



GarrettCom[®]

Industrial Networking at Its BestSM

Magnum Network Software – DX

Software Release Notes

Software Revision 2.1.0 RC7 and Software Revision 2.1.1 RC2

GarrettCom, Inc.

www.GarrettCom.com

www.GarrettCom.com/techsupport

email: support@GarrettCom.com

This document contains Confidential information or Trade Secrets, or both, which are the property of GarrettCom. This document may not be copied, reproduced or transmitted to others in any manner, nor may any use of the information in the document be made, except for the specific purposes for which it is transmitted to the recipient, without the prior written consent of GarrettCom.

Copyright 2010, GarrettCom.

All Rights Reserved

Release 2.1.0 RC7 Release Notes

The following notes describe significant changes and quality enhancements in MNS-DX version 2.1.0 RC7 (since version 2.0.1 RC1).

Note that MNS-DX 2.1.0 RC7 was issued only as a Beta release.

1.0 INTRODUCTION

The following notes contain details related to the MNS-DX v2.1.0 RC7 software release. MNS-DX v2.1.0 RC7 includes many new features and quality enhancements over MNS-DX v2.0.1 RC1 as outlined below.

- 1.1 **CRITICAL CUSTOMER ADVISORY:** Please be sure to save your logs to an offline system if you wish to retain them before upgrading from an older software revision. There have been changes to the event log subsystem and any existing logs will be deleted during the upgrade process.
- 1.2 **CRITICAL CUSTOMER ADVISORY:** Please be aware that changes to the Firewall operation have been implemented that may affect operation compared to previous software releases. These changes and specific feature interactions are documented in section 2.0 and should be read and fully understood before upgrading any existing DX units that have Firewall operation enabled.

2.0 NEW FEATURES

2.1 Statefull Firewall

The Stateful Firewall operation has been expanded to operate on both incoming and outgoing connections.

2.1.1 Operation Details

Enabling the DX Firewall

The default state of the DX Firewall is that packet filtering is disabled for each IP interface.

The user may enable packet filtering on any or all of the configured IP interfaces. After packet filtering is enabled on an interface, the default behavior of the firewall on that interface is:

- REJECT all inbound IP packets
- PERMIT all outbound IP packets

The firewall can then be further configured to permit selected TCP, UDP, and ICMP traffic to flow through specific firewalled interfaces. Permitted traffic flows may be selected based on source IP address, destination IP address, protocol type, and port numbers. The rules for selecting permitted traffic flows are split into Inbound Connection Rules and Outbound Connection Rules.

Traffic Selectors

When specifying an Inbound or Outbound Connection Rule, traffic selectors must be chosen. The specific selectors are as follows:

- Source IP Address
- Source Mask
- Destination IP Address
- Destination Mask
- Protocol/direction
- TCP or UDP Ports or ICMP Types

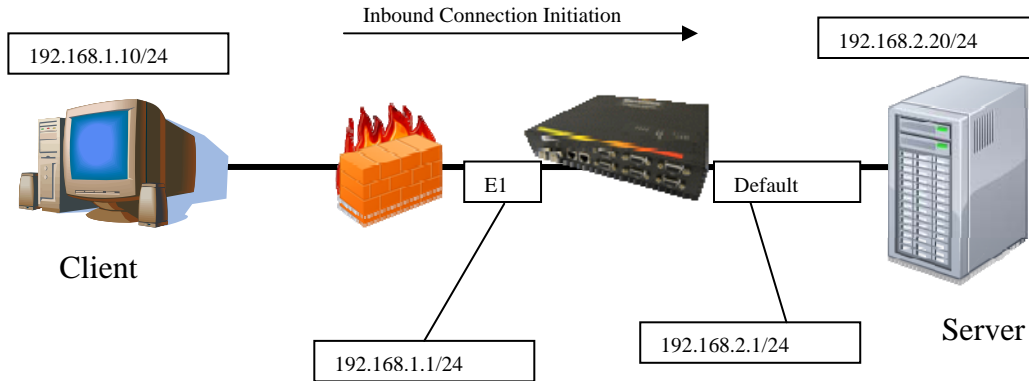
At a minimum, a Protocol/direction must be chosen. For TCP, UDP, and ICMP protocols, at least one TCP/UDP port or ICMP type must also be specified. Ports and types are specified as comma-delimited ranges (e.g. “22-25,80,443”). For stateless IP protocols like IPsec, OSPF, and VRRP the port/type field is not applicable.

Some limited wildcarding of the address information is supported. Subnets and supernets can be specified using appropriate IP address and mask combinations. Host addresses can be defined by either leaving the mask field blank or setting it to 255.255.255.255. If both the address and mask fields are left blank, the rule will match any IP address.

Allowing Inbound Connections

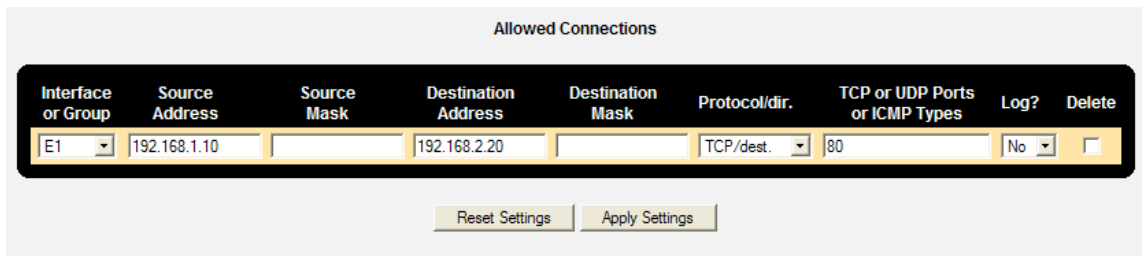
By definition, an inbound connection (or session) is initiated from “outside” the firewall. In other words, the initial packet for the session is received by the firewalled interface. In the case of TCP, this packet would be a SYN. In the case of UDP or ICMP, the packet will often be some sort of request message, e.g. an SNMP query or a ping. This sort of traffic can be permitted to pass through the firewalled interface by defining an Inbound Connection Rule. The traffic selectors in the rule should match the values expected in the initial received packet (i.e. the first packet in the selected flow).

For example, suppose you have the following network:



The DX is acting as a firewall between the outside network (“192.168.1.0/24”) attached to the routed port E1 and the inside network (“192.168.2.0/24”) attached to the Default IP interface (VLAN 1). In this scenario, since E1 is the outside interface, that is where the firewall should be enabled by the user. Once the firewall is enabled on E1, all packets from the host at 192.168.1.10 will be rejected and the Client is unable to access the Server.

Now suppose we would like to allow HTTP access from the Client to the Server. We would add an Incoming Connection rule for interface E1 specifying this permitted traffic flow as follows:

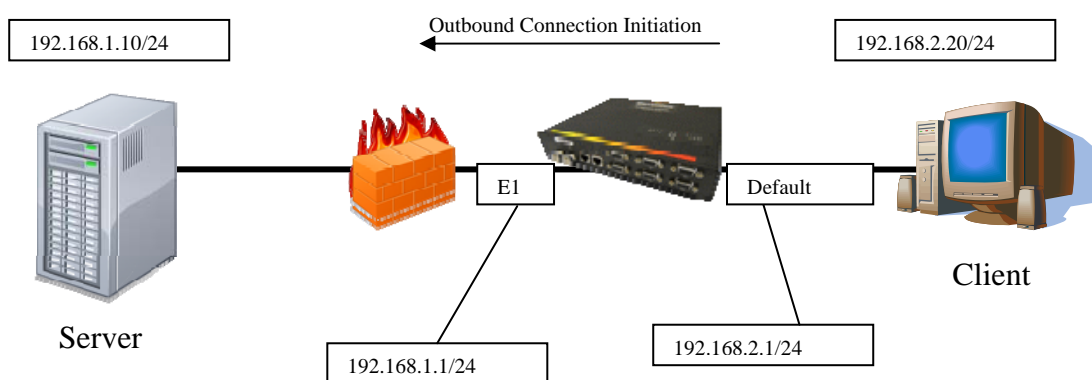


This rule permits HTTP connections initiated by the client at 192.168.1.10 to the server at 192.168.2.20.

Allowing Outbound Connections

By definition, an outbound connection (or session) is initiated from “inside” the firewall. In other words, the initial packet for the session is *transmitted out* a firewalled interface. In the case of TCP, this packet would be a SYN. In the case of UDP or ICMP, the packet will often be some sort of request message, e.g. an SNMP query or a ping. By default, all packets are allowed to pass out of a firewalled interface but the *return traffic* will be rejected after it is received at the firewalled interface. Return traffic can be allowed, thus permitting outbound connection initiation, by defining an Outbound Connection Rule. The traffic selectors in the rule should match the values expected in the initial packet *transmitted out* the firewalled interface (i.e. the first packet of the selected flow).

For example, suppose you have the following network:



The DX is acting as a firewall between the outside network (“192.168.1.0/24”) attached to the routed port E1 and the inside network (“192.168.2.0/24”) attached to the Default IP interface (VLAN 1). In this scenario, since E1 is the outside interface, that is where the firewall should be enabled by the user. Once the firewall is enabled on E1, the Client on the inside network can send packets out onto the “outside” network, but any response packets will be rejected after they are received on E1.

Now suppose we would like to allow HTTP access from the Client to the Server. We would add an Outgoing Connection rule for interface E1 specifying this permitted traffic flow as follows:

