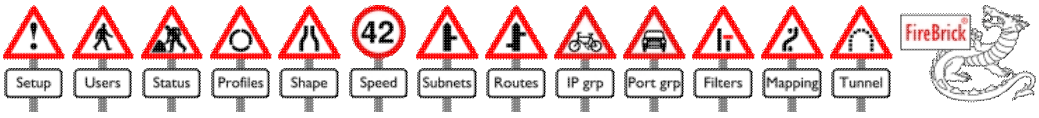


Introduction

The FireBrick 105 is a sophisticated router/firewall product that is designed to be the key device between the internet and your network. It provides state tracking firewalling and routing as well as useful features such as network address translation and automatic IP address allocation. The FireBrick has a number of optional extras making it invaluable at home or in an office. Whilst it is only a small box, it has the power to handle the fastest 8Mb/s ADSL internet links running flat out and handle hundreds of computers in a large office network.

Using this manual

This manual covers the basic operations clearly and simply, and acts as a reference. There are sections for each of the FireBrick configuration icons, and sections describing the underlying functionality of the FireBrick. Each section has at the end a Technical Reference which goes into much more detail about that section with a number of key technical points and notes listed. There is also a section describing each of the optional extra features that are available. Generally, the manual will describe the operation with most features installed, and so your FireBrick may be missing some of the options listed if you do not have all features.

Basic terms

There are some key terms used throughout the manual which it is useful to understand. Please read these first.

LAN	Local Area Network. This is a group of devices connected together, normally using ethernet, which can communicate directly with each other. It can include cables, hubs, switches, and even wireless access points.
LAN, WAN, DMZ	LAN, WAN (Wide Area Network) and DMZ (DeMiliterized Zone) are used to describe the sides of a firewall. They are all LANs but the WAN is used to describe the outside (connected to the rest of the world), The LAN is the inside connected to your network, and any DMZs are used for servers which are typically protected from the WAN but from which your LAN is protected in case such machines are compromised. Normally the single port on the left is the WAN and the 4 ports on the right are the LAN.
IP	Internet Protocol. An IP address is four parts with dots, e.g. 192.168.0.1. The FireBrick supports only conventional IP (version 4).
Mask	(Netmask, Subnet mask) is used to define the size of a local area network. Usually shown in the same format as an IP address, e.g. 255.255.255.192, but also shown as a bit count on the end of an IP address, e.g. 192.168.0.1/24. See Networks for more details.
Port	End point identity used by TCP and UDP protocols, a number 1 to 65535
TCP	Transport Control Protocol - used for most session based communications such as web pages, email, etc.
UDP	User Datagram Protocol - used for realtime and transaction based communications such as DNS and voice over IP.
DNS	Domain Name Service - the way in which machine names are converted to IP addresses, and various related functions.

Getting started

A quick start guide is included with your FireBrick (PDF).

It is very simple to connect your FireBrick to an existing network and make use of it's facilities with no additional configuration. Once connected, it is simple to access the configuration pages and make any

changes you wish.

There are 5 ethernet ports on the front of the FireBrick. The one of the left is normally the WAN side, and the 4 on the right are normally a high speed network switch connected to the LAN side. All ports support 10base-T and 100base-T as well as Full and half Duplex automatically and also have auto crossover to avoid any confusion with straight or crossover cables. The power connector is at the rear and should be used with the supplied power supply or equivalent.

Connecting a FireBrick in to an existing network

1. Check you have internet access from your computers.
2. Locate the router which connects your network to the internet. If you have ADSL, then this will probably also connect to a telephone socket. It should have a cable which connects from it to your network. It may have more (perhaps up to 4). If you find something with more cables, e.g. 8 or more, that is probably a switch or hub and not the router.
3. Place the FireBrick near this router and connect the power. The lights will cycle on the front.
4. Remove the cable(s) from the router which connect the router to your network, and plug them in to the right hand side of the FireBrick. It does not matter which of the 4 ports they connect to. As you connect each cable, the green light above the cable should light after a second or two.
5. Connect a cable (one is supplied) from the single left hand port on the FireBrick to the socket on the router from where the previous cables were removed. If there is more than one socket, any will do. When you do this the light above the port on the FireBrick should light up after a second or two.
6. Check you still have internet access from your computers.
7. Use one of the computers with web browser to access http://my.FireBrick.co.uk/ where you should see a configuration screen.

Connecting a FireBrick to a PC for stand alone configuration

1. Connect the power. The lights will cycle on the front.
2. Connect a cable from one of the 4 ports on the right to your PC. The light over the port on the FireBrick should come on after a second or two.
3. Configure your PC to have IP address 217.169.0.2 with netmask 255.255.255.252
4. Use a web browser on your PC to access http://217.169.0.1/ where you should see a configuration screen.

Factory reset

It is quite possible with any firewall product to mis-configure the unit so that you are unable to access it or make further configuration changes. Whilst this is unlikely, if this happens then the only option is a factory reset. As a security product, there are no back doors to help you if you forget the passwords you have set.

To factory reset:-

1. Disconnect the power and all network leads
2. Connect a network lead from the Left hand single port to the right hand port of the four ports on the right.
3. Connect the power and wait 2 seconds
4. The green POWER light should be blinking
5. Disconnect the network lead
6. The FireBrick will factory reset immediately

There are alternative factory resets which can be used depending on which of the 4 ports on the right are connected to the single port on the left. If the left hand port is used then the factory reset will include DHCP client on the WAN and DHCP server+client on the LAN. If the middle ports are used then they have the same effect as their adjacent end port but the WAN and LAN become reversed such that the single port is the LAN and the four port switch is the WAN.

Basic configuration

Accessing the FireBrick web pages there are a number of basic configurations steps which are recommended. You will find that the web pages have prompts to take you through these steps as follows:-

1. Setting an admin password. The FireBrick has a username/password security system, and you can define a number of users with different levels of access. Initially it is sensible to set a password on the admin user.
2. Logging in as Administrator. Having set a password you should log in using that password. This allows you access to all of the FireBrick features, and you will see many more icons on the administration pages once logged in.
3. Removing default view/edit rights from the nobody user. Without a password you still had some access to the FireBrick and it is sensible to now remove that access so that anyone accessing the FireBrick web configuration pages must login before they can do anything.
4. Once an IP address is set up you may find you have to log in again - this is because the FireBrick will have just set its clock from the internet.
5. The features menu under the Setup icon allows you to check you have all features installed. If you purchased any extra features with your FireBrick then they will be installed at this point.
6. Registration - by registering your FireBrick you can receive any notices by email advising of new software, features, or security alerts. Registration may also provide additional extended warranty.

Tips

Moving entries

Many of the configurations entries have a small green dot next to each entry - clicking on this dot allows the entry to be picked up and moved. Once picked up simply select one of the green arrows next to an entry to move it there. You can change to other pages of the same list first if necessary.

Entering IP ranges

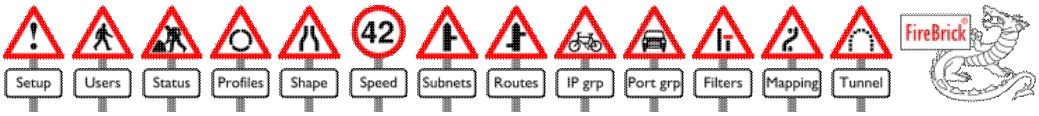
When entering IP ranges you can enter :-

1. Blank for any range
2. A single IP in thje left hand box for a single IP match
3. The lowest IP in the left box and the highest in the right box as a range of IPs
4. Any IP in the left box and a subnet mask in the right box for a range specified using a subnet mask
5. Any IP in the left box and a subnet bit count in the right box for a range specified using a subnet bitcount

In the last two cases the range is filled in when saved.

FireBrick Plus and FireBrick SoHo

If you have used a FireBrick Plus or SoHo model in the past, the FireBrick 105 has a number of new features. See the list of differences.
You can load a FireBrick SoHo or FireBrick Plus configuration in to the FireBrick 105 if you wish.



Setup

The setup function consists of a number of general setup facilities that can be selected from a sub menu.

Save config

This allows the current configuration of the FireBrick to be saved on your local PC. Selecting save config will normally cause your browser to pop up with a save box allowing you selected where to save the config. The default filename is the serial number of your FireBrick, allowing you to save many configs in one directory without risk of overwritting a different one. Once saved, the config can be reloaded in to the same or a different FireBrick. It is recommended that after any major changes you save your config

Clear Alert

If an alert is set (using Flash in any filters) then this stays set and the ALERT light continues to flash until you clear it using this link. The date/time is shown as when the alert was first set (if the clock is set).

Upload/Restore

This allows one of three types of files to be uploaded. Simply select the required file using the Browse button and click Send.

F	A flash file can be obtained from the FireBrick software web site. Uploading this will reprogram your FireBrick with a new version of software and usually then require the loading of a W file. The FireBrick will stop operating for up to a minute while flashing new softwate.
W	A web file contains all of the user interface (web pages) allowing you to manage your FireBrick. Without this you will see a User Interface Required page where you can load any of these 3 file types. Normally, for English web pages the file ends in WEN. You must load the version expected, or load a new flash file or config.
Config	A saved configuration file can also be loaded. This will completely replace the previous configuration with the new configuration.

LEDs

The LEDs (lights) over each port can be controlled in a number of different ways depending on your preference. There are 6 pre-defined combinations, a cycling lights option and the option to choose the yellow and greed LED functions directly. When cycling lights are selected the 4 ports on the right cycle the LEDs left/right/left all the time.

Ports

The Ports menu allows settings for all 5 ports to be controlled. With the 5PORT option the port configuration can be selected. Without the 5 port option, the WAN/LAN reverse can be selected. For normal use the settings should all be left on Auto.

Name	Interfaces are normally called WAN or LAN, but you can set the name yourself.
Crossover	Normally the FireBrick can be connected using a straight or crossover lead to a hub/switch or a computer directly. This allows specific select of the crossover mode (MDIX is a normal switch/hub connection and MDI is a normal PC connection).
Speed	Normally the FireBrick detects 10base-T or 100base-T automatically, but the port can be fixed to only one speed.
Duplex	Normally the FireBrick detects Full or Half duplex mode, but the port can be fixed to only one mode.
Disable	Causes the port to be disabled, allowing no traffic in or out.
Throttle	Causes the speed of traffic in and out to be cut to 128Kb/s. This is not traffic shaping but a crude packet limit which can be useful for network debugging.
B/Limit	Causes the speed of any broadcast traffic (or mulicast or flooded unicast) traffic to be limited to 128Kb/s. This can help track down and limit broadcast storms or loops and is mainly useful for network debugging.
Long	Ethernet cables are meant to only run 100m max. This option allows 10base-T sensitivity to be increased to allow use over longer cables (at your own risk).
Test	This causes a line test of the port (see below)
Reverse	This allows the WAN and LAN side to be reveresed. The change takes affect when you reset the FireBrick

Line test

The line test will take the port out of action for a few seconds and perform a time domain reflectometry measurement on the cable. The results are indicated on the right of the table when the tests are complete and remain visible until next reset/power cycle. This type of test can be effective on cables over 3m in length but the results should always be considered only an approximate indication.

If a cable is connected to a correct hub or switch or computer at the far end then the test simply indicates connected. If the cable is broken or shorted then this is indicated along with the distance.

Name/etc

This allows the identity of the FireBrick to be set.

Name	This names the FireBrick. Use a short name, usually related to the site name. To avoid problems with email, etc, use domain valid characters (a-z, A-Z, 0-9, and hyphen).
Domain	This provides a domain name. Again, use domain valid syntax. This is used for DHCP and with the name for emailed messages. Put your valid internet domain.
Administrator	Put the name of the administrator. This is for your own reference, but also reported if SNMP is enabled.
Location	Put the location. This is for your own reference, but also reported if SNMP is enabled.
SNMP Community	If this is not blank, then SNMP is enabled. Put the community name required, usually just public. Remember that you can use filters to restrict access to SNMP or any services on the FireBrick if required.
SNMP options	The ifDesc option causes the SNMP ifDesc to be a simple unique number (the SNMP interface index in the OID) rather than a description. This is because some tools expect it to be unique (e.g. cfgmaker for mrtg)

Gateway

This defines the general gateway IP address and interface. It is used if there are no matching routes or subnets.
The recommendation is to make this a subnet and not set a gateway address as such - the subnet can then have the gateway defined, which could be by DHCP.

Bonding

For full details see the bonding section. This allows up to two pseudo gateways to be specified, and up to four real gateways to be used in their place on a cyclic basis.

Stealth

This is not how you give the FireBrick and IP address. You can speficy the LAN stealth address on which the FireBrick will answer even for traffic passing through it. The FireBrick effectively hijacks traffic to this address. You can also set an address for the FireBrick to borrow on the WAN when setting it's clock, etc. This is normally the adderss of a machine on the LAN, and the FireBrick hijacks the replies to it's requests which would otherwise go back to that machine.

Disable ARP	Stops ARPs being sent automatically from one interface to another - this stops most stealth operations being possible in normal operation
Disable subnet broadcasts	Stops subnet broadcasts (i.e. last address in subnet) being treated as stealth
Disable local broadcasts	Stops local broadcasts (i.e. 255.255.255.255) being treated as stealth
Disable all stealth	Disables all stealth operation

Time

The FireBrick sets and maintains its clock from the internet. To set the time the FireBrick will need a gateway and IP (or stealth WAN IP) so as to be able to send time requests to the internet. The default settings are correct for UK and UK summer time.

Server	Specify the IP of the time server to try, normally 217.169.0.1
Backup	Specify a second time server to use if the first does not respond, normally 217.169.0.2
Time offset	Select the base time zone, e.g. for UK it is UTC+0
Summer time	Select if it is summer time, although this is normally set automatically.
Start summer time	Select the date and month, the Sunday on or after which the clocks go forward one hour. You can select manual to stop summer time being adjusted automatically. The time changes at 1am winter time.
End summer time	Select the date and month, the Sunday on or after which the clocks go backwards one hour.
Profile	The time is set every hour normally, although exactly when in the hour moves about deliberately. This profile allows this to be restricted to set the clock less often. On power up / restart, the clock is not set and so it continually tries until the clock is set, ignoring the profile selected.

Syslog

The FireBrick has an internal log, and can also log to a syslog server. This allows the IP and syslog type to be set.

Server IP	Specify the IP of the syslog server
Port	Specify the syslog port (normally 514)
Type	Select the syslog type, local0 to local7
Optional Interface	Specify the interface or interface and subnet on which the syslog is to be sent, otherwise normal routing rules apply

Optional Source IP	Specify the IP from which syslogs are sent - can be any IP as there is no reply to a syslog. Normally set automatically. Using a subnet for the interface sets the IP of that subnet
Optional Gateway IP	Specify the gateway IP to use. Normally set automatically. Setting a subnet for the interface sets the IP using the DHCP defined gateway for that subnet.

DNS

The FireBrick acts as a DNS relay, and uses DNS itself. This address defines the DNS server it uses.

Log/Filter Options

This allows defaults and options to be defined relating to logging and filtering. See filters for a description of Blink, Flash, Log, Syslog, and Email.

