

NetEqualizer User Guide

© Copyright 2005, 2006, 2007, 2008 APConnections. All rights reserved.

No part of this publication including text, examples, diagrams or illustrations may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical or otherwise, for any purpose, without prior written permission of APConnections, Inc.

Table of Contents

Where to install NetEqualizer	3
Setting your trunk size	3
Equalizing (default mode)	4
Default mode and some simple tuning ideas	4
The Basic three Parameters	5
Sizing internal Tables	6
Priority and Limits, NetEqualizer Application exception rules	8
Priority Techniques	8
Priority IP addresses	8
Traffic Control by Host/Subnet hard limits	9
Traffic Control by MAC address	9
MAC redirection	9
Traffic Control by VLAN	10
Masking off traffic (ignore traffic)	10
Setting Connection Limits	11
Bytes per day, week or month User Limits and Warnings	12
Monitoring and Reporting	13
To view all active Internet connections	13
To view the current log file and slowed down connections.	13
To see a list of active IP users and their associated MAC addresses	14
Graphical Reporting tools	14
Bandwidth Pools and the Virtual Equalizer	15
Adding IP Addresses to a Pool	15
Tips and Tricks	17
Disabling the default rules	17
NetEqualizer adjusts to traffic over several seconds	17
What to expect for your limit precision	17
Sometimes it's not NetEqualizer	17
Security Precautions	17
Persistence of Configurations, changing on the fly	18
Redundancy and Reliability	18
Appendix 1- Default parameter settings and units	19
Appendix 2 - Setting/Forcing LAN Speeds	20
Appendix 3 - Tuning Hard Limit and Pool Sensitivity	22
Appendix 4 - Packet Capturing for taps such as Calea	23
Appendix 5 - Network Access Control (NAC)	24
Appendix 6 - NetEqualizer User-Quota API (NUQ API) Programmer's Toolkit	26

Thank you for purchasing NetEqualizer. NetEqualizer offers a wide range of bandwidth control options, while at the same time allowing the user to keep it simple. Using NetEqualizer in default factory modes will take care of almost all network congestion and priority traffic flow requirements and is the recommended operational mode for most customers.

We understand you may need to get started right away. The basic setup details and minimal settings required are detailed in our Quick Start Guide. Most people will never need to go beyond that.

Once up and running, it is a good idea to review the entire user guide and all the advanced features available to you. Should you need assistance please visit <http://www.netequalizer.com> or contact your local reseller or distribution partner.

Where to install NetEqualizer

NetEqualizer can be installed on any link whose traffic you would like to shape. For maximum effectiveness, most users should install NetEqualizer between the network users and the Internet trunk.

Traffic running between your network and the Internet is generally a constriction point in traffic flow where many users compete for this limited resource. By placing your NetEqualizer at these junctions you will automatically optimize your Internet speed.

Note: For a detailed list of the steps necessary to get up and running, please see the NetEqualizer Quick Start Guide. If you do not have a copy of the Quick Start Guide, please contact your NetEqualizer sales rep or email support@apconnections.net.

Setting your trunk size

From the Web GUI screen select ->Modify Params

NetEqualizer allows for different speeds for incoming and outgoing links. The parameters are TRUNK_UP and TRUNK_DOWN. These parameters are set in **bytes/per second**. For a standard T1 trunk, it is recommended to set each of these parameters to 192000. If you have different speeds for downlink and uplink traffic, you would adjust these parameters accordingly.

Note: Do Misc/Stop NetEq & Misc/Start NetEq after changing any of these settings.

In the following sections we will discuss ways to tune your equalizing behavior by making these parameters larger or smaller than your actual link size.

Equalizing (default mode)

Default mode and some simple tuning ideas

Default mode is the initial mode that NetEqualizer comes configured in; it is automatically in this mode the first time you start it up. Default mode enables the network to deliver traffic equalizing for the most common situations--without the need for expertise in complex traffic shaping rules.

Once NetEqualizer is installed and up and running, a review of the standard log file will allow you to monitor and analyze how NetEqualizer is responding to your network's traffic.

When your network is experiencing moderate to heavy use, you will see entries containing the word PENALTY followed by two IP addresses. PENALTY indicates that NetEqualizer's built-in fairness rules have determined that the communication link between these two IP addresses (connection) is using too much bandwidth, so NetEqualizer has issued a penalty against this connection. The penalty causes all data on that connection to slow down. If NetEqualizer determines that this connection is still using too much bandwidth, it will increase the delay on the connection.

NetEqualizer bases its decision to issue penalties on built-in fairness rules:

- The persistence of the user.
- The length of time the connections been live. The longer the time, the more likely a penalty
- The amount of bandwidth used relative to the total size of the trunk
- The number of users on the trunk. The more users the less bandwidth NetEqualizer will allow per user before issuing a penalty.

There are advantages to using the default "fairness rules" over setting specific limits for p2p applications and other applications that burst.

To-date, no p2p application, has been able to subvert NetEqualizer fairness rules. NetEqualizer recognizes techniques such as HTTP tunneling (spoofing a p2p to look like web traffic)

The fairness rules act like a general antibiotic and will auto detect new p2p applications. As new applications are developed there is no need to purchase expensive software upgrades or manually maintain rules.

The built-in fairness rules require much less CPU. Therefore, if you use these rules, your NetEqualizer can run at the maximum specified traffic allotment

Using the Ratio parameter to influence default rules

NetEqualizer's "**RATIO**" allows you to influence the default rules without actually changing the rules. The RATIO parameter refers to the network utilization on a percentage basis. RATIO can be set from 1 to 100. A value of 100 tells NetEqualizer not to have the default rules kick in until the trunk is 100 percent utilized, a value of 85 would have the rules kick in at 85 percent utilized.

A powerful traffic controller, combining your custom rules with the default rules in the background

Sometimes the sheer volume of users on the network cannot be controlled by the custom rules you have implemented. Setting a per user limit of 512kbs will prevent a user from going over the 512kbs prescribed level; but if 20 of your users get on at one time with large downloads, a T1 trunk, for example, is quickly overwhelmed (to set a per user limit, please see the **Limit traffic by Host/Subnet soft limits** section). The default rules that kick in at 85 percent trunk utilization provide a unique safety valve for those busy hours when your trunk gets full. NetEqualizer is the only tool on the market with this flexibility.

- Individual bandwidth control rules (limit by user)
- Default rules only, balances your traffic all the time giving priority to web, chat, and e-mail
- A combination of custom bandwidth control limits, with the safety valve of default rules in the background

Note: The ratio parameter is only applied to the default rules; it has no effect on user defined limits.

