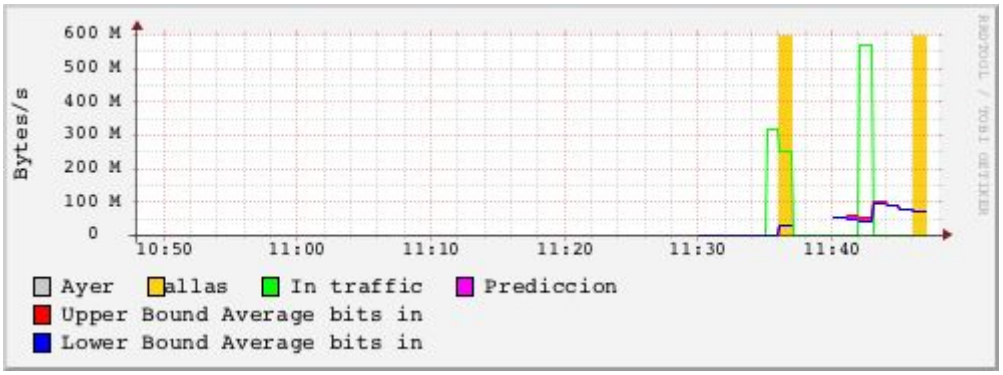
EXAMEN 3

Integrantes:

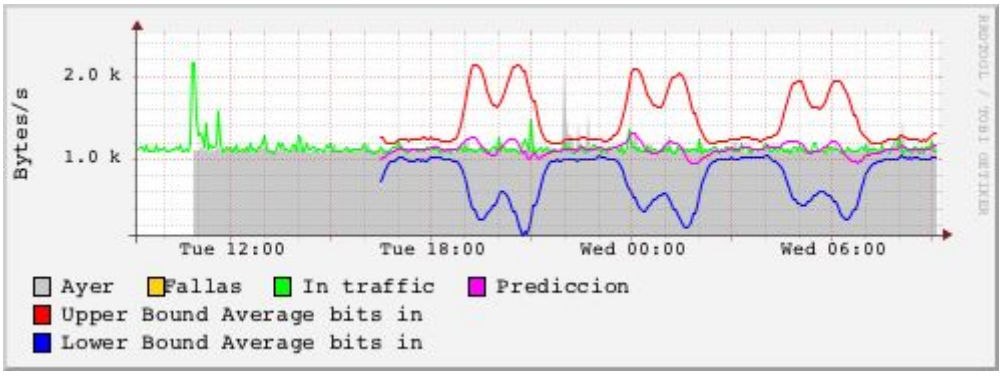
- Monroy Martos Elioth
- Hernández Pineda Miguel Angel
- Zúñiga Hernández Carlos

Equipo: 9
Grupo: 4CM1

Evidencia 2



En la siguiente gráfica se muestra un comportamiento anomalo, no se alcanza a mostrar el pasado debido a que los valores se encuentran en Megas, en lugar de Kilos.



