

**Rapport de**

**Réseau**

*Etude des options de sockets*

CASTEL Alexian - DIBON Josselin

Sommaire

[I. Option SO\_REUSEADDR 2](#_Toc472368573)

[1) Principe de l’option 2](#_Toc472368574)

[2) Impact 2](#_Toc472368575)

[3) Implémentation 2](#_Toc472368576)

[4) Analyse 2](#_Toc472368577)

[II. Option SO\_KEEPALIVE 2](#_Toc472368578)

[1) Principe de l’option 2](#_Toc472368579)

[2) Impact 2](#_Toc472368580)

[3) Implémentation 3](#_Toc472368581)

[4) Analyse 3](#_Toc472368582)

[III. Option SO\_RCVBUF 4](#_Toc472368583)

[1) Principe 4](#_Toc472368584)

[2) Impact 4](#_Toc472368585)

[3) Implémentation 4](#_Toc472368586)

[4) Analyse 4](#_Toc472368587)

[IV. Option SO\_SNDBUF 5](#_Toc472368588)

[1) Principe 5](#_Toc472368589)

[2) Impact 5](#_Toc472368590)

[3) Implémentation 5](#_Toc472368591)

[setsockopt(sock, SOL\_SOCKET, SO\_SNDBUF, &valOption, sizeof(valOption)); 5](#_Toc472368592)

[V. Option SO\_DEBUG 6](#_Toc472368593)

[1) Principe 6](#_Toc472368594)

[2) Impact 6](#_Toc472368595)

[3) Implémentation 6](#_Toc472368596)

[setsockopt(sock, SOL\_SOCKET, SO\_DEBUG, &valOption, sizeof(valOption)); 6](#_Toc472368597)

[VI. Option SO\_OOBINLINE 8](#_Toc472368598)

[1) Principe 8](#_Toc472368599)

[2) Impact 8](#_Toc472368600)

[3) Implémentation 8](#_Toc472368601)

[setsockopt(sock, SOL\_SOCKET, SO\_DEBUG, &valOption, sizeof(valOption)); 8](#_Toc472368602)

[VII. Option SO\_DONTROUTE 8](#_Toc472368603)

[1) Principe 8](#_Toc472368604)

[2) Impact 8](#_Toc472368605)

[3) Implémentation 8](#_Toc472368606)

[setsockopt(sock, SOL\_SOCKET, SO\_DONTROUTE, &valOption, sizeof(valOption)); 8](#_Toc472368607)

[VIII. Option SO\_BROADCST 9](#_Toc472368608)

[IX. Interface graphique 9](#_Toc472368609)

# I. Option SO\_REUSEADDR

## 1. Principe de l’option

Lorsque cette option est activée le serveur est autorisé à se lier à une adresse déjà utilisée (en mode TIME\_WAIT).

## 2. Impact

Cette option évite donc l’apparition de l’erreur « Address already in use ».

## 3. Implémentation

setsockopt(sock, SOL\_SOCKET, SO\_REUSEADDR, &valeurOption, sizeof(valeurOption));

## 4. Analyse

# II. Option SO\_KEEPALIVE

## 1. Principe de l’option

Cette option permet de maintenir une connexion TCP entre un serveur et un client. Lors de la création de la connexion un timeout est défini. A chaque fois que ce timeout est atteint, un paquet ne contenant que le bit ACK est émis afin de garder la connexion active

## 2. Impact

Une connexion sans activité est tout de même maintenu grâce à cette option.

