Chapter 4. Routing

This chapter describes how to configure IP routing in NetDefendOS.

• Overview, page 89
• Static Routing, page 90
• Policy-based Routing, page 98
• Dynamic Routing, page 103
• Multicast Routing, page 110
• Transparent Mode, page 119

4.1. Overview

IP routing capabilities belong to the most fundamental functionalities of NetDefendOS: any IP packet flowing through the system will be subjected to at least one routing decision at some point in time, and proper setup of routing is crucial for a NetDefendOS system to function as expected.

NetDefendOS offers support for the following types of routing mechanisms:

• Static routing.
• Dynamic routing.

NetDefendOS additionally supports route monitoring to achieve route and link redundancy with fail-over capability.
4.2. Static Routing

The most basic form of routing is known as Static Routing. The term static refers to the fact that entries in the routing table are manually added and are therefore permanent (or static) by nature.

Due to this manual approach, static routing is most appropriate to use in smaller network deployments where addresses are fairly fixed and where the amount of connected networks are limited to a few. For larger networks however (or whenever the network topology is complex), the work of manually maintaining static routing tables will be time-consuming and problematic. As a consequence, dynamic routing should be used in those cases.

For more information about the dynamic routing capabilities of NetDefendOS, please see Section 4.4, “Dynamic Routing”. Note however, that even if you choose to implement dynamic routing for your network, you will still need to understand the principles of static routing and how it is implemented in NetDefendOS.

4.2.1. Basic Principles of Routing

IP routing is the mechanism used in TCP/IP based networks for delivering IP packets from their source to their ultimate destination through a number of intermediary nodes, most often referred to as routers or firewalls. In each router, a routing table is consulted to find out where to send the packet next. A routing table usually consists of several routes, where each route in principle contains a destination network, an interface to forward the packet on and optionally the IP address of the next gateway in the path to the destination.

The images below illustrates a typical D-Link Firewall deployment and how the associated routing table would look like.

<table>
<thead>
<tr>
<th>Route #</th>
<th>Interface</th>
<th>Destination</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lan</td>
<td>192.168.0.0/24</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>dmz</td>
<td>10.4.0.0/16</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>wan</td>
<td>195.66.77.0/24</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>wan</td>
<td>all-nets</td>
<td>195.66.77.4</td>
</tr>
</tbody>
</table>

The above routing table provides the following information:

• Route #1: All packets going to hosts on the 192.168.0.0/24 network should be sent out on the lan interface. As no gateway is specified for the route entry, the host is assumed to be located on the network segment directly reachable from the lan interface.

• Route #2: All packets going to hosts on the 10.4.0.0/16 network are to be sent out on the dmz interface. Also for this route, no gateway is specified.

• Route #3: All packets going to hosts on the 195.66.77.0/24 network will be sent out on the wan interface. No gateway is required to reach the hosts.

• Route #4: All packets going to any host (the all-nets network will match all hosts) will be sent out on the wan interface and to the gateway with IP address 195.66.77.4. That gateway will then consult its routing table to find out where to send the packets next. A route with destination all-nets is often referred to as the Default Route as it will match all packets for which no specific route has been configured.

When a routing table is evaluated, the ordering of the routes is important. In general, a routing table is evaluated with the most specific routes first. In other words, if two routes have destination networks that overlap, the more narrow network will be evaluated prior to the wider one. In the above example, a packet with a destination IP address of 192.168.0.4 will theoretically match both the first route and the last one. However, the first route entry is a more specific match, so the evaluation will end there and the packet will be routed according to that entry.
4.2.2. Static Routing

This section describes how routing is implemented in NetDefendOS, and how to configure static routing.

NetDefendOS supports multiple routing tables. A default table called main is pre-defined and is always present in NetDefendOS. However, additional and completely separate routing tables can be defined by the administrator to provide alternate routing.

These user-defined extra routing tables can be used to implement Policy Based Routing which means the administrator can set up rules in the IP rule set which decide which of the routing tables will handle certain types of traffic. (see Section 4.3, “Policy-based Routing”).

The Route Lookup Mechanism

The NetDefendOS route lookup mechanism has some slight differences to how some other router products work. In many routers, where the IP packets are forwarded without context (in other words, the forwarding is stateless), the routing table is scanned for each and every IP packet received by the router. In NetDefendOS, packets are forwarded with state-awareness, so the route lookup process is tightly integrated into NetDefendOS's stateful inspection mechanism.

When an IP packet is received on any of the interfaces, the connection table is consulted to see if there is an already open connection for which the received packet belongs. If an existing connection is found, the connection table entry includes information on where to route the packet so there is no need for lookups in the routing table. This is far more efficient than traditional routing table lookups, and is one reason for the high forwarding performance of NetDefendOS.

If an established connection cannot be found, then the routing table is consulted. It is important to understand that the route lookup is performed before the various rules sections get evaluated. As a result, the destination interface is known at the time NetDefendOS decides if the connection should be allowed or dropped. This design allows for a more fine-grained control in security policies.

NetDefendOS Route Notation

NetDefendOS uses a slightly different way of describing routes compared to most other systems but this way is easier to understand, making errors less likely.

Many other products do not use the specific interface in the routing table, but specify the IP address of the interface instead. The routing table below is from a Microsoft Windows XP workstation:

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td></td>
<td></td>
<td>192.168.0.1</td>
<td>20</td>
</tr>
<tr>
<td>10.0.0.0</td>
<td></td>
<td>192.168.0.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.4.2.143</td>
<td></td>
<td>127.0.0.1</td>
<td>10.4.2.143</td>
<td>1</td>
</tr>
<tr>
<td>10.255.255.255</td>
<td>255.255.255.255</td>
<td>127.0.0.1</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>85.11.194.33</td>
<td>255.255.255.255</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>1</td>
</tr>
<tr>
<td>127.0.0.0</td>
<td></td>
<td>127.0.0.1</td>
<td>10.4.2.143</td>
<td>20</td>
</tr>
<tr>
<td>192.168.0.0</td>
<td></td>
<td>192.168.0.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192.168.0.0.0</td>
<td></td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>1</td>
</tr>
<tr>
<td>192.168.0.10</td>
<td>255.255.255.255</td>
<td>192.168.0.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192.168.0.255</td>
<td>255.255.255.255</td>
<td>192.168.0.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>224.0.0.0</td>
<td></td>
<td>10.4.2.143</td>
<td>10.4.2.143</td>
<td>50</td>
</tr>
<tr>
<td>224.0.0.0</td>
<td></td>
<td>192.168.0.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>255.255.255.255</td>
<td>255.255.255.255</td>
<td>10.4.2.143</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>255.255.255.255</td>
<td>255.255.255.255</td>
<td>192.168.0.10</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Default Gateway:</td>
<td></td>
<td>192.168.0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Persistent Routes:
None

The corresponding routing table in NetDefendOS is similar to this:

<table>
<thead>
<tr>
<th>Flags</th>
<th>Network</th>
<th>Iface</th>
<th>Gateway</th>
<th>Local IP</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>192.168.0.0/24</td>
<td>lan</td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>10.0.0.0/8</td>
<td>wan</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0.0.0.0/0</td>
<td>wan</td>
<td>192.168.0.1</td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>

The NetDefendOS way of describing the routes is easier to read and understand. Another advantage with this form of notation is that you can specify a gateway for a particular route without having a route that covers the gateway’s IP address or despite the fact that the route covers the gateway’s IP address is normally routed via another interface.

It is also worth mentioning that NetDefendOS allows you to specify routes for destinations that are not aligned with traditional subnet masks. In other words, it is perfectly legal to specify one route for the destination address range 192.168.0.5-192.168.0.17 and another route for addresses 192.168.0.18-192.168.0.254. This is a feature that makes NetDefendOS highly suitable for routing in highly complex network topologies.

Displaying the Routing Table

It is important to distinguish between the routing table that is active in the system, and the routing table that you configure. The routing table that you configure contains only the routes that you have added manually (in other words, the static routes). The content of the active routing table, however, will vary depending on several factors. For instance, if dynamic routing has been enabled, the routing table will be populated with routes learned by communicating with other routers in the network. Also, features such as route fail-over will cause the active routing table to look different from time to time.

Example 4.1. Displaying the Routing Table

This example illustrates how to display the contents of the configured routing table as well as the active routing table.