Default filter	This defines the default filter action if no other filters are matched.
Event	Certain events in the FirebBrick are logged as an "Event". This controls if/how such things are logged. Generally an event is something that happens that is non critical.
Alert	Alerts are normally more important events that are critical.
Debug	Debug messages are general additional detailed information.
Stats	Stats are generated automatically every 5 minutes showing usage of each filter and speed lane and interface.
Login OK	If a user login is successful it is logged using these options
Login Bad	If a user login fails, it is logged using these options
DHCP OK	If a DHCP address is allocated (rather than renewed, which is a debug message), then these options are used.
DHCP Bad	If a DHCP operation fails (e.g. no addresses left) then it is logged using these options.
Ping scan	If a ping based profile goes on or off line it is logged using these options.
Tunnel state	Log of tunnel state change (up/down), but does not exclude state changes for tunnels in "Timeout keep alive" mode as they would happen all the time.
Large sessions	Sessions where more than a specified amount of data is transferred are logged at the end of the session using these options.
Email server	This defines the IP of the email server to use to send emailed log entries
Test server	This sends a test email
From address	This defines the address from which the email is sent.
To address	This defines the address to which the email is sent.
Holdoff	Emails are not sent on the first emailable log event happening, there is an initial holdoff (in seconds) so that related events will appear in the same email. Once sent, there is then an additional holdoff which is mainly to limit the number of emails that can be sent when there is a recurring emaillable event.
Profile	Emails are only sent during a selected profile.
QOS TOS value	This allows the specific TOS (type of service) value that is considered to be priority traffic in bonded tunnels and speed lanes. This defaults to 160 which is typical for SIP phones. If using VoIP (Voice over IP) then ensure that you set all phones and links to use the same TOS and set the appropriate value here.

UI Options

Some general UI options can be set which affect the overall operation of the UI.

IP display/range	Various options allowing you to change the way IP addresses and in particular ranges of addresses are displayed.
Number grouping	This allows numbers to be shown with no grouping,or commas/dots or spaces every three digits from the right.
Decimal point	This allows numbers with a decimal point to use a dot or a comma
Speed	Select if you prefer to see speeds as KBytes/s (one decimal place) or Kbits/s
Date format	The date format can be an ISO format (YYYY-MM-DD), UK (DD-MM-YYYY), US (MM-DD-YYYY) or full, e.g. nth Month YYYY
Protocol input	The protocol selection in various places is normally TCP, UDP or ICMP only. This allows a full selection of all 254 protocols, or an input box to enter a protocol number.
Warning music	There is normally a tune played on a suitably configured PC which is trying to login to a FireBrick without the correct username or password. This can be disabled.

Security

See security for a more detailed description of the security model. This allows the general security settings for control of all of the main icons to be specified.

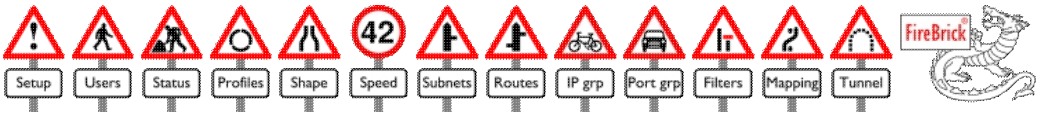
Features

See features for a more detailed description. This allows the current and availabel features to be listed, and the FireBrick to be updated with new features.

On a new FireBrick you should configure internet access and DNS and time setting, and then select Install Assigned Features to ensure you have the full set of features provided with yoru FireBrick installed.

Technical Reference

- The name and domain are used in the HELO of outgoing emails, and so should be set using domain valaid characters to avoid problems with some mail servers. Similarly the from and to email addresses need to be entered carefully.
- Emailed logs include the first message logged on the end of the subject line. This can be useful if emailing an SMS gateway.
- When cycling LEDs mode is used, the extra dot links control the phase of the lights. FireBricks connected using the WAN port and all set in the mode will synchronise themselves to product a dramatic effect.
- When cycling LEDs mode is used, the left hand port used LED mode 5.
- As soon as internet access from the FireBrick is possible (i.e. IP, gateway, etc) the clock may set and this will usually cause the logged on user to be logged off as the time jumps forward.
- If a port is set all manually (no auto) then auto negotiaton is disabled. This allows operation with some types of router which do not understand auto negotiation or have it disabled. You should ensure any manual settings agree with the settings at the other end to avoid problems.
- LED modes for SPEED and DUPLEX are lit for 100Mb/s and Full duplex.



Users

The FireBrick uses a username/password system to manage security. There can be a number of different users of the system, and each can have their own access permissions. One user is special, the nobody user, which defines the permissions before you are actually logged in as anyone else.

Login

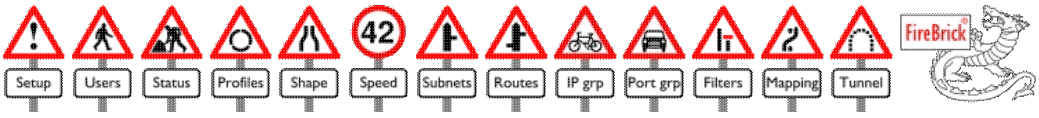
If you are not logged in as a specific user, you can select Login on the top left of the screen, and enter a user name and password. If you are logged in then you have a link to Logout instead. To change to another user, log out and then log in again.

User settings

Name	Allows you to give the user a full name for your reference
Security	Sets the security level of this user and so defines who can view or edit the users details
Profile	Defines the profile when the user can log in or use the FireBrick.
Login	The login name, i.e. what is typed in to the login box
Allow from	You can select one or more interfaces from which this user is allowed to log in.
Page colour	You can select the background colour for the FireBrick configuration pages when that user is logged in. This can be useful if you manage several different FireBricks as you can give each a different colour.
Timeout	This defines the timeout, in minutes, after which the user is automatically logged out if they have no accessed a page for that long.
Lines	This defines how many lines are shown on each page of entries in the administration pages.
View rights	There are 8 security levels, 1 to 8, and the check boxes define which security levels this user can view. This allows the user to see all details of any items at any level that is ticked, but not necessarily make changes.
Edit rights	This defines which security levels the user has permission to change. Any items with a level that is ticked can be changed by the this user.

Technical Reference

- It is important to also ensure that the nobody user also has permisions from at least the same interfaces as any specific user otherwise that user could not get to the login screen in order to login (as they are the nobody user until they have actually logged in). E.g. to allow WAN address to a user, also allow WAN access for the nobody user.
- It is not sensible to tick an edit right without a corresponding view right.



Status/information

The FireBrick contains a number of useful information and diagnostic tools as follows:-

Status

WAN/LAN	This shows the state of the port, green if it is connected, and the speed and duplex mode
Serial Number	This is also shown on the top left.
Base MAC	This is the base MAC address used by the FireBrick. The FireBrick uses different MACs depending on the subnet.
Time now	The current time, if set. Note that the setup/UI options allow the format of the time to be changed
Clock last set	The time when the clock was last set. The clock is normally set every hour.
Running since	The time when the FireBrick was last reset. Shown if the clock is now set.

DHCP

This lists all of the DHCP addresses the FireBrick has allocated, the time they are due for renewal (2 hours after they were issued), the MAC address and the machine name. Where there was no machine name you can set one manually by entering a name in the input box and pressing return.

The FireBrick tries to keep the same address for each device, and will only re-use an address if it is available and is the oldest (i.e. not used for the longest). You can manually clear an allocation by clicking on the interface.

ARP

This lists the FireBrick current ARP table, listing the interface, IP address and MAC address of any currently active IPs communicating with the FireBrick. If the MAC is all 0's then the device is not (or has not yet) responding and may be turned off or disconnected.

MAC

The MAC report shows the MAC addresses seen currently on each interface. Against the MAC the interface name is listed. This allows teh specific prot to be identified. The MAC list includes addresses sent by the FireBrick as well, and these are marked with the FireBrick's name and not a port name,

Sessions

The session list shows all currently active protocols, and also links for ALl, 1MB, and 10MB. The All link shows all current sessions, whilst 1MB and 10MB show sessions that have currently transferred more than 1MB or 10MB respectively. Selecting a protocol shows all sessions using that protocol.

The session table starts with an S for Stealth and R for Routed and then the protocol name. Clicking on the protocol kills the session.

Next, it shows the interface, IP(s), Port(s) and amount of data send from one end, and the same for the other end of the session. Normally there is only one IP and Port on each, but where NAT applies or address mapping is used there may be two IPs and/or ports listed.
Finally, it shows what filter and speed lane is applied to that session.

Log

The log contains everything marked log on the general log/filter setup or specific filters. If the clock is set then the log shows the time of each entry. There is space for over 1000 log entries, and the oldest entry is lost as new ones are added.

Selecting log shows the log on a page and keeps that page open watching the log in real time.

Selecting recent shows the log in the same way, but starting from only 5 minutes ago.

Selecting save shows the log but does not follow changes live and clears the log once displayed. This is useful for saving the log or clearing it.

Counters

The counters section shows counters for various statistics recorded against each port. These are mainly used for debugging network problems.

The core counters relate to traffic to/from the internal FireBrick routing/filtering core.

The change in the last second is also shown, which can be used to see instantaneous throughput on ports, etc.

Technical Reference

- Note that the status shows the names of the ports which could be WAN and LAN1-4, or LAN and WAN1-4 if the LAN/WAN reverse option applies, or could be any names you have given if you have the 5PORT option.
- The status of a port can show negotiated or not. If negotiated then this means that auto-negotiation was used. There are also notes for errors such as polarity reversed (where the wires within a pair are swapped). Some hubs/switches are wired incorrectly and show this, but this normally means the cable is incorrect. Polarity reversal is automatically compensated for. Other faults include jabber which means a faulty bit of equipment or cable, and remote fault which is a special flag in the auto negotiation to tell uss that the other end is seeing a fault in what we are sending.
- The MACs used are the Base MAC plus the subnet number, or plus 31 if no subnet can be found. This allows different subnets to act as DHCP clients on the same LAN if necessary, each getting a different address.
- When using stealth the MAC report can be a bit confusing. e.g. Traffic from the LAN to WAN shows as LANn on the LAN side, and as FireBrick on the WAN side because the FireBrick will have received it from LANn on the LAN and sent with the MAC unchanged out on the WAN.
- Counters are 32 bit wrapping numbers.

Profiles

FireBrick profiles are a very powerful feature. Most settings on the FireBrick have a profile, which is by default 24/7 (always active). There are standard profiles for 24/7, 9-5M-F and 2amSun. Additional profiles can be based on time, or pinging an address, or manually switch on the main dragon page along with quick filters.

Name	Allows you to name the profile. It will then appear against most other items with that name and with Not that name.
Security	Defines the security level controlling which users can view or edit this profile.
Profile	Allows this profile to depend on another profile, either AND or OR another [not] profile. If unsure, leave as AND 24/7
Mode	The profile can be time based, or manually controlled, ping based or checking tunnel state.
Alert LED	This allows the RED alert LED on the front of the FireBrick to be affected by the profile automatically.
Re-route	If this is set and a profile changes, traffic is re-evaluated for possible re-routing if routing rules are based on this profile.
Ping address	If a ping profile, this is the address to ping
Ping TTL	This allows the time to live to be set on the pings.
Optional Interface	Allows you to specify an interface or interface and subnet via which pings are to be sent, otherwise normal routing rules apply. This is also used for tunnel state mode to specify the tunnel.
Optional Source IP	Allows you to specify the source IP for the ping. This is normally set automatically, and setting a subnet/interface (above) will set for that subnet specifically
Optional Gateway IP	Allows you to specify the gateway IP for the ping. This is normally set automatically, and setting a subnet/interface (above) will the DHCP gateway for that subnet
24/7	Ticking this causes all hours of all days to be set on
As above	This allows one day to be set the same as the previous
9 to 5	This forces a day to be set for 9am to 5pm
Clr/24	The clr option causes the whole day to be cleared, and the 24 option causes the whole day to be set on
Hours	Each hour can be set on or off specifically. This affects timed profiles, and also when pinging is done.

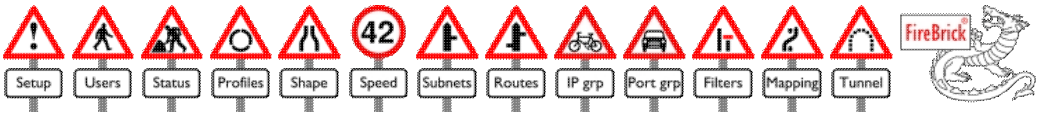
The profile types are:-

Timed	The profile is based on the time of day and day of week - you can select on an hour by hour basis. If the clock is not set then the state (active/inactive) of the profile does not change.
Enabled	The profile is permanently enabled, the time settings are not relevant. During the selected times, the profile appears as a check box on the login screen allowing to to be changed to disabled.
Disabled	The profile is permanently disabled, the time settings are not relevant. During the selected times, the profile appears as a check box on the login screen allowing it to be changed to enabled.

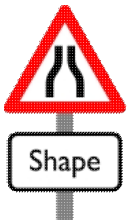
Ping	The profile is based on pinging an IP address which can be via a specific interface and gateway and also from as specific source addree. Pings are done during the enabled times.
Tunnel state	The profile reflects the state of the tunnel specified in the optional intergace. A specific tunnel must be selected.

Technical Reference

- Ping profiles cause a ping to be sent every second to the specified destination. If there is no response for 5 seconds, then the profile is down. Pings continue every second, and if there is one response then the profile is up.
- Avoid making profile inter dependance loops - it will not crash or hang the FireBrick but will have unpretictable results.



Shaping rules

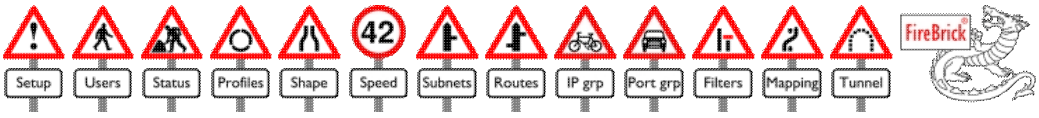


The FireBrick can be used to change the rate of traffic using speed lanes. The shaping rules define in to which speed lane each type of traffic is assigned. They operate much like filters. The first in-profile rule which matches the traffic in question is applied. If no rules match then the default master lane is used.

Name	Allows you to give a name to this rule
Security	Sets the security level of this rule and so defines who can view or edit the users details
Profile	Defines the profile when this rule applies.
Source	This allows you to specify one or more source interfaces from which the traffic may come
Target	This allows you to specify one of more target interfaces to which the trafficmay be going
Lane	This specifies the lane to be applied to the traffic
Both ways	This makes the rule work both ways saving making two rules.
Source ports	This allows a range of source ports to be specified. Applicable to TCP and UDP. Normally blank meaning any.
Target ports	This allows a range of target ports to be specifiied. Applicable to TCP and UDP. Typically just one port for the specific protocol, e.g. 80 for WWW
Protocol	This allows the specific protocol to be specified, or Any.
Port group	Instead of using a source port range, target port range and protocol, then a named port group can be selected.
Source IP range	Allows the range of source IPs to be specified, or blank for any.
Source IP group	Instead of an IP range, a named IP group can be selected.
Target IP range	Allows the range of target IPs to be specified, or blank for any.
Target IP group	Instead of an IP range, a named IP group can be selected.

Technical Reference

- Shaping rules are constantly rechecked in case profiles have moved traffic to a different speed lane. This can mean a few seconds delay in re-assigning traffic.
- The target IP and ports are those before any NAT or mapping, altough the rule is actually applied at the end of the process of setting up a new session
- Both ways operates on traffic from source IP, source Port and source interface to target IP, target port and target interface as well as from target IP, source Port and target interface to source IP, target Port, and source interface. I.e. the ports are not swapped as traffic is normally classified by target port regardless of direction of data.
- Selecting Any protocol and no ports set means any protocol. Selecting Any with ports set means TCP or UDP only.



Speed lanes

The FireBrick can manage the rate at which traffic flows using speed lanes. Traffic is placed in a speed lane using the shaping rules. All traffic in any speed lane is controlled by that lane - the speed is for the total traffic in the lane.

Speed lanes allow control of the rate at which traffic flows from and to each possible interface. You can select the interface you wish to view/control, the default being the interface current set for the default gateway (normally WAN).