The Basic three Parameters

In rare instances, NetEqualizer's default rules may need to be custom tuned for sensitivity. For example, if streaming music feeds break midstream at times when the total usage on the trunk is light, it might be because NetEqualizer is tuned to be too sensitive. Use the Modify Parameter Tab from the management GUI to change the parameters

Three parameters can be adjusted to shift sensitivity:

TRUNK_UP and TRUNK_DOWN

Making either of these parameters larger than your actual trunk size will make the shaping rules less restrictive.

Note: Do Misc/Stop NetEq & Misc/Start NetEq after changing one of these settings.

PENALTY_UNIT (units 100ths of seconds)

PENALTY_UNIT is the unit of time that NetEqualizer will start with when delaying a packet of Internet data. It iteratively increases penalties by this value should a "hog" not respond to the initial penalty. By increasing the size of this parameter the NetEqualizer will scale back hogs more quickly. Also please note, the higher your network speed, the more sensitive it is to PENALTY_UNIT. The default value of 15 will work fine on any networks, but if you see the NetEqualizer slowing streams too severely you may want to reduce this value to 8 or smaller.

Note: Here are some recommended settings based on trunk sizes:

1 to 3 MEG bits per second – Penalty Unit 5 to 15. HOGMIN 12000, HOGMAX 32000

3 to 10 Meg - PENALTY UNIT 4 to 10, HOGMIN 12000, HOGMAX 32000

10 to 45 Meg - PENALTY UNIT 1 to 5, HOGMIN 12000, HOGMAX 32000

Networks approaching 45 megabits may require a PENALTY UNIT resolution smaller than 100ths of seconds. In the default NetEqualizer the smallest Penalty that can be applied to an IP Packet is 1/100 of a second.

If you are finding that a default PENALTY of 1 is putting too much latency on your connections then you can adjust the PENALTY unit to 1/1000 of second with the following command from the command line

```
/bridge/bridge-utils/brctl/brctl rembrain my 99999
```

MAX_PENALTY (units 100ths of seconds)

This is the maximum delay that NetEqualizer will allow. It increments a delay by the value of PENALTY_UNIT every few seconds in the event a connection continues to use excessive bandwidth. A MAX_PENALTY of 200 (2 seconds) usually kills the connection altogether as most servers on the Internet give up communicating when communications lag for more than two seconds.

Sizing internal Tables

BRAIN_SIZE

BRAIN_SIZE determines how many connections NetEqualizer watches at one time. NetEqualizer keeps a mini history of the activity of all users on a trunk. It uses this database to make decisions on who is using too much bandwidth.

Set this to 3000 initially. Reports/Show Active Connections will show the contents of the brain table.

Many network administrators initially plan to set this parameter equivalent to the number of users on the network; however, this approach very often creates a very large table that may eventually cause performance problems. The bigger table, the more work NetEqualizer must do. We have found that NetEqualizer can do most of its shaping with a table size smaller than the total number of users. An undersized table will cause slightly less than optimal "equalizing" but should not be a cause for concern. NetEqualizer is able to function with a smaller table for a couple of reasons:

- NetEqualizer is primarily concerned with persistent data downloads; most connections are transient lasting a second or two at most
- Users often have large idle periods.

BUFFERS

BUFFERS controls the number of connections that can simultaneously be penalized (slowed down)

When NetEqualizer sets a penalty on a connection, it assigns a "delay" buffer to this connection to slow it down. NetEqualizer reserves a finite number of "delay" buffers when it powers up. The standard settings for BUFFERS are 10 percent the size of your brain table.

BUFFERS adds load to the CPU. Therefore, when NetEqualizer has more “connections” to slow than it has room in the buffer table, it will optimize by punishing the worst offenders with delays. Unless this value is vastly undersized, NetEqualizer can usually provide quality shaping on even busy networks.

Quick Fact: Reducing the number of buffers will only take affect after stopping and starting NetEqualizer. Increasing the parameter takes effect immediately.

Other parameters

ANCIENT

How long to keep a penalty in effect

INACTIVE_TICS

This is how long an entry in the BRAIN_TABLE will live before being removed if no activity is detected. Generally we are not interested in connections that are idle For example a value of 200 for this parameter instructs the NetEqualizer to “cancel” tracking a connection after 2 seconds.

MOVING_AVG

MOVING_AVG keeps NetEqualizer from penalizing short bursts of activity. For example, if this variable is set to 8 and the network is hit with a burst of 8000 bytes over a second from an IP address, the moving average for the second would be 8000/8 or 1000 bytes. If the burst persisted for four seconds, the average would be 32000/8 or 4000 bytes. Therefore, the larger this number, the longer a burst can be before it gets penalized. Note that if this parameter is set too high, nothing will ever get penalized. The min for this is 3 and the max is 40

HOGMIN

HOGMIN defines the minimum traffic level for which connections will not be penalized. In other words, a connection using less bandwidth in bytes per second than this number will never get penalized. The default value of 12000 bytes per second (96kbs) will ensure that most VoIP traffic is never accidentally throttled back when NetEqualizer reaches a congestion threshold. (VOIP with can be as high as 11000)

HOGMAX

HOGMAX defines the value in bytes per second that equalizing must kick in if the network is congested. The suggested value is 32000. HOGMAX exists to insure that NetEqualizer will penalize a totally congested system where HOGMIN or other rate caps may be preventing NetEqualizer from applying the equalizing rules.

Priority and Limits, NetEqualizer Application exception rules

NetEqualizer's default equalizing rules are able to handle congestion related traffic flow problems for most organizations. Most types of traffic that organizations want to be prioritized are prioritized by default just using the default equalizing rules. However, some organizations need exception rules for specific traffic priority/ limiting requirements.

NetEqualizer supports two types of exception rules:

Priority rules

Give traffic streams preferential treatment. Priority rules are most often used for voice or video traffic. An example would be Avaya IPoffice voice calls running over a WAN link that shares Internet traffic.

Limiting rules

Limit the amount of bandwidth a specific application type can use. An example would be Citrix printer traffic.

Note: NetEqualizer also provides priority for specific IP addresses.

Priority Techniques

How does NetEqualizer grant priority for IP addresses?

NetEqualizer recognizes two classes of traffic

- **Priority traffic**
- **Data traffic**

When priority traffic is detected, the bandwidth allocation for data traffic is reduced. When NetEqualizer identifies a priority IP address, it typically performs the following process:

1. A priority IP address becomes active
2. NetEqualizer dynamically reduces the data congestion ratio by a few percent (the actual ratio reduction depends upon the type of priority traffic) in order to make room for the priority traffic.
3. Priority traffic is given immunity to flow control—these streams will not be slowed.

Priority traffic is assured bandwidth and your data traffic is dynamically pushed into a smaller bandwidth window.

Note: factory delivered, NetEqualizer defaults are set to perform congestion control on your trunk when it becomes 85 percent full.

