

---

# ANEXO 6

## SCRIPTS PARA LA CAPTURA DE LOS DATOS

---

El presente anexo contiene la información correspondiente a los scripts desarrollados en el lenguaje de programación *perl* que permitieron la recolección de la información en las pruebas donde se midieron los valores de retardo y variación de retardo (OWAMP), uso de CPU y Velocidad.

### 1. OWAMP

---

#### 1.1. SCRIPT 1 – RrdCreateOWAMP

---

Este script es el encargado de la creación de las bases de datos de rrdtool.

```
#!/usr/bin/perl

use RRDs;

RRDs::create ("usoOWAMP1.rrd",
              "--start=N",
              "--step=1",
              "DS:delay:GAUGE:5:-1000:1000",
              "DS:min:GAUGE:5:-1000:1000",
              "DS:media:GAUGE:5:-1000:1000",
              "DS:max:GAUGE:5:-1000:1000",
              "DS:jitter:GAUGE:5:-100:1000",
              "DS:error:GAUGE:5:-100:1000",
              "DS:packet:GAUGE:5:-100:1100",
              "RRA:AVERAGE:0.5:1:1003");

my $err=RRDs::error;
if($err)
{
print "se presentó el siguiente problema al generar usoOWAMP1.rrd:
$err\n";
}

RRDs::create ("usoOWAMP1_1.rrd",
              "--start=N",
              "--step=1",
              "DS:delay:GAUGE:5:-1000:1000",
              "DS:min:GAUGE:5:-1000:1000",
```

```

        "DS:media:GAUGE:5:-1000:1000",
        "DS:max:GAUGE:5:-1000:1000",
        "DS:jitter:GAUGE:5:-100:1000",
        "DS:error:GAUGE:5:-100:1000",
        "DS:packet:GAUGE:5:-100:1100",
        "RRA:AVERAGE:0.5:1:1003");

my $err=RRDs::error;
if($err)
{
print "se presentó el siguiente problema al generar usoOWAMP1.rrd:
$err\n";
}

RRDs::create ("usoOWAMP2.rrd",
        "--start=N",
        "--step=1",
        "DS:delay:GAUGE:5:-1000:1000",
        "DS:min:GAUGE:5:-1000:1000",
        "DS:media:GAUGE:5:-1000:1000",
        "DS:max:GAUGE:5:-1000:1000",
        "DS:jitter:GAUGE:5:-100:1000",
        "DS:error:GAUGE:5:-100:1000",
        "DS:packet:GAUGE:5:-100:1100",
        "RRA:AVERAGE:0.5:1:1003");

my $err=RRDs::error;
if($err)
{
print "se presentó el siguiente problema al generar usoOWMAP2.rrd:
$err\n";
}

RRDs::create ("usoOWAMP2_2.rrd",
        "--start=N",
        "--step=1",
        "DS:delay:GAUGE:5:-1000:1000",
        "DS:min:GAUGE:5:-1000:1000",
        "DS:media:GAUGE:5:-1000:1000",
        "DS:max:GAUGE:5:-1000:1000",
        "DS:jitter:GAUGE:5:-100:1000",
        "DS:error:GAUGE:5:-100:1000",
        "DS:packet:GAUGE:5:-100:1100",
        "RRA:AVERAGE:0.5:1:1003");

my $err=RRDs::error;
if($err)
{
print "se presentó el siguiente problema al generar usoOWMAP2.rrd:
$err\n";
}