**CLI**

To see the configured routing table:

```
gw-world:/> cc RoutingTable main
```

```
gw-world:/main> show
```

```
Route
  # | Interface | Network      | Gateway | Local IP
----|-----------|--------------|---------|----------
1   | wan       | all-nets     | 213.124.165.1 | (none) |
2   | lan       | lannet       | (none)  | (none)   |
3   | wan       | wanet        | (none)  | (none)   |
```

To see the active routing table enter:

```
gw-world:/> routes
```

```
Flags Network  Iface Gateway | Local IP | Metric
------- ----------- --------- |---------- |----
192.168.0.0/24  lan        |          | 0
```
### Core Routes

NetDefendOS automatically populates the active routing table with Core Routes. These routes are present for the system to understand where to route traffic that is destined for the system itself. There is one route added for each interface in the system. In other words, two interfaces named lan and wan, and with IP addresses 192.168.0.10 and 193.55.66.77, respectively, will result in the following routes:

<table>
<thead>
<tr>
<th>Route #</th>
<th>Interface</th>
<th>Destination</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>core</td>
<td>192.168.0.10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>core</td>
<td>193.55.66.77</td>
<td></td>
</tr>
</tbody>
</table>

When the system receives an IP packet whose destination address is one of the interface IPs, the packet will be routed to the core interface. In other words, it is processed by NetDefendOS itself.

There is also a core route added for all multicast addresses:

<table>
<thead>
<tr>
<th>Route #</th>
<th>Interface</th>
<th>Destination</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>core</td>
<td>224.0.0.0/4</td>
<td></td>
</tr>
</tbody>
</table>

To include the core routes when you display the active routing table, you have to specify an option to the routing command.

### Example 4.2. Displaying the Core Routes

This example illustrates how to display the core routes in the active routing table.

**CLI**

```
gw-world:/> routes -all
```

<table>
<thead>
<tr>
<th>Flags</th>
<th>Network</th>
<th>Iface</th>
<th>Gateway</th>
<th>Local IP</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>127.0.0.1</td>
<td>core</td>
<td>(Shared IP)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>192.168.0.1</td>
<td>core</td>
<td>(Iface IP)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>213.124.165.181</td>
<td>core</td>
<td>(Iface IP)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>127.0.3.1</td>
<td>core</td>
<td>(Iface IP)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>127.0.4.1</td>
<td>core</td>
<td>(Iface IP)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>192.168.0.0/24</td>
<td>lan</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>213.124.165.0/24</td>
<td>wan</td>
<td>(Iface IP)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>224.0.0.0/4</td>
<td>core</td>
<td>(Iface IP)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.0.0.0/0</td>
<td>wan</td>
<td>213.124.165.1</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

93
### Web Interface

1. Select the **Routes** item in the **Status** dropdown menu in the menu bar.
2. Check the **Show all routes** checkbox and click the **Apply** button.
3. The main window will list the active routing table, including the core routes.

---

**Tip**

For detailed information about the output of the CLI `routes` command. Please see the CLI Reference Guide.

---

### 4.2.3. Route Failover

#### Overview

D-Link Firewalls are often deployed in mission-critical locations where availability and connectivity is crucial. A corporation relying heavily on access to the Internet, for instance, could have their operations severely disrupted if an Internet connection fails.

As a consequence, it is quite common to have backup Internet connectivity using a secondary Internet Service Provider (ISP). The connections to the two service providers often use different access methods to avoid a single point of failure.

To facilitate a scenario such as multiple ISPs, NetDefendOS provides a **Route Failover** capability so that should one route fail, traffic can automatically **failover** to another, alternate route. NetDefendOS implements Route Failover through the use of **Route Monitoring** in which NetDefendOS monitors the availability of routes and switches traffic to an alternate route should the primary, preferred one fail.

*Figure 4.1. A Route Failover Scenario for ISP Access*

---

#### Setting Up Route Failover

Route Monitoring should be enabled on a per-route basis. To enable the Route Failover feature in a scenario with a preferred and a backup route, the preferred route will have Route Monitoring enabled, however the backup route does not require it to be enabled since it will usually have no route to failover to. For a route with Route Monitoring enabled, one of two Route Monitoring
methods must be chosen:

**Interface Link Status**
NetDefendOS will monitor the link status of the interface specified in the route. As long as the interface is up, the route is diagnosed as healthy. This method is appropriate for monitoring that the interface is physically attached and that the cabling is working as expected. As any changes to the link status are instantly noticed, this method provides the fastest response to failure.

**Gateway Monitoring**
If a specific gateway has been specified as the next hop for a route, accessibility to that gateway can be monitored by sending periodic ARP requests. As long as the gateway responds to these requests, the route is considered to be functioning correctly.

**Setting the Route Metric**
When specifying routes, the administrator should manually set a route's Metric. The Metric is a positive integer that indicates how preferred the route is as a means to reach its destination. When two routes offer a means to reach the same destination, NetDefendOS will select the one with the lowest Metric value for sending data (if two routes have the same Metric, the route found first in the routing table will be chosen).

A primary, preferred route should have a lower Metric (for example "10"), and a secondary, failover route should have a higher Metric value (for example "20").

**Multiple Failover Routes**
It is possible to specify more than one failover route. For instance, the primary route could have two other routes as failover routes instead of just one. In this case the Metric should be different for each of the three routes: "10" for the primary route, "20" for the first failover route and "30" for the second failover route. The first two routes would have Route Monitoring enabled in the routing table but the last one (with the highest Metric) would not since it has no route to failover to.

**Failover Processing**
Whenever monitoring determines that a route is not available, NetDefendOS will mark the route as disabled and instigate Route Failover for existing and new connections. For already established connections, a route lookup will be performed to find the next best matching route and the connections will then switch to using the new route. For new connections, route lookup will ignore disabled routes and the next best matching route will be used instead.

The table below defines two default routes, both having *all-nets* as the destination, but using two different gateways. The first, primary route has the lowest Metric and also has Route Monitoring enabled. Route Monitoring for the second, alternate route isn't meaningful since it has no failover route.

<table>
<thead>
<tr>
<th>Route #</th>
<th>Interface</th>
<th>Destination</th>
<th>Gateway</th>
<th>Metric</th>
<th>Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>wan</td>
<td>all-nets</td>
<td>195.66.77.1</td>
<td>10</td>
<td>On</td>
</tr>
<tr>
<td>2</td>
<td>wan</td>
<td>all-nets</td>
<td>193.54.68.1</td>
<td>20</td>
<td>Off</td>
</tr>
</tbody>
</table>

When a new connection is about to be established to a host on the Internet, a route lookup will result in the route that has the lowest Metric being chosen. If the primary WAN router should then fail, this will be detected by NetDefendOS, and the first route will be disabled. As a consequence, a new route lookup will be performed and the second route will be selected with the first one being marked as disabled.

**Re-enabling Routes**
Even if a route has been disabled, NetDefendOS will continue to check the status of that route. Should the route become available again, it will be re-enabled and existing connections will...

---

4.2.3. Route Failover
Chapter 4. Routing
automatically be transferred back to it.

**Route Interface Grouping**

When using route monitoring, it is important to check if a failover to another route will cause the routing interface to be changed. If this could happen, it is necessary to take some precautionary steps to ensure that policies and existing connections will be maintained.

To illustrate the problem, consider the following configuration:

First, there is one IP rule that will NAT all HTTP traffic destined for the Internet through the **wan** interface:

<table>
<thead>
<tr>
<th>#</th>
<th>Action</th>
<th>Src Iface</th>
<th>Src Net</th>
<th>Dest Iface</th>
<th>Dest Net</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NAT</td>
<td>lan</td>
<td>lan-net</td>
<td>wan</td>
<td>all-nets</td>
<td>http</td>
</tr>
</tbody>
</table>

The routing table consequently contains the following default route:

<table>
<thead>
<tr>
<th>Route #</th>
<th>Interface</th>
<th>Destination</th>
<th>Gateway</th>
<th>Metric</th>
<th>Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>wan</td>
<td>all-nets</td>
<td>195.66.77.1</td>
<td>10</td>
<td>Off</td>
</tr>
</tbody>
</table>

Now a secondary route is added over a backup DSL connection and Route Monitoring is enabled for this. The updated routing table will look like this:

<table>
<thead>
<tr>
<th>Route #</th>
<th>Interface</th>
<th>Destination</th>
<th>Gateway</th>
<th>Metric</th>
<th>Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>wan</td>
<td>all-nets</td>
<td>195.66.77.1</td>
<td>10</td>
<td>On</td>
</tr>
<tr>
<td>2</td>
<td>dsl</td>
<td>all-nets</td>
<td>193.54.68.1</td>
<td>20</td>
<td>Off</td>
</tr>
</tbody>
</table>

Notice that Route Monitoring is enabled for the first route but not the backup, failover route.