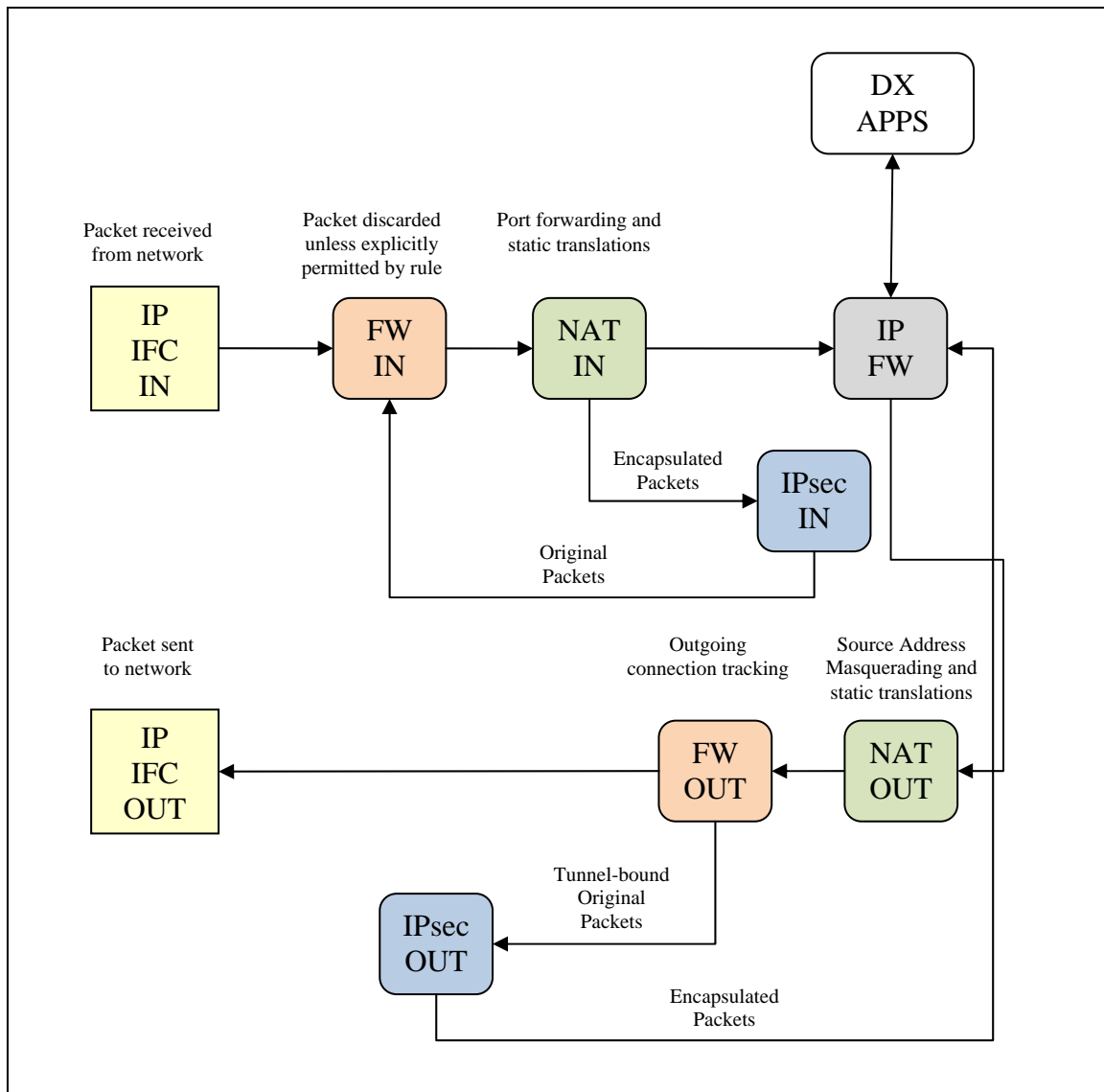
Allowed Connections

Interface or Group	Source Address	Source Mask	Destination Address	Destination Mask	Protocol/dir.	TCP or UDP Ports or ICMP Types	Log?	Delete
E1	192.168.2.20		192.168.1.10		TCP/dest.	80	No	<input type="checkbox"/>

This rule permits HTTP connections initiated by the client at 192.168.2.20 to the server at 192.168.1.10.

2.1.2 Firewall-NAT-IPSec Interaction

The DX Firewall, NAT, and/or IPsec functionality may be used together to implement various network security applications. When using some or all of these features in conjunction, it is critical to understand the way packets flow through the DX IP stack. Each stage in the flow can potentially discard, modify, or encapsulate packets, thus effecting the operation of the next stage in the flow. The diagram below is an illustration of the stages and the possible flows.



2.1.3 Packet Processing Stages

This section describes each of the packet processing stages diagrammed above.

IP IFC IN

By the time a packet reaches the IP IFC IN stage, the packet has been received from the network and has been mapped to a particular configured IP interface. Possible IP interfaces are:

- the “Default” interface. When VLANs are disabled, this is the single IP interface associated with all layer 2 bridged Ethernet ports. When VLANs are enabled, this is the IP interface associated with VLAN 1 (i.e. the default VLAN).
- An unbridged port (“routed”) interface. Ethernet ports may be isolated (i.e. conceptually removed from the rest of the layer 2 bridge functionality) by making them “unbridged” or “routed”. When a port is configured in this way, it becomes its own IP interface.
- A frame relay DLCI interface. If a frame relay DLCI is configured to run RFC 1490 encapsulation, that circuit is associated with its own IP interface.
- A PPP interface. If a PPP connection is configured, it is associated with its own IP interface.

FW IN

This is the DX Firewall Input stage. When a packet is received on a firewalled interface, that packet is discarded unless it is explicitly permitted by rule. A permitted packet may be part of a permitted and tracked Inbound or Outbound Connection. There are also a number of packet types that may be permitted that are not a part of a tracked connection or session. These types include IPsec ESP, IPsec AH, OSPF, and VRRP. While these packets are not part of a tracked connection, they are nevertheless configured as rules in the Inbound Connection table.

NAT IN

This is the DX NAT Input stage. This stage performs a number of possible translations:

- Port Forwarding. If a packet header matches a configured Port Forwarding rule, the destination address and port are translated.
- Static Translations. If a packet header matches a configured Static Translation rule, the destination address and port are translated.

- Reversing Source Address Masquerading. When dynamic NAPT is enabled on an interface, packets sent from that interface will have their source address and port translated in order to hide the internal addressing scheme. Return packets must have this translation reversed.

IPSEC IN

This is the DX IPsec Input stage. If an ESP tunneled packet is received, it is eventually passed to this stage for IPsec processing. The packet is decrypted and the inner original packet is then re-inserted into the stack flow, causing the firewall and NAT stages to be re-executed on the original packet.

IP FWD

This is the IP FWD stage. This is the point at which forwarding decisions are made in the IP stack. A packet may be sent to a waiting DX application process or it may be passed to a different IP interface for transmission. Packets sent by DX application processes are also first passed to this stage to determine the appropriate output interface.

NAT OUT

This is the DX NAT Output stage. This stage performs a number of possible translations:

- Reverse Port Forwarding. If an incoming packet had its destination address and port translated by a port forwarding rule, the reverse process must be applied to return packets, which means their source address and port is translated in order to hide the internal addressing scheme.
- Reverse Static Translations. This translation performs the reverse of the incoming static translation. That is, if a rule is matched, the source address and port are translated in order to hide the internal addressing scheme.
- Source Address Masquerading. When dynamic NAPT is enabled on an interface, this stage translates the packets source address and port in order to hide the internal addressing scheme.

FW OUT

This is the DX Firewall Output stage. This stage tracks configured Outgoing Connections. Packets for permitted, tracked connections are allowed by the FW IN stage.

IPSEC OUT

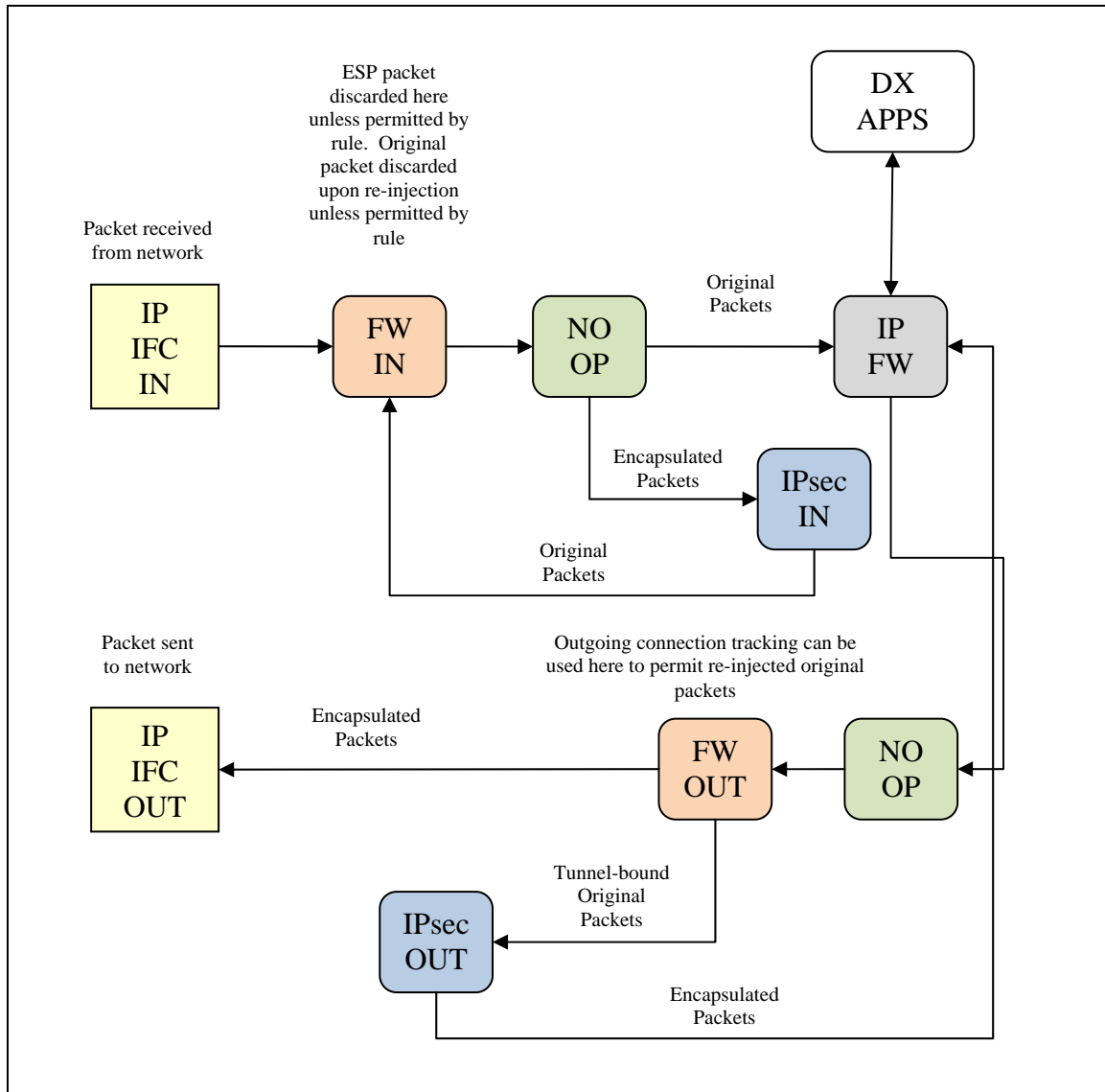
This is the DX IPsec Output stage. Packets that match the traffic selectors for a configured VPN tunnel are sent to this stage for encapsulation and encryption. The encapsulated packets are then re-inserted into the IP FWD stage to determine the next hop of the ESP packet.

IP IFC OUT

This is the IP interface upon which the packet is ultimately sent.

2.1.3 VPN + Firewall

This example describes the packet flow in a system where VPN and Firewall work in conjunction.



IKE

DX IPsec uses the Internet Key Exchange (IKE) protocol to set up security associations. This protocol runs over UDP on port 500. If IKE is to run over a firewalled interface, a specific permit rule must be defined for it in the firewall's Inbound Connection table. This rule can be as general or specific as required by the network application being implemented (e.g. the rule might only allow IKE sessions from specific IP addresses), but it must specify the UDP protocol with a destination port of 500.

ESP

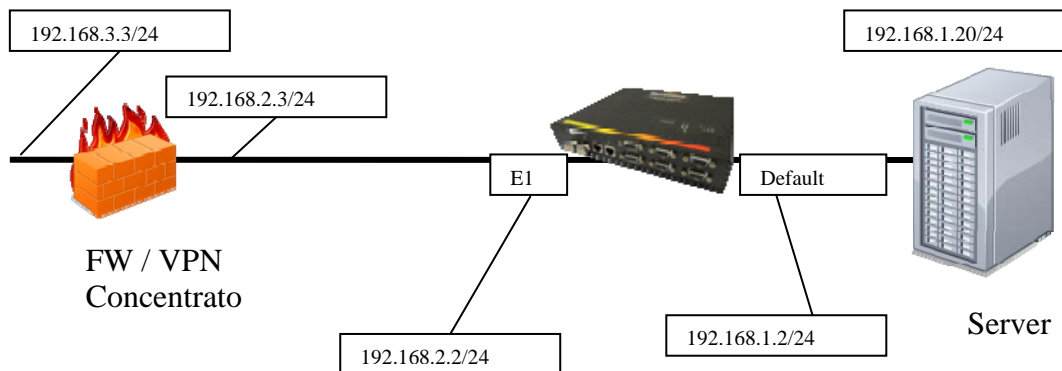
DX IPsec uses tunnel-mode ESP for encrypting data packets. ESP is a special IP protocol with protocol number 50. An Inbound Connection rule is required that permits ESP packets to flow through the firewalled interface. Again, this rule can be as general or specific (with respect to IP addressing) as desired as long as it specifies the ESP protocol type.