```

## 1.2. SCRIPT 2 – RrdUpdateOWAMP

---

Este script es el que obtiene la información de las peticiones owping realizadas por OWAMP y guarda la primer prueba en un archivo llamado *owamp1* y la segunda en el archivo de nombre *owamp2*.

```
#!/usr/bin/perl

use warnings;
use RRDs;
use threads;
use threads::shared;
use Time::HiRes qw(usleep nanosleep);

my $RRDTOOL='rrdupdate';
my $TARGET_RRD1='usoOWAMP1.rrd';
my $TARGET_RRD1_1='usoOWAMP1_1.rrd';
my $TARGET_RRD2='usoOWAMP2.rrd';
my $TARGET_RRD2_2='usoOWAMP2_2.rrd';

print "Ingresa la direccion del destino owping :";
my $dir=<STDIN>;
chomp($dir);
my $OWAMP="owping $dir -v -a 99.9 -c 1000 -i 0.3e ";
my $owamp1_txt="owamp1";
my $owamp2_txt="owamp2";
my $hilo;
my $counter1="true";
my $counter2="true";
my $erase="rm owamp";

my $delay2=0;
my $min1=0;
my $medial=0;
my $max1=0;
my $jitter1=0;
my $error1=0;
my $packet1=0;

my $delay1=0;
my $min2=0;
my $media2=0;
my $max2=0;
my $jitter2=0;
my $error2=0;
my $packet2=0;

my $time = time;
my $flag_percentiles1="";
my $flag_percentiles2="";

my $first_sample1=0;
my $first_sample2=0;

srand (time ^ $$ ^ unpack "%L*", `ps axww | gzip`);

while($counter2)
{
```

```

$random=60+int(rand(540));
print "esperando $random segundos" ;
select(undef, undef, undef, $random);

if($counter1)
{
    $in=`$OWAMP`;
}
if($in&&$counter1)
{
    #abrimos el archivo y copiamos la salida de owping
    open (owamp1_txt,"+>$owamp1_txt") || die "ERROR: No se
puede abrir el archivo\n";
    print owamp1_txt $in;
    close (owamp1_txt);

    print "##### hilo para RrdUpdateOWAMP2
#####";

    $hilo=threads->create(sub
    {

        $random=90+int(rand(510));
        print "esperando $random segundos" ;
        select(undef, undef, undef, $random);
        $in2=`$OWAMP`;

        if($in2&&$counter2)
        {
            #abrimos el archivo y copiamos la salida de owping
            open (owamp2_txt,"+>$owamp2_txt") || die "ERROR: No se
se puede abrir el archivo\n";
            print owamp2_txt $in2;
            close (owamp2_txt);

            #vamos a ir leyendo el archivo por region
            open (owamp2_txt,"<$owamp2_txt") || die "ERROR: No se
puede abrir el archivo\n";
            $counter2=0;

            #Abrimos la base de datos para que no senos pierda
ningun valor
            `$RRDTOOL $TARGET_RRD2
"$time:$delay2:$min2:$media2:$max2:$jitter2:$error2:$packet2"`;
            $time=$time+1;

            while(<owamp2_txt>)
            {
                chomp;
                $pos_delay=index($_,"delay=");
                $pos_ms=index($_,"ms");
                $pos_sync=index($_,"sync");

                if($pos_delay!=-1&&$pos_ms!=-1&&$pos_sync!=-1)
                {
                    $pos_delay=$pos_delay+6;
                    $delta=$pos_ms-$pos_delay;
                    $delay2=substr($_,$pos_delay,$delta)+0;

```

```

        if($delay2<0)
        {
            $delay2=$delay2*(-1);
        }
        print "delay2:$delay2:$counter2
min2:$min2 media2:$media2 max2:$max2 jitter2:$jitter2 err2:$error2
pac:$packet2\n";

        if($first_sample2==0)
        {
            `$RRDTOOL $TARGET_RRD2
"$time:$delay2:$min2:$media2:$max2:$jitter2:$error2:$packet2"`;
        }
        else
        {
            `$RRDTOOL $TARGET_RRD2_2
"$time:$delay2:$min2:$media2:$max2:$jitter2:$error2:$packet2"`;
        }

        print "respuesta de rrdtool:$?";
        if ( $? == '0' )
            {print "update done\n";}
        else
            {print "update failed\n";}

        $min2=0;
        $media2=0;
        $max2=0;
        $jitter2=0;
        $error2=0;
        $packet2=0;

        $time=$time+1;

    }

else
{
    $pos_sent=index($_,"sent,");
    if($pos_sent!=-1)
    {
        $pos_sent=$pos_sent+5;
        $pos_lost=index($_,"lost",$pos_sent);
        $delta=$pos_lost-$pos_sent;
        $packet2=substr($_,$pos_sent,$delta)+0;
    }

    $pos_min_media=index($_,"min/median/max
=");

    if($pos_min_media!=-1)
    {
        $pos_min_media=$pos_min_media+16;
        $pos_barra=index($_,"/", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;
        $min2=substr($_,$pos_min_media,$delta)+0;
        $pos_min_media=$pos_barra+1;

        $pos_barra=index($_,"/", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;

        $media2=substr($_,$pos_min_media,$delta)+0;
    }
}