Settings

Name	Specify a name for the speed lane, e.g. "slow"
Security	Controls which users can view or edit this speed lane
Master	Select a master speed lane which also applies to this traffic
Rate Min	The minimum speed setting in Kbits/s
Rate Max	The maximum allowed speed setting in Kbit/s or blank for no limit
Fast	This means that this speed lane is not also limited by the master speed lane, but the master lane is affected by this traffic.
ACK	Setting this means that the responses to TCP messages are priority and not delayed. They still affect the speed lane and so delay other traffic.
QOS	This means that traffic with a fast type of service are priority and not delayed. They still affect the speed lane and so delay other traffic.
ms	This allows a number of milliseconds latency to be added to all traffic to that interface via this lane.

The first speed lane is the default which is applied if there is no rule to allocate the traffic. It is also the default master speed lane of other lanes.

Traffic is assigned to a speed lane using the shaping rules, and the speed limits applied for that lane, but then the master speed lane is also applied. The master speed lane is usually used to allow the speed of an external internet feed (e.g. ADSL router) to be set, and allow some traffic to go in a lane marked fast to jump the queue.

This means that traffic may have up to 4 consecutive speed restrictions applied - from a lane and a master and to a lane and a master.

Statistics

If the clock is set, then various statistics are shown on the speed lanes...

Rate Set	The currently selected rate. Where take is selected, the current rate being used is shown in brackets
Rate Now	The instantaneous rate indication for the last whole second in KB/s or Kb/s.

Rate 5min	The average rate over the last whole 5 minute period in KB/s or Kb/s
Day This	The total transferred so far today, in MB
Day Last	The total transferred in the last whole day, in MB
Month This	The total transferred so far this month, in MB
Month Last	The total transferred in the last whole month, in MB

Note that the rate set and now can be displayed in Kbits/s or Kbytes/s depending on UI settings.

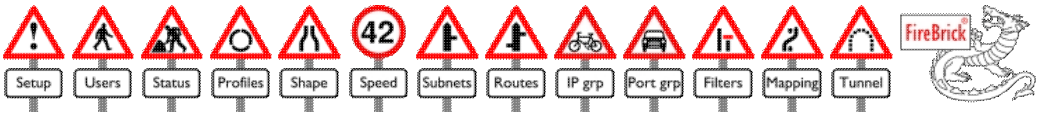
Technical Reference

- The Fast QOS flag works on packets with IP TOS (type of service) set to the value defined in the setup (default 160, typically for VoIP)
- The Fast ACK flag applies to any TCP packets with no payload. It applies if the specific speed lane has Fast ACK set.
- Traffic shaping operates by scheduling packets based on the time they would be sent if the link was the specified speed. This allows smooth operation with TCP and other protocols. Packets that queue jump (e.g. ACK or QOS) are counted and so slow later traffic hence maintaining the overall speed limits.
- Where a master speed lane has a minimum speed set, and there is spare unused capacity below that setting, then that spare capacity is added to the current usage of all subordinate lanes (subject to specified min and max speeds on each lane). This allows the spare capacity to be shared out. As the spare capacity is used, the subordinate lanes will move down to their minimum setting. By setting the max to the same as the min on any lane you can stop spare capacity being taken like this.
- Last day and Last month may be distorted if the FireBrick has been running for some time with the clock not set as these figures are updated on change of day or month.
- It is important to realise that the speed lane applies to the total traffic in that lane. E.g. if you have a lane of say 64Kb/s, and put web traffic (TCP port 80) down it, then this does not mean that each web page will be 64Kb/s but that the total of all sessions on that lane is limited to 64Kb/s - the more simultaneous sessions the less each gets. As such, you can have multiple lanes at the same speed and assign different traffic to different lanes.
- This allows simulation of long latency lines such as satellite links. Note that there is limited buffering capacity, so high bandwidth high latency links can result in packet loss. Latency settings are limited to a maximum of 5000ms and so not suitable for RFC1149 simulation.
- Some internally generated packets are never subject to speed lanes, such as DHCP request/replies, pings for profiles, and tunnel keep alives.

Master speed lanes

Traffic is assigned to a single speed lane using the shaping rules. Each speed lane may then also list a master speed lane which is also applied. The idea behind this is that you may want to control various types of traffic which all then go via a simple feed (e.g. an ADSL line). The master speed lane allows you to then control the rate at which the traffic is actually sent overall. By limiting this you can avoid the external routers buffers filling. This means that traffic marked fast is able to jump the queue on the FireBrick and then not hit another queue in the external router. This is particularly useful for voice over IP over ADSL, for example.

- Statistics record details of the traffic through the lane in each direction depending on the interface. They are also counted on the corresponding master speed lane.
- The from interface statistics includes traffic that has been discarded because of the speed restriction. The to interface statistics only shows traffic which is being sent.
- The statistics ar recorded as entries are recieved or placed in the queue, and so may show higher than the rate set even though the rate limiting is working perfectly.
- Traffic that is QOS or ACK is unaffected (i.e. not delayed) by its speed lane or the master, but it does contribute to the usage of both and so slows other traffic.
- Traffic that is Fast is affected by its own speed lane but not by the master. The master speed lane is however affected by the usage and so slows other traffic. I.e. traffic marked fast effectively stealths bandwidth from the master speed lane.
- Latency added is added for the specific speed lane and the master speed lane for the from and to interface making a total of 20s latency possible.



Subnets

The FireBrick can operate like any conventional network device with an IP address and netmask. However, the FireBrick can have multiple addresses and be on multiple networks at the same time even on the same physical network. The subnets allow the network address to be defined as well as DHCP and other settings.

Name	This allows the subnet to be given a name, but default it uses the name of the interface. The choice of name is important when used with the DHCP restrict feature
Security	This sets the security level and controls who can view or edit this subnet
Profile	The subnet can be subject to a profile, allowing the subnet to be visible part time
Interface	This defines on what interface the subnet operates
IP address	This specifies the IP address of the FireBrick on this subnet. As such it cannot be the network or broadcast address for the subnet
Subnet Mask	This defines the subnet mask applicable.
DHCP Client	If selected then this subnet is a DHCP client, and most settings will be overridden when the FireBrick obtains an address by DHCP. To make a subnet DHCP you do not need to fill in the IP or netmask or any other details.
Stealth	Set this if there is a subnet on the other side of the FireBrick with the same IP range and traffic is to pass through by stealth.
NAT	Set this if this a subnet using a private address range and address translation is to be used
VLAN ID	For advanced use
Allocation IP range	To make the subnet act as a DHCP server, an address or range of addresses can be specified. This sets the range of addresses that can be allocated.
DNS servers	As a DHCP server, you can specify the DHCP server to issue. Leave blank for the FireBrick to act as a DNS relay. As a DHCP client, this shows the DHCP servers the FireBrick received.
Gateway	This is the gateway applicable for any traffic routed to this subnet. As a DHCP client, this is filled in automatically. As a DHCP server, this is given out as the gateway, or if blank then the FireBricks IP is given out instead.
BOOTP server IP	For advanced use
BOOTP filename	For advanced use
Exclude gateway	Setting this means the FireBrick does not issue a gateway address as a DHCP server, and does not accept one as a client.
Exclude Time server	Setting this means the FireBrick does not issue a time server address as a DHCP server, and does not accept one as a client.
Exclude Syslog server	Setting this means the FireBrick does not issue a syslog server address as a DHCP server, and does not accept one as a client.
Exclude DNS server	Setting this means the FireBrick does not issue a DNS server address as a DHCP server, and does not accept one as a client.

Exclude Domain	Setting this means the FireBrick does not issue a Domain name as a DHCP server, and does not accept one as a client.
Backup DHCP	Setting this means the FireBrick will not answer the first time for any DHCP client, allowing another server to answer normally and making the FireBrick a fallback server.
Don't check	For advanced use
DHCP restrict	For advanced use
DHCP Mirror	For advanced use

Technical Reference

- The FireBrick uses a difference MAC address on each subnet, and so can be a DHCP client multiple times on the same interface.
- Subnets that are on an inactive profile do not answer ARPs for their IP, but previous ARPs may remain cached by other machines for a short period allowing traffic to be routed via the FireBrick after a profile becomes inactive.
- When a subnet is made active the FireBrick sends an ARP announcement.
- The FireBrick treats the network address on a subnet as a valid IP and not as a broadcast IP
- The subnet mask can be entered as a dotted quad (e.g. 255.255.255.0) or as a bit count (e.g. 24). It is normally displayed as a bit count.
- As a DHCP client the FireBrick sends its name and the subnet name concatenated with a space between as the subnet name.
- If stealth is set then the FireBrick will answer traffic for its IP, but other traffic on this subnet can be passed through the FireBrick (subnet to filtering) in stealth mode. If not set, then traffic for any of the IPs on this subnet are considered to be on that subnet and not passed through as stealth. In particular, this affects stealth transmission of ARP requests.
- The NAT setting causes any traffic from the subnet to be NATed if there was not an explicit route used to direct the traffic. If an explicit route is used then the NAT setting for that route applies. Note that this means traffic between two private subnets using subnet based routes will NAT if the subnets are set to NAT even though this may not be necessary if both networks use the FireBrick as a gateway.
- The DHCP range can be one address, a range, or an IP and mask or IP and bit count.
- DHCP allocations check that the address to be allocated is no in use by another machine (using an ARP) and abort if it is. As such addresses can be marked as in use in the status/DHCP report and not actually allocated. This avoids duplicate IPs.
- DHCP allocates are for 2 hours with 1 hour renewal, but allocations are persistent - using the same address each time unless all addresses were exhausted.
- As a DHCP client the FireBrick normally checks the address it was offered is not already in use (using ARP) and rejects it if another machine is using the address. This can be disabled with the Don't check option, and is relevant if something is proxy ARPing an entire block, for example. (e.g. cable modem services in some parts of Colombia).
- The BOOTP server and filername are sent in DHCP and BOOTP responses and allow network boot devices to obtain the information necessary to load.
- The FireBrick supports the use of a /31 (255.255.255.254) subnet mask to create a point to point link as per RFC3021. In this case the FireBrick can be either of the two addresses, and will ARP for the other address. Not all equipment is compatible with this mode of operation and so you should always test correct operation in such cases.
- The FireBrick also supports /32 (255.255.255.255) subnet mask. This means that the FireBrick will ARP for any other address, but that routing will be have to specifically directed to the subnet using routing rules.
- Note that the DHCP gateway is used when any routing sends traffic toa subnet without specifying a gateway - this allows per subnet gateways. It is set when used as a DHCP client.
- A more detailed description of routing is shown here.

DHCP restrict

The DHCP restrict mode allows the DHCP server to give different ranges of addresses to different machines on the network based on the name or MAC of those machines. The addresses could be on different subnets completely, or you could have multiple subnet entries with the same IP and netmask each with specific ranges to allocate on that subnet.

If a machine wanting an address quotes a name or MAC which starts with the restrict prefix or any subnet, then it can only have addresses from such subnets. If its name or MAC does not start with the restrict prefix of any subnet then it cannot use any subnet that has a restrict prefix set but can use any others that are unrestricted (restrict prefix is blank).

This is all within the restriction of subnets that are DHCP servers on the same interface (and same VLAN if using VLAN subnets). If you have VLAN subnets then that would normally a better way to manage allocations than using DHCP restrict.

The matching with the restrict prefix requires that the name quoted when requesting an address, of the full hex MAC address (no spaces or colons) starts with the prefix specified.

DHCP mirror

The DHCP mirror feature is specifically designed for cable modem situations where a single IP is available on the WAN using DHCP, but multiple machines may be required on the LAN using private addresses and NAT. In such cases it is often useful to have at least one machine on the LAN have the external IP address and not use NAT. This is simple enough except for the fact that the external address may change.

Typical use means that you set a WAN subnet as a DHCP client, and have a LAN subnet as private addresses DHCP server, but also have a LAN subnet set with DHCP mirror of the WAN subnet. This second LAN subnet is typically set to use DHCP restrict so that it only applies to one machine matching the subnet name (the machine that is to use the external address).

When the WAN gets an IP by DHCP, the mirroring LAN subnet is changed so that the FireBrick has the external gateway address, and it allocates only one DHCP address which is that received on the WAN. An address mapping entry can then be used to map traffic for the FireBrick on its WAN to the LAN hence passing through the external traffic (still subject to filtering).

When the WAN address changes, the mirroring LAN changes. The expiry on the mirroring LAN is set so as to be 10 seconds after the WAN and hence ensure a smooth change of IP on the LAN side as well.

VLAN subnets

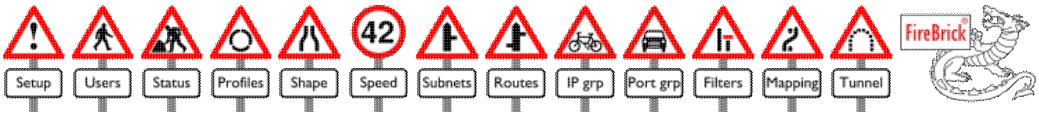
VLAN subnets allows the FireBrick to operate with an external VLAN tagging network switch. Any traffic sent to a subnet with a VLAN ID will be tagged with that VLAN ID, and this can be used on the switch to direct the traffic to specific ports. This allows groups of actual ports to be assigned to different subnets.

This is particularly useful with DHCP as it allows different ports to get different address ranges. Routing can also be used to direct traffic to specific VLAN subnets.

Using VLANs on a network switch also means that separate groups of ports can be separated, hence forcing any traffic between them via the FireBrick and hence subject to filtering rules.

Note that filtering rules apply based on the actual interface, not the VLAN, but can specify IP ranges or groups to allow control of traffic between specific groups of ports.

If VLAN subnets is not available, all VLAN tags are dropped and ignored by the FireBrick, even in stealth mode.



Routes

The FireBrick has to decide where to send traffic. This is done using routing rules. The rules are considered in order and the first appropriate matching route is applied. The routing list includes the subnets and the deafult gateway. The default gateway is always at the end, but the subnets can be moved to allow routes to be considered before or after the normal routes to subnets.

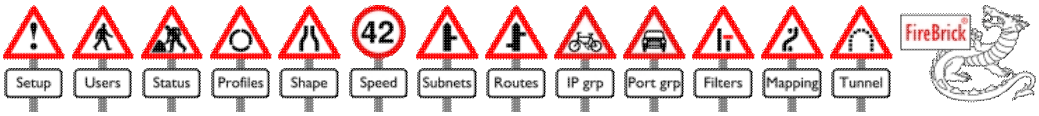
Name	Allows you to give a name to this rule
Security	Sets the security level of this rule and so defines who can view or edit the users details
Profile	Defines the profile when this rule applies.
Source	This allows you to specify one or more source interfaces from which the traffic may come
Send to	This allows you to say what interface the traffic will be sent
Gateway IP	The gateway to use on the specific interface. Blank if ARP is to be used.
Weight	For advanced use
NAT	Specifies that traffic is to be NATed when using this route
Proxy ARP	For advanced use
Source ports	This allows a range of source ports to be specified. Applicable to TCP and UDP. Normally blank meaning any.
Target ports	This allows a range of target ports to be specifiied. Applicable to TCP and UDP. Typically just one port for the specific protocol, e.g. 80 for WWW
Protocol	This allows the specific protocol to be specified, or Any.
Port group	Instead of using a source port range, target port range and protocol, then a named port group can be selected.
Source IP range	Allows the range of source IPs to be specified, or blank for any.
Source IP group	Instead of an IP range, a named IP group can be selected.
Target IP range	Allows the range of target IPs to be specified, or blank for any.
Target IP group	Instead of an IP range, a named IP group can be selected.