IP

After ESP packets are decrypted, the original tunneled IP packet is re-injected into the IP stack and passes back through the FW IN stage. The original IP packet flow must therefore be explicitly covered in the firewall rules. You may use an Incoming or Outgoing Connection rule (whichever is more applicable) to specify that the original IP packet flow is to be permitted to pass through the firewalled interface.

Example Rules

Suppose you have the following network.



The DX interface E1 is firewalled. In addition, a VPN is running between DX and a FW / VPN Concentrator at a central site. HTTP requests generated by hosts on the

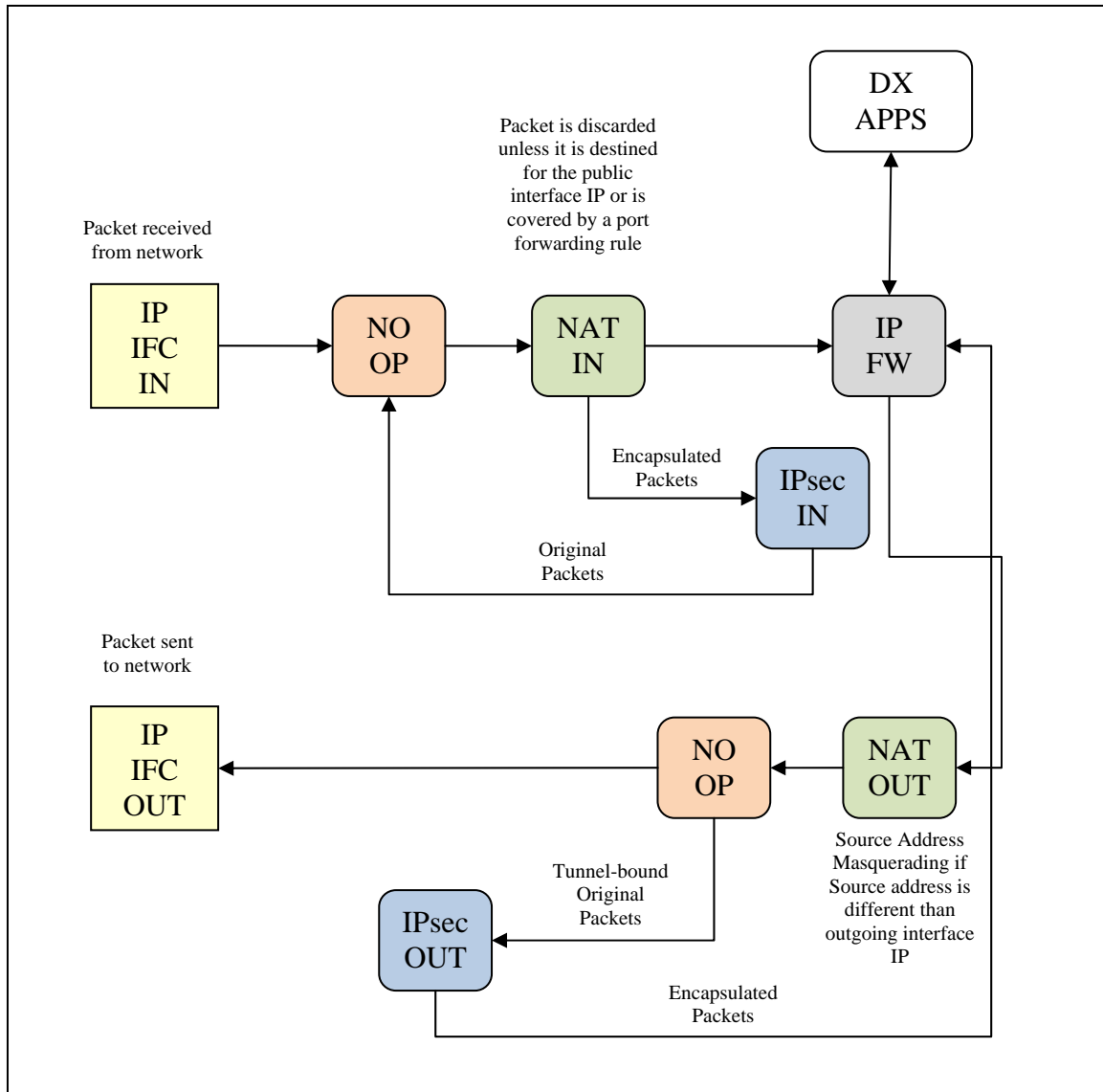
192.168.3.0 destined for the 192.168.1.0 network should be encrypted and passed through the tunnel.

On the DX, the following firewall rules are required to allow all of the IPsec related traffic to pass.

- 192.168.2.3 , 255.255.255.255 , 192.168.2.2 , 255.255.255.255 , UDP/dest. , port=500
- 192.168.2.3 , 255.255.255.255 , 192.168.2.2 , 255.255.255.255 , ESP
- 192.168.3.0, 255.255.255.0 , 192.168.1.0 , 255.255.255.0 , TCP/dest., port=80

VPN + NAPT

This example describes the packet flow in a system where VPN and Dynamic NAPT work in conjunction.



IKE

The operation of IKE is unaffected by the NAT. This is because all IKE traffic is sent and received on the dynamic NAPT public interface. The dynamic NAPT only effects traffic that passes between the public interface and a private interface.

ESP

The operation of ESP is unaffected by the NAT for the same reason that IKE traffic is unaffected. In tunnel-mode ESP, the packets do not pass between the public interface and a private interface. Instead, they terminate at the public interface, are decrypted and de-encapsulated, and then the original packet is re-injected into the stack at the NAT IN stage.

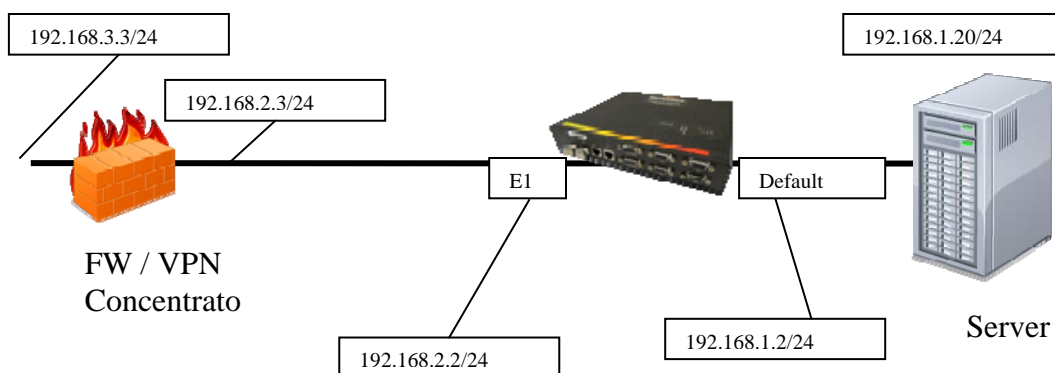
Port Forwarding and VPN Tunnel Specifications

Address translations are only performed on the original packets. The rules for translations apply directly to the original, unencapsulated traffic just as if the VPN did not exist. At this time, VPN tunnels do not have the option of bypassing the dynamic NAPT. This has two important consequences when defining your port forwarding rules and VPN tunnel specifications:

1. When packets leave the IPsec tunnel, they will be processed by the NAT. This means that a port forwarding rule must be defined to allow that packet to be translated and passed to the private network, just as you would normally do if you were only using NAT.
2. The addressing scheme used on the private network (i.e. the network “behind” the NAT) is still hidden by the DX NAT functionality. Hosts at the “outside”, e.g. at the remote VPN location, must still address their packets to the DX NAPT public interface address. In addition, due to Source Address Masquerading, packets received by outside hosts via the VPN tunnel will appear as if they were sent directly from the NAPT public interface. The VPN tunnel specification must be configured with these addresses in mind.

Example

In this example, the private network at 192.168.1.0 is hidden behind the DX NAT. Interface E1 is configured as the NAPT public interface. Interface Default is considered to be the private interface. Connections can be made out of the private network to the public network or port forwarding rules can be configured to allow specific connections to be made through the NAT from the public to the private network. In addition, packets between hosts on the 192.168.3.0 network and the server on the private network are tunneled via ESP.



First, we want to allow HTTP packets to pass through the NAT to the server at 192.168.1.20. First we choose a public port for our HTTP access. Since 80 is already used for DX management, we choose a new port, e.g. 10080. We then create a port forwarding rule that maps all public accesses to port 10080 to the server at 192.168.1.20, port 80.

Next, we create a VPN tunnel on the DX with the following specification:

Src = 192.168.3.0 / 255.255.255.0 , Dst = 192.168.2.2 / 255.255.255.255 , GW = 192.168.2.2

A complementary specification would be configured on the VPN concentrator at the central site.

Once these two rules (port forwarding and VPN) have been configured, all client access from the 192.168.3.0 network to 192.168.2.2 (port 10080) will be sent encrypted through the tunnel, translated at the DX, and then passed unencrypted to the server at 192.168.1.20 (port 80).

Note that this setup would allow other outside hosts to access the server via the port forwarding process at 192.168.2.2 (port 10080). This access could be denied by enabling the firewall on E1 and specifying the appropriate rules to only allow VPN access coming from the 192.168.3.0 network. An example set of firewall rules to accomplish this is as follows:

```
192.168.2.3 , 255.255.255.255 , 192.168.2.2 , 255.255.255.255 , UDP/dest. , port=500
192.168.2.3 , 255.255.255.255 , 192.168.2.2 , 255.255.255.255 , ESP
192.168.3.0 , 255.255.255.0 , 192.168.2.2 , 255.255.255.255 , TCP/dest., port=10080
```

2.2 Firewall Logging

Each Inbound and Outbound Connection rule has the option to be logged.

Important Note: Please take care in enabling the firewall log feature as it will increase the number of event log messages being locally stored as well as sent via methods such as syslog operation if enabled.

When a valid connection is permitted by a rule, with logging enabled for that rule, the connection state events are generated and sent to any configured logging systems (local DX log file, syslog, etc.). The specific events are defined below:

If the TCP handshake for a permitted connection is started but never finished:

TCP S.S.S.S (SP) -> D.D.D.D (DP), Session started

If the TCP handshake for a permitted connection is started and completed:

TCP S.S.S.S (SP) -> D.D.D.D (DP), Session established

If activity was detected on the connection during the TCP timeout interval, an update is generated:

TCP Update S.S.S.S (SP) -> D.D.D.D (DP), X packets

If activity was not detected on the connection during the TCP timeout interval, the current TCP flow state for the connection is silently discarded.

If the TCP connection is explicitly closed by one side or the other:

TCP S.S.S.S (SP) -> D.D.D.D (DP), Session closed

If the first packet of a permitted UDP session is detected:

UDP Start S.S.S.S (SP) -> D.D.D.D (DP), Session started

If activity was detected on the UDP flow during the UDP timeout interval, an update is generated:

UDP Update S.S.S.S (SP) -> D.D.D.D (DP), X packets

If activity was not detected on the session during the UDP timeout interval, the current UDP flow state for the session is silently discarded.