```

```

$pos_min_media=$pos_barra+1;

$pos_barra=index($_,"ms",$pos_min_media);
$delta=$pos_barra-$pos_min_media;
$max2=substr($_,$pos_min_media,$delta)+0;

}

$pos_err=index($_,"err=");
if($pos_err!=-1)
{
$pos_err=$pos_err+4;
$pos_ms=index($_,"ms",$pos_err);
$delta=$pos_ms-$pos_err;
$error2=substr($_,$pos_err,$delta)+0;

print "\n min2 $min2 media2 $media2 max2
$max2 error $error2 packet $packet2 ";
}

$pos_percentiles=index($_,"Percentiles:");
if($pos_percentiles!=-1)
{
$flag_percentiles2="true";
}
$pos_jitter=index($_,"99.9:");
if($pos_jitter!=-1&&$flag_percentiles2 eq
"true")
{
$pos_jitter=$pos_jitter+5;
$pos_ms=index($_,"ms");
$delta=$pos_ms-$pos_jitter;
$jitter2=substr($_,$pos_jitter,$delta)+0;

print "jitter $jitter2\n";

if($first_sample2==0)
{
`$RRDTOOL $TARGET_RRD2
"$time:U:$min2:$media2:$max2:$jitter2:$error2:$packet2"`;
#aprovechamos y abrimos la
base de datos usoOWMAP2_2 para que no se nos pierda ningun dato
`$RRDTOOL $TARGET_RRD2_2
"$time:U:U:U:U:U:U:U" `;
$time=$time+1;
`$RRDTOOL $TARGET_RRD2_2
"$time:U:U:U:U:U:U:U" `;

$first_sample2=1;
$time=$time+1;
}
else
{
`$RRDTOOL $TARGET_RRD2_2
"$time:U:$min2:$media2:$max2:$jitter2:$error2:$packet2"`;
$time=$time+1;
}
}

```

```

        $min2=0;
        $media2=0;
        $max2=0;
        $jitter2=0;
        $error2=0;
        $packet2=0;

        $flag_percentiles2="";
    }
}

}
close (owamp2_txt);

}
else
{
    $in2="no data";
}

$counter2="";

return $counter2;

});

print "#####TERMINO HILO#####";

#vamos a ir leyendo el archivo OWAMP1 por region
open (owamp1_txt,"<$owamp1_txt") || die "ERROR: No se puede
abrir el archivo\n";
$counter1=0;

#Abrimos la base de datos para que no senos pierda ningun
valor
`$RRDTOOL $TARGET_RRD1
"$time:$delay1:$min1:$medial:$max1:$jitter1:$error1:$packet1"`;
$time=$time+1;

while(<owamp1_txt>)
{
    chomp;
    $pos_delay=index($_,"delay=");
    $pos_ms=index($_,"ms");
    $pos_sync=index($_,"sync");

    if($pos_delay!=-1&&$pos_ms!=-1&&$pos_sync!=-1)
    {

        $pos_delay=$pos_delay+6;
        $delta=$pos_ms-$pos_delay;
        $delay1=substr($_,$pos_delay,$delta)+0;

```

```

        print "delay1:$delay1:$counter1 min1:$min1
medial:$medial max1:$max1 jitter1:$jitter1: err1:$error1
packet1:$packet1\n";

        if($delay1<0)
        {
            $delay1=$delay1*(-1);
        }

        if($first_sample1==0)
        {
            `RRDTOOL $TARGET_RRD1
"$time:$delay1:$min1:$medial:$max1:$jitter1:$error1:$packet1"`;
        }
        else
        {
            `RRDTOOL $TARGET_RRD1_1
"$time:$delay1:$min1:$medial:$max1:$jitter1:$error1:$packet1"`;
        }

        print "respuesta de rrdtool:$?";
        if ( $? == '0' )
            {print "update done\n";}
        else
            {print "update failed\n";}

        $min1=0;
        $medial=0;
        $max1=0;
        $jitter1=0;
        $error1=0;
        $packet1=0;

        $time=$time+1;
    }

else
{
    $pos_sent=index($_,"sent,");
    if($pos_sent!=-1)
    {
        $pos_sent=$pos_sent+5;
        $pos_lost=index($_,"lost",$pos_sent);
        $delta=$pos_lost-$pos_sent;
        $packet1=substr($_,$pos_sent,$delta)+0;
    }

    $pos_min_media=index($_,"min/median/max =");
    if($pos_min_media!=-1)
    {
        $pos_min_media=$pos_min_media+16;
        $pos_barra=index($_,"/", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;
        $min1=substr($_,$pos_min_media,$delta)+0;
        $pos_min_media=$pos_barra+1;

        $pos_barra=index($_,"/", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;
    }
}