As long as the preferred **wan** route is healthy, everything will work as expected. Route Monitoring will also be functioning, so the secondary route will be enabled should the **wan** route fail.

There are, however, some problems with this setup: if a route failover occurs, the default route will then use the **dsl** interface. When a new HTTP connection is then established from the **intnet** network, a route lookup will be made resulting in a destination interface of **dsl**. The IP rules will then be evaluated, but the original NAT rule assumes the destination interface to be **wan** so the new connection will be dropped by the rule set.

In addition, any existing connections matching the NAT rule will also be dropped as a result of the change in the destination interface. Clearly, this is undesirable.

To overcome this issue, potential destination interfaces should be grouped together into an **Interface Group** and the **Security/Transport Equivalent** flag should be enabled for the Group. The Interface Group is then used as the Destination Interface when setting policies. For more information on groups, see Section 3.3.6, “Interface Groups”.

**Gratuitous ARP Generation**

By default NetDefendOS generates a gratuitous ARP request when a route failover occurs. The reason for this is to notify surrounding systems that there has been a route change. This behaviour can be controlled by the advanced setting RFO_GratuitousARPOnFail.

**4.2.4. Proxy ARP**

As explained previously in Section 3.4, “ARP”, the ARP protocol facilitates a mapping between an IP address and the MAC address of a node on an Ethernet network. However, situations may exist where a network running Ethernet is separated into two parts with a routing device such as an installed D-Link Firewall, in between. In such a case, NetDefendOS itself can respond to ARP requests directed to the network on the other side of the D-Link Firewall using the feature known as Proxy ARP.

For example, host A on one subnet might send an ARP request to find out the MAC address of the
IP address of host B on another separate network. The proxy ARP feature means that NetDefendOS responds to this ARP request instead of host B. The NetDefendOS sends its own MAC address instead in reply, essentially pretending to be the target host. After receiving the reply, Host A then sends data directly to NetDefendOS which, acting as a proxy, forwards the data on to host B. In the process the device has the opportunity to examine and filter the data.

The splitting of an Ethernet network into two distinct parts is a common application of D-Link Firewall's Proxy ARP feature, where access between the parts needs to be controlled. In such a scenario NetDefendOS can monitor and regulate all traffic passing between the two parts.

**Note**

*It is only possible to have Proxy ARP functioning for Ethernet and VLAN interfaces.*
4.3. Policy-based Routing

4.3.1. Overview

Policy-based Routing (PBR) is an extension to the standard routing described previously. It offers administrators significant flexibility in implementing routing decision policies by being able to define rules so alternative routing tables are used.

Normal routing forwards packets according to destination IP address information derived from static routes or from a dynamic routing protocol. For example, using OSPF, the route chosen for packets will be the least-cost (shortest) path derived from an SPF calculation. Policy-based Routing means that routes chosen for traffic can be based on specific traffic parameters.

Policy-based Routing can allow:

- **Source based routing**: A different routing table may need to be chosen based on the source of traffic. When more than one ISP is used to provide Internet services, Policy-based Routing can route traffic originating from different sets of users through different routes. For example, traffic from one address range might be routed through one ISP, whilst traffic from another address range might be through a second ISP.

- **Service-based Routing**: A different routing table might need to be chosen based on the service. Policy-based Routing can route a given protocol such as HTTP, through proxies such as Web caches. Specific services might also be routed to a specific ISP so that one ISP handles all HTTP traffic.

- **User based Routing**: A different routing table might need to be chosen based on the user identity or the group to which the user belongs. This is particularly useful in provider-independent metropolitan area networks where all users share a common active backbone, but each can use different ISPs, subscribing to different providers.

Policy-based Routing implementation in NetDefendOS is based on two building blocks:

- One or more user-defined alternate Policy-based Routing Tables in addition to the standard default main routing table.
- One or more Policy-based routing rules which determines which routing table to use for which traffic.

4.3.2. Policy-based Routing Tables

NetDefendOS, as standard, has one default routing table called main. In addition to the main table, it is possible to define one or more, additional alternate routing tables (this section will sometimes refer to these Policy-based Routing Tables as alternate routing tables).

Alternate routing tables contain the same information for describing routes as main, except that there is an extra parameter ordering defined for each of them. This parameter decides how route lookup is done using alternate tables in conjunction with the main table. This is described further in Section 4.3.5, “The Ordering parameter” below.

4.3.3. Policy-based Routing Rules

A rule in the Policy-based Routing rule set can decide which routing table is selected. A
Policy-based Routing rule can be triggered by the type of Service (HTTP for example) in combination with the Source/Destination Interface and Source/Destination Network.

When looking up Policy-based Rules, it is the first matching rule found that is triggered.

4.3.4. Policy-based Routing Table Selection

When a packet corresponding to a new connection first arrives, the processing steps are as follows to determine which routing table is chosen:

1. The PBR Rules must first be looked up but to do this the packet's destination interface must be determined and this is always done by a lookup in the main routing table. It is therefore important a match for the destination network is found or at least a default all-nets route exists which can catch anything not explicitly matched.

2. A search is now made for a Policy-based Routing Rule that matches the packets's source/destination interface/network as well as service. If a matching rule is found then this determines the routing table to use. If no PBR Rule is found then the main table will be used.

3. Once the correct routing table has been located, a check is made to make sure that the source IP address in fact belongs on the receiving interface. The Access Rules are firstly examined to see if they can provide this check (see Section 6.1, “Access Rules” for more details of this feature). If there are no Access Rules or a match with the rules cannot be found, a reverse lookup in the previously selected routing table is done using the source IP address. If the check fails then a Default access rule log error message is generated.

4. At this point, using the routing table selected, the actual route lookup is done to find the packet's destination interface. At this point the ordering parameter is used to determine how the actual lookup is done and the options for this are described in the next section. To implement virtual systems, the Only ordering option should be used.

5. The connection is then subject to the normal IP rule set. If a SAT rule is encountered, address translation will be performed. The decision of which routing table to use is made before carrying out address translation but the actual route lookup is performed on the altered address. (Note that the original route lookup to find the destination interface used for all rule look-ups was done with the original, untranslated address.)

6. If allowed by the IP rule set, the new connection is opened in the NetDefendOS state table and the packet forwarded through this connection.

4.3.5. The Ordering parameter

Once the routing table for a new connection is chosen and that table is an alternate routing table, the Ordering parameter associated with the table is used to decide how the alternate table is combined with the main table to lookup the appropriate route. The three available options are:

1. Default - The default behaviour is to first look up the route in the main table. If no matching route is found, or the default route is found (the route with the destination all-nets - 0.0.0.0/0), a lookup for a matching route in the alternate table is done. If no match is found in the alternate table then the default route in the main table will be used.

2. First - This behaviour is to first look up the connection's route in the alternate table. If no matching route is found there then the main table is used for the lookup. The default all-nets route will be counted as a match in the alternate table if it exists there.

3. Only - This option ignores the existence of any other table except the alternate table so the alternate table is the only one used for the lookup. One application of this is to give the administrator a way to dedicate a single routing table to one set of interfaces. Only is the option to use when creating virtual systems since it can dedicate one routing table to a set of
interfaces.

The first two options can be regarded as combining the alternate table with the main table and assigning one route if there is a match in both tables.

**Important - Ensuring all-nets appears in the main table.**

A common mistake with Policy-based routing is the absence of the default route with a destination interface of all-nets in the default main routing table. If there is no route that is an exact match then the absence a default all-nets route will mean that the connection will be dropped.

---

**Example 4.3. Creating a Policy-Based Routing table**

In this example we create a Policy-based Routing table named "TestPBRTable".

**Web Interface**

1. Go to Routing > Routing Tables > Add > RoutingTable
2. Now enter:
   - **Name**: TestPBRTable
   - For **Ordering** select one of:
     - **First** - the named routing table is consulted first of all. If this lookup fails, the lookup will continue in the main routing table.
     - **Default** - the main routing table will be consulted first. If the only match is the default route (all-nets), the named routing table will be consulted. If the lookup in the named routing table fails, the lookup as a whole is considered to have failed.
     - **Only** - the named routing table is the only one consulted. If this lookup fails, the lookup will not continue in the main routing table.
3. If **Remove Interface IP Routes** is enabled, the default interface routes are removed, that is to say routes to the core interface (which are routes to NetDefendOS itself).
4. Click **OK**

---

**Example 4.4. Creating the Route**

After defining the routing table "TestPBRTable", we add routes into the table.