Technical Reference

- Stealth traffic already has a target MAC address on the other side of the FireBrick, and as such the FireBrick already knows the interface and target MAC to use. Stealth traffic is not subnet to the routing table.
- Some traffic has a partial route already, such as address mapped traffic which may be to LAN but not say which subnet, and return traffic for any sessions which should be via the interface on which it arrived. In such cases routes are only considered if they match the target interface correctly.
- Any route that sets an interface with a subnet, but for which a gateway is not defined will use the DHCP gateway defined for the subnet if specified. This is used in such cases before considering the default route. If there is not gateway defined, and the default is not the same general interface, then

an ARP is done for the target IP, even if outside the known IPs for a subnet.

- Proxy ARP is ignored on routes with protocol selection or group, as ARPs do not have an IP protocol.
- See Weighted rules for details on how to use the Weight option. This applies only with the bonding feature.
- If an explicit route is picked, then the NAT flag indicates if NAT applies or not. If a subnet route or the default route is picked then NAT is set based on the source IP/subnet being set for NAT.
- If a route is set for target Any, then the NAT flag may be set at that point, but the routing continues until an explicit target interface is found
- The proxy ARP setting causes the FireBrick to answer ARPs on the source interfaces for the IP range/group specified as the target
- Routing is done before filtering as filters operate on the apparent target interface (which is decided by routing)
- Routing is also done before any traffic shaping or address mapping wich are done after filtering. Address mapping may however cause routing to be done again if the addresses or interfaces are charged.
- A more detail description of routing is shown here.



IP groups

There are a number of places in the FireBrick configuration where a range of IP addresses can be specified, e.g. filters, shaping rules, address mapping, etc. In most places you can instead select a named IP group.

IP groups provide a convenient means of naming an IP address range or set of IP address ranges.

An example of an IP group which is included in the factory reset is the RFC1918 Private IPs. These are IP addresses reserved for priavte use and are 192.168.X.X, 172.16-31.X.X and 10.X.X.X.

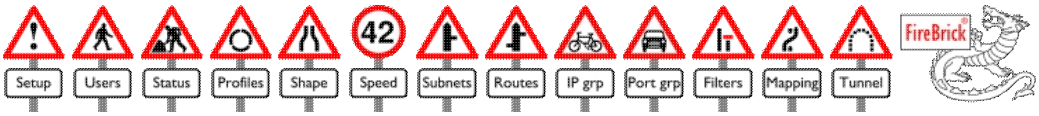
Name	Give the group a meaningful short name as this is what is shown in the list when groups can be used.
Security	This defines who can view or edit the group
Add new range	This allows a new range to be added to the group
Delete	This deletes a range from the group
Erase	This erases the whole IP group and all of the ranges within it

Logged in users

There is a special group which is not listed in the IP groups section but which can be selected in filters, etc. This IP of logged in users and is any IP address from which any user is logged in to the FireBrick. This can be useful to allow access from the location from which you have just logged in to the FireBrick. It also puts a time limit on the access as users automatically log out after a set time.

Technical Reference

- The security on a group does not affect whether a user can use the group.
- You can have a blank range for Any but this is pointless as the whole group becomes Any and you could simply not use a group instead.
- You cannot edit an entry, but you can add a new entry and then delete the old one.
- You cannot reorder entries as the order does not matter.



Port/Protocol groups

In many places, such as traffic shaping, filtering, routes, etc, you can set a range of ports and protocols to define the type of traffic being referenced. Some types of traffic use a number of different ports and protocols but you would want to allow all of these in one filter or use them all in one route. Port groups allow sets of protocol/port combinations to be defined as a named port group and used instead of specific ranges in filters, routes, etc.

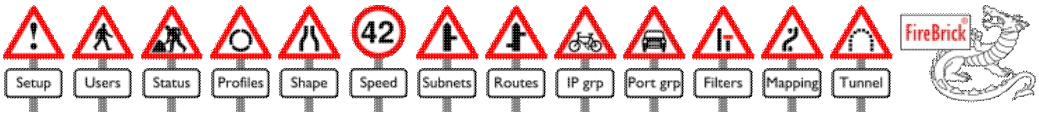
Name	Give the group a meaningful short name as this is what is shown in the list when groups can be used.
Security	This defines who can view or edit the group
Protocol	This allows the specific protocol to be specified, or Any.
Source ports	This allows a range of source ports to be specified. Applicable to TCP and UDP. Normally blank meaning any.
Target ports	This allows a range of target ports to be specifiied. Applicable to TCP and UDP. Typically just one port for the specific protocol, e.g. 80 for WWW
Add	Adds an entry to the group
Delete	Deletes the individual entry from the group
Erase	Erases the whole port group and all entries within it

FireBrick Tunnel Traffic

A default port group in the factory reset config is FireBrick tunnel traffic which uses UDP port 1.

Technical Reference

- The security on a group does not affect whether a user can use the group.
- You can add a blank Any entry but this is pointless as the whole group becomes Any and you could simply not use a group instead.
- You cannot edit an entry, but you can add a new entry and then delete the old one.
- You cannot reorder entries as the order does not matter.
- The extra handling for ICMP types as applicable to filters does not apply within port groups
- If port ranges are specified with protocol Any then this means TCP or UDP as ports only apply to these



Filters

Filters are the way that the FireBrick provides firewall protection. The FireBrick tracks every session, and applies filters when that session starts. Once checked against the filters, the session is then allowed even if the filters later change. Filters are considered in order until one matches, and then the filter rule applied to the session.

Name	Allows you to give a name to this rule
Security	Sets the security level of this rule and so defines who can view or edit the users details
Profile	Defines the profile when this rule applies
Source	This allows you to specify one or more source interfaces from which the traffic may come
Target	This allows you to specify one of more target interfaces to which the trafficmay be going
Action	This defines the main action which applies, Allow, Drop, Bounce or Reject as described belwo
Source ports	This allows a range of source ports to be specified. Applicable to TCP and UDP. Normally blank meaning any.
Target ports	This allows a range of target ports to be specifiied. Applicable to TCP and UDP. Typically just one port for the specific protocol, e.g. 80 for WWW
Protocol	This allows the specific protocol to be specified, or Any.
Port group	Instead of using a source port range, target port range and protocol, then a named port group can be selected.
Source IP range	Allows the range of source IPs to be specified, or blank for any.
Source IP group	Instead of an IP range, a named IP group can be selected.
Target IP range	Allows the range of target IPs to be specified, or blank for any.
Target IP group	Instead of an IP range, a named IP group can be selected.
Timeouts	For advanced use
TOS	For advanced use
Blink	Causes the ALERT light to blink every time this filter matches.
Flash	Causes the ALERT light to start blinking if this filter matches. It blinks until reset.
Log	Causes the filter to be logged in the internal log
Syslog	Causes the filter to be logged to an external syslog server
Email	Causes the filter to be logged to email
Quick setup	Shows this filter on the quick setup menu
Suspend	Causes this filter to be ignored, like profile Not 24/7.
SYN	For advanced use
Bypass	For advanced use

End-log	Causes logging of the end of the session regardless of the size of the session

Actions

The actions define what happens to the session. Only Allow casues a session to be created.

Allow	The traffic is allowed and the session created. All corresponding reply traffic and further traffic on the same session is allowed automatically even if the filters later change.
Drop	The packet is dropped / ignored
Reject	The packet is dropped, and an ICMP admin prohibited filter error message returned to the sender
Bounce	The packet is dropped and a valid response is sent to try and annoy the senders system. This is not a counter attack though.

Statistics

If the clock is set then a number of per filter statistics are available. These are a total of traffic both ways through the sessions associated with each filter.

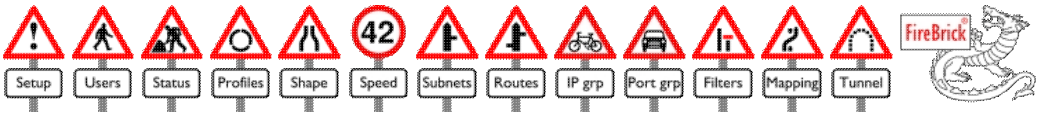
Rate Now	The instantaneous rate indication for the last whole second in KB/s or Kb/s.
Rate 5min	The average rate over the last whole 5 minute period in KB/s or Kb/s
Day This	The total transferred so far today, in MB
Day Last	The total transferred in the last whole day, in MB
Month This	The total transferred so far this month, in MB
Month Last	The total transferred in the last whole month, in MB

Note that the rate set and now can be displayed in Kbits/s or Kbytes/s depending on UI settings.

Technical Reference

- The timeouts, if not blank, define the initial and ongoing timeout for the session in seconds. If blank then defaults are used depending on the traffic. For UDP the default is a long initial timeout and short ongoing timeout. For TCP it is a short initial timeout and long ongoing timeout. The initial timeout is the session timeout before any reply has been received. The ongoing timeout is the session timeout after any reply packet has been received.
- If not blank, the TOS mask and value can be set to allow the traffci to be checked for TOS value. The TOS value is ANDed with the mask and check against the value. Note that the TOS of the initial packet in a session is checked only.
- If SYN is set then a TCP session will only match this filter if it has SYN and not ACK set.
- Bypass cuases the filter to be applied as normal, but does not allow a session to be created. This allows per packet rather than sessions based filtering if required. This is much slower but can be used where a machine is, for example, deliberately port scanning through a FireBrick and so many sessions would not timeout for a long time.
- End log will cause logging as per the log large sessions settings in the main setup menu, which may not be the same as those of the filter.
- If filters are moved, then statistics for on-going sessions may become distorted.
- For the Bounce action, the packet is dropped, but for ICMP pings a ping response is generated, and for TCP connections a SYN+ACK response is generated. This simulates a valid connection. The TCP response has a window size of 0 which will lock up some TCP stacks. The responses are also randomly slow (around 300ms) and so lower priority compared with real traffic. The bounce mode is intended to cause annoyance to a would be attacker.
- If no filter matches, the default filter is used as specified in the setup menu.
- Before the default filter there is an implicit filter allowing LAN to FireBrick port 80 and allowing FireBrick to any traffic. Explicit filtering rules can however block this default behaviour if required.
- Filters are applied after routing is done (as filters used the target interface that has been decided by the routing rules).

- Filtering is done before address mapping and so any IPs and ports apply before the addresses are changed.
- Filtering is done before NAT is applied
- Filtering is also done before traffic shaping.



Address mapping

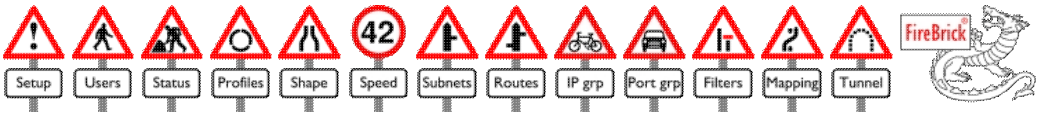
Address mapping allows a change to be made to the ports and IP addresses of traffic going via the FireBrick. This is the same principle as NAT, but allows more specific changes to be made. All new sessions are put through the address mapping table, and the first applicable entries is used.

Name	Allows you to give a name to this rule
Security	Sets the security level of this rule and so defines who can view or edit the users details
Profile	Defines the profile when this rule applies
Source	This allows you to specify one or more source interfaces from which the traffic comes from
Target	This allows you to specify one of more target interfaces to which the traffic was originally being sent (after any routing)
Map traffic to	This defines the new interface to which the traffic is to be sent
Weight	For advanced use
Source ports	This allows a range of source ports to be specified. Applicable to TCP and UDP. Normally blank meaning any.
Target ports	This allows a range of target ports to be specifiied. Applicable to TCP and UDP. Typically just one port for the specific protocol, e.g. 80 for WWW
Protocol	This allows the specific protocol to be specified, or Any.
Port group	Instead of using a source port range, target port range and protocol, then a named port group can be selected.
Source IP range	Allows the range of source IPs to be specified, or blank for any.
Source IP group	Instead of an IP range, a named IP group can be selected.
Target IP range	Allows the range of target IPs to be specified, or blank for any.
Target IP group	Instead of an IP range, a named IP group can be selected.
New target port	This allows the target port to be changed. Leave blank for no change
New source IP	This allows the source IP to be changed. Leave blank for no change. Set to 255.255.255.255 to cause NAT.
New target IP	This allows the target IP to be changed.

Technical Reference

- Both stealth and routed traffic is considered, and any mapped traffic becomes routed.
- Filters are applied before the traffic is mapped
- If a mapping rule matches, then routing is re-done with the new target IP, ports, etc.
- Speed lanes are applied to the pre-mapped IP and ports, but the mapped target interface
- If there is mapping entry the the routing table is reconsidered

- If a target IP range is specified (not an IP group) then the new target IP defines the value mapped for the first of the target range specified. The next is the new IP plus one and so on. i.e. a block of target IPs is remapped to a blank of new IPs. If this is not required then use an IP group to specifyc the range.
- Setting the new interface to a partial interface, e.g. just WAN with no subnet selected, will restrict the routing rules considered to be on that interface.
- Setting the new interface to Any will allow the routing rules to decide the new target interface
- Setting the new source IP to 255.255.255.255 causes the FireBrick to use NAT, using an IP appropriate for the subnet on which is eventually sends the packet.
- See Weighted rules for details on how to use the Weight option. This applies only with the bonding feature.



Tunnels

Tunnels provide a means of accessing remote networks and creating a virtual private network (VPN) to other FireBricks.

Once a tunnel is created it becomes available in the routing rules as a destination, allowing traffic to be routed down a tunnel.

Name	Give the tunnel a meaningful name - this appears in the list of target interfaces for routing, etc.
Authentication security	This defines who can view and edit this tunnel
Profile	This defines the profile applicable to the tunnel
IP at far end	This is the IP address of the other end of the tunnel. This can be blank at one end but a secret should be used in such cases.
Reference	This is the tunnel number of the tunnel at the far end that matches this tunnel
Secret	This is an optional secret (password) which must match the tunnel at the far end
Optional Interface	Specify the interface and subnet on which the tunnel traffic is to be sent, otherwise normal routing applies. Note that you have to be careful not to send tunnel traffic down the tunnel!
Optional Source IP	Specify the source IP for the tunnel traffic. Normally set automatically. If an subnet is defined for the interface, uses the subnet IP.
Optional Gateway IP	Specify the gateway IP for tunnel traffic. Normally set automatically. If a subnet is defined for the interface then the DHCP gateway on the subnet is used.
MTU	For advanced use
Fix	For advanced use
Limit	For advanced use
Keep-Alives	Controls when keep alive messages are sent (see keep alives)
Auth-All	Set to use slower authentication of all outbound packets (see below)
Bonding Set	For advanced use
Bonding Bias	For advanced use
QOS	For advanced use
Reorder	For advanced use

Routing

Creating a tunnel does not automatically say what traffic is to be sent down a tunnel - you will need to make a routing rule to specify this. Typically you will have a range of addresses which you know are at the far end of the tunnel, and will make a route sending those addresses down the tunnel.

Filtering

Creating a tunnel does not automatically allow the traffic through the FireBrick. You will need to consider what traffic you wish to allow from or to the tunnel. In many cases all traffic from Tunnel to LAN or from LAN to tunnel will be wanted, and so two filters will need to be added to allow such traffic.

The tunnel traffic is wrapped up in another packet to send over the internet to the other FireBrick. This traffic needs to be allowed. The FireBrick can is allowed to send traffic (unless a filter explicitly blocks it), but it will not automatically allow such traffic. To allow tunnelled packets through, create a filter allowing traffic to interface FireBrick on UDP port 1 (or use the FireBrick Tunnel Traffic port group).

Keep alives

Every second the FireBrick can send a small packet to the far end, which is a keep alive. This packet allows the far end to confirm the tunnel is all working. Both ends send these, and this is how the tunnel is shown as UP or DOWN. Normally keep alive packets are sent if they are being received, or if the far end has a fixed IP address. However there are other options. Master causes keep alives to always be sent. Slave causes them to only be sent if being received. Timeout causes them to be sent if there has been any traffic in the last minute - this is useful when operating on dialup/ISDN links allowing the link to drop if there is no activity.