## 3. Implémentation

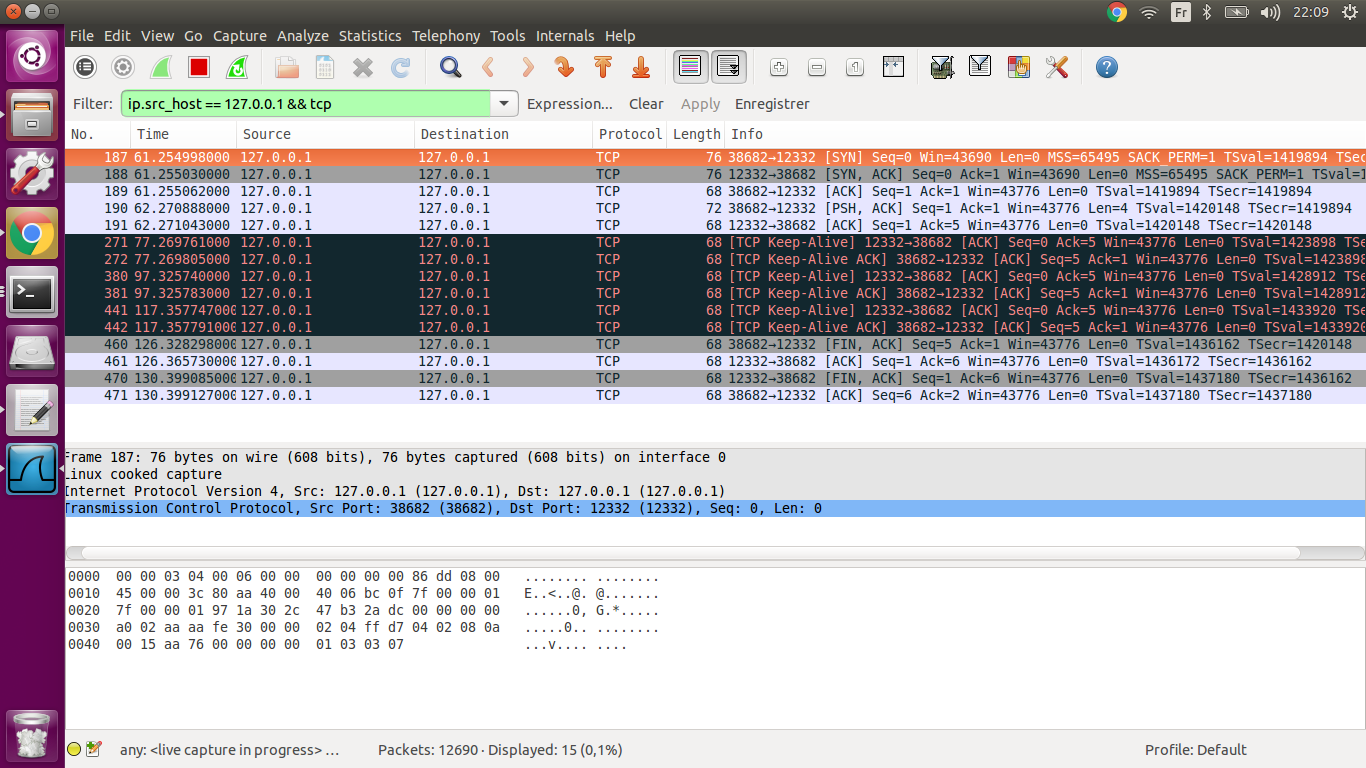
setsockopt(sock, SOL\_SOCKET, SO\_KEEPALIVE, & valeurOption, sizeof(valeurOption)) ;

Setsockopt(sock, IPPROTO\_TCP, TCP\_KEEPIDLE, &idle, sizeof(int)) ;

Setsockopt(sock, IPPROTO\_TCP, TCP\_KEEPINTVL, &interval, sizeof(int)) ;

Setsockopt(sock, IPPROTO\_TCP, TCP\_KEEPCNT, &maxpkt, sizeof(int)) ;

## 4. Analyse



Cette capture wireshark montre que l’option keepalive se lance pour la première fois 15 secondes après l’émission d’une trame et se répète toutes les 20 secondes. Ces paramètres sont réglables grâce aux options de TCP : TCP\_KEEPIDLE (délais avant de déclencher), TCP\_KEEPCNT (nombre d’occurrence avant la fermeture de la connexion) et TCP\_KEEPINTVL (délais avant répétition).

