	155363	5548.678223	6027.134766	131	0	0	618
17	3960	0.156000	4130	26474.35937	26474.35937	5	0	0	20
18	3960	2.156000	55354	25674.39453	25611.99804	67	0	0	20
19	3960	4.156000	109136	26259.86523	26891.00000	132	0	0	18
20	3960	6.156000	158810	25797.59570	24837.00000	195	0	0	17
21	3960	8.156000	210956	25865.12890	26073.00000	259	0	0	18
22	3960	10.156000	263676	25962.58398	26360.00000	325	1	7	20
23	3960	12.156000	314376	25861.79687	25350.00000	397	1	0	19
24	3960	14.187000	360414	25404.52539	22667.65039	454	1	0	36
25	3960	16.187000	403862	24949.77343	21724.00000	522	0	0	27
26	3960	18.187000	472596	25985.37304	34367.00000	600	0	0	24
27	3960	20.125002	520848	25880.64257	24897.81250	660	0	0	24
28	3960	22.156000	571426	25791.02734	24903.02539	723	1	4	28
29	3960	24.156002	628976	26038.08398	28774.97265	789	1	7	28
30	3960	26.187002	665442	25411.15625	17954.70117	852	1	0	27
31	3960	28.200001	726782	25772.41015	30471.95312	926	1	0	24
32									

Figure 15: Importation des données de QoS dans Excel