**Web Interface**

1. Go to Routing > Routing Tables > TestPBRTable > Add > Route
2. Now enter:
   - **Interface**: The interface to be routed
   - **Network**: The network to route
   - **Gateway**: The gateway to send routed packets to
   - **Local IP Address**: The IP address specified here will be automatically published on the corresponding interface. This address will also be used as the sender address in ARP queries. If no address is specified, the firewall's interface IP address will be used.
   - **Metric**: Specifies the metric for this route. (Mostly used in route fail-over scenarios)
3. Click **OK**
Example 4.5. Policy Based Routing Configuration

This example illustrates a multiple ISP scenario which is a common use of Policy-based Routing. The following is assumed:

- Each ISP will give you an IP network from its network range. We will assume a 2-ISP scenario, with the network 10.10.10.0/24 belonging to "ISP A" and "20.20.20.0/24" belonging to "ISP B". The ISP gateways are 10.10.10.1 and 20.20.20.1, respectively.
- All addresses in this scenario are public addresses for the sake of simplicity.
- This is a "drop-in" design, where there are no explicit routing subnets between the ISP gateways and the D-Link Firewall.

In a provider-independent network, clients will likely have a single IP address, belonging to one of the ISPs. In a single-organization scenario, publicly accessible servers will be configured with two separate IP addresses: one from each ISP. However, this difference does not matter for the policy routing setup itself.

Note that, for a single organization, Internet connectivity through multiple ISPs is normally best done with the BGP protocol, where you do not need to worry about different IP spans or policy routing. Unfortunately, this is not always possible, and this is where Policy Based Routing becomes a necessity.

We will set up the main routing table to use ISP A, and add a named routing table, "r2" that uses the default gateway of ISP B.

### Interface

<table>
<thead>
<tr>
<th>Interface</th>
<th>Network</th>
<th>Gateway</th>
<th>ProxyARP</th>
</tr>
</thead>
<tbody>
<tr>
<td>lan1</td>
<td>10.10.10.0/24</td>
<td></td>
<td>wan1</td>
</tr>
<tr>
<td>lan1</td>
<td>20.20.20.0/24</td>
<td></td>
<td>wan2</td>
</tr>
<tr>
<td>wan1</td>
<td>10.10.10.0/32</td>
<td></td>
<td>lan1</td>
</tr>
<tr>
<td>wan2</td>
<td>20.20.0.1/32</td>
<td></td>
<td>lan1</td>
</tr>
<tr>
<td>lan1</td>
<td>all-nets</td>
<td>10.10.10.1</td>
<td></td>
</tr>
</tbody>
</table>

Contents of the named Policy-based Routing table r2:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Network</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>wan2</td>
<td>all-nets</td>
<td>20.20.20.1</td>
</tr>
</tbody>
</table>

The table r2 has its Ordering parameter set to Default, which means that it will only be consulted if the main routing table lookup matches the default route (all-nets).

Contents of the Policy-based Routing Policy:

<table>
<thead>
<tr>
<th>Source Interface</th>
<th>Source Range</th>
<th>Destination Interface</th>
<th>Destination Range</th>
<th>Service</th>
<th>Forward VR</th>
<th>Return VR</th>
</tr>
</thead>
<tbody>
<tr>
<td>lan1</td>
<td>10.10.10.0/24</td>
<td>wan2</td>
<td>all-nets</td>
<td>ALL</td>
<td>r2</td>
<td>r2</td>
</tr>
<tr>
<td>wan2</td>
<td>all-nets</td>
<td>lan1</td>
<td>20.20.20.0/24</td>
<td>ALL</td>
<td>r2</td>
<td>r2</td>
</tr>
</tbody>
</table>

To configure this example scenario:

**Web Interface**

1. Add the routes found in the list of routes in the main routing table, as shown earlier.
2. Create a routing table called "r2" and make sure the ordering is set to "Default".
3. Add the route found in the list of routes in the routing table "r2", as shown earlier.
4. Add two VR policies according to the list of policies shown earlier.
   - Go to **Routing > Routing Rules > Add > Routing Rule**
   - Enter the information found in the list of policies displayed earlier
   - Repeat the above to add the second rule
4.3.5. The Ordering parameter

**Note**
Rules in the above example are added for both inbound and outbound connections.
4.4. Dynamic Routing

4.4.1. Dynamic Routing overview

Dynamic routing is different to static routing in that the D-Link Firewall will adapt to changes of network topology or traffic load automatically. NetDefendOS first learns of all the directly connected networks and gets further route information from other routers. Detected routes are sorted and the most suitable routes for destinations are added into the routing table and this information is distributed to other routers.

Dynamic Routing responds to routing updates on the fly but has the disadvantage that it is more susceptible to certain problems such as routing loops. In the Internet, two types of dynamic routing algorithm are used: the Distance Vector(DV) algorithm and the Link State(LS) algorithm. How a router decides the optimal or "best" route and shares updated information with other routers depends on the type of algorithm used.

Distance Vector Algorithms

The Distance vector (DV) algorithm is a decentralized routing algorithm that computes the "best" path in a distributed way. Each router computes the costs of its own attached links, and shares the route information only with its neighbor routers. The router will gradually learns the least-cost path by iterative computation and information exchange with its neighbors.

The Routing Information Protocol (RIP) is a well-known DV algorithm and involves sending regular update messages and reflecting routing changes in the routing table. Path determination is based on the "length" of the path which is the number of intermediate routers [also known as "hops"]. After updating its own routing table, the router immediately begins transmitting its entire routing table to neighboring routers to inform them of changes.

Link State Algorithms

In contrast to DV algorithms, Link State (LS) algorithms enable routers to keep routing tables that reflect the topology of the entire network. Each router broadcasts its attached links and link costs to all other routers in the network. When a router receives these broadcasts it runs the LS algorithm and calculates its own set of least-cost paths. Any change of the link state will be sent everywhere in the network, so that all routers keep the same routing table information.

Open Shortest Path First

Open Shortest Path First (OSPF) is a widely used LS algorithm. An OSPF enabled router first identifies the routers and subnets that are directly connected to it and then broadcasts the information to all the other routers. Each router uses the information it receives to build a table of what the whole network looks like. With a complete routing table, each router can identify the subnetworks and routers that lead to any destination. Routers using OSPF only broadcast updates that inform of changes and not the entire routing table.

OSPF depends on various metrics for path determination, including hops, bandwidth, load and delay. OSPF can provide a great deal of control over the routing process since its parameters can finely tuned.

Comparing Dynamic Routing Algorithms

Due to the fact that the global link state information is maintained everywhere in a network, LS algorithms offer a high degree of configuration control and scalability. Changes result in broadcasts of just the updated information to other routers which means faster convergence and less possibility of routing loops. OSPF can also operate within a hierarchy, whereas RIP has no knowledge of sub-network addressing. NetDefendOS uses OSPF as its dynamic routing algorithm because of the many advantages it offers.

Routing metrics
Routing metrics are the criteria a routing algorithm uses to compute the "best" route to a destination. A routing protocol relies on one or several metrics to evaluate links across a network and to determine the optimal path. The principal metrics used include:

- **Path length**: The sum of the costs associated with each link. A commonly used value for this metric is called "hop count" which is the number of routing devices a packet must pass through when it travels from source to destination.

- **Bandwidth**: The traffic capacity of a path, rated by "Mbps".

- **Load**: The usage of a router. The usage can be evaluated by CPU utilization and throughput.

- **Delay**: The time it takes to move a packet from the source to the destination. The time depends on various factors, including bandwidth, load, and the length of the path.

### 4.4.2. OSPF

#### Overview

*Open Shortest Path First* (OSPF) is a routing protocol developed for IP networks by the Internet Engineering Task Force (IETF). The NetDefendOS OSPF implementation is based upon RFC 2328, with compatibility to RFC 1583.

The way OSPF works is that it routes IP packets based only on the destination IP address found in the IP packet header. IP packets are routed "as is", that is they are not encapsulated in any further protocol headers as they transit the Autonomous System (AS). OSPF is a dynamic routing protocol, it quickly detects topological changes in the AS (such as router interface failures) and calculates new loop-free routes after a period of time.