Priority IP addresses

Selected by ->Add Rules->Priority Host

Priority Host allows you to select a specific IP address for priority treatment. Once set, this IP address, and any connection it is part of, will receive priority. The VAL field in the set up tab specifies how much bandwidth to allocate for each connection using this IP address.

Traffic Control by Host/Subnet hard limits

The previous section had the same name with suffix "soft limits."

The hard limits of this section differ from soft limits in two distinct ways

1. Hard limits do not allow bursting beyond the set limit, soft limits take a few seconds to kick in and hence allow a burst of data through
2. You can have many more hard limits than soft limits on a system before you exhaust your CPU resource. Hard limits consume 1/20 the cost in CPU power. Most reasonably equipped systems can handle up to several thousand hard limits, while the same system might only be able to handle several hundred soft limits.

Traffic Control by MAC address

MAC addresses are the identifiers of Ethernet cards on user or client machines. Usually the MAC address of an Ethernet card is printed on the card. When NetEqualizer shapes traffic by MAC address, it limits traffic to and from a specific host based on the MAC address located on the Ethernet card of the host. NetEqualizer takes this approach to traffic shaping rather than using an IP address because users may come in over DHCP, and their IP address may vary.

MAC redirection

To use MAC redirection you go under the MAC shaping menu and setup MAC redirection then add your MACs with the Add MAC redirection. Once you have added all the authorized MACs you will go under Firewall and Start/Restart Firewall. Be sure you also add any web or DNS or mail servers that will be going through the unit.

You can give the MACs a description or meaningful name as well. Each MAC and associated name or description must be unique.

You can remove MACs from the macs.allow file with the Remove Redirection menu.

When enabled, MAC redirection looks at the macs.allow file when an outgoing connection is made from your network out to the Internet and if they are using a browser at that time it will redirect the browser to a website of your choosing. This is typically done to inform people how to subscribe to your service or who to contact about your network and its use.

You can also elect to just drop all non allowed MACs instead of redirecting them.

Packets are allowed into your network from outside by default.

The website you redirect port 80 to if they are not authorized MACs is coded to also be on port 80 but you could change this by editing the firewall rules file after you setup redirection.

Traffic Control by VLAN

To set a VLAN shaping rule

From the GUI select ->Add Rule->VLAN Hard Limit

Select a VLAN id from 1 to 2000
Set the incoming bytes per second
Set the outgoing bytes per second

This will create a shaping rule and cause the NetEqualizer to enforce your rate limit such that the aggregate bandwidth usage of all current users will not exceed the values selected for incoming and outgoing bytes per second.

In addition to enforcing the VLAN rate limits the NetEqualizer will perform equalizing across all users on the VLAN. (when default rules are on <see REFERENCE to default rules)

For example if the you set the download limit to 190000 bytes per second (T1) and the VLAN usage level reaches 85 percent. The NetEqualizer will begin to Penalize any connection exceeding the value of HOGMIN (see param section). By tuning HOGMIN to 12000 or higher you can assure Web Pages, most e-mails and VOIP calls will receive priority over larger download amongst users sharing a VLAN.

Masking off traffic (ignore traffic)

The masking features on NetEqualizer were originally implemented to exclude local traffic crossing the NetEqualizer link from being considered for any shaping decisions. Masked traffic is "invisible" to NetEqualizer...

There are two types of masking, "paired" and "absolute." A host or subnet assigned as a "paired" mask will only be ignored if it is talking to another host or subnet that is also assigned as a paired mask. By design, this will cause NetEqualizer to ignore hosts within the same network talking to each other, while at the same time subject the same hosts to NetEqualizer's bandwidth shaping rules if they make a connection with a server on the Internet.

Absolute masks ignore all traffic to or from the masked host or subnet regardless of the connection.

Note MASKs must be /32 or /24 or /16 also be aware that MASKs will not bypass connection limits that are set in place.

Setting Connection Limits

Background: There are more reasons for system administrators to limit connections to a server than we can possibly include in this discussion. The APconnections' design team developed this feature within NetEqualizer to lessen the affects of Peer to Peer traffic and denial of service (DoS) attacks.

Peer to Peer traffic attempts to create hundreds of simultaneous connections to absorb lots of bandwidth. Setting connection limits is a good way to control Peer to Peer traffic on your network.

In a DoS attack, storms of incoming connections are generated by hackers with the intention of overwhelming a server or servers. An attacker will spoof requests, sending storms of erroneously addressed connection requests to your server. These request storms create overwhelming administrative overhead, crippling the server and requiring a reboot by IT staff. While there are techniques that attempt to validate the incoming requests by sending queries back to the sending IP address for verification, these approaches create more traffic on the network. Instead, we chose to address the issue by setting DoS protection connection limits.

NetEqualizer connection limiting feature keeps a total count of active connections (of any type) per host. Additional connections are dropped.

Connection limits can be set per individual IP or Globally. Select **Global Connection Limit** to set a Global connection limit. Simply enter a connection limit value. The connection limits will be set to half of this value for IN traffic and half for OUT traffic. The global connection limit applies to each individual IP that NetEqualizer sees.

Note: Normal users typically have 10 to 15 connections each for IN and OUT traffic. Setting a Global limit of 50 (25 IN and 25 OUT) is a good recommendation and excellent at controlling most Peer to Peer traffic.

Individual connection limits can also be set by selecting **Connection Limits**. You may find this useful to limit an individual user or required to define an exception to the Global Connection Limit. For example, a server in your network that needs more simultaneous connections than your Global Connection Limit setting. You can create a connection limit exception by setting a very large individual connection limit. Note: Individual connection limits must be set BEFORE the overall global connection limit is set (you can also reorder these rules by editing the configuration file, then doing Misc/Stop NetEq and then Misc/Start NetEq).

Note: You can see all active connections by selecting **Active Connections** in reports and Graphing. For connection limits to work, Brain size (in modify parameters) needs to be set large enough to track all connections. The maximum Brain size is currently 10000.

Bytes per day, week or month User Limits and Warnings

For many ISPs, WISPs, and other resellers of bandwidth, there is a market for selling bandwidth as a monthly, daily or hourly allotment. The user limit utility is a core set of flexible features that gives administrators the ability to create these allotments. Features:

- Select users by IP address
- Set bandwidth usage allotted by hour, day or month
- Two usage levels per interval, WARN1 and WARN2, detailed in the example below
- Automatically restrict a user's download speed if they exceed their total byte count for the day, month or year.
- Automatically removes usage restriction at the end of the time period

When the user limit utility starts up (from the Miscellaneous tab), it must access a configuration file. You should be very careful to follow the example below in formatting. Do not leave extra spaces and be sure to include all separator characters as shown.

