R5.02: QOS

Jean-Marc Pouchoulon

Septembre 2023

Ce TD a pour objet de faire des mesures de bandes passantes sur le réseau au travers de l'utilitaire iperf. On se servira de NETEM qui est inclu dans le kernel LINUX afin de simuler des problèmes sur le réseau (délais, pertes, duplications). On travaillera sur une VM Debian et on utilisera les Network Name Space afin de créer deux cartes réseaux ethernet VETH jumelées. Vous aurez donc deux devices ethernet virtuels (veth-server veth-client) avec deux adresses IP (10.12.0.1 10.12.0.2) entre lesquelles vous effectuerez des tests de performances avec iperf en faisant varier à tour de rôle les délais, les pertes et les duplications de paquets sur ce réseau interne à la machine. A partir de ces résultats vous aller grapher la bande passante sous CALC. Ces trois graphes sont à remettre à la fin du TD sur l'ENT de l'IUT. Vous donnerez aussi vos conclusions ou remarques.

1 Installation de l'environnement de test

1. installer iperf à l'aide de apt-get. Réaliser quelques tests sur la même machine afin de comprendre comment iperf fonctionne. voir `http://fr.wikipedia.org/wiki/Iperf`.Testez les quelques exemples proposés.
2. Le script suivant simule.sh va permettre de créer un Network Name Space. (source `http://www.pocketnix.org/posts/Simulating%20latency%20under%20Linux`).

```
#!/bin/bash

# Fail hard and fast if any intermediate command pipeline fails
set -e

NETNS=latency-network
SERVER_IF=veth-server
CLIENT_IF=veth-client
# The delay in uSecs
DELAY=100000

function help {
    echo "$0 <Server IP> <Client IP> [CMD [ARGS ....]]"
    echo ""
    echo "    Server IP: IP address assigned to the server"
    echo "    Client IP: IP address assigned to the client, Must be"
    echo "               in the same /24 subnet as the server"
    echo "    CMD/ARGS:  The command/server to execute at the other"
    echo "               end of the high latency link, DEFAULT=/bin/sh"
}

if [ "$1" == "-h" ]; then
    help
    exit 0
fi

if [ "$1" == "" ]; then
    echo "Error: Please specify an IPv4 Address for the server"
```

```
    echo
    help
    exit 1
fi
if [ "$2" == "" ]; then
    echo "Error: Please specify an IPv4 Address for the client"
    echo
    help
    exit 1
fi
SERVER_IP=$1
CLIENT_IP=$2
shift 2

if [ "$1" == "" ]; then
    echo "No command specified, using /bin/sh"
    CMD=/bin/sh
    ARGS=""
else
    CMD=$1
    shift
    ARGS=$*
fi

# Create the networking pair
ip li add $SERVER_IF type veth peer name $CLIENT_IF
# Automatically clean up interfaces on script exit
trap "ip li del $CLIENT_IF" EXIT

# Assign the requested IP addresses
ip ad add $CLIENT_IP/24 dev $CLIENT_IF

# Bring the interfaces up in the correct order
ip li set $CLIENT_IF up

# Create a net namespace and set it up with the server interface
ip netns add $NETNS
trap "ip li del $CLIENT_IF; ip netns del $NETNS" EXIT
ip li set $SERVER_IF netns $NETNS

# Set IP networking in the container
ip netns exec $NETNS ip ad add $SERVER_IP/24 dev $SERVER_IF
ip netns exec $NETNS ip li set $SERVER_IF up
ip netns exec $NETNS ip ro add default via $CLIENT_IP

# Execute the command in the namespace
ip netns exec $NETNS $CMD $ARGS
```

Lancez le:

```
./simul.sh  10.12.0.1 10.12.0.2
No command specified, using /bin/sh
# ip a
13: veth-server: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 42:35:f3:1e:5a:09 brd ff:ff:ff:ff:ff:ff
    inet 10.12.0.1/24 scope global veth-server
    inet6 fe80::4035:f3ff:fe1e:5a09/64 scope link
       valid_lft forever preferred_lft forever
14: lo: <LOOPBACK> mtu 16436 qdisc noop state DOWN
```

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

2 Réalisation des graphes "xy"

1. Vous lancerez le serveur iperf via:
   `./simul.sh  10.12.0.1 10.12.0.2 'iperf  -B 10.12.0.1 -s'`
2. Vous lancerez le test ensuite en vous "bindant" sur l'IP du device veth client.
3. Le script suivant vous permettra de dégrader les conditions du réseau afin de réaliser vos tests.

```
#!/bin/bash
# add_penalty.sh
# author :JMP dec 2014

if [ "$#" -ne 2 ]
   then
      echo "Usage: $0 delay|loss|duplicate nombre"
      exit 1
fi

TYPE=$1
VALEUR=$2

if [ $TYPE == "delay" ]
then
   DELAY="$2ms"
   echo -ne "ajout d'un delais de $DELAY \n"
   tc qdisc replace dev veth-client root netem delay  $DELAY
   ip netns exec latency-network  tc qdisc replace dev veth-server root netem delay $DELAY
elif [ $TYPE == "loss" ]
then
   LOSS="$2%"
   echo -ne "ajout d'une perte de $LOSS \n"
   tc qdisc replace dev veth-client root netem loss  $LOSS
   ip netns exec latency-network  tc qdisc replace dev veth-server root netem loss $LOSS
elif [ $TYPE == "duplicate" ]
then
   DUP="$2%"
   echo -ne "ajout d'une duplication de $DUP \n"
   tc qdisc replace dev veth-client root netem duplicate  $DUP
   ip netns exec latency-network  tc qdisc replace dev veth-server root netem duplicate $DUP
else echo -ne  "delay loss duplicate manquant\n"
fi

SHOWTCC=$(tc -p -s -d  qdisc show)
SHOWTCS=$(ip netns exec latency-network  tc -p -s -d  qdisc show)
echo -ne "$SHOWTCC \n"
echo -ne "$SHOWTCS \n"
```