If the first packet of a permitted ICMP flow is detected:

ICMP Start S.S.S.S (T) -> D.D.D.D, Session started

If activity was detected on the ICMP flow during the ICMP timeout interval, an update is generated:

ICMP Update S.S.S.S (T) -> D.D.D.D, X packets

If activity was not detected on the session during the ICMP timeout interval, the current ICMP flow state for the session is silently discarded.

In addition to logging connections for permitted flows the system can be configured for the logging of packet rejections. This is accomplished through a global enabling or disabling of the feature.

Important Note: The logging of packet rejections is enabled by default.

2.2.1 “Log Rejects” CLI commands

Syntax:

```
set log-rejects <y/n>
```

Examples:

```
MagnumDX(firewall)# set log-rejects n
```

2.2.2 “Log Rejects” Web interface

Navigate to the page “ Security:Firewall:Global Settings” to view and edit the various settings for maximum connections, maximum tracked rejects, various timeouts and to globally enable or disable logging of rejected packets.

Security : Firewall : Global Settings

Maximum Connections:	<input type="text" value="100"/>
TCP Timeout (secs):	<input type="text" value="120"/>
UDP Timeout (secs):	<input type="text" value="10"/>
ICMP Timeout (secs):	<input type="text" value="5"/>
Maximum Tracked Rejects:	<input type="text" value="100"/>
Tracked Reject Timeout (secs):	<input type="text" value="120"/>
Log Rejects?:	<input type="button" value="Yes"/> <input type="button" value="No"/> <input type="button" value="Yes"/>

When logging of packet rejections is enabled, the following events can be generated:

When the first packet belonging to a particular TCP flow is rejected:

TCP Denied S.S.S.S (SP) -> D.D.D.D (DP), First packet

When subsequent packets in the same TCP flow are rejected:

TCP Denied S.S.S.S (SP) -> D.D.D.D (DP), X packets in Y-second interval

When the first packet belonging to a particular UDP flow is rejected:

UDP Denied S.S.S.S (SP) -> D.D.D.D (DP), First packet

When subsequent packets in the same UDP flow are rejected:

UDP Denied S.S.S.S (SP) -> D.D.D.D (DP), X packets in Y-second interval

When the first packet belonging to a particular ICMP flow is rejected:

ICMP Denied S.S.S.S (T) -> D.D.D.D, First packet

When subsequent packets in the same ICMP flow are rejected:

ICMP Denied S.S.S.S (T) -> D.D.D.D, X packets in Y-second interval

The maximum number of denied flows that can be tracked at one time is configurable by the user (there is also a hard maximum dictated by system resources). When the denied flow cache is full:

Warning: Maximum denied flows (N) are being tracked

The maximum number of permitted flows that can be tracked at one time is configurable by the user (there is also a hard maximum dictated by system resources). When the permitted flow cache is full:

Warning: Maximum permitted flows (N) are being tracked

2.3 Appropriate Use Banner

MNS-DX now supports the configuration and display of a user customizable login banner for Web, CLI and SFTP access in support of NERC CIP requirements.

Display of the banner can be enabled or disabled. When upgrading from a previous software release to v2.1.0, the default mode will be “disabled” in order to preserve the behavior of the previous release. When a new v2.1.0 system is booted or the configuration of a v2.1 system is set to defaults, the default mode will be “enabled”.

The banner consists of text (up to a maximum of 750 characters) that can be customized by the system administrator. The default text reads:

This system is for the use of authorized users only.
Individuals using this system are subject to having their
activities monitored and recorded by authorized company personnel.

Anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of criminal activity, company personnel may provide the evidence of such monitoring to enforcement officials.

2.3.1 “Appropriate Use Banner” CLI commands

Display of the banner can be enabled or disabled using:

Syntax:

```
session set banner mode <enabled|disabled>
```

Example:

```
MagnumDX# session set banner mode enabled
```

The banner text can be set using:

Syntax:

```
session set banner text
```

Example:

```
MagnumDX# session set banner text
```

After issuing this command, the user will be prompted with “Enter banner text (up to 750 characters). Use two blank lines to finish.” The banner text can then be entered, one line at a time. At the end, the user presses return twice and the banner is set.

The current banner text and mode can be displayed by using the “session show banner” command as shown below:

```
MagnumDX# session show banner
```

```
Banner Mode: Enabled
```

```
Banner Text:
```

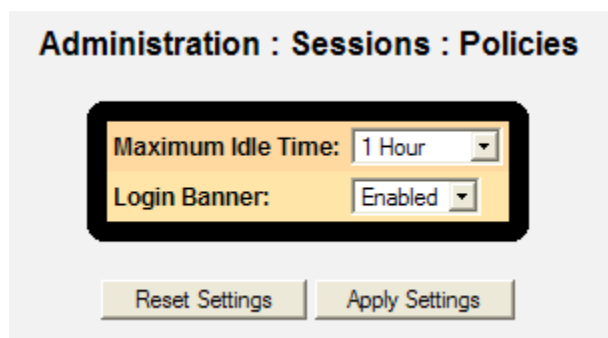
```
This system is for the use of authorized users only.  
Individuals using this system are subject to having their  
activities monitored and recorded by authorized company personnel.
```

```
Anyone using this system expressly consents to such monitoring and is  
advised that if such monitoring reveals possible evidence  
of criminal activity, company personnel may provide the evidence  
of such monitoring to enforcement officials.
```

2.3.2 “Appropriate Use Banner” Web interface

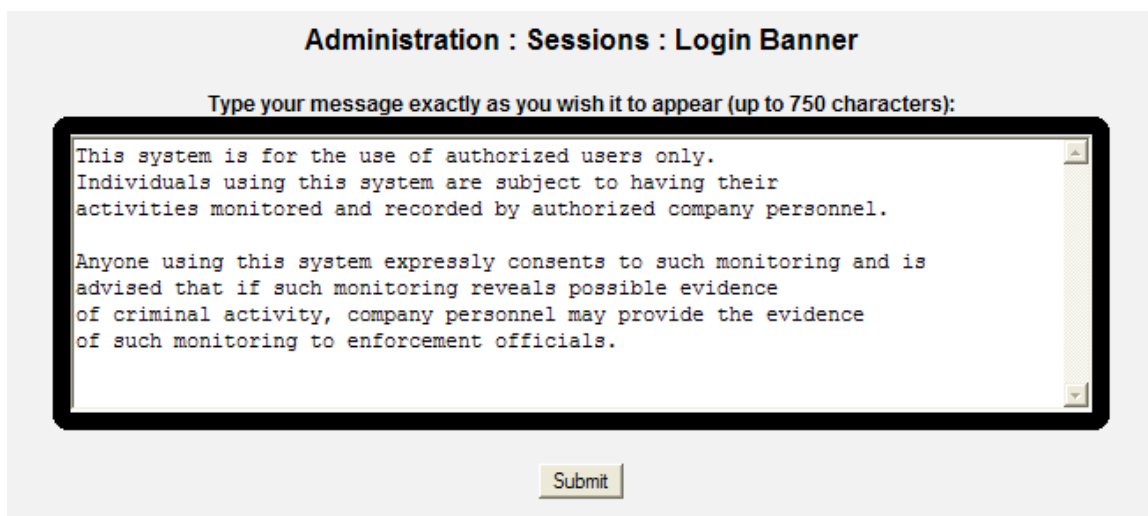
There are two enhancements to the web interface related to the Appropriate Use Banner operation.

The first change is to the existing “Administration : Sessions : Policies” page as follows:



The Login Banner parameter can be set to “Enabled”, in which case the banner will be displayed when a user connects to any of the supported user interfaces. If the parameter is set to “Disabled” no banner will be displayed.

The second change is the addition of a new page for setting the customizable banner text. This page can be found in the hierarchical menu as “Administration : Sessions : Login Banner”.



The user simply modifies the text in the box and presses the Submit button to apply the new text. The text will appear as shown (including new lines) when the banner is displayed to a connecting user.

2.4 DHCP Client

DHCP client functionality has been added to MNS-DX v2.1. A single IP interface on the device may be configured to receive its address via DHCP. The DHCP client cannot be enabled on multiple interfaces in this release.

When an IP interface is configured to receive its address via DHCP, the DHCP client process sends a DISCOVERY message and waits for an OFFER message from one or more DHCP servers. If no offers are received, it continues sending DHCP DISCOVERY messages. If an offer is received, it waits 5 seconds to see if any other offers are forthcoming. It then chooses

between the received offers and sends a REQUEST message to the chosen server. The server then responds with an ACK. At this point, the address has been assigned.

The initial state of the DHCP client is “Disabled”. If an IP interface is configured for DHCP, the state changes to “Pending” and remains in this state until a DHCP exchange is completed. At that point, the state changes to “Bound” and the address information that was received is available to be displayed to the user. In addition, the following actions are taken:

- IP address and netmask are assigned to the interface in the IP table
- If the DHCP server sent router information, a default route is created using that router address as the next hop
- If the DHCP server sent DNS server/domain information, this information is cached. If the DX is also acting as a DHCP server to downstream clients, this DNS information will be passed along if the DHCP server has been configured with the “Default” profile. If the DHCP server is using a custom profile, the DNS information is statically defined in that profile and so the cached information is not used.

The DHCP client status also displays the time at which the current lease was obtained as well as the time at which the lease will expire. When a lease is about to expire, the DHCP client will automatically try to renew it. If the DHCP client cannot renew the lease (e.g. no acceptable response from a DHCP server), the lease expires and the DHCP information is invalidated (including removing the DHCP IP address from the IP address table). After the lease expires, the DHCP client will solicit new offers by broadcasting a DISCOVERY message.

If the DHCP client does not get a response from a server, it will send 5 DISCOVERY retry messages with exponential back-off, starting at a retry interval of 2 seconds. If none of these messages are answered with an OFFER, a DHCP_TIMEOUT event is logged and the client restarts the discovery process. This means that it will take up to 1 minute for the DHCP client to time out and log an event when there is no server available.

If the DHCP client receives a response from a server but the assigned IP address conflicts with an IP already configured on the DX (i.e. no two interfaces can be assigned IP addresses that are on the same subnet), the DHCP client will attempt to get a better offer. If it cannot, it will accept the conflicting offer and log a DHCP_CONFLICT event. The conflicting IP address will not be added to the IP address table but it will show up in the DHCP client status display. To resolve this situation, the user must re-configure the network DHCP server or the reconfigure the IP addresses on the DX and then request a DHCP client renewal.

The user has the ability to manually request that DHCP renew its lease. Unlike an automatic renewal, a manual renewal will cause the client to release the current lease and attempt to negotiate a new lease from scratch with any available DHCP servers.