OSPF is a link-state routing protocol that calls for the sending of link-state advertisements (LSAs) to all other routers within the same area. In a link-state routing protocol, each router maintains a database describing the Autonomous System's topology. This database is referred to as the link-state database. Each router in the same AS has an identical database. From the information in the link-state database, each router constructs a tree of shortest paths with itself as root. This shortest-path tree gives the route to each destination in the Autonomous System.

OSPF allows sets of networks to be grouped together, this is called an area. The topology of an area is hidden from the rest of the AS. This information hiding reduces the amount of routing traffic exchanged. Also, routing within the area is determined only by the area's own topology, lending the area protection from bad routing data. An area is a generalization of an IP subnetted network.

All OSPF protocol exchanges can be authenticated. This means that only routers with the correct authentication can join the Autonomous System. Different authentication schemes can be used, like none, passphrase or MD5 digest. It is possible to configure separate authentication methods for each Autonomous System.

#### OSPF Areas

The Autonomous System is divided into smaller parts called *OSPF Areas*. This section describes what an area is, and its associated terms.

**Areas**

An area consists of networks and hosts within an AS that have been grouped together. Routers that are only within an area are called internal routers, all interfaces on internal routers are directly connected to networks within the area. The topology of an area is hidden from the rest of the AS.

**ABRs**

Routers that have interfaces in more than one area are called Area Border Routers (ABRs), these maintain a separate topological database for each area.
to which they have an interface.

**ASBRs**

Routers that exchange routing information with routers in other Autonomous Systems are called Autonomous System Boundary Router (ASBRs). They advertise externally learned routes throughout the Autonomous System.

**Backbone Areas**

All OSPF networks need to have at least the backbone area, that is the area with ID 0. This is the area that all other areas should be connected to, and the backbone make sure to distribute routing information between the connected areas. When an area is not directly connected to the backbone it needs a virtual link to it.

**Stub Areas**

Stub areas are areas through which or into which AS external advertisements are not flooded. When an area is configured as a stub area, the router will automatically advertises a default route so that routers in the stub area can reach destinations outside the area.

**Transit Areas**

Transit areas are used to pass traffic from a area that is not directly connect to the backbone area.

**The Designated Router**

Each OSPF broadcast network has a designated router and a backup designated router. The routers uses OSPF hello protocol to elect the DR and BDR for the network based on the priorities advertised by all the routers. If there already are a DR on the network, the router will accept that one, regardless of its own router priority.

**Neighbors**

Routers that are in the same area become neighbors in that area. Neighbors are elected via the Hello protocol. Hello packets are sent periodically out of each interface using IP multicast. Routers become neighbors as soon as they see themselves listed in the neighbor's Hello packet. This way, a two way communication is guaranteed.

The following **Neighbor States** are defined:

- **Down** This is the initial stat of the neighbor relationship.
- **Init** When a HELLO packet is received from a neighbor, but does NOT include the Router ID of the firewall in it, the neighbor will be placed in Init state. As soon as the neighbor in question receives a HELLO packet it will know the sending routers Router ID and will send a HELLO packet with that included. The state of the neighbors will change to 2-Way state.
- **2-Way** In this state the communication between the router and the neighbor is bi-directional. On Point-to-Point and Point-to-Multipoint interfaces, the state will be changed to Full. On Broadcast interfaces, only the DR/BDR will advance to Full state with their neighbors, all the remaining neighbors will remain in the 2-Way state.
- **ExStart** Preparing to build adjacency.
- **Exchange** Routers are exchanging Data Descriptors.
- **Loading** Routers are exchanging LSAs.
- **Full** This is the normal state of an adjacency between a router and the DR/BDR.

**Aggregates**

OSPF Aggregation is used to combine groups of routes with common addresses into a single entry
in the routing table. This is commonly used to minimize the routing table.

**Virtual Links**

Virtual links are used for:

- Linking an area that does not have a direct connection to the backbone.
- Linking the backbone in case of a partitioned backbone.

**Areas without direct connection to the backbone**

The backbone always need to be the center of all other areas. In some rare case where it is impossible to have an area physically connected to the backbone, a virtual link is used. This virtual link will provide that area with a logical path to the backbone area. This virtual link is established between two ABRs that are on one common area, with one of the ABRs connected to the backbone area. In the example below two routers are connected to the same area (Area 1) but just one of them, fw1, is connected physically to the backbone area.

**Figure 4.2. Virtual Links Example 1**

In the above example, the Virtual Link is configured between fw1 and fw2 on Area 1, as it is used as the transit area. In this configuration only the Router ID has to be configured. The diagram shows that fw2 needs to have a Virtual Link to fw1 with Router ID 192.168.1.1 and vice versa. These Virtual Links need to be configured in Area 1.

**A Partitioned Backbone**

OSPF allows for linking a partitioned backbone using a virtual link. The virtual link should be configured between two separate ABRs that touch the backbone are from each side and having a
Figure 4.3. Virtual Links Example 2

The Virtual Link is configured between fw1 and fw2 on Area 1, as it is used as the transit area. In the configuration only the Router ID have to be configured, as in the example above show fw2 needs to have a Virtual Link to fw1 with the Router ID 192.168.1.1 and vice versa. These VLinks need to be configured in Area 1.

**OSPF High Availability Support**

There are some limitations in High Availability support for OSPF that should be noted:

Both the active and the inactive part of an HA cluster will run separate OSPF processes, although the inactive part will make sure that it is not the preferred choice for routing. The HA master and slave will not form adjacency with each other and are not allowed to become DR/BDR on broadcast networks. This is done by forcing the router priority to 0.

For OSPF HA support to work correctly, the D-Link Firewall needs to have a broadcast interface with at least ONE neighbor for ALL areas that the firewall is attached to. In essence, the inactive part of the cluster needs a neighbor to get the link state database from.

It should also be noted that is not possible to put an HA cluster on the same broadcast network without any other neighbors (they won't form adjacency with each other because of the router priority 0). However, it may be possible, depending on the scenario, to setup a point to point link between them instead. Special care must also be taken when setting up a virtual link to an HA firewall. The endpoint setting up a link to the HA firewall must setup 3 separate links: one to the shared, one the master and one to the slave router id of the firewall.

4.4.3. Dynamic Routing Policy

Overview
In a dynamic routing environment, it is important for routers to be able to regulate to what extent they will participate in the routing exchange. It is not feasible to accept or trust all received routing information, and it might be crucial to avoid that parts of the routing database gets published to other routers.

For this reason, NetDefendOS provides a *Dynamic Routing Policy*, which is used to regulate the flow of dynamic routing information.

A Dynamic Routing Policy rule filters either statically configured or OSPF learned routes according to parameters like the origin of the routes, destination, metric and so on. The matched routes can be controlled by actions to be either exported to OSPF processes or to be added to one or more routing tables.

The most common usages of Dynamic Routing Policy are:

- Importing OSPF routes from an OSPF process into a routing table.
- Exporting routes from a routing table to an OSPF process.
- Exporting routes from one OSPF process to another.

**Note**  
By default, NetDefendOS will not import or export any routes. In other words, for dynamic routing to be meaningful, it is mandatory to define at least one Dynamic Routing Policy rule.

### Example 4.6. Importing Routes from an OSPF AS into the Main Routing Table

In this example, the routes received using OSPF will be added into the main routing table. First of all a Dynamic Routing Policy filter needs to be created. The filter needs to have a name, in this example `ImportOSPFRoutes` is used, as it explains what the filter does.

The filter must also specify from what OSPF AS the routes should be imported. In this example, a pre-configured OSPF AS named `as0` is used.

Depending on how your routing topology looks like you might want to just import certain routes using the `Destination Interface/Destination Network` filters, but in this scenario all routes that are within the `all-nets` network (which is the same as specifying the IP address `0.0.0.0/0`) are allowed.

**CLI**
```
gw-world:/> add DynamicRoutingRule OSPFProcess=as0 Name=ImportOSPFRoutes DestinationNetworkExactly=all-nets
```

**Web Interface**

1. Go to **Routing > Dynamic Routing Rules > Add > Dynamic routing policy rule**
2. Specify a suitable name for the filter, in this case `ImportOSPFRoutes`
3. In the **Select OSPF Process**, select `as0`
4. Choose all-nets in the **Exactly Matches** dropdown control
5. Click **OK**

The next step is to create a Dynamic Routing Action that will do the actual importing of the routes into a routing table. Specify the destination routing table that the routes should be added to, in this case `main`.