```



```

$medial=substr($_,$pos_min_media,$delta)+0;
    $pos_min_media=$pos_barra+1;

    $pos_barra=index($_,"ms",$pos_min_media);
    $delta=$pos_barra-$pos_min_media;
    $max1=substr($_,$pos_min_media,$delta)+0;

}

$pos_err=index($_,"err=");
if($pos_err!=-1)
{
    $pos_err=$pos_err+4;
    $pos_ms=index($_,"ms",$pos_err);
    $delta=$pos_ms-$pos_err;
    $error1=substr($_,$pos_err,$delta)+0;

    print "\n min1 $min1 medial $medial max1
$max1 error1 $error1 packet1 $packet1";
}

$pos_percentiles=index($_,"Percentiles:");
if($pos_percentiles!=-1)
{
    $flag_percentiles1="true";
}
$pos_jitter=index($_,"99.9:");
if($pos_jitter!=-1&&$flag_percentiles1 eq
"true")
{
    $pos_jitter=$pos_jitter+5;
    $pos_ms=index($_,"ms");
    $delta=$pos_ms-$pos_jitter;
    $jitter1=substr($_,$pos_jitter,$delta)+0;

    print "jitter $jitter1\n";

    if($first_sample1==0)
    {
        `RRDTOOL $TARGET_RRD1
"$time:U:$min1:$medial:$max1:$jitter1:$error1:$packet1"`;
        #aprovechamos y abrimos la base de
datos usoOWMAP1_1 para que no se nos pierda ningun dato
        `RRDTOOL $TARGET_RRD1_1
"$time:U:U:U:U:U:U:U" `;
        $time=$time+1;
        `RRDTOOL $TARGET_RRD1_1
"$time:U:U:U:U:U:U:U" `;

        $first_sample1=1;
        $time=$time+1;
    }
    else
    {
        `RRDTOOL $TARGET_RRD1_1
"$time:U:$min1:$medial:$max1:$jitter1:$error1:$packet1"`;
        $time=$time+1;
    }
}

```

```

    }

    $min1=0;
    $medial=0;
    $max1=0;
    $jitter1=0;
    $error1=0;
    $packet1=0;

    $flag_percentiles1="";
}

}

}
close (owamp1_txt);

}
else
{
    $in="no data";
}

$counter1="";

if($hilo->join())
{
    print "matando al hilo";
    $hilo->kill('pailas');
    threads->exit();
}
else
{
    print "no entro a matar el hilo";
}
}
}

```

### 1.3. SCRIPT 3 – RrdObtain

---

Es el encargado de obtener a partir de los dos archivos previamente obtenidos, cuatro archivos de texto los cuales contienen los datos de las pruebas realizadas pero con valores positivos y además como cada prueba de Owamp consta de una prueba de ida y una prueba de vuelta, entonces hay archivos separados para ellas.

```

#!/usr/bin/perl

use warnings;
use RRDs;
use threads;
use threads::shared;
use Time::HiRes qw(usleep nanosleep);

my $TARGET1='usoOWAMP1';
my $TARGET1_1='usoOWAMP1_1';
my $TARGET2='usoOWAMP2';

```

```

my $TARGET2_2='usoOWAMP2_2';

my $owamp1_txt="owamp1";
my $owamp2_txt="owamp2";
my $hilo;
my $counter1="true";
my $counter2="true";

my $delay2=0;
my $min1=0;
my $media1=0;
my $max1=0;
my $jitter1=0;
my $error1=0;
my $packet1=0;

my $delay1=0;
my $min2=0;
my $media2=0;
my $max2=0;
my $jitter2=0;
my $error2=0;
my $packet2=0;

my $flag_percentiles1="";
my $flag_percentiles2="";

my $first_sample1=0;
my $first_sample2=0;
my $in="true";
my $in2="true";

while($counter2)
{

    if($in&&$counter1)
    {

        print "##### hilo para RrdUpdateOWAMP2
#####";

        $hilo=threads->create(sub
        {

            if($in2&&$counter2)
            {

                #vamos a ir leyendo el archivo por region
                open (owamp2_txt,"<$owamp2_txt") || die "ERROR: No se
puede abrir el archivo\n";
                $counter2=0;

                open (TARGET2,"+>$TARGET2") || die "ERROR: No se
puede abrir el archivo\n";
                open (TARGET2_2,"+>$TARGET2_2") || die "ERROR: No se
puede abrir el archivo\n";

                while(<owamp2_txt>)