2.4.1 “DHCP Client” CLI

DHCP can be enabled on an interface using the following command:

```
ip set dhcp <interface> <y|n>
```

The DHCP client status can be checked using the following command:

```
dhcp show client
```

```
State : Bound
Client IP : 192.168.1.143
Netmask : 255.255.255.0
Router : 192.168.1.45
Primary DNS : 208.67.222.222
Secondary DNS :
Domain : garrettcom.com
Lease Origin : Fri Nov 13 13:01:57 2009
Lease Expire : Fri Nov 13 13:02:57 2009
```

Manual lease renewal can be achieved by executing the following command:

```
dhcp renew
```

2.4.2 “DHCP Client” Web Interface

DHCP can be enabled on a single IP interface using the IP address table page:

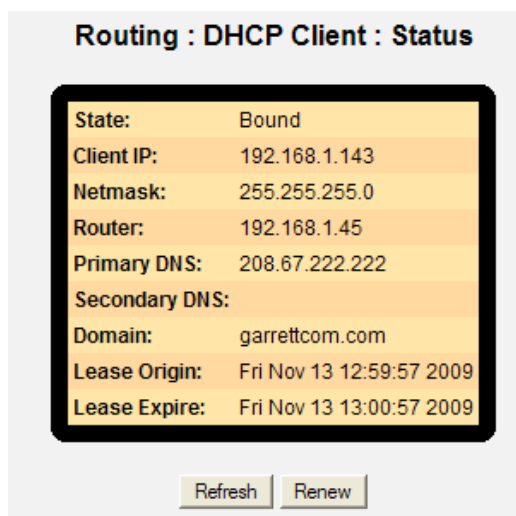
Routing : IP Addresses

Interface	DHCP?	Address	Subnet Mask	Remote Address	System	Status
Default	No	192.168.2.2	255.255.255.0		<input checked="" type="radio"/>	Down
E1	Yes	192.168.1.143	255.255.255.0		<input type="radio"/>	Up

[Other Options](#)

3.0

DHCP client status can be checked at page “Routing:DHCP Client: Status” as follows:



2.5 Configurable Administrative Route Distance

MNS-DX v2.1 software supports the concept of Administrative Distance (AD) as described and implemented by Cisco. The AD is a way of setting the preference/weight/priority of routes learned from different sources. The AD may be set for each routing protocol as well as for each individual static route. The default ADs used by DX are as follows:

- Local (Direct) Route – 0
- Static Route – 1
- VPN AutoRoute – 10 (not implemented by Cisco)
- eBGP – 20
- OSPF – 110
- RIP – 120
- iBGP – 200

Important Note: If there are two routes to the same destination in the DX routing table, the route with the lower AD is preferred.

In addition to the routes supported by Cisco, a feature of MNS-DX is that it automatically inserts routes to remote virtual private networks into the route table. These routes are derived from the current route table information and the address of the remote security gateway. VPN routes are assigned their own AD, which is configurable, but is set to 10 by default.

A common use of AD is to create “static floating routes” that support backup paths over ephemeral links such as dial-up connections. A typical scenario consists of a router with two IP interfaces, a primary and a backup. The primary interface is running OSPF and is connected to a permanent WAN link such as a T1. The backup interface is not running a routing protocol and is typically connected via a lower speed and/or higher cost technology such as dial-up or wireless. A static route is used to specify that a valid path exists over the

backup link. If the AD of the static route is higher than the AD of OSPF, then the primary path (OSPF routes) will always be preferred as long as that path is valid. If the primary path goes down, the OSPF routes will be removed from the route table and the backup path will be used.

Administrative Distance may be configured at any time. Changes to static and VPN routes take effect immediately. Changes to RIP or OSPF restart the routing process and so require these protocols to relearn the routes. Once the routes are relearned, they will appear in the route table with the new AD.

Route table entries with the same destination network are sorted by AD. The route with the lowest AD will appear first in the table and will be used for IP forwarding to that destination.

2.5.1 “Administrative Distance” CLI

This section shows the extension of existing CLI command trees to handle the distance parameter.

```
ospf
  set
    distance
      <1-255>

rip
  set
    distance
      <1-255>

ip
  add
    route
      <A.B.C.D> : destination network
      <A.B.C.D> : net mask
      <A.B.C.D> : next hop
      [distance <1-255>]

vpn
  set
    distance
      <1-255>

ospf show settings
```

```
Enabled? : Yes
Router ID : 0.0.0.2
```

AS Border Router? : No
Import BGP Routes? : No
Default BGP Route Metric : 1
Import RIP Routes? : No
Default RIP Route Metric : 20
Import Static Routes? : No
Default Static Route Metric : 20
Administrative Distance : 110

2.5.2 “Administrative Distance” Web Interface

The following changes have been made to the web management pages to support AD:

- Add “Administrative Distance” field to page “Routing : OSPF : Global Settings”
 - Default value is “110”
- Add “Administrative Distance” field to page “Routing : RIP : Global Settings”
 - Default value is “120”
- Add “Administrative Distance” column in page “Routing : Static Routes” table
 - Default value is “1”
- Add “Administrative Distance” column in page “Routing : Table” table
- Add “Administrative Distance” field to page “Security : VPN : Global Settings”
 - Default value is “10”

2.6 BGP MD5 Authentication

BGP sessions can be protected from tampering and spoofing by using TCP MD5 authentication. This type of packet authentication is specified in RFC 2385: “Protection of BGP Sessions via the TCP MD5 Signature Option”.

BGP MD5 authentication is enabled simply by specifying a password in the “MD5 Password” field of a particular peer in the Peer Settings table via the Web interface on page “Routing: BGP: Peer Settings” or through the CLI via the BGP “add peer” and edit peer” commands.

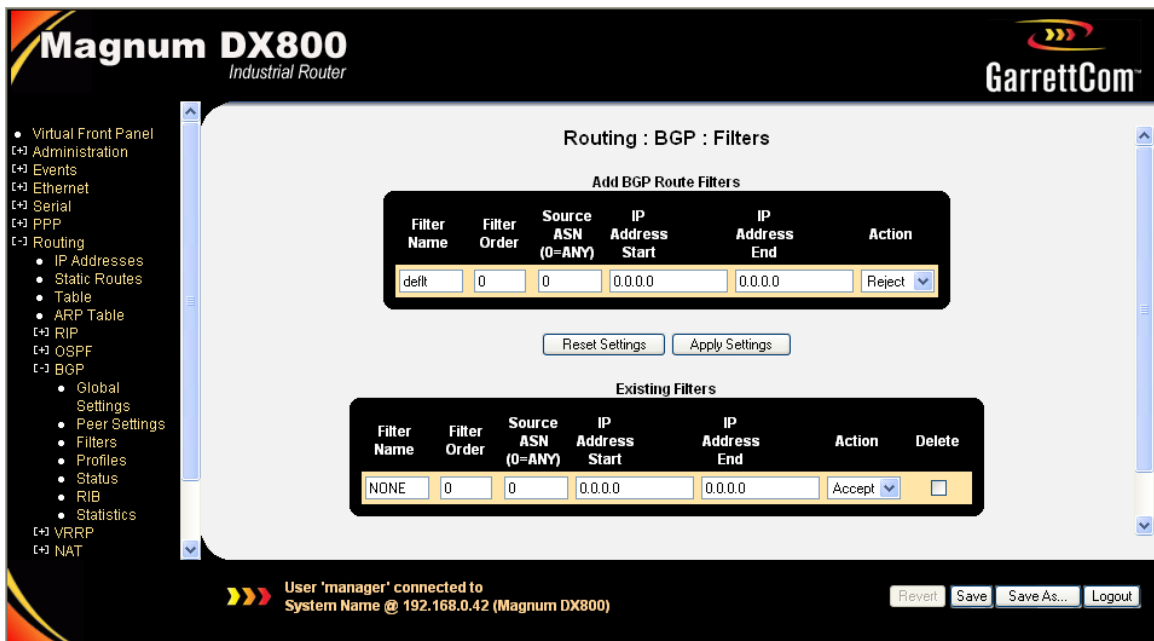
2.7 BGP Filters

BGP Input and Output filters refer to filters applied to BGP updates traveling into and out of the DX router. BGP Input filters refers to BGP updates coming from an external peer into the DX router while BGP Output filters refers to BGP updates leaving the DX router for an external peer.

A BGP Filter consists of one or more set of rules defined under a common name. The name of the filter is then used as an Input or Output filter for a BGP peer. The name of the filter is specified in a “Peer Setting” under the parameter “Input Filter” or “Output Filter”. The names are selected in a drop down menu so that only valid names can be used. Thus before a filter can be used in a peer setting, it must be defined as a BGP Filter.

A user can define one or more filters in the profile list. A defined filter profile may or may not be used by any specific peer setting. On the other hand, a filter cannot be deleted if it is being used in any peer setting.

A BGP Filter is first defined as a profile under “Routing: BGP: Filters” as shown below:



The user specifies a Filter Name, Filter Order, Source ASN, IP Address Start, IP Address End and the Action. Each of these parameters are defined below.

Filter Name: The Filter Name is the name associated with a set of rules that can be used for either Input or Output Filters.

Filter Order: This is a number that specifies the position of the rule in a set of rules. For example if a specific filter has 10 rules, and another rule is to be inserted in the middle, the user types in 5 as the filter order. This reorders the rest of the rules behind it. This is discussed later in rule creation and manipulation.

Source ASN: This specifies the Autonomous System Number (ASN) that the rule is to be applied to. An ASN of 0 specifies any ASN, and that rule is applied to ALL ASNs. Otherwise if a non-zero ASN is specified, then the rule is applied only to that ASN.

IP Address Start and IP Address End: This specifies the range of IP addresses that a rule is applied to. For example, 10.0.0.0 (Start Address) – 10.255.255.255 (End Address) would cause this instance of a rule to be applied to all IP addresses with a starting prefix of 10.

Action: Specifies whether to ACCEPT or REJECT the set of IP Addresses as defined by the previous set of parameters.

A filter with one rule is a simple filter. When more than one rule is entered under a common name, we refer to this as a complex filter.

For a simple filter a single rule is applied to a BGP prefix. The result of the filter application is an ACCEPT or REJECT. If ACCEPT then the prefix is inserted in the BGP route table otherwise it is not inserted into the BGP route table.

For a complex filter, multiple rules are applied to a BGP prefix. Each application results in an ACCEPT or REJECT.

The rules are applied sequentially to a specific prefix starting with rule 0. Once all the rules have been processed the result is an ACCEPT or REJECT of that net.

An ACCEPT means that that net is inserted into the BGP route table, a REJECT means that that net is NOT inserted into the BGP route table.

The default behavior is to ACCEPT. If a rule is not able to be applied to a specified net then the behavior is to ACCEPT that net.

The best way to describe the operation of BGP Rules operation is by example. Let's say the user wants to reject ALL network addresses with a prefix of 10 EXCEPT for 10.1.1.0. Also assume that this rule should apply to ALL ASNs. Since this is a complex filter, we need to specify a couple of rules.

- 1) Give it a name, Filt1.
- 2) First rule, so the filter order is 0
- 3) ASN is 0 since we want to apply this to ALL ASNs.
- 4) Start IP Address is 10.0.0.0
- 5) End IP Address is 10.255.255.255
- 6) Action is REJECT

This rule rejects ALL IP addresses that start with 10. Now we want to create a rule that would allow 10.1.1.0 to be accepted by the system. So a second rule is entered,

- 1) Same name as used previously, Filt1
- 2) This is the second rule, so filter order is 1
- 3) ASN is 0 as before
- 4) Start IP Address is 10.1.1.0
- 5) End IP Address is 10.1.1.0 also
- 6) Action is ACCEPT

With these two rules all IP addresses starting with 10 would be REJECTED except for 10.1.1.0 which would be accepted. This is shown below:

The screenshot shows the configuration interface for a Magnum DX800 Industrial Router. The main area displays the configuration for BGP filters. There are two sections: 'ADD NEW BGP FILTERS' and 'Existing Filters'.