Security

The authentication secret can be used to authenticate packets sent. This secret is used to digitally sign packets so the far end can be sure they are authentic and have not been changed in any way. You have to enter the same secret at each end.

If a secret has not been set, then the packets are not authenticated. However, the far end fixed IP is still checked, and this can be adequate security for most purposes.

There is an overhead to digitally signing and checking all of the packets sent, which can mean slower throughput. To reduce this effect the FireBrick will normally send unsigned data packets once the tunnel is established both ways. These are checked to be from same IP address as the previous (authenticated) keep alive packets recently received. There is an option Auth-All which forces all of the transmitted packets to be digitally signed. In this mode, the receiving FireBrick checks all packets so they cannot be spoofed by someone else and sent unsigned. This also applies automatically if the FireBrick at the other end is an older model (older software or a Plus or SoHo) and cannot accept unsigned packets.

Note that tunnel traffic is not encrypted.

Tunnel status

If the tunnel receives keep alive messages or any data it will shows as UP. If not then it shows as DOWN. In some cases you may see UP/DOWN where the FireBrick receives data from the far end, but the far end is not receiving data from this FireBrick. If a tunnel is not UP, then it is normally excluded from consideration as part of a tunnel set - the exception is when Timeout keep alive mode is used as you cannot be sure if the tunnel is UP or DOWN as it may simply have timed out. There are also tunnel set statistics (used with Bonding) which show the ordering of packets received, and how effective packet reordering is.

Technical Reference

- FireBrick tunnels work using a FireBrick Lightweight Tunnelling Protocol which works with other FireBricks and for which there is also an open source linux implementation.
- Tunnelled packets are sent from and to UDP port 1. Address mapping rules can be used to change this if necessary.
- FireBrick tunnels will work via NAT and reply to the port from which traffic was sent.
- It should be noted that if two ends of a tunnel are set correctly with specified IPs and send keep alives enabled, then there is no need to set an incoming filter to allow UDP port 1 as the traffic from the other end appears as replies to the outgoing traffic and so matches the session.

- The MTU defines the size of packet that can always reach the far end without being fragmented. On most networks this can be set to 1500. It cannot be set below 576. Any packet that would be too big once encapsulated is broken up first and sent in shorter tunnel packets. This is obviously less efficient.
- The Limit option will drop/reject any packets that are to big to send within the MTU when encapsulated. An ICMP can't fragment error is returned
- The Fix option causes the MSS in any TCP SYN packets to be adjusted if too high so that it will fit the MTU specified when encapsulated. This makes the tunnel more efficient. You should set this at both ends as it only affects MSS settings on SYN packets sent in to the tunnel.
- The local end IP can be set. This is rarely necessary, but in some cases it can be useful as it is checked at the far end to match the far end IP.
- If all tunnels in a set are inactive then the traffic goes to the originally routed tunnel only (i.e. as if there was not tunnel set), if it is enabled by profiles.
- With bonded tunnels, sessions reported in the session table relate to the tunnel the traffic was routed to, the redirect to another tunnel is done at the last stage when the traffic is sent down the tunnel and does not affect session tracking.
- QOS and Reorder relate to packet reordering on bonded tunnel sets, see below
- The bonding bias can be used to adjust the relative load of each tunnel in a set, e.g. 1, 1/2. 1/3, 1/4, etc. This is intended for mixed speed bonded sets, and its usefulness is somewhat experimental.
- A tunnel is treated as down if no packets of any sort for 5 seconds, or no keep alives for 12 seconds.

Tunnel bonding

With the bonding feature it is possible to select a set of tunnels which work together. When traffic is sent to any tunnel in a set, it is actually sent down one of the tunnels to share out the load. This means you can make use of multiple internet links connecting two FireBricks and send traffic down multiple links at once for a higher overall throughput.

Note that you should set MTU, Fix, and other settings the same for all of the tunnels in a set to avoid any unexpected behaviour.

If any of the tunnels are disabled (based on profile) or not active (expect keep-alives is set but none received) then it is omitted from the set and the traffic spread between the working tunnels in the set.

When bonding channels to achieve high speeds you must also consider the impact of windows size and speed/latency. Your TCP settings may need tuning to make the best use of a fast link to the internet.

Packet reordering

It is a fundamental principle of Internet Protocol (IP) that there is no guarantee packets will all arrive, arrive in order nor be duplicated. The higher protocols must handle this. For example, TCP is able to retransmit dropped packets and ignore duplicate packets. Unfortunately, not all protocols and applications work well when packets are reordered. Two key examples are that (a) some TCP stacks are much less efficient in the face of packet reordering so giving lower than expected throughput, and (b) some voice over IP systems do not handle packet reordering well resulting in distorted speech on a call.

Packet reordering in a FireBrick can happen as a result of the bonding features. These are typically used to send packets down multiple internet links. The difference in packet sizes alone can therefore result in packets arriving in a different order. Differences in the latency on each link can also have an impact.

To address this the FireBrick has two features.

- The first (now deprecated) is the QOS option. This affects the sending of traffic down a bonded set. Any traffic that matches the programmed TOS value in the FireBrick will not be bonded but will stick to the tunnel to which it was routed, load balancing other traffic around it. This is normally used to address VoIP issues by making calls stick to one or other link. Typically a load sharing routing rule set is used to send traffic to one of the links in a bonded set so some calls go down one link and some down another, etc. If traffic is sent to a link that is unusable (down by profile or keep alive) then the bonding is enabled again for that traffic until the link is usable.

- The second option is Reorder. This affects the receiving of packets on a bonded set. Packets arriving out of order are held until the missing packets are received (within limits). This can be less efficient if there are dropped packets, and so it is recommended that speed lanes be used to minimise any risk of dropped traffic between bricks. The result is that most, if not all, of the received packets come out of the FireBrick in the correct order. When this is used effecively, the QOS option becomes redundant and so is deprecated. Note that this will not work if the far end is a FireBrick which does not have the Reorder option available (i.e. older software).

Note that reordering is only available with both bonding and traffic shaping. Its use without shaping traffic to avoid inter-brick packet loss is not recommended as the far end will wait several packets before confirming a missed packet is in fact missed, hence hindering performance.

FireBrick Lightweight Tunnelling Protocol

The tunnelling protocol uses UDP port 1 packets, with one of two packet formats (single and multi-segment). In each format the first byte in the UDP payload is of the form SFPPVVV where S indicates the packet is signed, F indicates the last segment, PP indicates the part (0, 1 or 2) and VVVV is the version (2). The second byte is a tunnel reference - which tunnel at the far end to use, and starts from 1.

For signed packets, a 16 byte MD5 signature is at the end of the packet. This is a signature of the whole UDP payload up to this signature, followed by the secret. The signature covers the UDP payload only so does not include IP or port numbers as they could change in transit via NAT, etc.

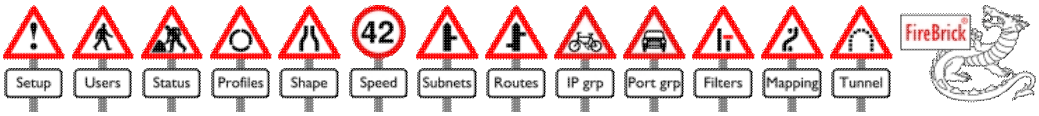
For packets that will fit in the sending MTU, the packet is sent in one go. In this case the tunnel reference is followed by the data in the packet, and then 30 bytes (before any MD5 signature) of header data. To reassemble the packet, simply move 30 bytes from the end to the start of the packet. This is a light weight tunnelling system which does not involve moving much data in the packet (only 30 bytes). These packets have F=1 (final) and PP=0 (first part).

For packets that will not fit, then the next two bytes after the tunnel reference are a cycling packet reference. Then there are up to 512 bytes of data. The packet may be in 2 parts (F=0/PP=0 and F=1/PP=1) or 3 parts (F=0/PP=0, F=0/PP=1, and F=1/PP=2) all with the same packet reference. All but the final packet are exactly 512 bytes. This is not as efficient because data has to be moved in the packet, and the 2 or 3 parts have to be reassembled. All segments in a packet are expected to arrive within 2 seconds.

For keep alive packets, F=0/PP=3 and one byte follows the tunnel reference with flags 000TRSU1. U mean the tunnel is up (i.e. sending end is seeing incoming keep alives). S means the sending end expects to continuously send keep alives (i.e. master or auto when far end is fixed IP). T means the sending end will send unsigned payload packets. R means the sending end can accept unsigned payload packets.

The IP header ID field should be contiguously increasing in the low byte on all packets sent to the same tunnel set so as to ensure the Reorder option works correctly.

Debug level logging on the FireBrick will report if there are any unexpected tunnel packets received, etc.



Quick setup

When you connect to the FireBrick admin pages, or click on the dragon, you will access the quick setup page. If you are not logged in then this may show no information as you may have no access to any filters or profiles.

The quick setup page contains a number of check boxes, which can be changed and updated with the update quick settings button.

Profiles

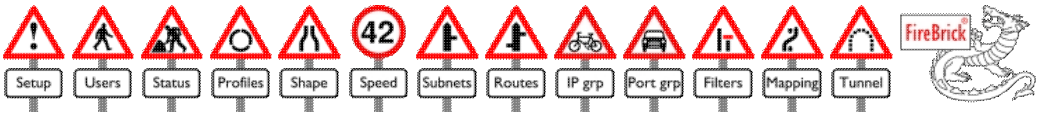
Any profiles which are set to manual mode are shown on the quick setup. The check box, if ticked, sets the profile to Manually enabled, otherwise the profile is set to Manually disabled.

Filters

Any filter with quick setup selected is show. If ticked then the filter is enabled, otherwise the filter is suspended.

Technical Reference

- Note that a suspended filter is not considered. This is not the same as a drop filter as that would stop checking of any later filters and drop teh traffic. A suspended filter means that later filters are considered. As such a filter may be unticked on the quick setup pages, but traffic of the type specified may be allowed (by another filter).
- The profiles and filters listed are only those that the logged in user can access.
- This screen also shows an startup tips, such as setting an admin password, or logging in.



FireBrick SoHo and Plus

This page lists the key differences between the FireBrick SoHo/Plus models and the newer FireBrick 105.

Configuration files from the SoHo or Plus models can be loaded into the FireBrick 105.

Key differences:-

- The 105 is fasters and has more capacity
- All ports are auto 10/100 with auto crossover and built in cable tester
- Many features have general enhancements and improvements
- IP groups and port groups added
- The SoHo used to include one tunnel and bonding features which are now optional extras
- Improved web configuration pages and tool tips
- There are prompts to suggest basic initial configuration steps, simplifying initial setup for new users

Technical Reference

General differences:-

- The 105 is significantly faster than the SoHo/Plus models allowing throughputs in excess of 10Mb/s
- The 105 has more capacity - allowing more sessions and DHCP addresses, etc.
- The 4 port switch on the 105 is a high speed network switch capable of 100Mb/s full duplex on all ports. On the SoHo/Plus this was a 10Mb/s hub
- All ports are auto 10/100 with autocrossover
- All ports have a built in cable tester
- There is no rear serial connector on the 105, which was only for factory use on the SoHo/Plus
- The bar graph mode for the LEDs is no longer available on the 105
- The MAC list on the 105 provides information on which port each MAC has been seen
- There are additional features available and upgrading can be done on line
- The 105 now has IP groups
- The 105 now has Port/Protocol groups
- There are more detailed per port counters which apply to each port, not just WAN and LAN
- Traffic routes can now be based on port/protocol as well as IP
- The 105 allows routing to individual subnets
- The 105 allows multiple DHCP clients on the same interface by use of per subnet MAC addressing
- The web configuration pages include tool tips
- Context based prompts suggest the initial setting admin password, logging in, clearing nobody rights, setting up IP and checking features

SoHo

The key differences bettween the SoHo and the FireBrick 105 with no optional features installed are:-

- The SoHo included a single FireBrick tunnel. To use tunnels on the 105 you will need to purchase the tunnel feature.
- The SoHo included multiple gateway bonding and weighted routes. These are now only part of the optional bonding feature on the 105. The limited subnets on the SoHo made these features of limited use anyway
- The 105 has all of the general additional features listed above

Plus

The Plus is roughly equivilant to the 105 with Extras, Profiles, Shaping, Tunnels, Reporting and Bonding with the following key differences:-

- There are more profiles, routes, tunnels, address mapping rules, and speed lanes
- SNMP now provides details of all 5 ports distinctly
- The 105 has all of the general additional features listed above

Ethernet Networking

This is a guide to basic networking, covering cables, hubs, switches, routers, IP, DNS, netmasks, firewalls and gateways. It is meant as a basic guide and does not cover every aspect in great detail - just enough to get you started when setting up a network for the first time - especially if you have something like an ADSL router.

Cables and wires

A network can connect using cables, coax, fibre, or radio, but we will concentrate on cables. Generally, the way a network operates once connected is the same whatever method you use. The speed may change, but it is still effectively an ethernet network we are talking about.

The cables you will have heard of are "cat 5" or "category 5" or "cat 5e". Cat 5 is basically a system of structure cabling, and a cat5 lead or patch lead is just part of that. A patch lead is just a lead with a plug each end that connects things together. Cat5 is a specification for teh cables and installation, and Cat5e is an enhanced version of that spec. Cat5 cables run up to 100m maximum to be within spec.

The cable itself has 8 wires arranged as 4 pairs. The fact that they are pairs is important to the way the electrical signals are carried. Each pair has a colour: blue, orange, green and brown. Each pair is two wires twisted together (see picture on right). One is coloured (maybe with a thin white stripes) and the other is normally white with thin coloured stripes.

The colours themselves don't matter to the electrical signals but the fact that they are pairs does. There are conventions which you should follow. It is important that the pairs are not split or rearranged.

If you look at the plugs on the end of the cable, with the lead back towardes you, they are 8 way, usually clear plastic and you can see the wires in the plug. The colours normally used, are as follows, with pin 1 on the left:-

1	White/Orange
2	Orange/White
3	White/Green
4	Blue/White
5	White/Blue
6	Green/White
7	White/Brown
8	Brown/White

Crossover cables

For normal 10/100 networking only the Orange and Green pairs are used. One pair is *transmit* and one is *receive*. Normally a cable is straight, i.e. the same wiring at each end - pin 1 to pin 1, pin 2 to pin 2, etc. There are two types of sockets - the ones you find on a computer (MDI) and the ones you find on a hub (MDIX). The two types of sockets are wired differently so that a straight cable can be used to connect between them correctly. This means that at one end pins 1 and 2 are transmit and at the other end they are receive, so it works.

In some cases you need to connect a computer port to a computer port or a hub to a hub. Some hubs have an alternative or switched port to allow hub to hub connections with a straight lead, but if this is not the case, or it is already in use, you need a crossover lead. A crossover lead has transmit connected to receive. i.e. 1 to 3, 2 to 6, 3 to 1, and 6 to 2. You can tell a crossover lead as the colours of the cables are different at each end, e.g. one end normally starts White then Green rather than White then Orange as normal.

The FireBrick 105 has auto-crossover ports so you can use either straight or crossover cables to connect to a hub or computer without having to worry which is which. When two auto crossover ports are connected then they work out between them which will be MDI and which will be MDIX.

Cables give us a way to connect two devices together.

Hubs and switches

Having a network connection on a PC is all very well, and with a crossover cable you can connect directly to another PC, and they can communicate. However, you normally want to connect several PCs, or other network devices, together in to a group (a *network*). To do this you need hubs and/or switches.