La gráfica anterior muestra que el programa si gráfica el historico. Solo como demostración.
El código usado para hacer la detección fue el siguiente:

```
from pysnmp.hlapi import *
import rrdtool
import time
from convertidor import convertir_a_numero
from notificador import Notificador
from logger import Logger
#Funcion para hacer consultas a un solo objeto con referencias al nombre
```

```
def consultav2SNMP(comunidad,host,version,interfaz,grupo,objeto,puerto):
        resultado=""
        errorIndication, errorStatus, errorIndex, varBinds =
next(getCmd(SnmpEngine(),CommunityData(comunidad,mpModel=version),UdpTransport
Target((host, puerto)),ContextData(),

ObjectType(ObjectIdentity(grupo,objeto,interfaz).addAsn1MibSource('file:///usr/share/snm
p','http://mibs.snmplabs.com/asn1/@mib@')))

        if errorIndication:
            print(errorIndication)
        elif errorStatus:
            print('%s at %s' % (errorStatus.prettyPrint(),
                        errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
        else:
            for varBind in varBinds:
                VarB=(' = '.join([x.prettyPrint() for x in varBind]))
                resultado=VarB.partition(' = ')[2]
        return resultado

def consultaSNMP(comunidad,host,version,interfaz,grupo,objeto1,objeto2,puerto):
        resultado=[]
        errorIndication, errorStatus, errorIndex, varBinds =
next(getCmd(SnmpEngine(),CommunityData(comunidad,mpModel=version),UdpTransport
Target((host, puerto)),ContextData(),

ObjectType(ObjectIdentity(grupo,objeto1,interfaz).addAsn1MibSource('file:///usr/share/sn
mp','http://mibs.snmplabs.com/asn1/@mib@')),

ObjectType(ObjectIdentity(grupo,objeto2,interfaz).addAsn1MibSource('file:///usr/share/sn
mp','http://mibs.snmplabs.com/asn1/@mib@')))

        if errorIndication:
            print(errorIndication)
            resultado.append("")
            resultado.append("")
        elif errorStatus:
            print('%s at %s' % (errorStatus.prettyPrint(),
                        errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
        else:
            for varBind in varBinds:
                VarB=(' = '.join([x.prettyPrint() for x in varBind]))
                resultado.append(VarB.partition(' = ')[2])
        return resultado
def crearRRDHW(nombre,inicio,step,tiempo,steps1,row1):
        ret=rrdtool.create(nombre,'--start',inicio,'--step',step,
                "DS:inoctets:COUNTER:"+tiempo+":U:U",
                "RRA:AVERAGE:0.5:"+steps1+":"+row1,
                "RRA:HWPREDICT:300:0.1:0.0035:10:3",
                "RRA:SEASONAL:10:0.1:2",
```

```
                "RRA:DEVSEASONAL:10:0.1:2",
                "RRA:DEVPREDICT:300:4",
                "RRA:FAILURES:300:3:5:4")
        if ret:
                print(rrdtool.error())

#crearRRDHW("prueba.rrd","N","1","600","1","600")
final=False
fecha_inicio=fecha_final=""
noti=Notificador("Compu Profa")
log=Logger("Compu Profa")
archivo_rdd="predict.rrd"
while 1:
        # consulta=consultav2SNMP("public","localhost",1,1,'IF-MIB','ifInOctets',161)

consulta=consultav2SNMP("variation/virtualtable","10.100.71.200",1,1,'IF-MIB','ifInOctets',
1024)
        valor = "N:" + str(consulta)
        # print (valor)
        ultimo=rrdtool.last(archivo_rdd)
        inicio=ultimo-3600
        ayerInicio=(inicio-86400)
        ayerFinal=ultimo-86400
        primero=rrdtool.first(archivo_rdd)
        rrdtool.update(archivo_rdd, valor)
        rrdtool.dump(archivo_rdd,'prueba.xml')
        ret = rrdtool.graphv( "prueba.png",
             "--start",str(inicio),
            "--end",str(ultimo),
             "--vertical-label=Bytes/s",
               '--slope-mode',
               "DEF:obs="+archivo_rdd+":inoctets:AVERAGE",
               "DEF:pred="+archivo_rdd+":inoctets:HWPREDICT",
               "DEF:dev="+archivo_rdd+":inoctets:DEVPREDICT",
               "DEF:fail="+archivo_rdd+":inoctets:FAILURES",
               "DEF:yvalue="+archivo_rdd+":inoctets:AVERAGE:start=" + str(ayerInicio) +
":end=" + str(ayerFinal),
                           'SHIFT:yvalue:86400',
               "CDEF:scaledobs=obs,8,*",
               "CDEF:upper=pred,dev,2,*,+",
               "CDEF:lower=pred,dev,2,*,-",
               "CDEF:scaledupper=upper,8,*",
               "CDEF:scaledlower=lower,8,*",
               "CDEF:scaledh=yvalue,8,*",
               "CDEF:scaledpred=pred,8,*",
               "VDEF:lastfail=fail,LAST",
                                       "PRINT:lastfail:%6.2lf %S ",
                                       "PRINT:lastfail: %c :strftime",
            "AREA:scaledh#C9C9C9:Ayer",
            "TICK:fail#FDD017:1.0:Fallas",
```

```
            "LINE1:scaledobs#00FF00:In traffic",
            "LINE1:scaledpred#FF00FF:Prediccion\\n",
            "LINE1:scaledupper#ff0000:Upper Bound Average bits in\\n",
            "LINE1:scaledlower#0000FF:Lower Bound Average bits in")
        if "nan" not in ret["print[0]"]:
                ultima_falla=float(ret["print[0]"])
                if ultima_falla==0:
                        # print("No soy una falla")
                        if final:
                                final=not final
                                # print("Fecha inicio: "+fecha_inicio)
                                # print("Fecha final: "+fecha_final)
                                noti.enviarCorreo(4,fecha_final,"prueba.png")
                                log.escribirLog(4,fecha_final)
                else:
                        # print("Soy una falla")
                        fecha_final=ret["print[1]"]
                        if not final:
                                # print("Falla detectada")
                                fecha_inicio=ret["print[1]"]
                                final=not final
                                noti.enviarCorreo(3,fecha_inicio,"prueba.png")
                                log.escribirLog(3,fecha_inicio)

if ret:
    print (rrdtool.error())
    time.sleep(300)
```