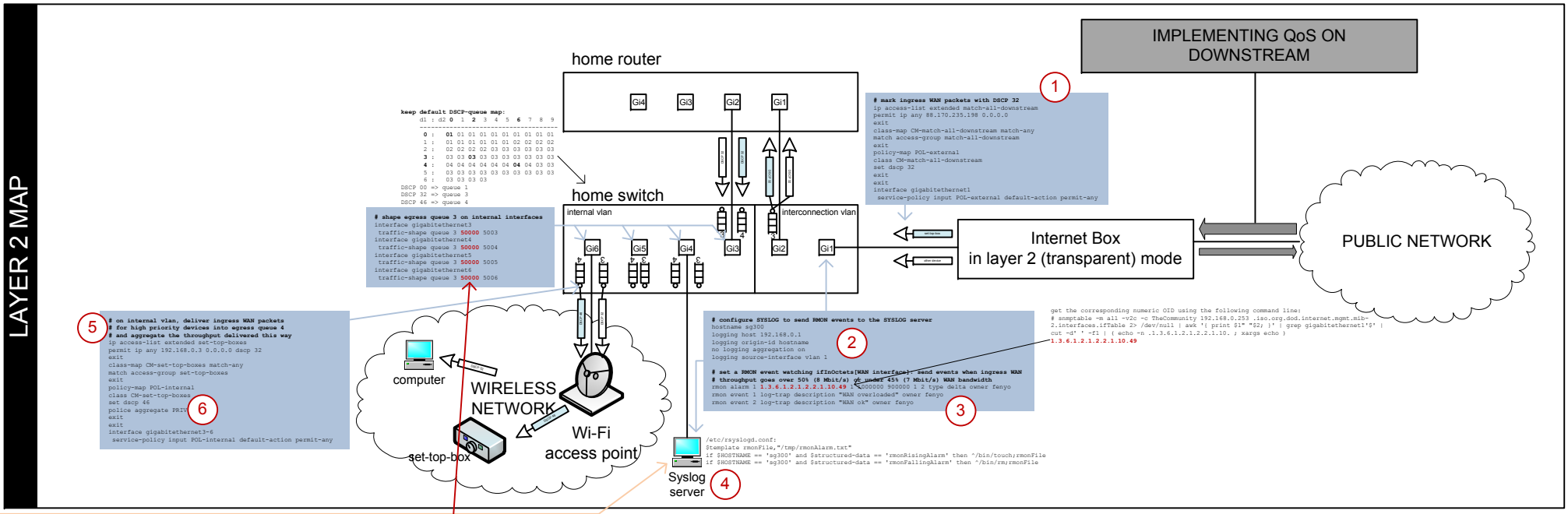
IMPLEMENTING QoS ON DOWNSTREAM

LAYER 2 MAP




```
keep default DSCP-queue map:
     d1 : d2 0  1  2  3  4  5  6  7  8  9
    ----------------------------------------
     0 :  01 01 01 01 01 01 01 01 01 01
     1 :  01 01 01 01 01 01 02 02 02 02
     2 :  02 02 02 02 03 03 03 03 03 03
     3 :  03 03 03 03 03 03 03 03 03 03
     4 :  04 04 04 04 04 04 04 04 03 03
     5 :  03 03 03 03 03 03 03 03 03 03
     6 :  03 03 03 03
DSCP 00 => queue 1
DSCP 32 => queue 3
DSCP 46 => queue 4
```

```
# mark ingress WAN packets with DSCP 32
ip access-list extended match-all-downstream
permit ip any 88.170.235.198 0.0.0.0
exit
class-map CM-match-all-downstream match-any
match access-group match-all-downstream
exit
policy-map POL-external
class CM-match-all-downstream
set dscp 32
exit
exit
interface gigabitethernet1
 service-policy input POL-external default-action permit-any
```

```
# shape egress queue 3 on internal interfaces
interface gigabitethernet3
 traffic-shape queue 3 50000 5003
interface gigabitethernet4
 traffic-shape queue 3 50000 5004
interface gigabitethernet5
 traffic-shape queue 3 50000 5005
interface gigabitethernet6
 traffic-shape queue 3 50000 5006
```