Hubs

I hub is a device that connects several network connections together. Typically available in 4, 5, 8, 16, 24, 32 ports they work by allowing data sent from one connection to go out to all of the others. You can connect hubs together, but as a general rule you should not have more than 4 hubs from any one machine to any other as the accuracy and delay of the signals cannot be guaranteed and the network might not work properly. Hubs are very simple to use as you just plug a patch lead between a port on the hub and the PC. Hubs have the limitation that only one device can talk at once, and so as the network gets larger the capacity is reduced. Thankfully, hubs are less and less common these days.

Uplink

Some hubs have a switch by one of the ports, or an alternative port (e.g. 8X next to 8). This is an uplink port. On the X port the wiring is crossed over so that the port works like a PC connection and a straight lead can be used to another hub. It is important that if the port is an alternative rather than having a switch that you use one or the other socket and not both.

If you try and connect two uplink ports together, that would need a crossover lead as both ends are crossed over in that case. You can use a crossover lead to connect two of the normal ports one two hubs together.

Link light

Hubs, and must PCs, have a link light. This helps you make sure you have the cable right. If you do, then both ends with light up the link light. If the cable is faulty it is possible for only one end to light up and it wont work. If you have the wrong sort of cable (crossover / straight) then neither end link light will be lit.

10/100/1000

Different speeds of networking are available. The most basic is 10Mb/s (10,000,000 bits per second), and is called *10baseT*. Faster networks operate at 100Mb/s, and even 1Gb/s. It is possible to get 100Mb/s only hubs, but normally anything that handles 100Mb/s will also handle 10Mb/s and is also a *switch* rather than a hub. Usually the switch/hub will automatically work out if the connection is 10Mb/s or 100Mb/s and usually has a light to tell you. The FireBrick 105 works out if a port is 10Mb/s or 100Mb/s automatically, although you can set it manually if necessary.

Switches

A switch looks much like a hub, and may initially seem to do exactly the same job. A switch has a number of sockets to connect network devices such as PCs, and allows the devices to send data to each other. The big

difference between a switch and a hub is performance. The way ethernet is designed to work is any packet of information sent on the network is actually seen by all of the other machines. This means they cannot actually all talk at the same time. With a hub, this is how it works - if one device is sending data, all of the others are receiving that data (and can't be sending data themselves). This means on a 10Mb/s hub, all of the devices share the network capacity (the 10Mb/s).

A switch on the other hand operates by receiving the message and sending it out on the right ports. This means that if A is sending data to B, then C can be sending data to D at exactly the same time. It means the speed is not shared between all of the devices. The switch uses MAC addresses (see below) to work out which ports to send data to, and keeps track of what machines you have plugged in where.

As a switch can receive and send data independently on each port, it allows 10/100 switching. I.e. a mixture of 10Mb/s and 100Mb/s devices. It can receive at 10Mb/s on one port, and at the end of the data being received, send it at 100Mb/s on other ports, and the other way around. Obviously a switch has some memory internally to hold these messages.

The LAN port on a FireBrick 105 is a full 10/100 network switch allowing high speed switching of data between the ports.

Full duplex

It is possible on 100Mb/s to operate in *full duplex* mode. This means you can send and receive data at the same time on a port. This again adds performance, but it only works where the PC can handle full duplex. The switch will normally work this out, and some have a light to say if a port is full duplex or not. If one end is full duplex and the other is not (e.g. the switch or PC got it wrong) then you will lose packets and have an unreliable link. The FireBrick 105 works out the duplex automatically, but it can be set manually if necessary.

Auto crossover

Most modern switches have auto crossover, so that you can connect directly to a PC or another switch or hub with either a straight or crossover cable. The FireBrick 105 has auto crossover, but it can be set manually if necessary.

Managed switches

A managed switch simply means that the ports can be set up manually, and normally that data on the traffic being carried can be viewed (how busy each port is). This means you could fix a port at 100Mb/s or full duplex, or whatever instead of trusting the automatic negotiation (which is not always foolproof). The FireBrick 105 is a managed switch.

With hubs and switches, we now have a way of physically connecting several devices, such as a PC, together.

LAN/Ethernet

Having covered some of the basics of how computers and network devices physically connect to each other using cabling and hubs/switches, we need to look at what they say to each other and how a *network* is created. Ethernet has been around a long time, and evolved to allow different types of physical connection. Originally it used thick yellow expensive coax cable, but can now operate over Cat5 cables and even radio network as well. The network is called a LAN (Local Area Network) as it is a network of machines that are normally physically close to each other (local area) allowing direct communications between those machines.

Packets

The internet, and ethernet work on *packets*. These are small chunks of data (up to 1.5K) that contain information about where they are to go, and where they are from, and some data. All communications on a network or the internet is broken down in to small packets like this.

Wrapping

Breaking down information in to small chunks, packets, is only part of the story. These packets are then sent in various ways, and ethernet packets are just a wrapping around another type of packets (normally an IP packet). So the ethernet packet contains information about where the packet is from and to on the LAN, and also some check data to ensure it is not corrupted in any way.

MAC Addressing

As mentioned above, the basic principle of ethernet is a *broadcast* medium - i.e. when one device sends a packet, all of the rest see it. In practice you are normally sending a packet from A to B, i.e. it is destined for only one other device on the LAN. To achieve this the destination address is included in the start of the packet, and all of the devices seeing the packet look at that address to see if it is for them. If it is not they ignore it, and if it is then they accept it and process it. A switch uses this to work out which port each MAC address is on and so where to send packets rather than sending to all ports at once.

The address is a MAC address (Media Access Control). Every network device as a unique MAC address fixed by the manufacturer of the network card. MAC addresses are only used on the local network (LAN) to allow packets to be sent from one device to another.

A MAC address is actually a 48 bit number, and is usually written using hexadecimal, e.g. 00:03:79:12:33:57. You don't normally have to deal with MAC addresses.

Now we have a way of connecting several devices, such as PC, physically together in a local network and a way that allows them to send specifically addresses packets of data to each other.

IP

IP (Internet Protocol) is the basis of the whole of the internet. It allows packets of information to be sent over a wide area network (WAN), i.e. anywhere in the world.

IP addresses

Unlike MAC addresses which are automatically allocated by the manufacturer of a network card, and which are used for addressing machines on a **local** network, **IP** addresses are allocated in a structure which allows packets to them to find their way from the other side of the world if necessary.

An IP address is a 32 bit number, and is normally written as a dotted quad, e.g. 192.168.0.1. This means that the 32 bit number is broken in to 4 parts, each of 8 bits, and each part is written in decimal (a number from 0 to 255) with dots in between. The address 192.168.0.1 is 11000000101010000000000000000001.

IP is a means to send packets to a specific IP address. Again, like ethernet, the IP packet wraps up the data it contains, and adds addressing information and checking information to make an IP packet.

Private / public addresses

You may have seen addresses starting 192.168, or 10, or even 172.16 in many places. This is because they are *private addresses*. They are just like any other IP addresses, except they will never be allocated to anyone, and so can safely be used in private networks. The actual ranges are 192.168.x.x, 10.x.x.x, and 172.16-31.x.x.

If you are setting up a private network, and need some IP addresses, you should always use these ranges. If you just make up addresses (e.g. 100.100.100.x) then they could be allocated to a real place on the internet - perhaps a customers mail server, or your favourite web site, and that part of the internet would not be accessible to you as you would have hi-jacked their addresses for your private use.

Special addresses

There are special addresses. 255.255.255.255 is a special address used to send packets to every device on a local network (not to the whole internet!). Addresses starting 224-239 are *multicast* addresses used for some special applications to send data to more than one machine at a time. The address 127.0.0.1 is special in that it is another address for yourself, i.e. on any machine this is a way to talk to itself. Address 0.0.0.0 is not valid but is used like 127.0.0.1 on many machines. Addresses 1.x.x.x to 223.x.x.x are otherwise public internet addresses.

Where addresses come from

If you need to accept data from the internet (even if it is just the data in a web page you have requested) you need a public internet address. With a modem this is typically allocated on the fly when you connect to the internet from your PC. With a leased line or ADSL connection these may be permanently allocated to you.

The actual addresses are allocated by regional internet registries (RIRs), of which there are only 4 such registries. There is one for the whole of Europe, called RIPE. RIPE allocates large blocks of several thousand addresses at a time to local internet registries (LIRs) which are typically internet providers. The whole world has to know where these addresses are to go to, so allocating large blocks to ISPs allows this to be done without too much difficulty.

The ISP will then assign addresses in small blocks to you. The ISP makes sure that any packets that are sent for your addresses get to you, via your ADSL or leased line, or whatever.

ARP

If you have several PCs on a local network, and they each have their own IP address, then they may want to send messages (IP packets) to each other. They do this by sending a packet to a specific address. E.g. 192.168.1.2 wants to send a packet to 192.168.1.3. The IP packet is created with these addresses, but to send on the local network this has to be put in to an ethernet packet. The ethernet packet needs to say which MAC address to send the packet to.

In order to find the MAC address for an IP address on the local network, a special packet called an ARP request (Address Resolution Protocol) is sent. This is a broadcast packet to all of the machines on the local network asking, e.g. "where is 192.168.1.3". The machine with that address answers and tells it's MAC address. This is remembered for a while to avoid sending loads of these broadcast messages, and any packets for that IP address can now be correctly addressed to the right MAC address.

Creating a subnet, and subnet masks

We have said that IP addresses have a structure. Large blocks allocated to an ISP, and an ISP then routes a smaller block to a customer. But how do you group IP addresses in to a "block"?

The answer is that a sequential set of IP addresses are treated as a block of addresses. However, it is not as simple as saying that a certain customer has 1.2.3.5 to 1.2.3.17 as their addresses (that would be too simple <-:). In practice a subnet is created. A subnet is a group of addresses based on the binary IP address.

What happens is that part of the 32 bit address is assigned to a network. e.g. if we have machines all starting 192.168.5. in a network (that's 192.168.5.0 to 192.168.5.255) then that is a 24 bit subnet. This is because the first 24 bits, the 192.168.5 part are the same for all machines on that network.

Looking at it in binary that network is 11000000101010000000010100000000 to 11000000101010000000010111111111, i.e. the first 24 bits (110000001010100000000101) are the same, and the last 8 bits changed (from 00000000 to 11111111). To define this a subnet mask is used. This says which bits are the same, e.g. 11111111111111111111111100000000 which is 255.255.255.0.

All machines in a network have their own unique IP address within the same block, and they also know their subnet mask. This is important so that machines know if another address is on the same network. If it is, then packets to another address can be sent directly and ARP used to find the MAC address to send them to. If not

on the same network, then the packet will have to be sent elsewhere (see below).

When the address block is 256 addresses, then it easy. e.g. 192.168.5.x. However, you will often have smaller blocks allocated (as IP addresses need to be carefully allocated so they don't run out). This means you could have, say, a 28 bit netmask (which is 255.255.255.240) allowing you only 16 addresses.

Special addresses (again)

In any subnet, the first and last address are special. The first is effectively unused, but the last is a *broadcast* address which all devices listen for in addition to their own address. Broadcast addresses like this stay on the same local network and are used for machines to tell other machines about them (windows network neighbourhood uses this, for example). This means if you have a block of, say, 16 addresses, you can only use 14 of them anyway. Note that the FireBrick 105 treats the first (network)address like a normal address, but most other equipment does not.

DHCP

When setting up a network, each machine has to have several parameters set. We know the IP must be set uniquely for each machine but must be within the same group (subnet) of addresses. The same subnet mask must also be set on every machine, and we will see below that a gateway and DNS server are normally needed as well.

DHCP (Dynamic Host Configuration Protocol) is a way for one machine on a local network to take on the job of giving out addresses to other machines. If you set up a DHCP server, you tell it a range of addresses to give out, and the details of subnet masks, gateway, DNS, and several other settings. Once a machine is then plugged in to the network, having been set to get details *automatically* then it will be allocated the necessary details from the DHCP server. This can save a lot of time and effort.

The FireBrick 105 can be a DHCP server and allocate addresses to machines on your network automatically.

Getting off the network

So far we have seen how we can physically connect machines on a network; how they can send messages to each other by MAC address; how they can have IP addresses from a small group (subnet) and how (using ARP) they can send messages to each other by IP address.

By having a subnet mask, it is possible for a machine to work out that a packet is intended for a machine outside its own network. e.g. if the network is 192.168.5.0-255, a machine sending to 217.169.0.1 knows it is not on the same network - so what does it do.

Considering a PC with only its network connection, it has to send the IP packet in an ethernet packet, and so has to put a MAC address in that packet to say where it is going. It knows the destination is not on the same network, so it knows there is no point sending an ARP to find the MAC address. Instead it sends an ARP to find a *gateway* machine. That is a machine that knows what to do with this packet. This means that each machine needs a gateway address - the IP address of a machine on the local network which talks to the rest of the world.

Sending the packet to that machine (still addressed at the IP level to the original IP address), means that it can then be sent on its way via modem, ISDN, leased line, ADSL or whatever.

Routing

There is more to life that just getting the packet off the network. There are sometimes situations where you need to send some packets one way and some another. This is where routing tables come in. A routing table says which blocks of IP addresses go via what gateway or route. You can set these on windows PCs (from DOS), but usually you will have one machine on your network that is a router, and it will have routing rules to say what goes down (say) the ADSL, what goes via another box (perhaps an ISDN router) and so on. The PCs on the network can then live with just a gateway address and the gateway bounces the traffic to the right place.

The FireBrick 105 can act as a router directing traffic to other gateways as necessary.

In the internet there are lots of routes that say which blocks of IP addresses go where and this is how the IP addresses for you network get to come down the right line to your router.

More detail on routing within the FireBrick is here.

With IP addresses we have a way to send packets of data to any IP address on the planet.

DNS

So far we have seen how a network of networks (an internet) can be set up. Each local area network has a small group of IP addresses (a subnet) and some gateways or routers carrying traffic off its network to the rest of the world. The Internet is a collection of networks and links owned by ISPs and companies.

This is all very well if I want to send a packet to 217.169.0.1, It will get there. What if I want to go to a web site? That has a name not a number. What if I want to send an email - that uses a name not a number..

To solve this DNS (Domain Name Service) is used. This allows names to be turned in to numbers. Normally every PC will list one or two DNS servers (which have to be listed by IP address and not name or else you get a bit of a chicken and egg problem). The only reason for two is resilience. These are servers, usually provided by your ISP, which will look up names for you and get the addresses.

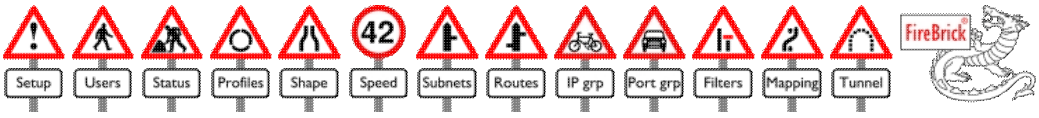
So, when you put www.me.uk in to a web browser, the browser uses DNS to look up that as 217.169.20.29 and then fetch the page.

The FireBrick 105 can act as a DNS relay. This means you can use it as a DNS server, and it will send your requests on to the DNS server it has been configured to use.

With DNS we can call machines by names, and so we can go to web sites, send email, and communicate with the world

---

If you want to know more about IP networking, ack your supplier about IP training courses.



FireBrick routing, mapping and filtering

This is a general description of the way in which the FireBrick establishes and routes a new session.

What is a session?

The FireBrick tracks sessions. These are a set of packets in each direction that belong together. The initial packet will start the session and then there are replies and further packets in the same session.

In the case of TCP, there is a specific sequence of packets to establish, maintain and drop a session. The FireBrick tracks this from the initial SYN packets through to the final FIN packets.