**CLI**
```
gw-world:/> cc DynamicRoutingRule ImportOSPFRoutes
```
Example 4.7. Exporting the Default Route into an OSPF AS

In this example, the default route from the main routing table will be exported into an OSPF AS named as0. First, add a dynamic routing policy filter that matches the main routing table and the default route:

**CLI**

```
gw-world:/ImportOSPFRoutes> add DynamicRoutingRule OSPFProcess=as0 name=ExportDefRoute RoutingTable=MainRoutingTable DestinationInterface=wan DestinationNetworkExactly=all-nets
```

**Web Interface**

1. Go to Routing > Dynamic Routing Rules
2. Click on the recently created ImportOSPFRoutes
3. Go to OSPF Routing Action > Add > DynamicRoutingRuleAddRoute
4. In Destination, add the main routing table to the Selected list
5. Click OK

Then, create an OSPF Action that will export the filtered route to the specified OSPF AS:

**CLI**

```
gw-world:/ExportDefRoute/> cc DynamicRoutingRule ExportDefRoute
```

```
gw-world:/ExportDefRoute/> add DynamicRoutingRuleExportOSPF ExportToProcess=as0
```

**Web Interface**

1. Go to Routing > Dynamic Routing Rules
2. Click on the newly created ExportDefRoute
3. Go to OSPF Action > Add > DynamicRoutingRuleExportOSPF
4. For Export to process choose as0
5. Click OK
4.5. Multicast Routing

4.5.1. Overview

Certain types of Internet interactions, such as conferencing and video broadcasts, require a single client or host to send the same packet to multiple receivers. This could be achieved through the sender duplicating the packet with different receiving IP addresses or by a broadcast of the packet across the Internet. These solutions waste large amounts of sender resources or network bandwidth and are therefore not satisfactory. An appropriate solution should also be able to scale to large numbers of receivers.

Multicast Routing solves the problem by the network routers themselves, replicating and forwarding packets via the optimum route to all members of a group. The IETF standards that enable Multicast Routing are:

1. Class D of the IP address space which is reserved for multicast traffic. Each multicast IP address represent an arbitrary group of recipients.
2. The Internet Group Membership Protocol (IGMP) allows a receiver to tell the network that it is a member of a particular multicast group.
3. Protocol Independent Multicast (PIM) is a group of routing protocols for deciding the optimal path for multicast packets.

Multicast routing operates on the principle that an interested receiver joins a group for a multicast by using the IGMP protocol. PIM routers can then duplicate and forward packets to all members of such a multicast group, thus creating a distribution tree for packet flow. Rather than acquiring new network information, PIM uses the routing information from existing protocols, such as OSPF, to decide the optimal path.

A key mechanism in the Multicast Routing process is that of Reverse Path Forwarding. For unicast traffic a router is concerned only with a packet's destination. With multicast, the router is also concerned with a packets source since it forwards the packet on paths which are known to be downstream, away from the packet's source. This approach is adopted to avoid loops in the distribution tree.

By default multicast packets are routed by NetDefendOS to the core interface. SAT Multiplux rules are set up in the IP rule set in order to perform forwarding to the correct interfaces. This is demonstrated in the examples which follow.

Note

For multicast to function with an Ethernet interface on any D-Link Firewall, that interface must have multicast handling set to On or Auto. For further details on this see Section 3.3.2, “Ethernet”.

4.5.2. Multicast Forwarding using the SAT Multiplex Rule

The SAT Multiplex rule is used to achieve duplication and forwarding of packets through more than one interface. This feature implements multicast forwarding in NetDefendOS, where a multicast packet is sent through several interfaces. Note that, since this rule overrides the normal routing tables, packets that should be duplicated by the multiplex rule needs to be routed to the core interface.

By default, the multicast IP range 224.0.0.0/4 is always routed to core and does not have to be manually added to the routing tables. Each specified output interface can individually be configured with static address translation of the destination address. The Interface field in the Interface/Net Tuple dialog may be left empty if the IPAddress field is set. In this case, the output interface will be determined by a route lookup on the specified IP address.
The multiplex rule can operate in one of two modes:

**Use IGMP**

The traffic flow specified by the multiplex rule must have been requested by hosts using IGMP before any multicast packets are forwarded through the specified interfaces. This is the default behaviour of NetDefendOS.

**Not using IGMP**

The traffic flow will be forwarded according to the specified interfaces directly without any inference from IGMP.

*Note*

Since the Multiplex rule is a SAT rule, an Allow or NAT rule has to be specified together with the Multiplex rule.

### 4.5.2.1. Multicast Forwarding - No Address Translation

This scenario describes how to configure multicast forwarding together with IGMP. The multicast sender is 192.168.10.1 and generates the multicast streams 239.192.10.0/24:1234. These multicast streams should be forwarded from interface wan through the interfaces if1, if2 and if3. The streams should only be forwarded if some host has requested the streams using the IGMP protocol. The example below only covers the multicast forwarding part of the configuration. The IGMP configuration can be found below in Section 4.5.3.1, “IGMP Rules Configuration - No Address Translation”.

*Figure 4.4. Multicast Forwarding - No Address Translation*

*Note*

Remember to add an Allow rule matching the SAT Multiplex rule.
Example 4.8. Forwarding of Multicast Traffic using the SAT Multiplex Rule

In this example, we will create a multiplex rule in order to forward the multicast groups 239.192.10.0/24:1234 to the interfaces if1, if2 and if3. All groups have the same sender 192.168.10.1 which is located somewhere behind the wan interface. The multicast groups should only be forwarded to the out interfaces if clients behind those interfaces have requested the groups using IGMP. The following steps need to be executed to configure the actual forwarding of the multicast traffic. IGMP has to be configured separately.

Web Interface

A. Create a custom service for multicast called multicast_service:

1. Go to Objects > Services > Add > TCP/UDP
2. Now enter:
   - Name: multicast_service
   - Type: UDP
   - Destination: 1234

B. Create an IP rule:

1. Go to Rules > IP Rules > Add > IP Rule
2. Under General enter:
   - Name: a name for the rule, eg. Multicast_Multiplex
   - Action: Multiplex SAT
   - Service: multicast_service
3. Under Address Filter enter:
   - Source Interface: wan
   - Source Network: 192.168.10.0/24
   - Destination Interface: core
   - Destination Network: 239.192.10.0/24
4. Click the Multiplex SAT tab and add the output interfaces if1, if2 and if3 one at a time. For each interface, leave the IPAddress field blank since no destination address translation is wanted.
5. Make sure the forwarded using IGMP checkbox is set
6. Click OK

4.5.2.2. Multicast Forwarding - Address Translation Scenario

Figure 4.5. Multicast Forwarding - Address Translation
This scenario is based on the previous scenario but now we are going to translate the multicast group. When the multicast streams 239.192.10.0/24 are forwarded through the if2 interface, the multicast groups should be translated into 237.192.10.0/24. No address translation should be made when forwarding through interface if1. The configuration of the corresponding IGMP rules can be found below in Section 4.5.3.2, “IGMP Rules Configuration - Address Translation”.

**Caution**
As previously noted, remember to add an Allow rule matching the SAT Multiplex rule.

---

**Example 4.9. Multicast Forwarding - Address Translation**

The following SAT Multiplex rule needs to be configured to match the scenario described above:

**Web Interface**

A. Create a custom service for multicast called `multicast_service`:

1. Go to `Objects > Services > Add > TCP/UDP`
2. Now enter:
   - **Name**: `multicast_service`
   - **Type**: UDP
   - **Destination**: 1234

B. Create an IP rule:

1. Go to `Rules > IP Rules > Add > IP Rule`
2. Under **General** enter:
   - **Name**: a name for the rule, eg. `Multicast_Multiplex`
   - **Action**: Multiplex SAT
   - **Service**: multicast_service
3. Under **Address Filter** enter:
   - **Source Interface**: wan
   - **Source Network**: 192.168.10.1
4.5.3. IGMP Configuration

IGMP signaling between hosts and routers can be divided into two categories:

**IGMP Reports** Reports are sent from hosts towards the router when a host wants to subscribe to new multicast groups or change current multicast subscriptions.

**IGMP Queries** Queries are IGMP messages sent from the router towards the hosts in order to make sure that it will not close any stream that some host still wants to receive.

Normally, both these types of rules has to be specified for IGMP to work. One exception to this is if the multicast source is located on a network directly connected to the router. In this case, no query rule is needed.

A second exception is if a neighbouring router is statically configured to deliver a multicast stream to the D-Link Firewall. In this case also, an IGMP query would not have to be specified.

NetDefendOS supports two IGMP modes of operation - Snoop and Proxy.