# III. Option SO\_RCVBUF

## 1. Principe

Le but de cette option est de paramétrer la taille du buffer de réception

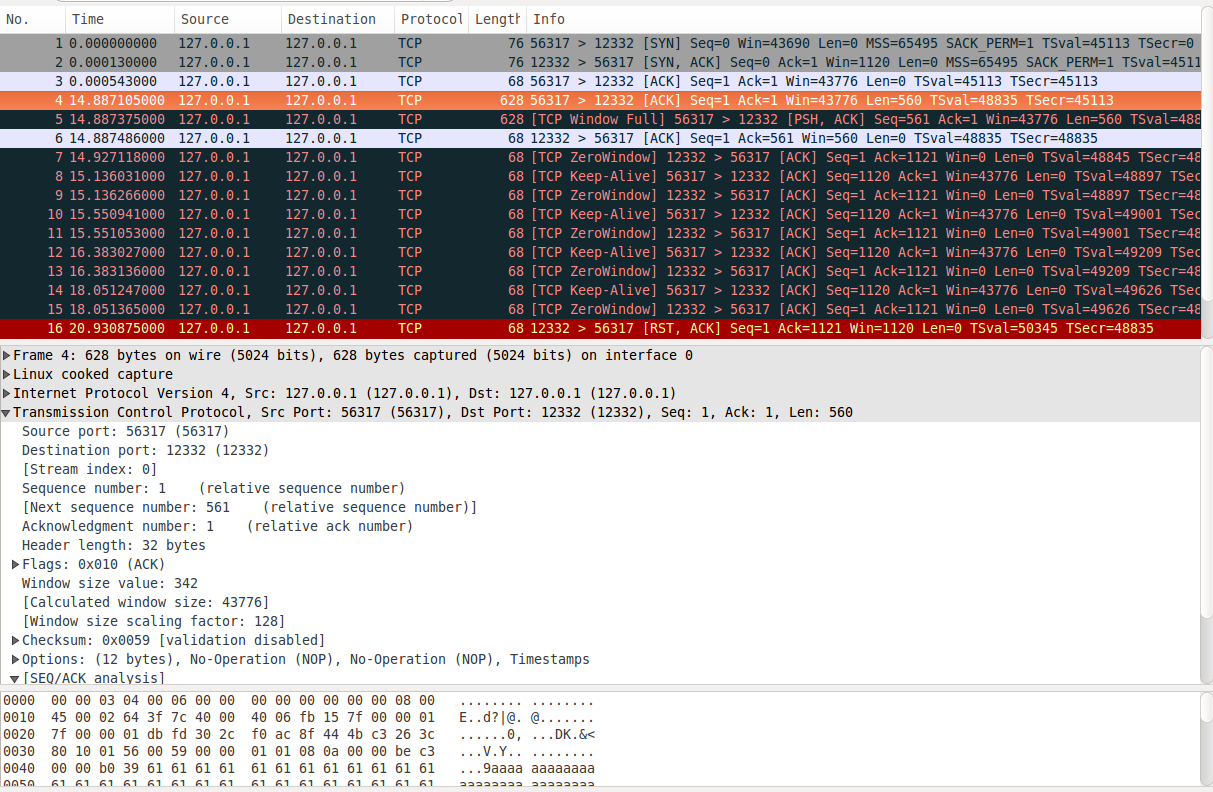
## 2. Impact

Si un client envoie un message supérieur à la taille maximale du buffer de réception du serveur, ce dernier ne recevra pas le message dans son intégralité.

## 3. Implémentation

setsockopt(sock, SOL\_SOCKET, SO\_RCVBUF, & valeurOption, sizeof(valeurOption));

## 4. Analyse



# IV. Option SO\_SNDBUF

## 1. Principe

Le but de cette option est de paramétrer la taille du buffer d’envoi

## 2. Impact

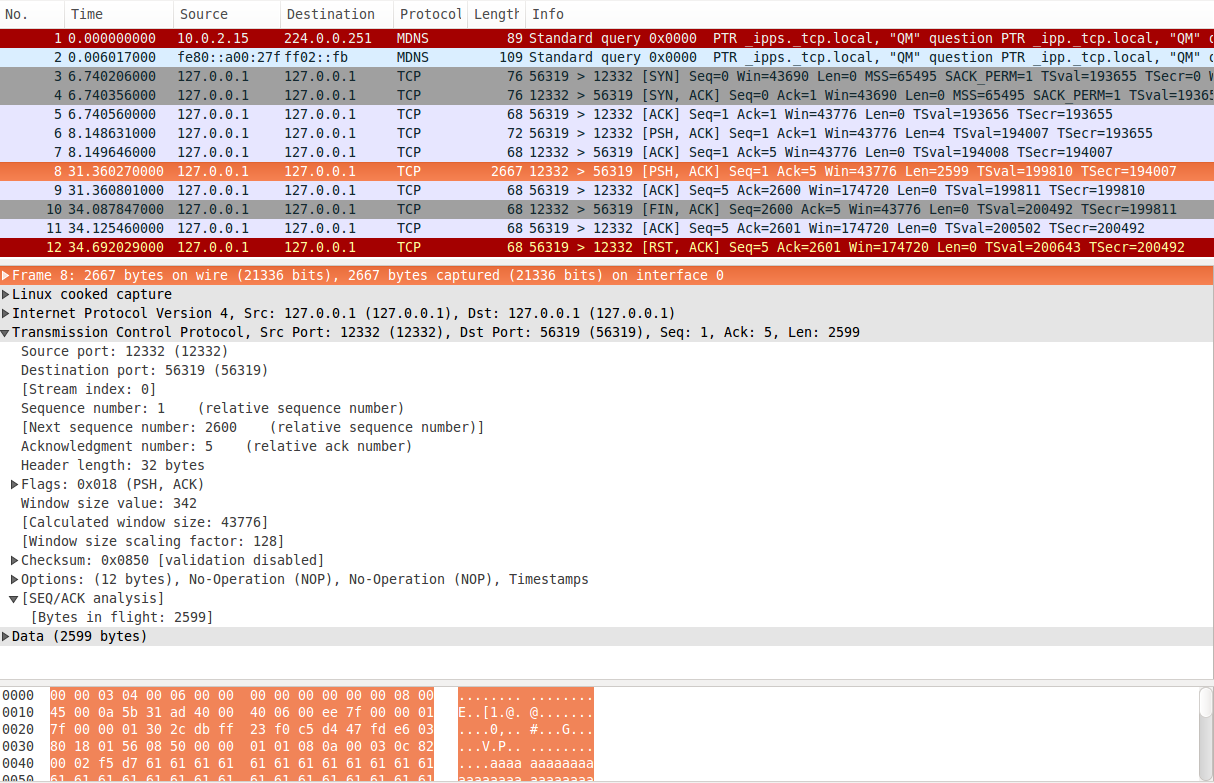
Cette option limite la taille des données envoyées sur le serveur.