Toggle reporting on from the Start Stats tab under reports and graphing from the GUI
Edit the userlimit config file, which is accessed from a tab under Miscellaneous on the GUI
Toggle userlimit utility tab from the Miscellaneous tab on the Web GUI

Sample configuration file and description of fields.

```
IP%10.0.0.4&INTERVAL%HOURL&WARN1%2000%art@apconnections.net%You Are A  
toad%250  
IP%10.0.0.1&INTERVAL%DAY&WARN1%3000%stevew@apconnections.net%You Are  
toad2%100&WARN2%60000%jerry@apconnections.net%You Are A toad3%120  
MAC%0:2:3f:37:29:11&INTERVAL%DAY&WARN1%22000%mail@this%Over Day Limit%200
```

IP/MAC is the type of address followed by a "%" and then the MAC or IP address. Note that there are no leading 0s on the MAC (same format entered in the GUI).

INTERVAL can be month, day or hour. Interval refers to the length of time bytes are counted for the user before the count is incremented back to 0.

WARN1 is the first byte count limit. The utility will take action when this user has moved this many total bytes in and out.

- ❖ **WARN1** has the following subfields separated by "%"
Field1: the number of bytes needed to set off this warning (bi-directional total)
Field2: the email of the person to send a memo to
Field3: the text of the e-mail
Field4: and the last number is the kbs to set a limit to for this user

WARN2 just like WARN1 a second level of warning.

Notes: The email address is required in the configuration, but is not an implemented feature. Secondly START STATS must be executed for User Limits to work. See <http://www.netequalizer.com/tsfaq.htm> for details on how to automatically have START STATS enabled when NetEqualizer starts.

Monitoring and Reporting

NetEqualizer supports basic real time reporting and then provides two reporting tools for more information.

To view all active Internet connections

To view all active Internet connections your NetEqualizer is currently seeing from the reports and graphing tab on the GUI select **Active Connections**.

Index	SRCP	DSTP	Wavg	Avg	IP1	IP2	Prot
0	3027	3027	48	178	62.216.31.50	172.16.0.7	UDP 1
1	3027	3027	360	1308	172.16.0.7	62.216.31.50	UDP 2
2	3388	80	470	1968	17.112.152.32	172.16.0.7	TCP 1
3	80	3388	3583	11328	172.16.0.7	17.112.152.32	TCP 2
4	3381	80	7736	27509	213.248.112.115	172.16.0.7	TCP 1
5	80	3385	2248	7997	172.16.0.7	213.248.112.115	TCP 2
6	3027	3027	69	17	212.138.47.14	172.16.0.7	UDP 1
7	3384	110	126	340	62.129.139.40	172.16.0.7	TCP 1
8	110	3384	189	507	172.16.0.7	62.129.139.40	TCP 2
9	3389	80	850	2269	216.109.119.252	172.16.0.7	TCP 1
10	80	3389	31076	82871	172.16.0.7	216.109.119.252	TCP 2
11	3391	80	530	2120	216.52.17.116	172.16.0.7	TCP 1
12	3392	80	226	906	213.248.112.116	172.16.0.7	TCP 1
13	80	3392	142	572	172.16.0.7	213.248.112.116	TCP 2
14	3396	80	1672	6688	63.88.212.82	172.16.0.7	TCP 1
15	80	3391	524	2100	172.16.0.7	216.52.17.116	TCP 2

Definitions for field headers are as follows:

Index	=	The index into the table
SRCP	=	The source port for this connection
DSTP	=	The destination port for this connection (the service being requested http, FTP, etc.)
Wavg	=	A weighted average of total bytes on this connection per second for the last eight seconds
Avg	=	the average in bytes per second since this IP pair came into the table
IP1	=	source IP address
IP2	=	Destination IP address
Prot	=	The protocol ICMP,TCP/IP,UDP

To view the current log file and slowed down connections.

In this log you will see entries for traffic up and down, PENALTY THRESHOLD per pool and entries containing the word PENALTY followed by two IP addresses.

Approximately every twenty seconds the log will contain a date and time stamped entry for traffic UP and traffic DOWN. This is instantaneous bytes per second of traffic in each direction.

[Explanation of PENALTY THRESHOLD goes here]

The PENALTY THRESHOLD shows threshold where penalties will occur (by pool). When the trunk (or pool) is not congested the UP and Down values are simply your defined trunk (pool) size. When congestion is occurring, UP and DOWN are the values used to determine how

much traffic a user (connection) has to pull to be eligible for a PENALTY. The smallest this value can be is HOGMIN, the largest it can be 10 times HOGMIN.

The Penalty entries mean is that NetEqualizer has decided that the communication link between these two IP addresses (connection) is using too much bandwidth, so NetEqualizer has levied a penalty against this connection. The penalty causes all data on this connection to slow down. If this connection continues to use too much bandwidth, NetEqualizer will increase the amount of this delay.

Below an example of the logfiles depending on model.

```
09/10/04 12:39:30 Traffic up: 10352 Traffic down: 480 NUMBER_ACTIVE 24
09/10/04 12:39:30 PENALTY IP : 216.166.39.234 172.16.0.2 LPEAK: 480 WAVG: 3296 BUFF: 499 DELAY: 15
09/10/04 12:39:30 PENALTY IP : 68.222.209.130 172.16.0.2 LPEAK: 480 WAVG: 3341 BUFF: 498 DELAY: 15
09/10/04 12:39:30 PENALTY IP : 67.173.185.63 172.16.0.2 LPEAK: 480 WAVG: 3535 BUFF: 497 DELAY: 15
09/10/04 12:39:35 Traffic up: 9692 Traffic down: 539 NUMBER_ACTIVE 37
09/10/04 12:39:36 INCREASE PENALTY IP: 216.166.39.234 172.16.0.2 BUFF: 499 DELAY: 30
09/10/04 12:39:36 INCREASE PENALTY IP: 68.222.209.130 172.16.0.2 BUFF: 498 DELAY: 30
09/10/04 12:39:36 INCREASE PENALTY IP: 67.173.185.63 172.16.0.2 BUFF: 497 DELAY: 30

PENALTY THRESHOLD pool 0 up 384000 down 384000
02/15/06 06:04:33 PENALTY DECREASE: 192.168.1.150 192.168.1.140 to 15 POOL: 0
02/15/06 06:04:37 PENALTY REMOVE: 192.168.1.150 192.168.1.140 POOL: 0
02/15/06 06:04:38 PENALTY IP : 192.168.1.150 192.168.1.140 POOL: 0 WAVG: 660609 BUFF: 122 DELAY: 15
02/15/06 06:04:40 Traffic up: 11381 Traffic down: 645937 POOL 0
PENALTY THRESHOLD pool 0 up 384000 down 32000
02/15/06 06:04:58 PENALTY REMOVE: 192.168.1.150 192.168.1.140 POOL: 0
02/15/06 06:05:00 Traffic up: 167 Traffic down: 8470 POOL 0
PENALTY THRESHOLD pool 0 up 384000 down 384000
02/15/06 06:05:03 PENALTY IP : 192.168.1.150 192.168.1.140 POOL: 0 WAVG: 1426006 BUFF: 122 DELAY: 15
02/15/06 06:05:09 INCREASE PENALTY IP: 192.168.1.150 192.168.1.140 POOL: 0 BUFF: 122 DELAY: 30
02/15/06 06:05:21 Traffic up: 171 Traffic down: 7606 POOL 0
PENALTY THRESHOLD pool 0 up 384000 down 384000
02/15/06 06:05:24 PENALTY DECREASE: 192.168.1.150 192.168.1.140 to 15 POOL: 0
```

To see a list of active IP users and their associated MAC addresses

From the GUI select **show get MAC IP**

The above command will give you a dump of all active IP addresses and their associated MAC addresses. Note: The report shows MAC addresses without leading zeros if there was one.