**Figure 4.6. Multicast Snoop**
In Snoop mode, the router will act transparently between the hosts and another IGMP router. It will not send any IGMP Queries. It will only forward queries and reports between the other router and the hosts. In Proxy mode, the router will act as an IGMP router towards the clients and actively send queries. Towards the upstream router, it will be acting as a normal host, subscribing to multicast groups on behalf of its clients.

4.5.3.1. IGMP Rules Configuration - No Address Translation

This example describes the IGMP rules needed for configuring IGMP according to the No Address Translation scenario described above. We want our router to act as a host towards the upstream router and therefore we configure IGMP to run in proxy mode.

Example 4.10. IGMP - No Address Translation

The following example requires a configured interface group IfGrpClients including interfaces if1, if2 and if3. The ip address of the upstream IGMP router is known as UpstreamRouterIP.

Two rules are needed. The first one is a report rule that allows the clients behind interfaces if1, if2 and if3 to subscribe for the multicast groups 239.192.10.0/24. The second rule, is a query rule that allows the upstream router to query us for the multicast groups that the clients have requested. The following steps need to be executed to create the two rules.

**Web Interface**

A. Create the first IGMP Rule.

1. Go to Routing > IGMP > IGMP Rules > Add > IGMP Rule
2. Under General enter:
   - Name: A suitable name for the rule, eg. Reports
   - Type: Report
   - Action: Proxy
   - Output: wan (this is the relay interface)
3. Under Address Filter enter:
   - Source Interface: IfGrpClients
4.5.3. IGMP Configuration

4.5.3.2. IGMP Rules Configuration - Address Translation

The following examples illustrate the IGMP rules needed to configure IGMP according to the Address Translation scenario described above in Section 4.5.2.2, “Multicast Forwarding - Address Translation Scenario”. We need two IGMP report rules, one for each client interface. If1 uses no address translation and if2 translates the multicast group to 237.192.10.0/24. We also need two query rules, one for the translated address and interface, and one for the original address towards if1.

Two examples are provided, one for each pair of report and query rule. The upstream multicast router uses IP UpstreamRouterIP.

Example 4.11. Configuration if1

The following steps needs to be executed to create the report and query rule pair for if1 which uses no address translation.

**Web Interface**

A. Create the first IGMP Rule.

1. Go to Routing > IGMP > IGMP Rules > Add > IGMP Rule
2. Under General enter:
   - **Source Interface**: wan
   - **Source Network**: UpstreamRouterIp
   - **Destination Interface**: core
   - **Destination Network**: auto
   - **Multicast Source**: 192.168.10.1
   - **Multicast Group**: 239.192.10.0/24
3. Click OK

B. Create the second IGMP Rule:

1. Again go to Routing > IGMP > IGMP Rules > Add > IGMP Rule
2. Under General enter:
   - **Name**: A suitable name for the rule, eg. Queries
   - **Type**: Query
   - **Action**: Proxy
   - **Output**: IfGrpClients (this is the relay interface)
3. Under Address Filter enter:
   - **Source Interface**: if1net
   - **Source Network**: if1net, if2net, if3net
   - **Destination Interface**: core
   - **Destination Network**: auto
   - **Multicast Source**: 192.168.10.1
   - **Multicast Group**: 239.192.10.0/24
4. Click OK
3. Under **Address Filter** enter:
   - **Source Interface:** if1
   - **Source Network:** if1net
   - **Destination Interface:** core
   - **Destination Network:** auto
   - **Multicast Source:** 192.168.10.1
   - **Multicast Group:** 239.192.10.0/24

4. Click **OK**

B. Create the second IGMP Rule:

1. Again go to **Routing > IGMP > IGMP Rules > Add > IGMP Rule**
2. Under **General** enter:
   - **Name:** A suitable name for the rule, eg. *Queries_if1*
   - **Type:** Query
   - **Action:** Proxy
   - **Output:** if1 *(this is the relay interface)*
3. Under **Address Filter** enter:
   - **Source Interface:** wan
   - **Source Network:** UpstreamRouterIp
   - **Destination Interface:** core
   - **Destination Network:** auto
   - **Multicast Source:** 192.168.10.1
   - **Multicast Group:** 239.192.10.0/24
4. Click **OK**

---

**Example 4.12. Configuration if2 - Group Translation**

The following steps need to be executed to create the report and query rule pair for if2 which translates the multicast group. Note that the group translated therefore the IGMP reports include the translated IP addresses and the queries will contain the original IP addresses.

**Web Interface**

A. Create the first IGMP Rule.

1. Go to **Routing > IGMP > IGMP Rules > Add > IGMP Rule**
2. Under **General** enter:
   - **Name:** A suitable name for the rule, eg. *Reports_if2*
3. Under Address Filter enter:
   - Source Interface: if2
   - Source Network: if2net
   - Destination Interface: core
   - Destination Network: auto
   - Multicast Source: 192.168.10.1
   - Multicast Group: 239.192.10.0/24
4. Click OK

B. Create the second IGMP Rule:

1. Again go to Routing > IGMP > IGMP Rules > Add > IGMP Rule
2. Under General enter:
   - Name: A suitable name for the rule, eg. Queries_if2
   - Type: Query
   - Action: Proxy
   - Output: if2 (this is the relay interface)
3. Under Address Filter enter:
   - Source Interface: wan
   - Source Network: UpstreamRouterIp
   - Destination Interface: core
   - Destination Network: auto
   - Multicast Source: 192.168.10.1
   - Multicast Group: 239.192.10.0/24
4. Click OK

**Advanced IGMP Settings**

There are a number of advanced settings which are global and apply to all interfaces which do not have IGMP settings explicitly specified for them. These global settings can be found in Chapter 13, Advanced Settings. Individual IGMP settings are found in the IGMP section of the administration interface.
4.6. Transparent Mode

4.6.1. Overview of Transparent Mode

Deploying D-Link Firewalls operating in Transparent Mode into an existing network topology can significantly strengthen security. It is simple to do and doesn't require reconfiguration of existing nodes. Once deployed, NetDefendOS can then allow or deny access to different types of services (for example HTTP) and in specified directions. As long as users of the network are accessing permitted services through the D-Link Firewall they are not aware of its presence. Transparent Mode is enabled by specifying a Switch Route instead of a standard Route.

A typical example of Transparent Mode's ability to improve security is in a corporate environment where there might be a need to protect different departments from one another. The finance department might require access to only a restricted set of services (HTTP for example) on the sales department's servers whilst the sales department might require access to a similarly restricted set of applications on the finance department's network. By deploying a single D-Link Firewall between the two department's networks, transparent but controlled access can be achieved using the Transparent Mode feature.

Another example might be an organisation allowing traffic between the external Internet and a range of public IP address' on an internal network. Transparent mode can control what kind of service is permitted to these IP addresses and in what direction. For instance the only services permitted in such a situation may be HTTP access out to the Internet.

4.6.2. Comparison with Routing mode

The D-Link Firewall can operate in two modes: Routing Mode or Transparent Mode. In Routing Mode, the D-Link Firewall performs all the functions of a Layer 3 router; if the firewall is placed into a network for the first time, or if network topology changes, the routing configuration must therefore be thoroughly checked to ensure that the routing table is consistent with the new layout. Reconfiguration of IP settings may be required for pre-existing routers and protected servers. This mode works well when complete control over routing is desired.

In Transparent Mode, where Switch Route is used instead of Route, the firewall acts in a way that has similarities to a switch; it screens IP packets and forwards them transparently to the correct interface without modifying any of the source or destination information on the IP or Ethernet levels. Two benefits of Transparent Mode are:

- When a client moves from one interface to another without changing IP address, it can still obtain the same services as before (for example HTTP, FTP) without routing reconfiguration.
- The same network address range can exist on several interfaces.

**Note**

*D-Link Firewalls need not operate exclusively in Transparent Mode but can combine Transparent Mode with Routing Mode to operate in a hybrid mode. That is to say, the firewall can have both Switch Routes as well as standard routes defined. It is also possible to create a hybrid case by applying address translation on otherwise transparent traffic.*

4.6.3. Transparent Mode Implementation

In transparent mode, NetDefendOS allows ARP transactions to pass through the D-Link Firewall, and determines from this ARP traffic the relationship between IP addresses, physical addresses and interfaces. NetDefendOS remembers this address information in order to relay IP packets to the correct receiver. During the ARP transactions, neither of the endpoints will be aware of the firewall's presence.
When beginning communication, a host will locate the target host's physical address by broadcasting an ARP request. This request is intercepted by NetDefendOS and it sets up an internal ARP Transaction State entry and broadcasts the ARP request to all the other switch-route interfaces except the interface the ARP request was received on. If NetDefendOS receives an ARP reply from the destination within a configurable timeout period, it will relay the reply back to the sender of the request, using the information previously stored in the ARP Transaction State entry.