```

```

{
  chomp;
  $pos_delay=index($_,"delay=");
  $pos_ms=index($_,"ms");
  $pos_sync=index($_,"sync");

  if($pos_delay!=-1&&$pos_ms!=-1&&$pos_sync!=-1)
  {
    $pos_delay=$pos_delay+6;
    $delta=$pos_ms-$pos_delay;
    $delay2=substr($_,$pos_delay,$delta)+0;

    if($delay2<0)
    {
      $delay2=$delay2*(-1);
    }
    print "delay2:$delay2:$counter2
min2:$min2 media2:$media2 max2:$max2 jitter2:$jitter2 err2:$error2
pac:$packet2\n";

    if($first_sample2==0)
    {
      print TARGET2 "$delay2 \n";
    }
    else
    {
      print TARGET2_2 "$delay2 \n";
    }

    $min2=0;
    $media2=0;
    $max2=0;
    $jitter2=0;
    $error2=0;
    $packet2=0;

    $time=$time+1;

  }

  else
  {
    $pos_sent=index($_,"sent,");
    if($pos_sent!=-1)
    {
      $pos_sent=$pos_sent+5;
      $pos_lost=index($_,"lost",$pos_sent);
      $delta=$pos_lost-$pos_sent;
      $packet2=substr($_,$pos_sent,$delta)+0;
    }

    $pos_min_media=index($_,"min/median/max
=");

    if($pos_min_media!=-1)
    {
      $pos_min_media=$pos_min_media+16;
      $pos_barra=index($_,"/", $pos_min_media);
      $delta=$pos_barra-$pos_min_media;
      $min2=substr($_,$pos_min_media,$delta)+0;
      $pos_min_media=$pos_barra+1;
    }
  }
}

```

```

        $pos_barra=index($_,"/", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;

    $media2=substr($_, $pos_min_media, $delta)+0;
        $pos_min_media=$pos_barra+1;

        $pos_barra=index($_, "ms", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;
        $max2=substr($_, $pos_min_media, $delta)+0;

    }

    $pos_err=index($_, "err=");
    if($pos_err!=-1)
    {
        $pos_err=$pos_err+4;
        $pos_ms=index($_, "ms", $pos_err);
        $delta=$pos_ms-$pos_err;
        $error2=substr($_, $pos_err, $delta)+0;

        print "\n min2 $min2 media2 $media2 max2
$max2 error $error2 packet $packet2 ";
    }

    $pos_percentiles=index($_, "Percentiles:");
    if($pos_percentiles!=-1)
    {
        $flag_percentiles2="true";
    }
    $pos_jitter=index($_, "99.9:");
    if($pos_jitter!=-1&&$flag_percentiles2 eq
"true")
    {
        $pos_jitter=$pos_jitter+5;
        $pos_ms=index($_, "ms");
        $delta=$pos_ms-$pos_jitter;
        $jitter2=substr($_, $pos_jitter, $delta)+0;

        print "jitter $jitter2\n";

        if($first_sample2==0)
        {
            $first_sample2=1;
            print TARGET2 "$min2 $media2
$max2 $jitter2 $error2 $packet2 \n" ;
        }
        else
        {
            print TARGET2_2 "$min2
$media2 $max2 $jitter2 $error2 $packet2 \n";
        }

        $min2=0;
        $media2=0;
        $max2=0;
        $jitter2=0;
        $error2=0;
    }

```

```

        $packet2=0;

        $flag_percentiles2="";
    }

}

}
close (owamp2_txt);
close (TARGET2);
close (TARGET2_2);

}
else
{
    $in2="no data";

}

$counter2="";

return $counter2;

});

print "#####TERMINO HILO#####";

#vamos a ir leyendo el archivo OWMAP1 por region
open (owamp1_txt,"<$owamp1_txt") || die "ERROR: No se puede
abrir el archivo\n";
$counter1=0;

open (TARGET1,"+>$TARGET1") || die "ERROR: No se puede
abrir el archivo\n";
open (TARGET1_1,"+>$TARGET1_1") || die "ERROR: No se puede
abrir el archivo\n";

while(<owamp1_txt>)
{
    chomp;
    $pos_delay=index($_,"delay=");
    $pos_ms=index($_,"ms");
    $pos_sync=index($_,"sync");

    if($pos_delay!=-1&&$pos_ms!=-1&&$pos_sync!=-1)
    {

        $pos_delay=$pos_delay+6;
        $delta=$pos_ms-$pos_delay;
        $delay1=substr($_,$pos_delay,$delta)+0;

        print "delay1:$delay1:$counter1 min1:$min1
medial:$medial max1:$max1 jitter1:$jitter1: err1:$error1
packet1:$packet1\n";

        if($delay1<0)
        {