Graphical Reporting tools

The 1u rack mount models (NE2000 and NE3000 series) support NTOP, an open source reporting tool with excellent graphics and tables for detailed reports.

Select **View ntop reports** to see the NTOP reports. (note: you must Start ntop first)

To see an overview of ntop please use this link to their overview page:

<http://www.ntop.org/overview.html>

Note: for 1u units shipped prior to December 2005, additional memory must be added before using NTOP with an updated software version.

Bandwidth Pools and the Virtual Equalizer

A pool is a collection of IP addresses that share a bandwidth allocation. Once IP addresses are contained within a pool, the sum total of bandwidth for all the IP addresses will not be allowed to exceed more than the total bandwidth allocated to that pool. For example if four IP addresses are set in a pool; the pool bandwidth is set at 1Mbps, then the total bandwidth for all four IPs is 1Mbps (the total, not per IP). Pools were added to NetEqualizer for the marketplaces where bandwidth is advertised and sold as "you are one of n customers sharing x bandwidth".

Prior to Virtual NetEqualizing, you were required to have one NetEqualizer on every internet trunk where congestion might occur. Now you can group users into logical trunks by IP address and apply the same equalizing technology to each logical group. For the example above - equalizing will occur across the four IPs in the pool that are sharing 1Mbps. When the total bandwidth (for that pool) threshold is reached (set by the RATIO parameter) then the connection(s) associated with those IPs using the most bandwidth will be penalized. The penalties show up in the NetEqualizer log file.

Virtual Equalizing adds the capability to equalize within pools. You can choose to equalize or simply limit bandwidth within pools. When DEFAULT RULES (in Modify Parameters) is ON then all pools will be equalized. When DEFAULT RULES is OFF, then each pool of uses will only be bandwidth limited to the bandwidth specified for each pool.

Virtual equalizing was added for network topologies where bandwidth congestion is occurring at nodes in the network, not necessarily at the WAN/LAN connection. For example, in a wireless network where bandwidth congestion occurs at the wireless hotspots or in the backhaul connections. Individual pools can be defined with the IPs of users at each hotspot and equalizing applied per hotspot. (this assumes that the IP addresses are visible to NetEqualizer). Thus a single NetEqualizer unit can provide equalizing to multiple congestion points.

Adding IP Addresses to a Pool

Before you start adding IP addresses to a pool you must first create the logical identity (pool) to associate them with. This can be done from the bandwidth pools tab on the main gui <create pool>. The other tabs on this menu are self explanatory and are shown on the screen shot above.

Once a pool is in the system you can add and remove members to that pool

The bandwidth restriction on a pool may fluctuate a bit depending on the type of traffic. Heavy use of UDP traffic tends to run over the limit, and heavy TCP/IP (FTP for example) will tend to be held below the limit.

Tips on fine tuning the behavior of HARD LIMITS and pools can found in APPENDIX 4.

Pools can number from 1 to 40, up to 40 different pools per NetEqualizer. Use of more than 2 pools at a time must be licensed. Licensing restriction will be enforced in future releases of the NetEqualizer

Once you have created a pool you can begin to add IP addresses to the pool.

Pool trivia

- Once an IP address is in a pool it may not exist as an individual hard limit. You may add additional soft limits to an IP address. You will get an error if you try to add an IP addresses to a pool that already exists as a hard limit.
- IP addresses within a pool need not be contiguous you can add members to a pool in any order.
- When Equalizing is turned on the members of the pool will be equalized, that is, priority and fairness rules will be applied within the pool. For example:
- If you have 10 members in a T1 sized pool and the bandwidth within the pool is at 100 percent utilization, the NetEqualizer will create temporary policies to the heavy users within the pool to slow them down so they do not squeeze out other users.
- If you choose to create a Priority IP address this IP address will receive priority over other IP addresses within the pool.

What happens if I do not specify any pools?

- There is always a default pool in place on the NetEqualizer (pool 0) All IP addresses are by default part of pool 0. Pool 0 is controlled by the overall trunk size of the pipe the NetEqualizer is sitting on and is what you defined in the initial setup. By definition all legacy NetEqualizer systems have a pool 0 defined and no steps need to be taken
- If I put an IP address in a specific pool is it still part of the default pool 0?

Yes nothing escapes the default pool. If you do not want the default pool to affect the IP address you can achieve this effect by making the default `TRUNK_UP` and `TRUNK_DOWN` excessively large.

- How is the trunk size of default pool 0 defined?
It is defined by the traditional parameters on the Modify Parameters page of the GUI. `TRUNK_UP` and `TRUNK_DOWN` control the size.
- How do pools show up in the log?
Bandwidth usage for defined pools is reported every 20 seconds in the standard log, Misc->showlog.

When equalizing is on and a POOL is over the threshold defined by `RATIO`, then penalties will appear for IP addresses within that pool and will be identified in the log by IP address and also pool number.

Tips and Tricks

Disabling the default rules

From the GUI select-> Modify Params and set the default rules -> off. Turning the rules off will ensure that no default shaping or limiting will take place on NetEqualizer. If you have not set any specific rules, turning the default rules off deactivates all automatic shaping.

NetEqualizer adjusts to traffic over several seconds

Because NetEqualizer adjusts to traffic over several seconds, attempts to set limits on short traffic bursts will have limited affect. NetEqualizer is designed to allow short bursts of traffic through. For most users, allowing these bursts is the desired effect. Short bursts have relatively little affect on overall traffic and should be given priority.

When you do your initial testing on Limits, use file transfers that persist for more than 15 seconds to allow NetEqualizer to come to a steady rate of data transfer.