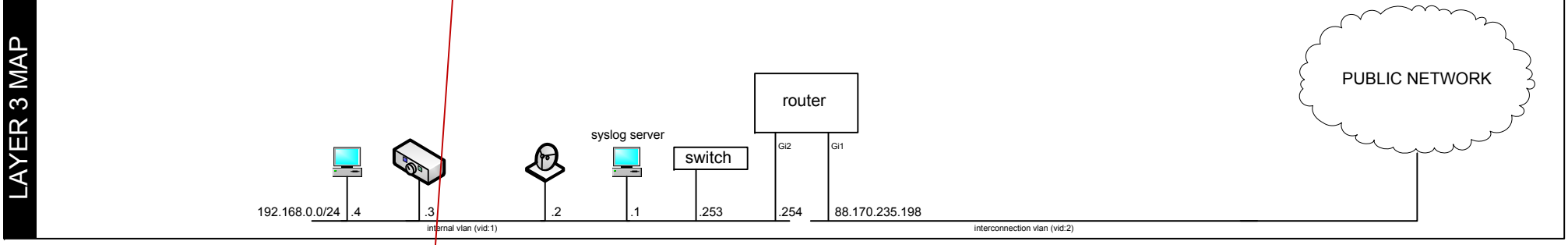
```
# on internal vlan, deliver ingress WAN packets
# for high priority devices into egress queue 4
# and aggregate the throughput delivered this way
ip access-list extended set-top-boxes
permit ip any 192.168.0.3 0.0.0.0 dscp 32
exit
class-map CM-set-top-boxes match-any
match access-group set-top-boxes
exit
policy-map POL-internal
class CM-set-top-boxes
set dscp 46
police aggregate PRIV
exit
exit
interface gigabitethernet3-6
 service-policy input POL-internal default-action permit-any
```

```
# configure SYSLOG to send RMON events to the SYSLOG server
hostname sg300
logging host 192.168.0.1
logging origin-id hostname
no logging aggregation on
logging source-interface vlan 1

# set a RMON event watching ifInOctets[WAN interface] : send events when ingress WAN
# throughput goes over 50% (8 Mbit/s) or under 45% (7 Mbit/s) WAN bandwidth
rmon alarm 1 1.3.6.1.2.1.2.2.1.10.49 7000000 900000 1 2 type delta owner fenyo
rmon event 1 log-trap description "WAN overloaded" owner fenyo
rmon event 2 log-trap description "WAN ok" owner fenyo
```

```
get the corresponding numeric OID using the following command line:
# snmptable -m all -v2c -c TheCommunity 192.168.0.253 .iso.org.dod.internet.mgmt.mib-
2.interfaces.ifTable 2> /dev/null | awk '{ print $1" "$2; }' | grep gigabitethernet'$' |
cut -d' ' -f1 | { echo -n .1.3.6.1.2.1.2.2.1.10. ; xargs echo }
1.3.6.1.2.1.2.2.1.10.49
```

```
/etc/rsyslogd.conf:
$template rmonFile,"/tmp/rmonAlarm.txt"
if $HOSTNAME == 'sg300' and $structured-data == 'rmonRisingAlarm' then ^/bin/touch;rmonFile
if $HOSTNAME == 'sg300' and $structured-data == 'rmonFallingAlarm' then ^/bin/rm;rmonFile
```