During the ARP transaction, NetDefendOS learns the source address information for both ends from the request and reply. NetDefendOS maintains two tables to store this information: the Content Addressable Memory (CAM) and Layer 3 Cache. The CAM table tracks the MAC addresses available on a given interface and the Layer 3 cache maps an IP address to MAC address and interface. As the Layer 3 Cache is only used for IP traffic, Layer 3 Cache entries are stored as single host entries in the routing table.

For each IP packet that passes through the D-Link Firewall, a route lookup for the destination is done. If the route of the packet matches a Switch Route or a Layer 3 Cache entry in the routing table, NetDefendOS knows that it should handle this packet in a transparent manner. If a destination interface and MAC address is available in the route, NetDefendOS has the necessary information to forward the packet to the destination. If the route was a Switch Route, no specific information about the destination is available and the firewall will have to discover where the destination is located in the network. Discovery is done by NetDefendOS sending out ARP as well as ICMP (ping) requests, acting as the initiating sender of the original IP packet for the destination on the interfaces specified in the Switch Route. If an ARP reply is received, NetDefendOS will update the CAM table and Layer 3 Cache and forward the packet to the destination.

If the CAM table or the Layer 3 Cache is full, the tables are partially flushed automatically. Using the discovery mechanism of sending ARP and ICMP requests, NetDefendOS will rediscover destinations that may have been flushed.

4.6.4. Enabling Transparent Mode

Two steps are normally required to have NetDefendOS operate in Transparent Mode:

1. If desired, create a group of the interfaces that are to be transparent. Interfaces in a group can be marked as Security transport equivalent if hosts are to move freely between them.

2. Create Switch Routes and if applicable use the interface group created earlier. For the Network parameter, specify the range of IP addresses that will be transparent between the interfaces. When the entire firewall is working in Transparent Mode this range is normally all-nets.

4.6.5. High Availability with Transparent Mode

Switch Routes cannot be used with High Availability and therefore true transparent mode cannot be implemented with a NetDefendOS High Availability Cluster.

Instead of Switch Routes the solution in a High Availability setup is to use Proxy ARP to separate two networks. This is described further in Section 4.2.4, “Proxy ARP”. The key disadvantage with this approach is that clients will not be able to roam between NetDefendOS interfaces, retaining the same IP address.

4.6.6. Transparent Mode Scenarios

Scenario 1

The firewall in Transparent Mode is placed between an Internet access router and the internal network. The router is used to share the Internet connection with a single public IP address. The internal NAT:ed network behind the firewall is in the 10.0.0.0/24 address space. Clients on the internal network are allowed to access the Internet via the HTTP protocol.
Example 4.13. Setting up Transparent Mode - Scenario 1

**Web Interface**
Configure the interfaces:

1. Go to Interfaces > Ethernet > Edit (wan)
2. Now enter:
   - **IP Address**: 10.0.0.1
   - **Network**: 10.0.0.0/24
   - **Default Gateway**: 10.0.0.1
   - **Transparent Mode**: Enable
3. Click **OK**
4. Go to Interfaces > Ethernet > Edit (lan)
5. Now enter:
   - **IP Address**: 10.0.0.2
   - **Network**: 10.0.0.0/24
   - **Transparent Mode**: Enable
6. Click **OK**

Configure the rules:

1. Go to Rules > IP Rules > Add > IPRule
2. Now enter:
   - **Name**: HTTPAllow
   - **Action**: Allow
   - **Service**: http
   - **Source Interface**: lan
Scenario 2

Here the D-Link Firewall in Transparent Mode separates server resources from an internal network by connecting them to a separate interface without the need for different address ranges.

Figure 4.9. Transparent mode scenario 2

All hosts connected to LAN and DMZ (the lan and dmz interfaces) share the 10.0.0.0/24 address space. As this is configured using Transparent Mode any IP address can be used for the servers, and there is no need for the hosts on the internal network to know if a resource is on the same network or placed on the DMZ. The hosts on the internal network are allowed to communicate with an HTTP server on DMZ while the HTTP server on the DMZ can be reached from the Internet. The firewall is transparent between the DMZ and LAN while traffic can subjected to the IP rule set.

Example 4.14. Setting up Transparent Mode - Scenario 2

Configure a Switch Route over the LAN and DMZ interfaces for address range 10.0.0.0/24 (assume the WAN interface is already configured).

Configure the interfaces:

Similar as shown in the previous example, first, we need to specify the involving interfaces lan, and dmz using the example IP addresses for this scenario.

Interface Groups:

Similar as shown in the previous example. Configure both interfaces lan and dmz into the same group.
Switch Route:
Similar as shown in the previous example. Set up the switch route with the new interface group created earlier. Configure the rules:

1. Go to Rules > New Rule
2. The Rule Properties dialog will be displayed
3. Specify a suitable name for the rule, for instance HTTP-LAN-to-DMZ
4. Enter following:
   - **Action**: Allow
   - **Source Interface**: lan
   - **Destination Interface**: dmz
   - **Source Network**: all-nets
   - **Destination Network**: 10.1.4.10
5. Under the **Service** tab, choose http in the **Pre-defined** control
6. Click OK
7. Go to Rules > New Rule
8. The Rule Properties dialog will be displayed
9. Specify a suitable name for the rule, for instance HTTP-WAN-to-DMZ
10. Enter following:
    - **Action**: SAT
    - **Source Interface**: wan
    - **Destination Interface**: dmz
    - **Source Network**: all-nets
    - **Destination Network**: wan_ip
11. Under the **Service** tab, choose http in the **Pre-defined** control
12. Under the **Address Translation** tab, choose **Destination IP Address** and enter 10.1.4.10 in the **New IP Address** control.
13. Click OK
14. Go to Rules > New Rule
15. The Rule Properties dialog will be displayed
16. Specify a suitable name for the rule, for instance HTTP-LAN-to-DMZ
17. Enter following:
    - **Action**: Allow
    - **Source Interface**: wan
    - **Destination Interface**: dmz
    - **Source Network**: all-nets
    - **Destination Network**: wan_ip
18. Under the **Service** tab, choose http in the **Pre-defined** control
19. Click OK

**Web Interface**
Configure the interfaces:
1. Go to Interfaces > Ethernet > Edit (lan)
2. Now enter:
   - IP Address: 10.0.0.1
   - Network: 10.0.0.0/24
   - Transparent Mode: Disable
   - Add route for interface network: Disable
3. Click OK
4. Go to Interfaces > Ethernet > Edit (dmz)
5. Now enter:
   - IP Address: 10.0.0.2
   - Network: 10.0.0.0/24
   - Transparent Mode: Disable
   - Add route for interface network: Disable
6. Click OK

Configure the interface groups:

1. Go to Interfaces > Interface Groups > Add > InterfaceGroup
2. Now enter:
   - Name: TransparentGroup
   - Security/Transport Equivalent: Disable
   - Interfaces: Select lan and dmz
3. Click OK

Configure the routing:

1. Go to Routing > Main Routing Table > Add > SwitchRoute
2. Now enter:
   - Switched Interfaces: TransparentGroup
   - Network: 10.0.0.0/24
   - Metric: 0
3. Click OK

Configure the rules:

1. Go to Rules > IP Rules > Add > IPRule
2. Now enter:
   - Name: HTTP-LAN-to-DMZ
   - Action: Allow
   - Service: http
   - Source Interface: lan
   - Destination Interface: dmz
   - Source Network: 10.0.0.0/24
   - Destination Network: 10.1.4.10
3. Click OK
4. Go to Rules > IP Rules > Add > IPRule
5. Now enter:
   - Name: HTTP-WAN-to-DMZ
   - Action: SAT
   - Service: http
   - Source Interface: wan
   - Destination Interface: dmz
   - Source Network: all-nets
   - Destination Network: wan_ip
   - Translate: Select Destination IP
   - New IP Address: 10.1.4.10
6. Click OK
7. Go to Rules > IP Rules > Add > IPRule
8. Now enter:
   - Name: HTTP-WAN-to-DMZ
   - Action: Allow
   - Service: http
   - Source Interface: wan
   - Destination Interface: dmz
   - Source Network: all-nets
   - Destination Network: wan_ip
9. Click OK