## 

## 3. Implémentation

## setsockopt(sock, SOL\_SOCKET, SO\_SNDBUF, & valeurOption, sizeof(valeurOption));

## 4. Analyse



# V. Option SO\_DEBUG

## 1. Principe

Active les informations de débogage, cette option a une valeur entière.

## 2. Impact

Permet d’avoir des informations sur les informations de débogage.

## 3. Implémentation

## setsockopt(sock, SOL\_SOCKET, SO\_DEBUG, & valeurOption, sizeof(valeurOption));

## 4. Analyse

# VI. Option SO\_OOBINLINE

## 1. Principe

L'activation de cette option fait que les données hors bande sont placées dans la file d'attente normale d'entrée de données à mesure qu'elles sont reçues.

## 2. Impact

## 3. Implémentation

## setsockopt(sock, SOL\_SOCKET, SO\_DEBUG, & valeurOption, sizeof(valeurOption));

## 4. Analyse

# VII. Option SO\_DONTROUTE

## 1. Principe

Demande que les messages sortants contournent les fonctionnalités de routage standard. La destination doit être sur un réseau directement connecté et les messages sont dirigés vers l'interface réseau appropriée en fonction de l'adresse de destination. Cette option prend un booléen (sous forme d’*int*) en paramètre.

## 2. Impact

## 3. Implémentation

## setsockopt(sock, SOL\_SOCKET, SO\_DONTROUTE, & valeurOption, sizeof(valeurOption));

## 4. Analyse

# VIII. Option SO\_BROADCST

## 1. Principe

Cette option permet de définir si la diffusion de datagrammes est autorisée, ou non.

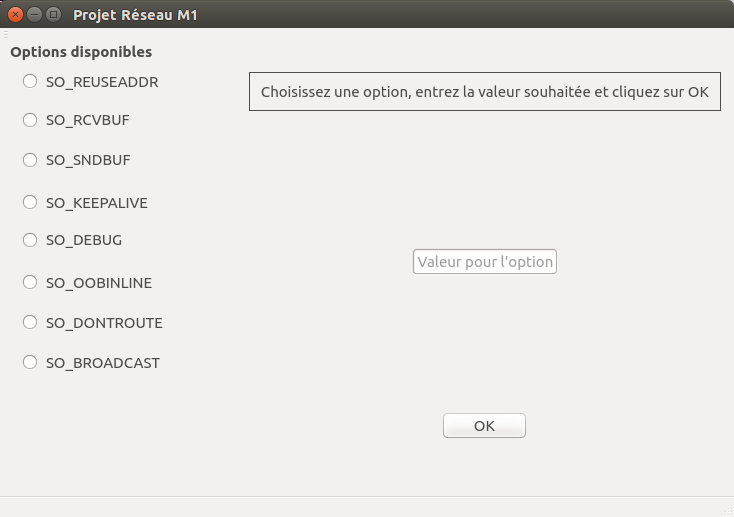
## 2. Impact

## 3. Implémentation

## setsockopt(sock, SOL\_SOCKET, SO\_BROADCAST, & valeurOption, sizeof(valeurOption));

## 4) Analyse

# IX. Interface graphique



*Figure 1 - Interface graphique*

L’interface graphique (visible en figure 1) a été réalisée en C++, à l’aide de la librairie graphique *Qt*. Elle se compose de quatre parties :

* La liste des options, sur la gauche
* Le bandeau d’aide en haut
* La zone de saisie de la valeur souhaitée pour l’option
* Le bouton « OK » permettant de valider la saisie

Le fonctionnement est simple, il suffit de sélectionner l’option à tester, entrer la valeur voulue, puis cliquer sur « OK ».

Suite à cela, deux terminaux s’ouvrent, un représentant le serveur, et un autre représentant le client.