```

```

        $delay1=$delay1*(-1);
    }

    if($first_sample1==0)
    {
        print TARGET1 "$delay1 \n";
    }
    else
    {
        print TARGET1_1 "$delay1 \n";
    }

    $min1=0;
    $medial=0;
    $max1=0;
    $jitter1=0;
    $error1=0;
    $packet1=0;

    $time=$time+1;

}

else
{

    $pos_sent=index($_,"sent,");
    if($pos_sent!=-1)
    {
        $pos_sent=$pos_sent+5;
        $pos_lost=index($_,"lost",$pos_sent);
        $delta=$pos_lost-$pos_sent;
        $packet1=substr($_,$pos_sent,$delta)+0;
    }

    $pos_min_media=index($_,"min/median/max =");
    if($pos_min_media!=-1)
    {
        $pos_min_media=$pos_min_media+16;
        $pos_barra=index($_,"/", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;
        $min1=substr($_,$pos_min_media,$delta)+0;
        $pos_min_media=$pos_barra+1;

        $pos_barra=index($_,"/", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;

    $medial=substr($_,$pos_min_media,$delta)+0;
        $pos_min_media=$pos_barra+1;

        $pos_barra=index($_,"ms", $pos_min_media);
        $delta=$pos_barra-$pos_min_media;
        $max1=substr($_,$pos_min_media,$delta)+0;

    }

    $pos_err=index($_,"err=");
    if($pos_err!=-1)
    {
        $pos_err=$pos_err+4;
    }
}

```

```

        $pos_ms=index($_,"ms",$pos_err);
        $delta=$pos_ms-$pos_err;
        $error1=substr($_,$pos_err,$delta)+0;

        print "\n min1 $min1 medial $medial max1
$max1 error1 $error1 packet1 $packet1";
    }

    $pos_percentiles=index($_,"Percentiles:");
    if($pos_percentiles!=-1)
    {
        $flag_percentiles1="true";
    }
    $pos_jitter=index($_,"99.9:");
    if($pos_jitter!=-1&&$flag_percentiles1 eq
"true")
    {
        $pos_jitter=$pos_jitter+5;
        $pos_ms=index($_,"ms");
        $delta=$pos_ms-$pos_jitter;
        $jitter1=substr($_,$pos_jitter,$delta)+0;

        print "jitter $jitter1\n";

        if($first_sample1==0)
        {
            print TARGET1 "$min1 $medial $max1
$jitter1 $error1 $packet1 \n";

            $first_sample1=1;
        }
        else
        {
            print TARGET1_1 "$min1 $medial
$max1 $jitter1 $error1 $packet1 \n";
        }

        $min1=0;
        $medial=0;
        $max1=0;
        $jitter1=0;
        $error1=0;
        $packet1=0;

        $flag_percentiles1="";
    }
}

}
close (owamp1_txt);
close (TARGET1);
close (TARGET1_1);
}
else

```



```

    {
        $in="no data";
    }

    $counter1="";

    if($hilo->join())
    {
        print "matando al hilo";
        $hilo->kill('pailas');
        threads->exit();
    }
    else
    {
        print "no entro a matar el hilo";
    }
}

```

#### 1.4. SCRIPT 4 – RrdGraphOWAMP

---

A pesar que este script no fue utilizado para las pruebas definitivas debido a razones estéticas, su funcionamiento se comprobó en pruebas previas; es el encargado de graficar los resultados de Owamp y puede ser utilizado en otros proyectos si se desea.

```

#!/usr/bin/perl
use RRDs;
#creamos la grafica
$start_time=time();
RRDs::graph("USO_DE_OWAMP1.png","--title=OWAMP Router 2",
    "--vertical-label=ms",
    "--right-axis-format=%lf",
    "--start=1272341837",
    "--end=1272341937",
    "DEF:probe1-graph=usoOWAMP1.rrd:delay:AVERAGE",
    "DEF:probe2-graph=usoOWAMP1.rrd:jitter:AVERAGE",
    "DEF:probe3-graph=usoOWAMP1.rrd:media:AVERAGE",
    "DEF:probe4-graph=usoOWAMP1.rrd:error:AVERAGE",
    "DEF:probe5-graph=usoOWAMP1.rrd:packet:AVERAGE",
    "VDEF:valor_jitter=probe3-graph,MAXIMUM",
    "AREA:probe1-graph#114b92:Delay ", #6d1a85
    "GPRINT:probe1-graph:MIN: Min\\: %5.3lf ",
    "GPRINT:probe3-graph:MAX: Avg\\: %5.3lf ", #no puede ser el
minimo, porque esta columna solo tien un valor en todos las 1000 filas
    "GPRINT:probe1-graph:MAX: Max\\: %5.3lf ",
    "COMMENT: \n",
    "LINE2:valor_jitter#fbed0a:Jitter",
    "GPRINT:probe2-graph:MAX: \ %5.3lf ",
    "GPRINT:probe4-graph:MAX:Error\\: %5.3lf ",
    "GPRINT:probe5-graph:MAX:Lost Packet\\: %5.0lf ",
    "COMMENT: ",);

my $err=RRDs::error;
if($err)
{
print "se presento el siguiente problema al generar la grafica:
$err\n";
}