In the case of UDP, the FireBrick tracks corresponding replies back to the same port and IP from the same port and IP as the initial packet, and uses timeouts to end the session.

In the case of ICMP, replies are tracked, like echo replie to echo requests, and also ICMP errors are tracked as part of the initial session to which they relate.

In the case of other protocols, the session is tracked based soley on the source and target IP addresses, which places some restrictions on using NAT with protocols like GRE.

The FireBrick will route the initial packet in a session based on routing rules, and will apply filters and mapping rules. The replies undo any mapping or NAT and are routed back. The first reply packet is routed within some constraints so that it will head for the same interface as the original packet, for example.

What order are things done?

The order things are done is as follows - this is also the same order in which the routing, filtering and mapping icons are shown above.

1. Routing rules are applied. This defines the interface to which the packet is to be sent - which is necessary to for filtering and mapping
2. Filtering is done. This is done before mapping as mapping will potentially lose some information (i.e. many ports could be mapped to one, etc)
3. Mapping is done, potentially changing the IPs and ports
4. If mapping was done, then this may mean routing is re-applied

Routing rules

Routing is the process or working out which interface, sub interface (e.g. subnet or tunnel) and gateway (for ethernet interfaces). For tunnels the traffic goes to the specified tunnel, and there is no need for a gateway. For routes to ethernet interfaces (e.g. LAN/WAN) there needs to be a gateway address for any traffic to an address off subnet.

Routing is done by looking through the routing rules in order and trying to match the traffic to the routing rule. This can mean checking ports and protocols even. At the point that subnets are routed, the subnets are checked in order for an address on one of the subnets. Then the routing rules resume. Ultimately if there is no match, the default routing rule is applied.

In some cases there is already some routing information. This applies when routing the initial reply of a session, but also when traffic like pings, tunnels, etc, have some inforation specified such as an interface. Where an interface is know, the routing and subnet rules are only considered if they route to that interface.

Routing can mean sending traffic to a specific interface. It can further mean specifying a specific subnet. It can also mean specifying a specific gateway for an ethernet interface.

The normal recommendation is to route traffic that is for the ethernet to a specified subnet, and not specify a gateway. If the routing comes outs as being a subnet and no gateway, then the routing could be to an IP on that subnet - in which case ARP is used. If not, then the gateway sepcified in the subnet config (under DHCP server) is used. Failing an ARP is done for the off-subnet target IP address. ARPs are done from the FireBricks IP/MAC on that subnet.

Routing to a subnet and not specifying a gateway but allowing the subnet specific gateway to be used helps avoid duplication. This is very useful when the external subnets are allocated by DHCP and so the gateway can be changed. This is particularly useful when multiple subnets are used for multiple external interfaces and tunnels are set to go to specific subnets with DHCP gateways.

If routing is to an ethernet interface with no gateway or subnet then the FireBrick will ARP for the off-subnet address. It may use its stealth address as the source for the ARP and the base MAC address. This is not ideal, and a gateway or subnet should be specified.

If routing to an ethernet interface with a gateway but no subnet specified, then the subnets are checked for the first subnet containing the gateway address, and this is used.

NAT

NAT simply means that the source IP is changed to that of the FireBrick. The FireBricks IP will depend on wheich subnet it is sending traffic to. If to a tunnel, then the tunnel can set the NAT IP to be used, else the NAT is applied when leaving the far end brick via an ethernet interface. This means the source IP is set at the last stage, after mapping, and so filtering on the source IP is not possible.

NAT is applied if the routing rule is to the default gateway or matched a subnet, and the source subnet has NAT selected. Failing that NAT is applied if the explicit routing rule used has NAT set. This means that any explicit routing rule that is used without NAT ticked will not use NAT even if the source is a subnet that has NAT ticked. Sometimes this means different explicit routing rules for traffic from some addresses than others even if the destication is the same, as one set may need NAT and one set may not.

Filtering rules

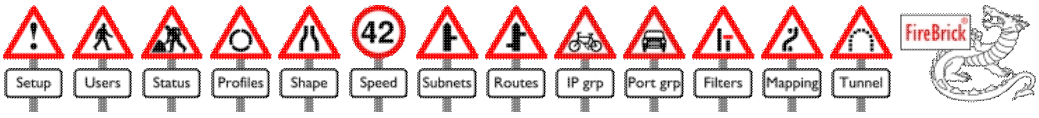
Filering rules simply match the assigned source and target interface (from the routing rules), IPs, ports, etc, and decide if the session is allowed. If allowed, all replies are allowed and further traffic within the session.

Mapping rules

Mapping rules are then applied, and again work on similar critera to filtering. They then change the target interface, source and/or target IPs and/or ports. If changing the source IP then a new source port is assigned. Setting the new source IP of 255.255.255.255 has special meaning and makes the FireBrick apply NAT.

Key points

- We recommend that all subnets are set up with IP and mask, but also with a gateway applicable for that subnet. This allows routing to the subnet without having to specify the gateway address again in the routing rules.
- Any explicit rules used for routing must consider NAT as it will not be automatically applied unlike routes to subnets or the default gateways
- The order routing, filtering and mapping are applied are the same as the icons on the top of the config, routing -> filtering -> mapping.



Bonding

The FireBrick 105 has an optional bonding feature which allows both load sharing (typically to share downlink on multiple internet feeds) and multiple gateway operation (typically to share uplink on multiple internet feeds).

Uplink bonding

Uplink bonding means making use of multiple internet routes to provide a true aggregate uplink. This applies even to a single session or file transfer as the bonding is done on a per packet basis.

To provide a mechanism for sending packets to multiple routers the FireBrick uses a pseudo router address. This is a normal IP address that could be on the same interface as your routers. Whenever the FireBrick uses this as the gateway for traffic, it will follow all of the normal rules for any gateway address and determining which interface to send traffic.

At the last moment when it would get the MAC address for this gateway (checking the ARP cache or sending an ARP), it sustitutes one of the real gateway addresses. It then uses the corresponding MAC address for the real gateway and sends the packet. The packet still goes on the interface it has decided based on the pseudo gateway address defined in the routing.

The FireBrick can be configured with one or two pseudo gateway addresses, and up to six real router addresses. When it tries to send traffic to either gateway address it will in fact send to one of the real addresses to share out the load.

The real addresses can be specified as an IP address, or by reference to a subnet which has a gateway address defined. Using a gateway on a subnet avoids dublication and also use with DHCP subnets.

The reason there are two pseudo addresses is that you may want to use NAT with separate external addresses. When you route NAT traffic to a gateway the gateway address is used to work out which subnet applies and hence which of the FireBricks own IP addresses to use for NAT. Having two pseudo addresses allows for NAT via two different subnets and hence two different source addresses whilst still sending traffic via multiple actual gateways.

Without NAT, it is normal to have a separate block of addresses on the LAN routed via one or both of the external internet connections. It is sensible for any FireBrick originated traffic (time setting, emails, etc) to be sent using an address in this separate routed block. As such it is normal to have this subnet on the LAN and the WAN. The LAN first allowing use of the addresses for PCs, and the WAN so that the FireBrick has an address it can use when talking to the WAN. Then, one of the routed block of addresses is used as a pseudo gateway address.

The pseudo address could be any address which the FireBrick can route to the interface, but is ideally one of the addresses on a WAN subnet to allow the FireBrick to use a sensible address itself. However, this would waste one otherwise usable address purely for internal operation. This is particularly annoying if the external subnets are /30s which would not leave space for a third address for this purpose. For this reason it is recommended that the otherwise unusuable network address (first address in the subnet) is used as the pseudo gateway. The FireBrick will treat a network address as a host and so allow it to be a valid gateway address. Remeber, this pseudo address is purely internal and not actually used outside, so the router does not have to be configured specially.

Note: Some routers seem to take offence to an ARP being sent for their IP apparently from an IP not on their subnet. This can be avoided by using the subnet with gateway to set the real address as the ARP is from the subnet address. It can also be avoided by making the routers think they are all part of the same larger subnet even if they are in fact using adjacent /30 blocks.

Different speed uplinks

There is no allowance for different speeds of uplink, but this can be accomodated to some extent. e.g. if uplinks of 250K and 500K are available, list the 500K gateway twice and the 250K gateway once, and this will cause the traffic to be sent in the right proportions. Mixed speed links may not give the performance you expect though as there will be mre packet reordering.

Packet reordering

IP does not guarantee packet order (or packet route, reliable delivery or non duplication of packets). It is the stack, e.g. TCP, which ensures correct order and integrity of streams of data. As such TCP has to cope with dropped, repeated and reordered packets. Sending packets via two paths could result in packet re-ordering if the latency on the two paths is different.

Unfortunately, some TCP stacks are less efficient when there is packet re-ordering and so it is not always possible to achieve the full speed of the two links combined. In practice we find that this is generally not an issue and bodning uplink works well.

Packet reordering has also been seen to have some detrimental impact on voice over IP allocations, although they should be able to cope within the jitter delay. It also has impact on some RFC2833 handling where it can appear to indicate new DTMF digits as a result of duration going backwards. As such you may want to force VoIP traffic via one router or use load sharing to send individual call consistently via one route.

More than 6 gateways

In theory you could cascade multiple FireBricks to allow more than 6 final gateways to be used. If you need more than 6 links, tunnel bonding may be more appropriate. However, more links will probably start to make packet reordering a major factor.

Fallback

With the profiles feature, bonded uplink can have a profile set for each gateway allowing a graceful fallback to fewer gateways.

Downlink load sharing

If using downlink load sharing (see below), you typically NAT sessions to be from one of two separate FireBrick addresses via two internet connections. If uplink bonding is also required then you can use two pseudo gateways - one for each of the subnets for each NAT session, and list both as pseudo gateways. This means the uplink bonding applies to the traffic regardless of which IP it is NATed via. Again, profiles allow fallback to only use the correct real gateway instead of a pseudo gateway in the event of one link failing.

Downlink bonding

Obviously an ISP could offer a service of downlink bonding, either using a FireBrick at their end of multiple links or some similar device. This has no impact on the FireBrick as it simple receives the packets.

Downlink load sharing

The FireBrick can be very useful in a large office to allow use of aggregate downlink capacity on multiple internet feeds. This is applied on a per session basis, so the capacity on any one session will be limited to the feed it uses, but the overal capacity for all sessions can reach that of all feeds combined.

Probability based routing

The typical usage is with two internet feeds, each with a small block of IPs so that the FireBrick has a real IP. The LAN then uses private IPs. The FireBrick NATs traffic to the internet.

To achieve the load sharing, the routing rules are used instead of a default gateway. Two rules are defined setting the gateway to each of teh external routers with a 50% loading. This means that each session may use one or the other route. NAT is selected on the route, and the IP used will depend on the FireBricks IP for the subnet associated with the gateway selected. The reply traffic will then arrive for that IP via that gateway, and so half the sessions will have data arriving via one route and half via another.

Fallback

It is recommended that the profiles feature is used to monitor gateways and take out the routes used for any that are not working. This can distort the load sharing on the remaining routes when there are more than 2 gateways remaining, although suitable use of 50% routes as 33% routes, and then default routes can allow for this if configured carefully.

Tunnel bonding

The FireBrick can be used to establish tunnels to other FireBricks (tunnel feature). With the bonding feature as well, the tunnels can be linked in to a set allowing multiple tunnels to be used as one. Normally this would be configured so that each tunnel is sent via a different physical link.

Balancing different speeds

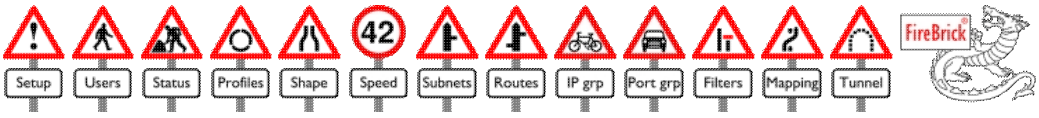
The tunnels in a set are equally weighted for traffic sent down them. If you have a mixture of physical links you can make multiple tunnels down the same link to bias the traffic to the faster links.

Fallback

Using tunnel send and expect keep alives, the tunnels are automatically removed from the set when not active both ways. This allows the remaining tunnels to share the load automatically without the need for the profiles feature. Tunnels can also be used with profiles to control if they are active or inactive.

Packet reordering

Packet reordering has also been seen to have some detrimental impact on voice over IP allocations, although they should be able to cope within the jitter delay. It also has impact on some RFC2833 handling where it can appear to indicate new DTMF digits as a result of duration going backwards. To help remove this problem tunnels have a QOS option which will force packets with TOS bits 7 or 4 set (as is common with VoIP traffic) to be sent consitently to the selected tunnel rather than bonding. If the selected tunnel is not available then this flag is ignored. You can use load sharing to send sessions to different starting points in your set of bonded tunnels to share out VoIP calls between them.



FireBrick 105 User security model

Overall model

There are security levels 1, 2, 3, 4, 5, 6, 7, 8. These are not levels as such, i.e. 8 is not better that 1 - they are just 8 different settings.

Each of the settings in the FireBrick has a level defined. It is in the setup for each filter, mapping, route, etc.

Each user has a 8 check boxes, one for each level, defining the view settings. When logged in as a user, you can only view settings on the levels which you have ticked. If you have all 8 levels ticked you can view anything.

Each user also has 8 check boxes, one for each level, defining the edit settings. If you can view a setting, you can access the setting page for it, but you can only save changes to that setting if it is on a security level where you have edit selected. If you have all 8 levels ticked then you can change anything.

There are also security settings for the top level menus. These also define the default setting applied when you erase something. They also define whether you can see the icon and list of items. E.g. if filters are set for level 1, then you can only see the filters icon if you have level 1 view rights ticked.

Changing security levels

You can change the security level of anything which you have edit rights to. If you change it to a level for which you do not have edit rights then you will not be able to change it back. If you change it to a level for which you do not have view rights, then you will not even be able to see it.

Changing your view/edit rights

You can edit any users view and edit rights if you have edit rights for the security level for that user. i.e. users have a security level just like any other settings. However, you cannot give or take away any rights for any user which you do not yourself have. You do not see the check boxes for those levels when you edit that user. This applies to your own settings too. So, if you remove your own rights you cannot give yourself them back!

Nobody user

The nobody user is a special user - it defines the rights that apply when not logged in. These can be as comprehensive as any other user. You can make a brick allow complete view and edit or all 8 levels without logging in even, if you want to, though this is not recommended.

Initially the nobody user can view and edit level 1, which allows it to set a password for the admin user. You can then log in as the admin user which gives full view and edit rights for all 8 levels. It is recommended that you always have at least one user with all rights. If not, you can never get those rights back as there is no login that will be able to tick them in the user settings.

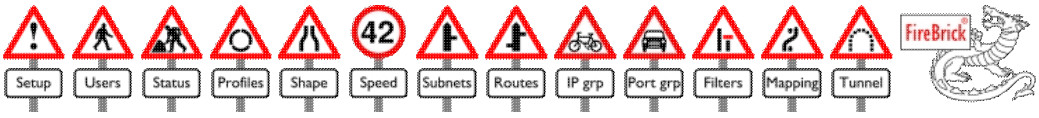
Special settings

There are settings for upgrade and config in the setup menu under security. These specifically relate to loading new software, and to loading or saving the config. This means you could make a user that, for example, can save the config or load new software and nothing else. This may be useful for some remote script that regularly archives the config, and updates bricks with new releases.

Settings for top level (icons)

For each top level setting there is a security level under settings/security. This controls if the top level icon (e.g. filters) is visible.

However, it is possible for a edit to have view or edit rights to an individual setting and not have the top level ability to list the settings. This can be useful, for example, for manual controlled profiles. If a user has rights to change one, then it will appear in the main login page as a check box, but not offer the profiles icon itself to allow listing of profiles. You can also time control when this is visible even.