ADD NEW BGP FILTERS

Filter Name	Filter Order	Source ASN (0=ANY)	IP Address Start	IP Address End	Action
Filt1	1	0	10.1.1.0	10.1.1.0	Accept

Existing Filters

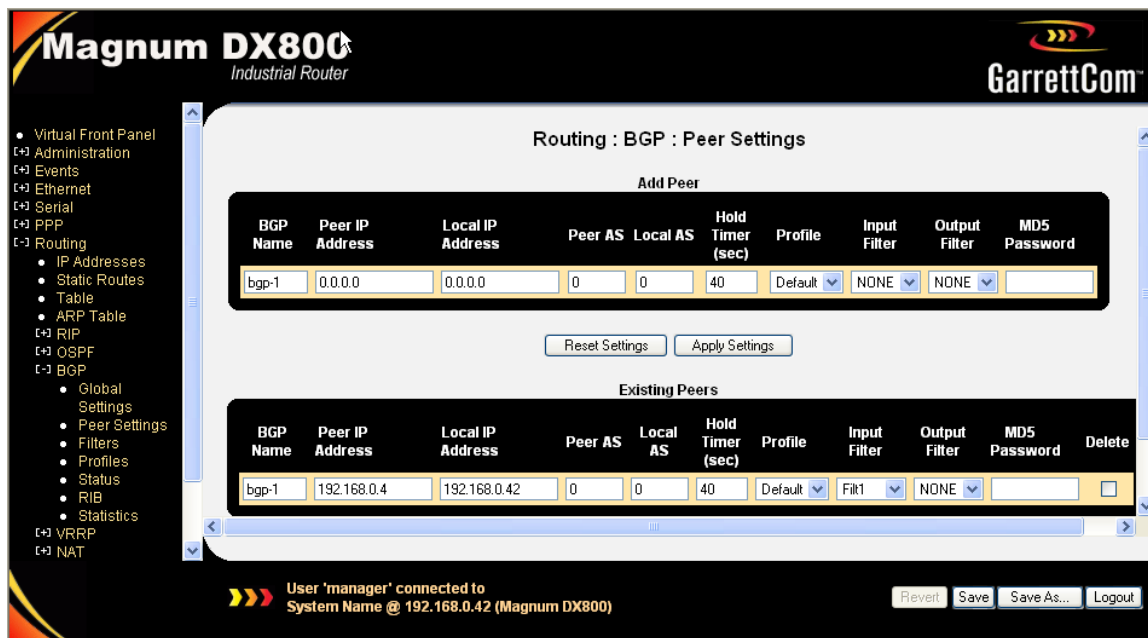
Filter Name	Filter Order	Source ASN (0=ANY)	IP Address Start	IP Address End	Action	Delete
NONE	0	0	0.0.0.0	0.0.0.0	Accept	<input type="checkbox"/>
Filt1	0	0	10.0.0.0	10.255.255.255	Reject	<input type="checkbox"/>
Filt1	1	0	10.1.1.0	10.1.1.0	Accept	<input type="checkbox"/>

The interface also shows a navigation menu on the left with options like Administration, Events, Ethernet, Serial, PPP, Routing, IP Addresses, Static Routes, Table, ARP Table, RIP, OSPF, BGP, Global Settings, Peer Settings, Filters, Profiles, Status, RIB, Statistics, VRRP, and NAT. At the bottom, it indicates 'User 'manager' connected to System Name @ 192.168.0.42 (Magnum DX800)' and provides buttons for Revert, Save, Save As..., and Logout.

The default behavior of the system is to ACCEPT thus all other nets would be accepted,

i.e. all nets starting with a number other than 10. Below is the filter profile describing for the DX. Of course, this filter profile must be applied to a specified BGP peer. This is done in the peer setting.

The figure below shows the filter name specified in the peer parameter. The name “Filt1” is inserted under the Input Filter parameter. This specifies to use the filter “Filt1” as the BGP Input Filter and use the rules as defined in that filter profile for all BGP updates from BGP peer “bgp-1”.



2.8 Frame Relay end-to-end Keep-alive

The end-to-end keep-alive (EEK) feature operates over frame relay WAN links.

EEK implements a keep-alive mechanism that uses a simple request and response model. Either side or both sides can be configured to send EEK requests. When an EEK request is received, an EEK response is sent.

A DLCI may be placed in one of five EEK modes:

1. Disabled – EEK is disabled
2. Bidirectional – the device sends EEK requests and also responds to received EEK requests
3. Request – The device only sends EEK requests. It does not respond to received EEK requests
4. Reply – The device only replies to received EEK requests. It does not send EEK requests.
5. Passive-Reply – The device only responds to received EEK requests. It does not set any timers or keep track of any events.

The following EEK parameters may be configured on a per-port basis. Window, Errors, and Successes are applied to both the send and receive side EEK processes.

1. Request Timer – The number of seconds to wait before sending the next EEK request
2. Receive Timer – The number of seconds to wait for an EEK request before adding a receive error event to the EEK window
3. Window – The size of a sliding window containing the total number of EEK events that will be examined to determine if an end-to-end path is up or down.
4. Errors – The number of error events in the EEK window that cause the path to be marked as down.
5. Successes – The number of consecutive successful events required to mark the path as up.

The EEK status for each DLCI can be monitored by the user. The EEK status consists of the following variables:

1. State – The current EEK state of the DLCI. Valid states are: Disabled (EEK is disabled for this DLCI), Up, Dn-Snd (down due to send side errors), Dn-Rcv (down due to receive side errors), Dn-S/R (down due to both send and receive errors)
2. Total Send Events – the total number of EEK request polls that have been sent.
3. Total Receive Events – the total number of EEK request polls that have been received.
4. Send Error Events – the total number of send error events currently present in the EEK window
5. Receive Error Events – the total number of receive error events currently present in the EEK window
6. Consecutive Send Successes – the current number of consecutive send success events.
7. Consecutive Receive Successes – the current number of consecutive receive success events

The values are reset to 0 when a DLCI becomes active, or when a DLCI's EEK setting is changed.

2.8.1 “EEK” CLI

Some new commands have added for EEK as well as additional fields to existing commands.

```
fr set eek
    <W1|...>
    [request-timer <1-255>] : seconds
    [receive-timer <1-255>] : seconds
    [window <1-32>] : events
    [errors <1-32>] : events
    [successes <1-32>] : events

fr add dlc
    <W1|...>
    <1-1022> : circuit identifier
    [eek <none|bi-dir|request|reply|passive-reply>]
```

```

fr edit dlcI
    <W1|...>
    <1-1022> : circuit identifier
    [eek <none|bi-dir|request|reply|passive-reply>]

```

fr show eek settings

```

Port Request Receive
ID  Timer   Timer   Window Errors Successes
==== =====
W1  10      15      3      2      2

```

fr show eek status

```

Port
ID  DLCI  State      Total Send Total Recv Send      Receive Consec  Consec
ID  DLCI  State      Events     Events     Err Ev   Err Ev   Snd Suc Rcv Suc
==== =====
W1  100  Dn-Rcv    12         7         0       3       9      0

```

fr show dlcI settings

```

Port
ID  DLCI  CIR      IP  EEK
==== =====
W1  100          Yes Bi-dir
W1  200          Yes None

```

fr show dlcI status

```

Port
ID  DLCI  State      Rx      Rx      Tx      Tx      Drops
ID  DLCI  State      Packets Octets  Packets Octets
==== =====
W1  100  EEK TO    113    1243   114    1254    0
W1  200  Active     0       0      0       0      0

```

2.8.2 “EEK” Web Interface

This section describes enhancements to the web pages required to support the addition of EEK.

A new web page, “WAN : EEK Settings” has been added to configure the per-port EEK parameters.

WAN : EEK Settings

Port ID	Request Timer	Receive Timer	Window	Errors	Successes
W1	10	15	3	2	2

A new web page, “WAN : EEK Status” has been added to provide counters and also the current state of the EEK event window.

WAN : EEK Status

Port ID	DLCI	State	Total Send Events	Total Receive Events	Send Error Events	Receive Error Events	Consec Send Successes	Consec Receive Successes
W1	100	Up	0	0	0	0	0	0

A new column, “EEK”, has been added to the DLCI table on the “WAN : DLCI Settings” page to configure the EEK mode for the DLCI. The possible choices are:

1. None
2. Bi-Dir (Bidirectional)
3. Request (Send requests only)
4. Reply (Reply to requests only)
5. Passive (reply to requests if present, but don't count errors if requests are not seen)

Finally, a DLCI shown in the “WAN : DLCI Status” page may now be in a state of “EEK TO” if EEK is enabled and has timed out.

2.9 PPP over WAN

MNS-DX v2.1.0 software supports PPP operation over WAN links.

A DDS or T1/E1 port provides a raw, physical layer communication capability that requires some type of higher-level framing in order to carry packetized data. Starting in version 1.4, MNS-DX supported frame relay (HDLC) as a basic framing mechanism and the encapsulation technique described in RFC 1490 as a way of carrying IP traffic of a WAN

link. In version 2.1.0, MNS-DX supports PPP (with HDLC-like framing) as an alternative for carrying IP traffic over a WAN.

A WAN port can be configured for either frame relay or for PPP operation. A WAN link cannot support both protocols simultaneously. If a DLCI or LMI is configured on a WAN port, then the port is automatically set up for frame relay operation. If a PPP connection is configured on a WAN port, then the port is automatically set up for PPP operation. If a user attempts to configure a DLCI on a PPP-enabled WAN port or conversely, a PPP connection on a frame relay-enabled WAN port, an error will be returned by the software indicating the port is already in use.

To configure PPP over WAN, the user should do the following:

1. Administratively enable the desired WAN port and configure any appropriate port settings (e.g. whether to use the local or received clock for timing).
2. Add a new PPP connection. Specify the WAN port, the appropriate PPP profile, and the username/password for authentication if necessary. The default WAN PPP profile can be used if CHAP authentication is desired.
3. Set the local and remote IP address of the PPP connection in the IP address table. The local network mask should be set to 255.255.255.255. The remote IP address should be the IP address assigned to the other side of the PPP connection.

A default PPP profile for PPP over WAN is automatically added to the PPP profile table upon initial system startup (or whenever the system is booted with zero profiles defined). The default PPP profile for WAN has the following parameters:

- Name – WAN
- LCP Echo Interval – 30
- Auth Type – None
- Assign IP – No
- Use Hayes Modem – No
- Match Speed – No
- Compress TCP Headers – Yes
- Modem Init String - Empty

2.10 Configurable Event Specifications

The event management capabilities of DX have been enhanced in v2.1.0.

Each event type generated by the system now has a unique identifier that consists of an event category and an event number within that category. When an Event ID is displayed to or referenced by a user, it has the form X-Y, where X is the event category and Y is the event number within the category.

Each event type has following attributes association with it:

- Severity level (0-7)
- Remote Logging Mode: Disabled, Syslog
- Local Logging Mode: Disabled, Volatile, Persistent

- Local Target Log

Event severity levels match the levels defined for Syslog as shown:

- 0 – emergencies
- 1 – alerts
- 2 – critical
- 3 – errors
- 4 – warnings
- 5 – notifications
- 6 – information
- 7 - debugging

If Remote Logging is enabled for a particular event type, when that event is generated it will be sent to all configured Syslog collectors.

If Local Logging is set to Volatile or Persistent, when that event is generated it will be written to the active log file (in RAM) for the configured Target Log.