```

```

RRDs::graph("USO_DE_OWAMP1_1.png", "--title=OWAMP Router 2",
  "--vertical-label=ms",
  "--start=1272341938",
  "--end=1272342038",
  "DEF:probe1-graph=usoOWAMP1_1.rrd:delay:AVERAGE",
  "DEF:probe2-graph=usoOWAMP1_1.rrd:jitter:AVERAGE",
  "DEF:probe3-graph=usoOWAMP1_1.rrd:media:AVERAGE",
  "DEF:probe4-graph=usoOWAMP1_1.rrd:error:AVERAGE",
  "DEF:probe5-graph=usoOWAMP1_1.rrd:packet:AVERAGE",
  "VDEF:valor_jitter=probe3-graph,MAXIMUM",
  "AREA:probe1-graph#fbed0a:Delay ",
  "GPRINT:probe1-graph:MIN: Min\\: %5.3lf ",
  "GPRINT:probe3-graph:MAX: Avg\\: %5.3lf ",
  "GPRINT:probe1-graph:MAX: Max\\: %5.3lf ",
  "#AREA:probe1-graph#fbed0a:Delay bits ", #6d1a85
  "COMMENT:      \n",
  "LINE2:valor_jitter#114b92:Jitter",
  "GPRINT:probe2-graph:MAX: \ %5.3lf ",
  "GPRINT:probe4-graph:MAX:Error\\: %5.3lf ",
  "GPRINT:probe5-graph:MAX:Lost Packet\\: %5.0lf ",
  "COMMENT: ",);

my $err=RRDs::error;
if($err)
{
print "se presento el siguiente problema al generar la grafica:
$err\n";
}

##### SEGUNDA PRUEBDA
#####

RRDs::graph("USO_DE_OWAMP2.png", "--title=OWAMP Router 2",
  "--vertical-label=ms",
  "--start=1272341837",
  "--end=1272341937",
  "DEF:probe1-graph=usoOWAMP2.rrd:delay:AVERAGE",
  "DEF:probe2-graph=usoOWAMP2.rrd:jitter:AVERAGE",
  "DEF:probe3-graph=usoOWAMP2.rrd:media:AVERAGE",
  "DEF:probe4-graph=usoOWAMP2.rrd:error:AVERAGE",
  "DEF:probe5-graph=usoOWAMP2.rrd:packet:AVERAGE",
  "VDEF:valor_jitter=probe3-graph,MAXIMUM",
  "AREA:probe1-graph#114b92:Delay ", #6d1a85
  "GPRINT:probe1-graph:MIN: Min\\: %5.3lf ",
  "GPRINT:probe3-graph:MAX: Avg\\: %5.3lf ", #no puede ser el
minimo, porque esta columna solo tien un valor en todos las 1000 filas
  "GPRINT:probe1-graph:MAX: Max\\: %5.3lf ",
  "COMMENT:      \n",
  "LINE2:valor_jitter#fbed0a:Jitter",
  "GPRINT:probe2-graph:MAX: \ %5.3lf ",
  "GPRINT:probe4-graph:MAX:Error\\: %5.3lf ",
  "GPRINT:probe5-graph:MAX:Lost Packet\\: %5.0lf ",
  "COMMENT: ",);

my $err=RRDs::error;
if($err)
{

```

```
print "se presento el siguiente problema al generar la grafica:
\$err\n";
}
```

```
RRDs::graph("USO_DE_OWAMP2_2.png","--title=OWAMP Router 2",
"--vertical-label=ms",
"--start=1272341938",
"--end=1272342038",
"DEF:probe1-graph=usoOWAMP2_2.rrd:delay:AVERAGE",
"DEF:probe2-graph=usoOWAMP2_2.rrd:jitter:AVERAGE",
"DEF:probe3-graph=usoOWAMP2_2.rrd:media:AVERAGE",
"DEF:probe4-graph=usoOWAMP2_2.rrd:error:AVERAGE",
"DEF:probe5-graph=usoOWAMP2_2.rrd:packet:AVERAGE",
"VDEF:valor_jitter=probe3-graph,MAXIMUM",
"AREA:probe1-graph#fbed0a:Delay ",
"GPRINT:probe1-graph:MIN: Min\\: %5.3lf ",
"GPRINT:probe3-graph:MAX: Avg\\: %5.3lf ",
"GPRINT:probe1-graph:MAX: Max\\: %5.3lf ",
#"AREA:probe1-graph#fbed0a:Delay bits ", #6d1a85
"COMMENT: \n",
"LINE2:valor_jitter#114b92:Jitter ",
"GPRINT:probe2-graph:MAX:\ %5.3lf ",
"GPRINT:probe4-graph:MAX:Error\\: %5.3lf",
"GPRINT:probe5-graph:MAX:Lost Packet\\: %5.0lf ",
"COMMENT: ",);
```

```
my $err=RRDs::error;
if($err)
{
print "se presento el siguiente problema al generar la grafica:
\$err\n";
}
```