What to expect for your limit precision

NetEqualizer does a decent job over time (five minute averages) of keeping bandwidth within specification. Please note:

1. NetEqualizer will allow some bursts through. As noted above, NetEqualizer takes a few seconds to adjust to changing traffic situations. If you are testing with one or two large downloads, the bursts will be more pronounced than traffic on a busy network
2. Some tuning may be required to override the background shaping rules (which may be more restrictive than your desired limits)
3. On higher speed networks, the default tuning in NetEqualizer may reduce traffic rates more than an acceptable margin of error (acceptable error margin to us is 10 percent, we do not claim to have billable quality rate limiting). Reduce the size of PENALTY_UNIT to compensate.

Sometimes it's not NetEqualizer

There are some streaming utilities that are all or nothing. As they get penalized, they compensate by sending bigger packets, and then they die and restart. As a result of this effect, you may see jumpy traffic flows when running simple tests with certain applications. Fortunately, the applications that react this way are typically streaming music applications that are not bandwidth intensive. Most of them try to hold steady at 56kbs or so. Streams in this range should not hit the penalty radar like p2p traffic. You should keep this in mind if you are using a streaming music or video download (i.e., real player) when you do your early testing. NetEqualizer will attempt to slow the stream gracefully; however some traffic streams may drop off quickly. This happens when the parent application decides a data rate is too slow to continue. Typically this would be limited to streaming video

Security Precautions

Firewall rules are provided to prohibit unauthorized users from accessing the NetEqualizer IP and thus SSH access and the Web GUI screen. The firewall rule settings are accessible

from the Web GUI screen by selecting firewall->edit firewall rules. Below is a section of this file that appears on the GUI admin screen in a default system before any firewall rules are set.

```
# Uncomment and edit the following lines to allow certain computers to access the GUI
#/sbin/iptables -A INPUT -s 192.168.1.100 -j ACCEPT
#/sbin/iptables -A INPUT -s 192.168.1.101 -j ACCEPT
#/sbin/iptables -A INPUT -s 192.168.1.20 -j ACCEPT
#
# Uncomment the following line to tell the firewall to drop everything else not in the lines
above
#/sbin/iptables -A INPUT -p tcp -j DROP
```

If the network admin always uses IP address 140.32.22.5 when accessing the system, you could limit access to NetEqualizer with the following changes. Notice we have removed the “#” characters to activate these rules

```
# Uncomment and edit the following lines to allow certain computers to access the GUI
/sbin/iptables -A INPUT -s 140.32.22.5 -j ACCEPT
#/sbin/iptables -A INPUT -s 192.168.1.101 -j ACCEPT
#/sbin/iptables -A INPUT -s 192.168.1.20 -j ACCEPT
#
# Uncomment the following line to tell the firewall to drop everything else not in the lines
above
/sbin/iptables -A INPUT -p tcp -j DROP
```

Persistence of Configurations, changing on the fly

NetEqualizer supports persistence of configurations. NetEqualizer will not only allow you to change configurations from the GUI on the fly, it will also save your current settings for you. Simply select -> save NetEqualizer config, when you have your configuration the way you like it.

Redundancy and Reliability

NetEqualizer’s bridge architecture fully supports network redundancy; it has an automatic failover in 30 seconds using STP with another STP capable switch or another NetEqualizer unit. All you need to do is configure a second NetEqualizer or STP capable switch on your network in parallel with your primary machine.

Appendix 1- Default parameter settings and units

RATIO=85	% of TRUNK bandwidth
PENALTY_UNIT=15	100 ^{ths} of seconds
MAX_PENALTY=120	100 ^{ths} of seconds
BUFFERS=123	simultaneous penalties
ANCIENT=20	seconds
BRAIN_SIZE=350	simultaneous connections tracked
INACTIVE_TICS=1000	100 ^{ths} of seconds (1000 = 10 seconds)
MOVING_AVG=8	seconds
HOGMIN=8000	Bytes per second
HOGMAX=32000	Bytes per second

Appendix 2 - Setting/Forcing LAN Speeds

Most NetEqualizer Configurations will auto sense the LAN speed to your router. However it is not uncommon to have the NetEqualizer LAN ports not synch correctly with your router LAN speeds. The most common symptom indicating you have a LAN speed sense problem is unexplained dropped packets.

Below are the advanced instructions for setting LAN speeds. This feature is not available from the GUI

To find out what your 10/100 lan ports are set at run

```
/sbin/mii-tool
```

from the console or from the web GUI Misc/Run a Linux Command page.

To see if your ports are dropping packets, run:

```
/sbin/ifconfig
```

To see what your ports details are run:

```
/usr/sbin/ethtool eth0
```

and

```
/usr/sbin/ethtool eth1
```

For 1000 speeds you must use ethtool and not mii-tool.

```
ethtool -s DEVNAME \  
        [ speed 10|100|1000 ] \  
        [ duplex half|full ]    \  
        [ autoneg on|off ]
```

Here are some examples to force a WAN interface to a certain speed:

```
/usr/sbin/ethtool -s eth0 speed 1000 duplex full autoneg off
```

```
/usr/sbin/ethtool -s eth1 speed 1000 duplex full autoneg off
```

If you need to put some setup commands in an auto startup file you can put them into /art/autostart by editing the file from the console or SSH.

Login as root with the default password of neteq unless you changed it. You can use nano the text editor to edit the /art/autostart file. Just put any lines you need at the very bottom of that file. The command to edit it would be:

```
nano -w /art/autostart
```

Put in lines at the very bottom of /art/autostart such as:

```
/usr/sbin/ethtool -s eth0 speed 1000 duplex full autoneg off
```

```
/usr/sbin/ethtool -s eth1 speed 1000 duplex full autoneg off
```

use the backspace and delete and arrow keys just like Notepad. Save with Ctrl-o and Enter and exit with Ctrl-x. There is a menu at the bottom of nano that shows these commands.

You can also edit the autostart file with the web GUI by replacing the 192....143 with your IP to the web GUI and running this link:

<http://192.168.1.143/cgi-bin/arbi/doEditauto.cgi>

Appendix 3 - Tuning Hard Limit and Pool Sensitivity

The hard limit in shaping rules in the NetEqualizer are factory set to be accurate in most environments; however sometimes it is important to tune them more accurately especially when customers are running speed tests or you just like to tinker.

Hard limits work by keeping track of how many bytes a connection has used every second. When a byte count approaches the limit for that second a time delay is imposed on remaining packets.

The following command line allows you to set how responsive the hard limit and bandwidth pooling utilities will react in different situations. This is done by changing the amount of delay put on a connection once their allocation per second is exceeded.

```
/bridge/bridge-utils/brctl sethardval my <val>
```

Val can be configured three different ways to handle combinations of UDP streams and TCP/IP streams. Some UDP speed tests do not respond to delayed packets while TCP streams over respond.

1) To put the same delay on TCP and UDP streams val can be a number in the range 1-200 this would make all packets exceeding their hard limit quota delayed <val> hundredths of seconds. The default current version has this hard coded to 110 (for a reference)

2) To have the NetEq just drop packets when a user is over their 1 second quota Val can be 999999

This will cause all buffering to cease and packets to be dropped for both TCP and UDP packets when a hard limit is exceeded for a second. The next second the connection starts counting over.