If Local Logging is set to Persistent, when generated, that particular event will additionally be written to a temporary RAM log buffer associated with the Target Log. If at any time the current log file is closed so that a new log file can be opened, the temporary log buffer is written to a new file in flash memory. Files written in this way will have their status marked as “Saved” and will survive system reboots and power-cycles.

As an example, when an “warm start” event is logged by the Syslogger, the text will have the following form:

```
<2>Aug 13 10:20:34 2009 90.0.0.1 %WARM_START-1-2: Warm start, software version: 2.1.0,  
config: 'config0.xml'
```

When the same event is logged to a file, the IP address is omitted:

```
<2>Aug 13 10:20:34 2009 %WARM_START-1-2: Warm start, software version: 2.1.0, config:  
'config0.xml'
```

Event Categories

The following table is a list of defined event categories:

1	SYSTEM
2	POWER SUPPLY
3	LOG MANAGEMENT
4	SOFTWARE MANAGEMENT
5	CONFIGURATION MANAGEMENT
6	AUTHENTICATION
7	SESSIONS
8	PHYSICAL LINK
9	IP INTERFACE
10	PORT SECURITY
11	FIREWALL
12	TERMINAL SERVER
13	RSTP
14	OSPF
15	BGP
16	VRRP
17	PPP
18	VPN
19	CERTIFICATE MANAGEMENT

Event Descriptions

EVENT_ID	EVENT_TAG	EXAMPLE DESCRIPTIVE TEXT
1-1	COLD_BOOT	<i>Cold start, software version: 2.1.0, config: 'config0.xml'</i>
1-2	WARM_BOOT	<i>Warm start, software version: 2.1.0, config: 'config0.xml'</i>
2-1	PS_DOWN	<i>Power Supply PS1 is down.</i>
2-2	PS_UP	<i>Power Supply PS2 is up.</i>
3-1	LOG_CREATE	<i>Log file 'Default-20090819-154502.log' was created.</i>
3-2	LOG_DELETE	<i>Log file 'Default-20090819-154502.log' was deleted by user 'manager'.</i>
3-3	LOG_AUTO_DELETE	<i>Log file 'Default-20090819-154502.log' was automatically deleted.</i>
4-1	SOFTWARE_UPLOAD	<i>Software image version 2.1.0 uploaded by user 'manager'.</i>
4-2	SOFTWARE_FINALIZED	<i>Upgrade finalized by user 'manager', current version: 2.1.0, fallback: 2.0.1</i>
5-1	CONFIG_SWITCH	<i>Config switched to 'config1.xml' by user 'manager'.</i>
5-2	CONFIG_UPLOAD	<i>Config 'config10.xml' uploaded by user 'manager'.</i>
5-3	CONFIG_DELETE	<i>Config 'config10.xml' was deleted by user 'manager'.</i>
5-4	CONFIG_AUTO_DELETE	<i>Config 'config15.xml' was automatically deleted.</i>
5-5	CONFIG_SAVE	<i>Config 'config20.xml' was saved by user 'manager'.</i>
5-6	CONFIG_CHANGE	<i>Current config table 'Ethernet/PortSettingsTable' was changed by user 'manager'.</i>
5-7	CONFIG_RESTORE	<i>Config was restored to defaults by user 'manager'.</i>
6-1	PASSWORD_CHANGE	<i>Password was changed for user 'maint1'.</i>
6-2	USER_DELETE	<i>User 'maint1' was deleted.</i>
6-3	MAX_USERS	<i>Maximum number of users reached.</i>
6-4	NEW_USER	<i>New user 'tech' was created.</i>
6-5	USER_SUSPENDED	<i>User 'tech' was suspended.</i>
6-6	SUSPENSION_LAPSED	<i>Suspension timeout has elapsed for user 'tech'.</i>
6-7	PASSWORD_EXPIRED	<i>User 'maint1' password expired.</i>
6-8	INVALID_USERNAME	<i>Login attempt via SSH (192.168.1.2) with invalid username 'trythisuser'.</i>

6-9	INVALID_PASSWORD	<i>Login attempt via TELNET (192.168.2.3) with username 'someuser' and invalid password 'password123'.</i>
6-10	HACKING_ATTEMPT	<i>Possible hacking attempt, 20 failed login attempts in 5 minutes.</i>
7-1	LOGIN	<i>User 'manager' logged in via HTTPS (192.168.1.42).</i>
7-2	LOGOUT	<i>User 'manager' logged out via HTTPS (192.168.1.42).</i>
7-3	IDLED_OUT	<i>User 'manager' idled out via HTTPS (192.168.1.42).</i>
7-4	CONSOLE_DISCONNECT	<i>User 'manager' disconnected from CONSOLE and was logged out.</i>
8-1	ETHERNET_DOWN	<i>Ethernet port E2 is down.</i>
8-2	ETHERNET_UP	<i>Ethernet port E2 is up.</i>
8-3	SERIAL_DOWN	<i>Serial port S3 is down.</i>
8-4	SERIAL_UP	<i>Serial port S3 is up.</i>
8-5	WAN_DOWN	<i>WAN port W1 is down.</i>
8-6	WAN_UP	<i>WAN port W2 is up.</i>
9-1	INTERFACE_DOWN	<i>IP interface Default is down.</i>
9-2	INTERFACE_UP	<i>IP interface E3 is up.</i>
10-1	PORT_LOCKED	<i>Ethernet port E4 has been locked out by port security.</i>
11-1	MAX_PERMIT_FLOWS	<i>Warning – Maximum permitted flows (100) are being tracked.</i>
11-2	MAX_DENIED_FLOWS	<i>Warning – Maximum denied flows (100) are being tracked.</i>
11-3	TCP_START	<i>TCP 192.168.1.1 (43532) -> 192.168.2.2 (23), Session started</i>
11-4	TCP_ESTAB	<i>TCP 192.168.1.1 (43532) -> 192.168.2.2 (23), Session established</i>
11-5	TCP_UPDATE	<i>TCP 192.168.1.1 (43532) -> 192.168.2.2 (23), 20 packets.</i>
11-6	TCP_END	<i>TCP 192.168.1.1 (43532) -> 192.168.2.2 (23), Session closed.</i>
11-7	TCP_DENIED	<i>TCP Denied 192.168.1.1 (43532) -> 192.168.2.2 (23), First packet.</i>
11-8	TCP_DENIED_UPDATE	<i>TCP Denied 192.168.1.1 (43532) -> 192.168.2.2 (23), 100 packets in 120 second interval.</i>
11-9	UDP_START	<i>UDP 192.168.1.1 (43532) -> 192.168.2.2 (123), Session started</i>
11-10	UDP_UPDATE	<i>UDP 192.168.1.1 (43532) -> 192.168.2.2 (123), 10 packets.</i>
11-11	UDP_DENIED	<i>UDP Denied 192.168.1.1 (43532) -></i>

		<i>192.168.2.2 (123), First packet.</i>
11-12	UDP_DENIED_UPDATE	<i>UDP Denied 192.168.1.1 (43532) -> 192.168.2.2 (123), 52 packets in 60 seconds.</i>
11-13	ICMP_START	<i>ICMP 192.168.1.1 (43532) -> 192.168.2.2 (123), Session started</i>
11-14	ICMP_UPDATE	<i>ICMP 192.168.1.1 (43532) -> 192.168.2.2 (123), 10 packets.</i>
11-15	ICMP_DENIED	<i>ICMP Denied 192.168.1.1 (43532) -> 192.168.2.2 (123), First packet.</i>
11-16	ICMP_DENIED_UPDATE	<i>ICMP Denied 192.168.1.1 (43532) -> 192.168.2.2 (123), 16 packets in 60 seconds.</i>
12-1	TS_HOST_UNREACH	<i>Serial port S1 reports that the host at 192.168.1.1 is unreachable.</i>
12-2	TS_HOST_DOWN	<i>Serial port S1 reports that the host at 192.168.1.1 is down.</i>
12-3	TS_CONN_REFUSED	<i>Serial port S1 reports that the connection to the host at 192.168.1.1 (10232) was refused.</i>
12-4	TS_LOST_CONNECTION	<i>Serial port S1 lost connection with host at 192.168.1.1 (10232).</i>
12-5	TS_NO_SSL	<i>Serial port S1 reports that the host at 192.168.1.1 (10232) did not respond to the SSL handshake.</i>
12-6	TS_SSL_NOTICE	<i>Serial port S1 received a notification (<notification text>) from the host at 192.168.1.1 (10232).</i>
12-7	TS_SSL_PROBLEM	<i>Serial port S1 experienced a problem (<problem-text>) while connecting to the host at 192.168.1.1 (10232).</i>
12-8	TS_SSL_CERT_INVALID	<i>Serial port S1 reports that the certificate presented by the host at 192.168.1.1 (10232) was invalid.</i>
13-1	RSTP_NON_EDGE_DOWN	<i>RSTP link down on non-edge port</i>
13-2	RSTP_SENT_TCN	<i>RSTP sent topology change notice</i>
14-1	OSPF_NBR_2WAY	<i>OSPF neighbor 192.168.3.2 transitioned to 2WAY.</i>
14-2	OSPF_NBR_FULL	<i>OSPF neighbor 192.168.3.2 transitioned to FULL.</i>
14-3	OSPF_NBR_DOWN	<i>OSPF neighbor 192.168.3.2 transitioned to DOWN.</i>
15-1	BGP_PEER_ESTAB	<i>BGP setting state to ESTAB for 192.168.5.9.</i>
15-2	BGP_KA_TIMEOUT	<i>BGP keepalive timeout disconnect for 192.168.5.9.</i>

16-1	VRRP_MASTER	<i>VRRP 2 transitioned to MASTER.</i>
16-2	VRRP_BACKUP	<i>VRRP 2 transitioned to BACKUP.</i>
17-1	PPP_HANGUP	<i>PPP PPP-S2 is hanging up.</i>
17-2	PPP_CONNECT	<i>PPP PPP-S2 has connected.</i>
17-3	PPP_SPEED_CHANGE	<i>PPP PPP-S2 changing serial speed to 1200.</i>
18-1	VPN_PHASE_1_SUCCESS	<i>VPN Src: 192.168.1.1 Dst: 192.168.1.2 IKE Phase I Success.</i>
18-2	VPN_PHASE_2_SUCCESS	<i>VPN Src: 192.168.1.42 Dst: 192.168.3.3 IKE Phase II Success.</i>
18-3	VPN_DEAD_PEER	<i>VPN detected dead peer: 192.168.1.2</i>
19-1	CERT_CREATE	<i>Certificate 'newcert.pem' was created by user 'manager'.</i>
19-2	CERT_DELETE	<i>Certificate 'newcert.pem' was deleted by user 'manager'.</i>
19-3	CERT_UPLOAD	<i>Certificate 'newcert.pem' was uploaded by user 'manager'.</i>
19-4	CERT_TRUST	<i>Certificate 'mycert.pem' was marked as trusted by user 'manager'.</i>
19-5	CERT_UNTRUST	<i>Certificate 'mycert.pem' was marked as untrusted by user 'manager'.</i>

Event Defaults

The following table is a list of defined event IDs and default specification values:

EVENT_ID	EVENT_TAG	DEFAULT SEVERITY	DEFAULT TARGET LOG	DEFAULT VOLATILITY
1-1	COLD_BOOT	2	DEFAULT	NV
1-2	WARM_BOOT	2	DEFAULT	NV
2-1	PS_DOWN	1	DEFAULT	NV
2-2	PS_UP	2	DEFAULT	NV
3-1	LOG_CREATE	5	DEFAULT	NV
3-2	LOG_DELETE	4	DEFAULT	NV
3-3	LOG_AUTO_DELETE	4	DEFAULT	NV
4-1	SOFTWARE_UPLOAD	6	DEFAULT	NV
4-2	SOFTWARE_FINALIZED	5	DEFAULT	NV
5-1	CONFIG_SWITCH	4	DEFAULT	NV
5-2	CONFIG_UPLOAD	6	DEFAULT	NV
5-3	CONFIG_DELETE	4	DEFAULT	NV
5-4	CONFIG_AUTO_DELETE	5	DEFAULT	NV
5-5	CONFIG_SAVE	4	DEFAULT	NV
5-6	CONFIG_CHANGE	4	DEFAULT	NV
5-7	CONFIG_RESTORE	4	DEFAULT	NV

6-1	PASSWORD_CHANGE	4	SECURITY	NV
6-2	USER_DELETE	2	SECURITY	NV
6-3	MAX_USERS	3	SECURITY	NV
6-4	NEW_USER	2	SECURITY	NV
6-5	USER_SUSPENDED	2	SECURITY	NV
6-6	SUSPENSION_LAPSED	4	SECURITY	NV
6-7	PASSWORD_EXPIRED	5	SECURITY	NV
6-8	INVALID_USERNAME	1	SECURITY	NV
6-9	INVALID_PASSWORD	1	SECURITY	NV
6-10	HACKING_ATTEMPT	1	SECURITY	NV
7-1	LOGIN	2	SECURITY	NV
7-2	LOGOUT	2	SECURITY	NV
7-3	IDLED_OUT	2	SECURITY	NV
7-4	CONSOLE_DISCONNECT	2	SECURITY	NV
8-1	ETHERNET_DOWN	2	DEFAULT	NV
8-2	ETHERNET_UP	2	DEFAULT	NV
8-3	SERIAL_DOWN	2	DEFAULT	NV
8-4	SERIAL_UP	2	DEFAULT	NV
8-5	WAN_DOWN	2	DEFAULT	NV
8-6	WAN_UP	2	DEFAULT	NV
9-1	INTERFACE_DOWN	2	DEFAULT	NV
9-2	INTERFACE_UP	2	DEFAULT	NV
10-1	PORT_LOCKED	2	DEFAULT	NV
11-1	MAX_PERMIT_FLOWS	2	FIREWALL	V
11-2	MAX_DENIED_FLOWS	2	FIREWALL	V
11-3	TCP_START	5	FIREWALL	NV
11-4	TCP_ESTAB	5	FIREWALL	NV
11-5	TCP_UPDATE	5	FIREWALL	NV
11-6	TCP_END	5	FIREWALL	NV
11-7	TCP_DENIED	2	FIREWALL	V
11-8	TCP_DENIED_UPDATE	2	FIREWALL	V
11-9	UDP_START	5	FIREWALL	NV
11-10	UDP_UPDATE	5	FIREWALL	NV
11-11	UDP_DENIED	2	FIREWALL	V
11-12	UDP_DENIED_UPDATE	2	FIREWALL	V
11-13	ICMP_START	5	FIREWALL	NV
11-14	ICMP_UPDATE	5	FIREWALL	NV
11-15	ICMP_DENIED	2	FIREWALL	V
11-16	ICMP_DENIED_UPDATE	2	FIREWALL	V
12-1	TS_HOST_UNREACH	4	DEFAULT	NV
12-2	TS_HOST_DOWN	4	DEFAULT	NV
12-3	TS_CONN_REFUSED	4	DEFAULT	NV
12-4	TS_LOST_CONNECTION	3	DEFAULT	NV
12-5	TS_NO_SSL	4	DEFAULT	NV
12-6	TS_SSL_NOTICE	5	DEFAULT	NV

12-7	TS_SSL_PROBLEM	3	DEFAULT	NV
12-8	TS_SSL_CERT_INVALID	3	DEFAULT	NV
13-1	RSTP_NON_EDGE_DOWN	2	DEFAULT	NV
13-2	RSTP_SENT_TCN	2	DEFAULT	NV
14-1	OSPF_NBR_2WAY	5	DEFAULT	NV
14-2	OSPF_NBR_FULL	5	DEFAULT	NV
14-3	OSPF_NBR_DOWN	2	DEFAULT	NV
15-1	BGP_PEER_ESTAB	5	DEFAULT	NV
15-2	BGP_KA_TIMEOUT	2	DEFAULT	NV
16-1	VRRP_MASTER	4	DEFAULT	NV
16-2	VRRP_BACKUP	4	DEFAULT	NV
17-1	PPP_HANGUP	2	DEFAULT	NV
17-2	PPP_CONNECT	5	DEFAULT	NV
17-3	PPP_SPEED_CHANGE	5	DEFAULT	NV
18-1	VPN_PHASE_1_SUCCESS	5	DEFAULT	NV
18-2	VPN_PHASE_2_SUCCESS	5	DEFAULT	NV
18-3	VPN_DEAD_PEER	2	DEFAULT	NV
19-1	CERT_CREATE	5	DEFAULT	NV
19-2	CERT_DELETE	4	DEFAULT	NV
19-3	CERT_UPLOAD	5	DEFAULT	NV
19-4	CERT_TRUST	4	DEFAULT	NV
19-5	CERT_UNTRUST	4	DEFAULT	NV

2.10 Multiple Circular Event Logs

The user may configure multiple logs and different event types may be targeted to different logs.

Support for multiple logs allows the user to partition the log space so that an abundance of high frequency / low importance events do not overwhelm (or kick out) a few low frequency / high importance events. Events can also be logged according to their function. For example, a separate firewall log can be created to separate those events from general system events.

Each log has the following attributes associated with it:

- Create new file: daily, weekly, monthly
- Max files: maximum number of files to maintain for the log
- Max file size: a new log file is created once this limit is reached
- Overwrite old: if set, old log files are deleted to make room for new log files
- Rate Limit: number of events to log per second, over the limit are dropped

All events configured as volatile or non-volatile are logged to files stored in RAM. There is a total of 1 MB reserved to store volatile logs.

Log file names follow the convention “<logname>-<date>-<time>.log” where logname is the configured name of the log, date is when the log was created in YYMMDD format and the time is the when the log was created in HHMMSS format.

When an event needs to be logged in RAM, the following procedure is followed to determine a new file should be created:

- If there is no current log file, a new file is created
- If there is a current log file:
 - If the “Create new file” attribute is set to daily and the last event was logged on the previous day, create a new file
 - If the “Create new file” attribute is set to weekly and the last event was logged during the previous week, create a new file
 - If the “Create new file” attribute is set to monthly and the last event was logged during the previous month, create a new file
 - If the “Max file size” for the current file has been reached, create a new file

The following procedure is followed to manage the “circular” nature of an event log:

- If the maximum number of files exists:
 - If the “Overwrite old” attribute is set to no, do not create a new file
 - If the “Overwrite old” attribute is set to yes, delete the oldest file in that log and then create the new file
- Otherwise create the new file

If an event needs to be logged and there is no more room left:

- If the “Overwrite old” attribute is set to no, do not log the event
- If the “Overwrite old” attribute is set to yes, delete the oldest file in that log and then log the event

Each log also maintains a temporary buffer where persistent events are stored. When a new log file is created in RAM, the temporary buffer containing all of the persistent events that were logged in the previously active RAM file are written to an identically named log file in the flash file system.

When a persistent log file needs to be written to the flash file system and there is no more room left:

- If the “Overwrite old” attribute is set to no, do not log the event
- If the “Overwrite old” attribute is set to yes, delete the oldest file in that log and then create the new persistent file

When an old log file needs to be automatically removed by the system to make room for a new file, both the RAM and flash file system version of the file are deleted.

Log files stored in the flash file system will persist across system reboots and power cycles. When the system starts back up, these files will be available as “Saved” log files. Log files stored in RAM will be lost on a reboot.

2.11 Route Re-distribution

Full route re-distribution between BGP, OSPF and RIP has been added to release v2.1.0.

3.0 QUALITY ENHANCEMENTS

- 3.1 Interoperability issues with OSPF MD5 operation.
- 3.2 Fiber Ethernet port status would show down after a system reboot even when the port was up and active.
- 3.3 Editing settings of an existing Radius server did not take effect until after a system reboot.
- 3.4 The CLI used to fail silently or with an ambiguous error when given a bad interface. It now will return error message that "Interface does not exist".
- 3.5 IP interface disappears after setting IP address with bad mask.
- 3.6 Power Supply 1 LED would not illuminate on a single power supply DX1000 system.
- 3.7 Needed system reboot to get SNMP v1 or v2 to work after system was set to v3 operation.
- 3.8 OSPF default route disappears when port is unbridged.
- 3.9 VRRP master/backup preemption does not negotiate with the same priority.
- 3.10 Add VRRP events to log system.
- 3.11 VRRP group is not refreshed after an IP address change.
- 3.12 Wan port state not correctly reflected in virtual front panel display.
- 3.13 Added a timer for Ethernet port up condition to help prevent repeated log entries for Up/Down conditions on problem links.
- 3.14 IPsec: Host address selector not specified correctly in identity payload.
- 3.15 Added a RADIUS local fallback mode.
- 3.16 File transfer fails when using Bitwise SFTP client.
- 3.17 Firewall doesn't filter packets when enabled after configuration restore.

- 3.18 Downloading or viewing a configuration file with spaces in its name fails.
- 3.19 Warn user of the requirement to finalize after a software upgrade.
- 3.20 Added last modified time stamp to configuration file.
- 3.21 Problem deleting IPSec SA w/ mix of host and subnet addresses.
- 3.22 Telnet task runs continuously in a loop if telnet session is disconnected while in VPN trace mode.
- 3.23 DX1000 front panel WAN LEDs not functional.
- 3.24 UTM-1 edge IPsec interoperability.
- 3.25 Fixed host address problem in IPSec tunnel specifications.
- 3.26 Added CPU utilization readout.
- 3.27 Memory utilization readout never changes.
- 3.28 OSPF NSSA does not work.
- 3.29 Route information not exchanged during initial BGP connection.
- 3.30 IP interface up after setting address on unbridged port that is down.
- 3.31 Increased maximum number of NAT port forwarding rules to 32 for TCP and 32 for UDP.
- 3.32 Software image name containing spaces does not boot.
- 3.33 Increased the maximum number of SSH sessions to 8 and the maximum port forwarding connections to 256.

Release 2.1.1 RC2 Release Notes

The following notes describe basic quality enhancements in MNS-DX version 2.1.1 RC2 (since version 2.1.0 RC1).

1.0 INTRODUCTION

The following notes contain details related to the MNS-DX v2.1.1 RC2 software release. MNS-DX v2.1.1 RC2 includes a couple quality enhancements over MNS-DX v2.1.0 RC2 as outlined below.

2.0 NEW FEATURES

3.0 QUALITY ENHANCEMENTS

- 3.1 An issue was introduced in MNS-DX v2.1.0 that prevents the initial configuration from being automatically written to the file system after starting from defaults. This has been resolved.
- 3.2 DX units that were manufactured prior to MNS-DX release v1.4.0 do not have a specific piece of information programmed into them. If this information does not exist, the operating software uses a default value, but this default value was incorrect and prevented the firewall outbound connection table from being accessible to the user. This has been resolved.