## 2. CPU – RrdObtainCPU

---

Se encarga de obtener la información de uso de CPU del router mediante peticiones snmp get, y de almacenar el valor en un archivo llamado *cpu*.

```
#!/usr/bin/perl

use warnings;

print "Ingresa la direccion del destino para monitorear CPU:";
my $dir=<STDIN>;
chomp($dir);
my $SNMPGET="snmpget -v2c -c ipsec $dir
1.3.6.1.4.1.9.9.109.1.1.1.1.3.1";
my $valor_cpu=0;
my $cpu_txt="cpu";

open (cpu_txt,"+>$cpu_txt") || die "ERROR: No se puede abrir el
archivo\n";

for(my $i=1;$i<=360;$i++)
{
    select(undef, undef, undef, 5);
```

```

$in=`$SNMPGET`;
if($in)
{
    $gauge=index($in,"Gauge32")+8;
    $valor_cpu=(substr($in,$gauge)+0);
}
else
{
    $in="no data";
    $valor_cpu="no data";
}

print "\n","respuesta del snmpget: ",$in;
print "valor de cpu:$valor_cpu T\n";

print cpu_txt "$valor_cpu \n";

}

close (cpu_txt);

```

### 3. SCRIPTS – RrdObtainVelocity

---

Se encarga de obtener la información de velocidad del router mediante peticiones snmp get, y de almacenar el valor en un archivo llamado *throughput*.

```

#!/usr/bin/perl

use warnings;
use RRDs;

print "Ingresa la direccion del destino para monitorear la velocidad:";
my $dir=<STDIN>;
chomp($dir);

my $SNMPGET_IN="snmpget -v2c -c ipsec $dir 1.3.6.1.2.1.2.2.1.10.2";
my $SNMPGET_OUT="snmpget -v2c -c ipsec $dir 1.3.6.1.2.1.2.2.1.16.2";
my $throguhput_txt="throughput";
my $throguhputIN_txt="throughputIN";
my $throguhputOUT_txt="throughputOUT";

open (throughput_txt,"+>$throguhput_txt") || die "ERROR: No se puede
abrir el archivo\n";
open (throughputIN_txt,"+>$throguhputIN_txt") || die "ERROR: No se
puede abrir el archivo";
open (throughputOUT_txt,"+>$throguhputOUT_txt") || die "ERROR: No se
puede abrir el archivo";

for(my $i=1;$i<=360;$i++)
{
    $in1=`$SNMPGET_IN`;

```

```

if($in1)
{
    $in_counter1=index($in1,"Counter32")+10;
    $in_bytes1=(substr($in1,$in_counter1)+0);
}

$out1=`$SNMPGET_OUT`;
if($out1)
{
    $out_counter1=index($out1,"Counter32")+10;
    $out_bytes1=(substr($out1,$out_counter1)+0);
}

select(undef, undef, undef, 5);

$in2=`$SNMPGET_IN`;

if($in2)
{
    $in_counter2=index($in2,"Counter32")+10;
    $in_bytes2=(substr($in2,$in_counter2)+0);
}

$out2=`$SNMPGET_OUT`;
if($out2)
{
    $out_counter2=index($out2,"Counter32")+10;
    $out_bytes2=(substr($out2,$out_counter2)+0);
}

if($in_bytes1&&$out_bytes2&&$in_bytes2&&$out_bytes1)
{
    $bps_in=int((( $in_bytes2-$in_bytes1)*8)/5);
    $bps_out=int((( $out_bytes2-$out_bytes1)*8)/5);
}
else
{
    $bps_in="no data";
    $bps_out="no data";
}

print "\n","respuesta del snmpget in1: ",$in1;
print "\n","respuesta del snmpget in2: ",$in2;
print "valor de bits in:",$bps_in,"\n";

print "\n","respuesta del snmpget out1: ",$out1;
print "\n","respuesta del snmpget out2: ",$out2;
print "valor de bits out:",$bps_out,"\n";

print throughput_txt "$bps_in \n";
print throughput_txt "$bps_out \n";

print throughputIN_txt "$bps_in\n";
print throughputOUT_txt "$bps_out\n";
}

close (throughput_txt);
close (throughputIN_txt);
close (throughputOUT_txt);

```