

BwctlScriptByTobyRodwell

```
#!/bin/bash

# This script runs a bwctl test and, concurrently runs the 'ss' (show socket command)
# at regular intervals during the test. It is recommended the 'ss' interval be 0.1 or
# or 0.2 (seconds)
#
# The syntax of the command are:
# tcpctest <target bwctl server> <length of test in sec> <'ss' frequency>
#
# The results are stored in two files in the user's home directory
# 1. tcp_test-<date/time of test>-<target host name>.results
# 2. stat_changes-<date/time of test>-<target host name>.results
#
# File 1 is a list of the 'ss' results, which can be graphed
# (with the use of a spreadsheet) for ease of interpretation.
# File 2 shows the difference between 'netstat -s' immediately before and after
# the bwctl test. This is particularly useful for spotting TCP retransmissions etc

mkdir $HOME/temp

echo $1 >$HOME/temp/target.tmp
echo $2 > $HOME/temp/test_length.tmp
echo $3 > $HOME/temp/ss_int.tmp

#BWCTL test port is 5001 by default
test_port=5001
host=`uname -n`

#This file is used to pick up just those lines from'ss' that are related to the test-port (e.g. 5001), without
adding a newline (feature of printf)
echo "/$test_port/ {printf \"%s\\t %s\\t %s\\t %s\\t %s\\t %s\\t %s\\t %s\\t %s\\t \\\" ,\\$1, \\$2, \\$3, \\$4, \\$5, \\$6, \\$7, \\$8, \\$9 }\" > $HOME/temp/awk_filter.tmp

#This file is used to pick up just those lines from'ss' that are related to the test-port (e.g. 5001), and adds
a newline (feature of print)
echo "/$test_port/ {print \\$0}\" > $HOME/temp/awk_filter2.tmp

test_length=`cat $HOME/temp/test_length.tmp`

netstat -s > $HOME/temp/stat_before.tmp
date +%y%m%d%H%M%S > $HOME/temp/time.tmp
test_time=`cat $HOME/temp/time.tmp`
# The use of brackets enables the running of concurrent sub-processes
(
test_time=`cat $HOME/temp/time.tmp`
target=`cat $HOME/temp/target.tmp`
test_length=`cat $HOME/temp/test_length.tmp`
test_interval=2

echo $test_time > $HOME/tcp_test-$test_time-$host.results

bwctl -c $target -t $test_length -i $test_interval >> $HOME/tcp_test-$test_time-$host.results
) &
(
# Sleep interval for 'ss'
ss_int=`cat $HOME/temp/ss_int.tmp`
test_length=`cat $HOME/temp/test_length.tmp`
bwctl_port=5001
test_time=`cat $HOME/temp/time.tmp`
echo $test_time > $HOME/temp/ss.tmp
uname -a >> $HOME/temp/ss.tmp
cat $HOME/temp/target.tmp > $HOME/temp/ss.tmp
testIterations=`echo "($test_length + 10) / $ss_int" | bc`

for ((i=0; i<$testIterations; i++))
```

```

do
    testTimeNow=`echo "$i * $ss_int" | bc`
    echo -n $testTimeNow '$\t' >> $HOME/temp/ss.tmp
    /usr/sbin/ss -i | awk -f $HOME/temp/awk_filter.tmp >> $HOME/temp/ss.tmp
    echo " " >> $HOME/temp/ss.tmp
    sleep $ss_int
done
)
wait

netstat -s > $HOME/temp/stat_after.tmp
target=`cat $HOME/temp/target.tmp`

# Get the RTT
ping -c 3 $target >> $HOME/tcp_test-$test_time-$host.results
# Headers for the 'ss' results
echo '$Time \t State \t Recv-Q \t Send-Q \t Local Socket \t Peer Socket \t RTO \t CWND \t ssthresh' >> $HOME
/tcp_test-$test_time-$host.results

# This filters out lines without '5001' on them
awk -f $HOME/temp/awk_filter2.tmp $HOME/temp/ss.tmp>> $HOME/tcp_test-$test_time-$host.results

# Display the BWCTL result
head -n 20 $HOME/tcp_test-$test_time-$host.results

# Save the difference between start and end netstat stats.
diff $HOME/temp/stat_before.tmp $HOME/temp/stat_after.tmp > $HOME/stat_changes-$test_time-$host.results

# Tidy up
rm $HOME/temp/*.tmp
rmdir $HOME/temp

exit 0

```