3) Or Val can be a constant between 1 and 200 plus 1000000

The constant will be used to set a buffer time (hundredths of seconds) for TCP packets and drop UDP packets.

Note to make this Command persist through a re-boot it should be entered as a command in /art/autostart

Appendix 4 - Packet Capturing for taps such as Calea

Under the Misc menu you will see Start Packet Capture. That is used on the NetEqualizer side to set it up.

Setting up the receiver for the tap (THIS MUST BE DONE FIRST):

Install netcat (nc) onto a computer. Then on the receiving computer you run the command line of:

```
nc -l -p XXXXX
```

where XXXXX is the port you want to listen on and that you setup on the NetEqualizer to send on.

Netcat can be installed on Ubuntu or Debian with:

```
apt-get update
```

```
apt-get install netcat
```

Netcat can also be installed on Windows by going up and finding the Windows version and installing it.

Netcat can be piped to a file or whatever you want as well using the > and | like any other command.

Appendix 5 - Network Access Control (NAC)

The NetEqualizer Network Access Control (NAC) module is an add-on module which runs concurrently on most standard NetEqualizer Bandwidth shapers. When activated it will force unknown users to login for access to your network.

The Authorization system will not run unless it is factory enabled. If you have an older system and wish to upgrade you will need to contact APConnections to determine if it is possible on your system.

There are two ways to restrict access to your network using the NAC module.

1) Administrators of the NetEqualizer can manually create accounts for users through the administrative interface.

An account is defined by a user id and can be shared by 1 or more users, the amount of simultaneous users allowed to share an account can be defined when the account is created. When multiple users share an account the NAC module keeps track of how many users are logged in. If the number of logins exceeds the account limit, additional users will be denied access.

A session is considered logged in if there is activity within the last 10 minutes, Inactive sessions will be automatically logged out. The inactivity time limit value is configurable.

2) The second option for creating an account is the automated creation. This option is designed for hotspots where users can sign up for access with a credit card.

How user accounts are enforced.

1) Each user account requires a Login ID (and an optional password) the default system does not require a password.

The reason for eliminating the password in the default setup was the desire to streamline and simplify system administration . The NAC is not meant to protect sensitive data in any way, it is simply a gate over Internet access. Since the NAC system limits the amount of simultaneous user sessions, it would NOT be in the interest of a paying customer to give out their Login ID. Using a simple Login ID also insures that Users will pick something simple and are less likely to forget, hence less administrative overhead without complex password recovery support. Again, having the password enabled is optional though if you need it.

2) Accounts are activated for a time period by hours or days. Timing is based on Calendar time (not a meter).

3) Administrators have full access to account records and may extend the time period upon request.

4) The NAC module will allow the administrator to set up a data rate associated with each account thus allowing different classes of service.

5) For flexibility purposes accounts are controlled by IP address. Each time a user logs in the NAC records the users IP address. The administration screen menu contains a report option for showing currently active sessions , this report will show also display the current IP address all active user sessions.

6) The NAC will time out inactive sessions (selectable time out period) for cases where users do not have a persistent IP (they can login with a new IP).

7) MAC authentication for access is not supported via the NAC module.

Check List for installation

You will need to know the following:

- 1) The management IP address for your NetEqualizer. This is the IP for web GUI and shell access.
- 2) An IP address for the redirection HTTP server (must be different from the Management IP address). The Redirection HTTP server comes with NAC module and runs on the NetEqualizer. There is no need for an additional web server. It requires an IP address on your local network. Unauthorized users on your network will be redirected to the redirection server.
- 3) A list of IP addresses of servers , management stations, and any users you want exempted from redirection. You will be allowed to exempt this set from the authorization server.
- 4) PS2 Keyboard and monitor, or SSH access from another computer.
- 5) Authorize.net account for automated credit card processing. Other processing vendors will be supported in the future or you can program your own interface.
- 6) Some basic knowledge of the Linux shell and default editors. Normal NetEqualizer bandwidth control setup does not require Linux knowledge; however setting up the authentication system does.
- 7) You will need to purchase the additional module license from your reseller or the factory, the authentication system is not included in the basic NetEqualizer bandwidth shaper; it is considered a separate product.

Appendix 6 - NetEqualizer User-Quota API (NUQ API) Programmer's Toolkit

The following article serves as the programmer's toolkit for the new NetEqualizer User-Quota API (NUQ API). Other industry terms for this process include bandwidth allotment, usage-based service, as well as others. **The NUQ API toolkit is available with NetEqualizer release 4.5** and a current [software subscription license \(NSS\)](#).

Background

Prior to this release, we provided a GUI-based user limit tool, but it was discontinued with release 4.0. The GUI tool did not have the flexibility for application development and was inadequate for customizations. The NetEqualizer User-Quota API (NUQ API) programmer's toolkit is our replacement for the GUI tool. The motivation for developing the toolkit was to allow ISPs, satellite providers, and other Internet management companies to customize their business processes around user limits. The NUQ API is a quick and easy way to string together a program of actions in unique ways to meet your needs. However, it does require basic programming/Linux skills.

Terms of Use

APconnections, the maker of the NetEqualizer, is an OEM manufacturer of a bandwidth shaper. The toolkit below provides short examples of how to use the NUQ API to get you started developing a system to enforce quota bandwidth limits for your customers. You are free to copy/paste and use our sample programs in the programmer's toolkit to your liking. However, questions and support are not covered in the normal setup of the NetEqualizer product (NSS) and must be negotiated separately. Please call 303.997.1300 x103 or email sales@apconnections.net to set up a support contract for the NUQ API programmer's toolkit.

Once you have upgraded to version 4.5 and have purchased a current NSS, please contact APconnections for installation instructions. Once installed, you can find the tools available in the directory/art/quota.

Step 1: Start the Quota Server

In order to use the NUQ API programmer's toolkit, you must have the main quota server running. To start the quota server from the Linux command line, you can do:

```
# /art/quota/quota &
```

Once the quota main process is running, you can make requests using the command line API.

The following API commands are available:

quota_create

Usage:

```
quota_create 102.20.20.2/24
```

Will cause the NetEqualizer to start tracking data for a block (subnet) of IP addresses in the range 10.20.20.0 through 10.20.20.255

quota_remove

Usage:

```
/art/quota/quota_remove 102.20.20.2/24
```

Will remove a block of IP addresses from the quota system.

Note: You must use the exact same IP address and mask to remove a block as was used to create the block.

quota_set_alarm

Usage:

```
/art/quota/quota_set_alarm 102.20.20.2/17 <down limit> <up limit>
```

Will set an alarm when an IP address reaches a defined limit.