FireBrick 105 Scripted access

The only configuration access to the FireBrick is via the web interface. However, there are a number of examples of cases where some automated access is required, e.g. configuring many FireBricks at ones; uploading new softtware to many FireBricks; or automated saving of configs from many FireBricks.

This page gives a basic outline of the way in which you can access FireBricks using scripts. These scripts assume you have access to curl which is a standard web page access function. Curl is standard on linux and is available for windows as well.

Except for config and software loading, all access is normal HTTP port 80 GET commands which may have a number of arguments in normal URL encoding. You can work out most commands and parameters by simple inspection of the web page source.

Using Curl

Curl has many arguments. In most examples here the arguments used are as follows :-

-s	Silent - don't print progress stats
-f	Fail silently on errors
-G	Do GET instead of POST
-d tag=value	Specify an argment to be sent
-o /dev/null	Do not actually write the output anywhere
http://ipornname/1/	The URL and session for the FireBrick

Login

Before you can access settings you must log in. Once logged in, all web page accesses much be from the same IP address. Also, they must also have the same session ID. If you are writing a script, the login is the first step.

Session

The session ID is a number that appears as a directory in front of the actual page name. e.g. http://my.firebrick.co.uk/1/main is accessing main with session ID 1. Normally when you access the FireBrick, a random session ID is allocated. However, you can pick a fixed number, such as 1, if you like, and use this throughout your script.

Logging in

dologin

user	User name
pass	Password

curl -s -f -G -o /dev/null -d user=admin -d pass=fred http://my.firebrick.co.uk/1/dologin

Info

Often, having logged in, the info function is useful. This produces a file (use -o filename, or use stdout to get the file). The file contains a number of values each of the form of a letter, =, and a string, on a line by itself.

Using and grep and sed these can be extracted easily.

The key values are:-

L	Language, normally EN
O	OEM/user interface, normally FB This is relevant when upgrading software
S	Serial number Often used as a file name in saved configs, etc
V	Version number
N	Version name
E	Internal electronic serial number
M	Base MAC address
I	Number of MACs (always 23)

Basic settings changes

To change settings, you must perform a GET on a command for the relevant settings, e.g. filter. The full range of arguments to this can be seen by inspection of the HTML for the edit page. You must not omit any settings.

filter

i	The filter number minus 1, e.g. for filter 1, i=0 This is common to most settings
g	The page number minus 1 - this shows the right page on Save so can be omitted
N	Filter name This is common to most settings
A	Security access level, minus 1 (i.e. 0 to 7) This is common to most settings
TP	The time profile. This is the profile number, 1 to 100 Profile 0 is 24/7 Profile 101 is 9-5 M-F Profile 102 is 2am Sunday Add 103 to the profile number to set Not the profile This is common to most settings
SAVE	Save, this is set to the string Save to cause the settings to be saved
ISL	IP Source first address
ITL	IP Target first address
ISG	IP Source group number, or 0 for not applied
ITG	IP Target group number, or 0 for not applied
ISH	IP Source last address, or netmask or bit count
ITH	IP Target last address, or netmask or bit count
PSL	Source port first address
PTL	Target port first address
PSH	Source port last address
PTH	Targte port last address
PPG	Port/protocol group number, or 0 for not applied
IS	Interface, source (intterface number, dot sub interface), see interfaces
IT	Interface, target
Z	Action, 1=Aloow, 0=Drop, 2=Reject, 3=Bounce

T1	Initial timeout, or 0 for default
T2	Ongoing timeout, or 0 for default
TV	TOS value
TM	TOS mask
O	Options, either multiple O= entries for each bit, e.g. O=16 for Log, etc, or one O= value with the sum of the bit values
O2	Options2, more options, see HTML

Saving config

To save a configuration, access download. e.g.:-

curl -s -f -o config.bin http://my.firebrick.co.uk/1/download

Always check that the file was created and is not zero length. Do not overwrite previous versions directly - check first. Any error in permissions, etc, may generate a zero length file.

Uploading config

To upload a config you need to make a POST containing the config file to upload with the file argument UPLOAD. e.g.

curl -s -f -o /dev/null -F UPLOAD=@config.dat http://my.firebrick.co.uk/1/upload

It is recommended you do a login again after the upload.

Uploading software

To upload software you do the same as config, but using the softare. First upload the F file, e.g.

curl -s -f -o /dev/null -F UPLOAD=@1740_2-02-569-Salorch.F http://my.firebrick.co.uk/1/upload

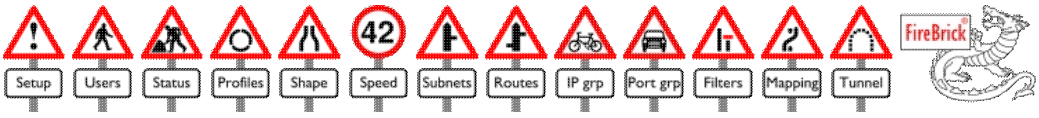
You then have to wait for it to finish flashing. You may simply want to make the next connand use --connect-timeout 60 in curl. It is recommended you do a login again after the upload. e.g.

curl -s -f --connect-timeout 60 -G -o /dev/null -d user=admin -d pass=fred http://my.firebrick.co.uk/1/dologin

Then you load the user interface. The FB in the filename is normally the OEM version. You may like to check the O field in info to confirm which is required. The WEN suffix means Web pages, EN (English).

curl -s -f -o /dev/null -F UPLOAD=@1740_2-02-569FB-Salorch.WEN http://my.firebrick.co.uk/1/upload

Again you need to wait for the flash to compleye, and the login again and fetch info to confirm the upload is complete. The V field in info says the version, and the L says the language. A language of NONE means the UI has not loaded.



FireBrick 105 Features

The FireBrick 105 has a number of standard features and also optional features.

Buying and installing features

To buy a new feature you must contact your FireBrick supplier. The supplier will obtain a feature token which can be assigned to a feature on your FireBrick. A feature token can then be assigned to a specific feature on a specific FireBrick. Once assigned the FireBrick can be updated to install all assigned features.

The supplier may simply supply the feature token to you. In which case you can assign and install the feature. The supplier may assign the feature to the specific FireBrick and feature you have requested. This is usually the case when ordering a new FireBrick with specific features.
If the supplier has access to the FireBrick, he may install the feature as well which means you do not have to do anything.

Ask you supplier how they handle feature upgrades. Some suppliers provide complete management of FireBricks and will do everything for you. Do not be alarmed if they simply provide the feature token as it is a very simple process for you to assign and install the feature on your FireBrick yourself.

Before buying a feature you should read the description here, and the description of the configuration (icons) to be sure it will do what you require. If in doubt, ask your supplier.

Obtaining a feature tokens

A feature token is three groups of 4 letters, like XXXX-XXXX-XXXX. It is unique and can be used only once to assign a feature to a FireBrick and is non transferable or refundable. You obtain a feature token from your FireBrick supplier.

Assigning a feature

Once you have a feature token you have to assign it to a specific feature on a specific FireBrick. You can do this in one of two ways:-

- If your FireBrick is on line to the internet, has a DNS server defined, and can set it's clock then you can use the setup/features configuration page to enter the feature token and select a feature required. This assigns the feature and does the installation on your FireBrick in one go and takes around 10 seconds.
- If your FireBrick is not on line or you do not wish to use the automated system, you can manually assign the feature token to a specific FireBrick and Feature using the feature tools on this web site. Once you have done this you will have to install the current features.

Installing current features

Once a feature has been assigned to a specific FireBrick, you need to install the feature on to the FireBrick itself. This can be done in one of two ways:-

- If your FireBrick is on line to the internet, has a DNS server defined, and can set it's clock then you can use the setup/features configuration page to select install assigned features.
- If your FireBrick is not on line or you do not wish to use the automated system, you can manually install the feature. To do this obtain an installation key from the feature tools on this web site, and entering it in to the setup/features page on your FireBrick. This will take about 10 seconds.

Standard Features

Standard features are included in all standard FireBricks. However, there are various OEM versions also available. OEM suppliers provide thheir own support and documentation, and they may have alternative standard features. As such it is possible to have a FireBrick which does not have these standard features.

Filtering

This is the core fire walling function of a FireBrick. It controls the filter icon, and the filtering table. Without this feature the FireBrick allows all traffic.

Grouping

This is the named IP and port group feature. Without it there are no IP groups and Port groups icons or options to select these rather than manually entering IP or port ranges.

Subnets

This is the subnets and DHCP feature. If controls the subnets icon. Without it the FireBrick can only operate in stealth mode.

Mapping

This is the address mapping feature. It controls the mapping icon. Without it the only address mapping that is possible is NAT as set in subnets.

Optional Features

Optional features can be installed by purchasing a feature token as described above. All of these are available on a standard FireBrick, however some OEM variants may not allow all of these features to be purchased.

Extras

This provides additional filters, routers administrative users, etc. It is useful for larger or more complex installations. It does not matter when you by extras (e.g. before or after buying another feature) as you will get the increased number of all features you have installed.

Menu	Normal	Extras
Administrative users	5	10 (including nobody user)
Profiles	10	100 (+3 pre-defined)
Shaping rules	30	100
Speed lanes	10	50
Subnets	5	30
Routing rules	5	100 (+subnets and default gateway)
IP groups	10	100 (up to 500 individual IP ranges in total)
Port groups	10	100 (up to 500 individual port/protocol ranges in total)
Filters	30	100
Mapping rules	5	100
Tunnels	10	100

Shaping

Traffic shaping provides a means to group different types of traffic in to speed lanes. The traffic grouping rules are much like filters in that they allow grouping on interface, IP source/target, protocol, and port source/target. The speed lanes themselves then allow the rate to each ethernet interface to be set in whole KB/s. There are also options to allow spare capacity on one or more speed lanes to be taken up by other speed lanes.

The shaping rules also allow a master rate control to which all lanes are subject unless marked otherwise. This allows, for example, a master lane to be set for an outgoing ADSL line, and then certain types of traffic, e.g. voice over IP, to queue jump that limitation.

Profiles

Profiles are a general way to turn on off almost any of the rules within the FireBrick. e.g. individual routing or filtering rules can be associated with a profile. There are standard profiles for 24/7 (always on), 9-5M-F, and 3am Sun. It is possible for a rule to be associated with not a profile, so Not 24/7 means always off. These pre-defined profiles are available in every FireBrick.

The profiles feature allows manual, timed and ping based profiles to also be used.

- Manual profiles are either on or off, and are controlled by a check box on the quick setup screen. This can be useful to allow a whole set of rules to be switched in one go.
- Time based profiles can be set on or off for each whole hour in a week. This allows aspects of the FireBrick to be time based.
- Ping profiles are on if there are responses to a ping being sent by the FireBrick, and off if there is no response. The pings can be via specific routes and gateways allowing the profile to be used to monitor an internet link or a server. When ping profiles change, a log can be generated.

Profiles can also be combined, making one profile dependent on another in some way. This allows complex combinations of time, manual switches and external availability to control operations of the FireBrick. A common use is for backup internet links allowing a profile control routing to a backup router if a main link stops working.

Tunnels

Tunnels are a way to create a virtual route from one FireBrick to another over an IP link. It allows virtual private networks (VPNs) to be created between FireBricks. The protocol used is proprietary but documented and there is at least one linux implementation freely available. The protocol allows authentication of tunnels (by IP and MD5/secret) but is not encrypted.

Reporting

Reporting provides a number of ways of extracting information from the FireBrick and includes:-

- SNMP reporting of each port, and also traffic through speed lanes
- Email log entries
- Syslog log entries

Bonding

Bonding provides two ways in which multiple links can be combined.:-

- Multiple gateway router bonding - typically used for bonding uplink on multiple ADSL lines. Packets can be sent round robin to up to 4 actual gateways allowing aggregation of the capacity even on a single data transfer session.
- Weighted routing - typically used for bonding downlink on multiple internet links. Sessions can be sent via more than one gateway on a random probability basis which can be used with NAT to ensure replies come via a specific route. This allows aggregation of overall traffic levels although individual data transfer sessions will be limited to the speed of one link only.

5Port

The FireBrick normally operates with a WAN port and a LAN port (on 4 port switch). In this mode the WAN and LAN can be reversed, putting the 4 port switch on the WAN. There are however only two interfaces for fire walling, WAN and LAN. The names of these can be changed as necessary.

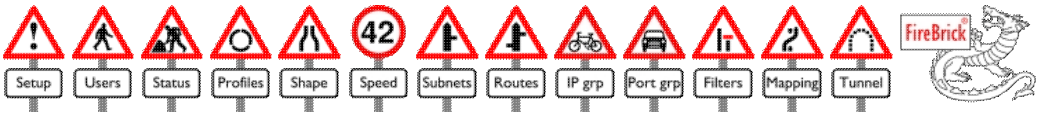
The 5Port option changes the FireBrick to allow each port to be separately configured to operate independently or as a switch. There are 5 separate interfaces for fire walling. This allows configurations with 1, 2 or 3 additional DMZs as well as WAN and LAN if required. Stealth mode still operates between the WAN and LAN interfaces. The factory default for a 5 port switch is to have all 5 ports as distinct interfaces.

VLAN

Normally any VLAN tags received by the FireBrick are ignored and stripped off any packets sent through the FireBrick.

With the VLAN subnets feature you can set each subject to have a VLAN identity. This means any traffic to that subnet is tagged with that VLAN tag. When used in conjunction with a VLAN capable switch this allows independent subnets to operate on different groups of ports on the switch. When the FireBrick acts as a DHCP server, it serves addresses based on the VLAN tag of the request and hence allows independent DHCP allocations for each group of ports. Routing rules allow traffic to be routed to specific subnets.

VLAN identities are not a part of shaping, mapping or filtering rules, but by careful allocation of IP ranges to different VLAN subnets, these rules can use IP ranges to identify each port group.



Additional URLs

The FireBrick web interface has a number of additional URLs. These can be accessed manually.

Session ID

Before any web page on the FireBrick can be accessed, you much have the necessary security setting, and this usually means logging in (unless the nobody user is set to have access). The Firebrick can be accessed with a number after the domain/IP, e.g. http://my.FireBrick.co.uk/123/. This number is normally allocated automatically when you access the base page of the FireBrick, and is a session ID for your connection. You should not normally use a manual session ID, but allow one to be created. However, when making scripts to access the FireBrick you may find it helpful to use a session ID which you create. The IDs are any non zero unsigned 32 bit number.

If you have already logged in, then you can simply place the required URL manually after the session ID and the training / character, e.g. http://my.FireBrick.co.uk/123/reboot

Login

If you are making any sort of script to access the FireBrick, you need to pick a suitable session ID, and complete a login. The login is done using the URL dologin with arguments of user and http://my.firebrick.co.uk/123/. e.g. http://my.FireBrick.co.uk/123/dologin?user=fred&pass=test

Once you have logged in, the same IP can access any pages using that session ID until the login timeout, or until you access logout.

Special URLs

Once logged in, or if the nobody user has appropriate read access, the following special URLs provide various information or functions :-

URL	Rights	Function / information
factoryinit	Edit Upload	Performs a full factory reset
datainit	Edit Upload	Resets all data area - includes statistics, DHCP allocations, dynamic tunnel endpoints and various dynamic information. Does not clear config and does not clear ethernet counters.
zapfilters	View Filters Edit for each filter	Erases all filters
reboot	Edit Upload	Performs immediate restart
download	View Upload	Returns current config file
upload	Edit Upload	With FILE= a file, posted as multipart/form-data, allows upload of config, code or UI image
info	View Setup	Provides a number of parameters used for factory testing
dump?DHCP	View Diagnostics	Reports table of DHCP allocations in CSV format
logt	View Diagnostics	Text dump of current log (optional argument BACK can be used to specify how many seconds ago from which the log should be started)