LAYER 3 MAP




```
Daemon process on the syslog server:

#!/bin/zsh
FULL=/tmp/rmonAlarm.txt
# (BW_MAX - BW_MIN) must be a multiple of BW_STEP
BW_NOLIMIT=50000 # must be 3 to 5 times the max downstream throughput
BW_MAX=10000 # must 50% of the max downstream throughput
BW_STEP=3000 # one step
BW_MIN=1000 # minimum bandwith shaped to each internal interface egress queue 3
BW_CUR=$BW_NOLIMIT

needshape() {
  test ! -f $FULL && return 1
  # find the OID using such a command line: snmpwalk -On -m all -v2c -c TheCommunity switch rCiscoAggregatePolicerStatisticsTable 2> /dev/null | fgrep .1.3.6.1.4.1.9.6.1.101.88.35.3.1.3 | cut -d' ' -f1
  OID=.1.3.6.1.4.1.9.6.1.101.88.35.3.1.3.2
  snmpget -Oq -v2c -c TheCommunity 192.168.0.253 $OID | read foo X
  sleep 1
  snmpget -Oq -v2c -c TheCommunity 192.168.0.253 $OID | read foo Y
  test $X -eq $Y && return 1
  # add 1000 (value of CIR) because we only get out-of-profile quantity with SNMP
  echo multimedia: $(( 1000 + (Y - X) * 8 / 1000 )) kbit/s
  return 0
}

shape() {
  date
  echo shaping to $1 kbit/s
  # use the committed-burst size values to find the committed rate CIDs in rCiscoPolicerTable and set the shaped throughput
  snmpwalk -On -v2c -c TheCommunity 192.168.0.253 .1.3.6.1.4.1.9.6.1.101.88.23.1.7 | egrep '500.$' | sed 's/23.1.7/23.1.5/' | sed 's/ / i /' | sed 's/ 500.$/ '$1'g/' | sargs snmpset -Oq -v2c -c TheCommunity 192.168.0.253 > /dev/null
}

slower() {
  test $BW_CUR -eq $BW_MIN && return
  if test $BW_CUR -eq $BW_NOLIMIT
  then
    BW_CUR=$BW_MAX
    return
  fi
  BW_CUR=$(( BW_CUR - BW_STEP ))
}

faster() {
  test $BW_CUR -eq $BW_NOLIMIT && return
  if test $BW_CUR -eq $BW_MAX
  then
    BW_CUR=$BW_NOLIMIT
    return
  fi
  BW_CUR=$(( BW_CUR + BW_STEP ))
}

shape $BW_NOLIMIT
while sleep 1
do
  LAST_CUR=$BW_CUR

  if needshape
  then
    slower
  else
    faster
  fi

  test $LAST_CUR -ne $BW_CUR && shape $BW_CUR
done
```

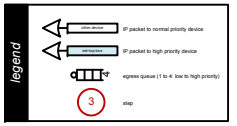
```
# create 2 vlans
vlan database
default-vlan vlan 1
exit
vlan database
vlan 2
exit
```

```
# configure 2 vlans: internal & interco
interface vlan 1
 name internal
 ip address 192.168.1.253 255.255.255.0
 no ip address dhcp
interface vlan 2
 name interco
interface gigabitethernet1
 description CPE
 switchport access vlan 2
interface gigabitethernet2
 description home router WAN
 switchport mode access
 switchport access vlan 2
interface gigabitethernet3-6
 switchport mode access
```

```
# put the device into layer 2 mode
set system mode switch

# packets which are not classified by policy map rules to a QoS
# action are mapped to an egress queue based on the packet's fields
qos advanced ports-trusted

# enable Weighted Random Tail Drop
qos wrr-queue wrtd

# create an aggregate-policer with CIR 1000
qos statistics aggregate-policer PRIV
qos aggregate-policer PRIV 1000 3000 exceed-action none
```

Seven steps:

1- **mark downstream WAN packets with DSCP 32 to match them later**
Their destination address is your NAT public address, so you can not discriminate those addressed to your high priority devices before crossing back the NAT translation made by your router.



2- **configure a RMON alarm to alert a server when downstream is overloaded**

3- **configure syslog to send RMON events to the server**
Since you need reliable alerts, avoid sending SNMP traps over UDP but use SYSLOG events on TCP.

4- configure the SYSLOG server to inform a daemon that downstream is overloaded (create a temporary File when overloaded)

5- on internal vlan, deliver downstream WAN packets addressed to high priority devices into high priority egress queue 4 by translating their DSCP to 46. Other WAN packets still have DSCP 32, such that they are delivered to egress queue 3.
Since this is processed on internal vlan interfaces, it is done after NAT reverse translation, then you can simultaneously match packets for high priority devices (use destination address) and those who come from the WAN (they match DSCP 32). After that step, packets coming from WAN to high priority devices have DSCP 46 (=> delivered to egress queue 4), those coming from WAN to low priority devices have DSCP 32 (=> delivered to egress queue 3) and locally generated packets going to other local hosts or to the WAN have DSCP 0 (=> delivered to egress queue 1).

6- count the previously matched packets and aggregate the results for every interfaces, using an aggregate policer

7- on the SYSLOG server, run a daemon that:
- sleep when the WAN downstream is not overloaded
- use SNMP sets to increase/decrease the committed rate of the traffic shaping on queue 3 on internal interfaces, step by step, until the WAN downstream is not overloaded anymore

Implementing downstream QoS with a Cisco Small Business switch using:

- SNMP
- a RMON probe
- an aggregate policer
- multiple egress queues
- traffic shaping
- DiffServ marking
- a SYSLOG server running a daemon receiving RMON events, collecting aggregate policer statistics and modifying the traffic shaping configuration depending on the network behaviour