Alarm notifications will be reported in the log /tmp/quotalog see the sample programs below for usage.

Note: All IPs in the subnet range will get flagged when/if they reach the defined limit. The limits are in bytes transferred. Alarm notifications are reported in the quotalog /tmp/quotalog see example below.

quota_remove_alarm

Usage:

```
/art/quota/quota_remove_alarm 102.20.20.2/17
```

Will remove all alarms in effect on the specified subnet.

Note: The subnet specification must match exactly the format used when the alarm was created -- same exact IP address and same exact mask.

quota_reset_ip

Usage:

```
/art/quota/quota_reset_ip 102.20.20.2/17
```

Will reset the usage counters for the specified subnet range

quota_status_ip

Usage:

```
/art/quota/quota_status_ip 102.20.20.2/24
```

Will show the current usage byte count for the specified IPs in the range to the console. The usage counters must be initiated with quota_create command.

Will also put usage statistics to the default log /tmp/quotalog

quota_rules

Will display all current rules in effect

usage

```
/art/quota/quota_rules
```

ADD_CONFIG

Used to set rate limits on IP's which would be the normal action should a user exceed their quota

Usage:

```
/art/ADD_CONFIG HARD <ip> <down> <up><subnet mask>
```

Where

HARD is a constant that specified the type of operation in this case HARD indicates "hard limit"

<ip> is the ip address format x.x.x.x

is the specified max download transfer speed for this ip in BYTES per second, this is not kbs

<up> is the specified up transfer speed in BYTES per second

specifies the subnet mask for the ip address for example 24 would be the same as x.x.x.x/24 notation ; however for this command the mask is specified as a separate parameter.

REMOVE_CONFIG

Usage:

```
/art/REMOVE_CONFIG HARD x.x.x.x
```

Where x.x.x.x is the base ip used in the ADD_CONFIG HARD command no other parameters are necessary on the removal of the rule.

Log:

```
/tmp/quotalog
```

Various status messages will get reported along with ALARMS and usage statistics

Examples and Sample sessions (assumes Linux shell and Perl knowledge)

From the command line of a running NetEqualizer first start the quota server

```
root@neteq:/art/quota# /art/quota/quota &
[1] 29653
#
```

Then I issue a command to start tracking byte counts on the local subnet, for this example I have some background network traffic running across the NetEqualizer.

```
root@neteq:/art/quota# ./quota_create 192.168.1.143/24
Created 192.168.1.143/24
root@neteq:/art/quota#
```

I have now told the quota server to start tracking bytes on the subnet 192.168.1.*

To see the transferred current byte count on an IP you can use the status_ip command

```
root@neteq:/art/quota# ./quota_status_ip 192.168.1.143/24
Begin status for 192.168.1.143/24
status for 192.168.1.255
start time = Fri Apr 2 21:23:13 UTC 2010
current date time = Fri Apr 2 21:55:28 UTC 2010
Total bytes down = 65033
```

```
Total bytes up = 0
status for 192.168.1.119
start time = Fri Apr 2 21:54:50 UTC 2010
current date time = Fri Apr 2 21:55:28 UTC 2010
Total bytes down = 3234
Total bytes up = 4695
End of status for 192.168.1.143/24
root@neteq:/art/quota#
```

Yes, the output is a bit cryptic, but everything is there. For example, the start time and current time since the data collection started on each IP reporting in.

Now let's say we wanted to do something useful when a byte count or quota was exceeded by a user.

First, we would set up an alarm.

```
root@neteq:/art/quota# ./quota_set_alarm 192.168.1.143/24 10000 10000
alarm block created for 192.168.1.143/24
```

We have now told the quota server to notify us when any IP in the range 192.168.1.* exceeds 10000 bytes up or 10000 bytes down.

Note: If an alarm is raised, the next alarm will occur at twice the original byte count. In the example above, we will get alarms at 10,000, 20,000, 30,000 and so forth for all IPs in the range. Obviously, in a commercial operation, you would want your quotas set much higher in the gigabyte range.

Now that we have alarms set, how do we know when they happen and how can we take action?

Just for fun, we wrote a little perl script to take action when an alarm occurs. So, first here's the perl script code and then an example of how to use it.

```
root@neteq:/art# cat test
#!/usr/bin/perl
while ( 1)
{ $line = readline(*STDIN);
print $line;
chomp ($line);
@foo=split(" ", $line);
if ( $foo[0] eq "ALARM")
{
print "send an email to somebody important here \n";
}
}
```

First, save the perl script off to a file. In our example, we save it to a file /art/test

Next, we will monitor the /tmp/quotalog for new alarms as they occur, and when we find one we will print the message "send an email to somebody important here" . To actually send an email you would need to set up an email server and call the command line smtp command with your message , we did not go that far here.

Here is how we use the test script to monitor the quotalog (where ALARM Messages get reported)

```
root@neteq:/art# tail -f /tmp/quotalog | ./test
```

Log Reset

```
ALARM 192.168.1.119 has exceeded up byte count of 160000
send an email to somebody important here
ALARM 192.168.1.119 has exceeded down byte count of 190000
send an email to somebody important here
ALARM 192.168.1.119 has exceeded up byte count of 170000
send an email to somebody important here
ALARM 192.168.1.119 has exceeded down byte count of 200000
send an email to somebody important here
ALARM 192.168.1.119 has exceeded up byte count of 180000
send an email to somebody important here
ALARM 192.168.1.119 has exceeded down byte count of 210000
send an email to somebody important here
ALARM 192.168.1.119 has exceeded up byte count of 190000
send an email to somebody important here
ALARM 192.168.1.119 has exceeded down byte count of 220000
send an email to somebody important here
```

Now, what if we just want to see what rules are in effect? Here is a sequence where we create a couple of rules and show how you can status them. Note the subtle difference between the command quota_rules and status_ip. Status_ip shows ip's that are part of rule and are actively counting bytes. Since a rule does not become active (show up in status) until there are actually bytes transferred.

```
root@neteq:/art/quota# ./quota_create 192.168.13.143/24
Created 192.168.13.143/24
root@neteq:/art/quota# ./quota_rules
Active Quotas -----
192.168.13.143/24
Active Alarms -----
root@neteq:/art/quota# ./quota_set_alarm 192.168.11.143/24 20000 20000
alarm block created for 192.168.11.143/24
root@neteq:/art/quota# ./quota_rules
Active Quotas -----
192.168.13.143/24
Active Alarms -----
```



```
192.168.11.0/24  
root@neteq:/art/quota#
```

That concludes the NetEqualizer User-Quota API (NUQ API) programmer's toolkit for now. We will be adding more examples and features in the near future. Please feel free to e-mail us with feature requests and bug reports on this tool.

Note: You must have a current NSS to receive the toolkit software. It is not enabled with the default system.

