Configuring Circuits and Virtual Circuits in the Exinda Optimizer

Introduction

The introduction of circuits and virtual circuits into the Exinda Optimizer series of products provides an extra two layers for optimization policies. Circuits and virtual circuits offer greater flexibility and control when setting up optimization policies, particularly for networks with multiple links, multiple locations and/or asynchronous connections (i.e. ADSL). This guide aims to present examples and guidance on how to setup and configure networks using the new circuit and virtual circuit technologies built into Exinda Optimizer as of version 4.50.

What are Circuits and Virtual Circuits?

Circuits can be thought of as physical links, while virtual circuits can be thought of as virtual links within a circuit. Using virtual circuits, a physical link can be divided up into one or more virtual links based on source/destination subnet(s) and/or direction.

The following parameters can be configured for circuits and virtual circuits.

Circuit
- Name
- Number
- Inbound (upstream) bandwidth
- Outbound (downstream) bandwidth

Virtual Circuit
- Name
- Number
- Bandwidth
- Source/destination subnet(s)
- Direction

The Exinda Optimizer is a hierarchical policy-based system, so the numbers assigned to circuits, virtual circuits and policies are important. A packet passing through the Exinda Optimizer will traverse the circuits, virtual circuits and policies, in order, until it is matched to a policy.

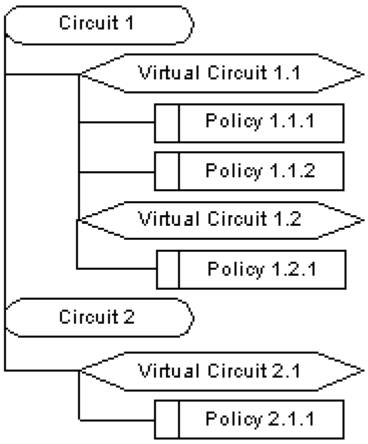


Figure 1: Structure of circuits, virtual circuits and policies.

Figure 1, above shows how circuits, virtual circuits and policies are structured within the Exinda Optimizer. These components are referenced using "dot notation", for example, Policy 1.2.1 belongs to circuit number 1 and virtual circuit number 2.

An Example: Configuring Simple Networks

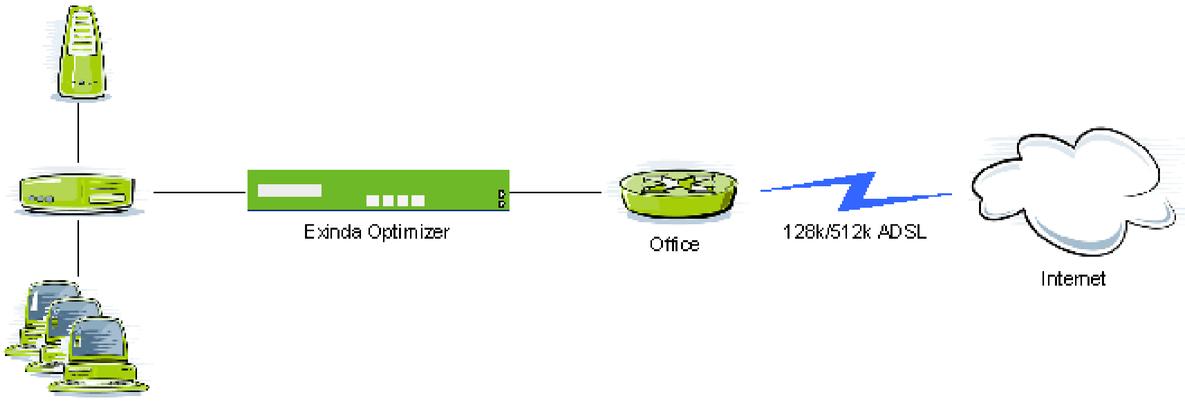


Figure 2: An example simple ADSL network diagram.

Figure 2, above, represents an office that has a single (128kbps upload, 512kbps download) ADSL connection to the Internet. The only circuit (link) present is the ADSL connection, which is setup as the only circuit in the Exinda Optimizer, viz

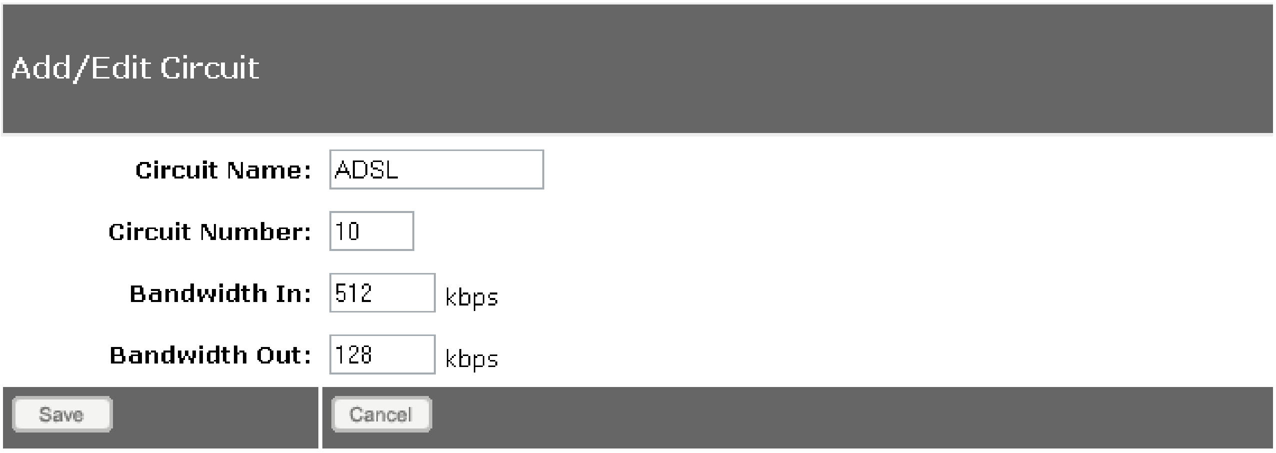


Figure 3: Setting up an ADSL circuit.

Since ADSL connections have different inbound and outbound transfer speeds, a virtual circuit must be setup for each direction within the ADSL circuit. This allows bandwidth to be allocated separately for inbound and outbound traffic. If the link was synchronous (such as ISDN or SDSL) only one virtual circuit would be required.

The following diagrams show the setup screens for each virtual circuit (inbound and outbound).

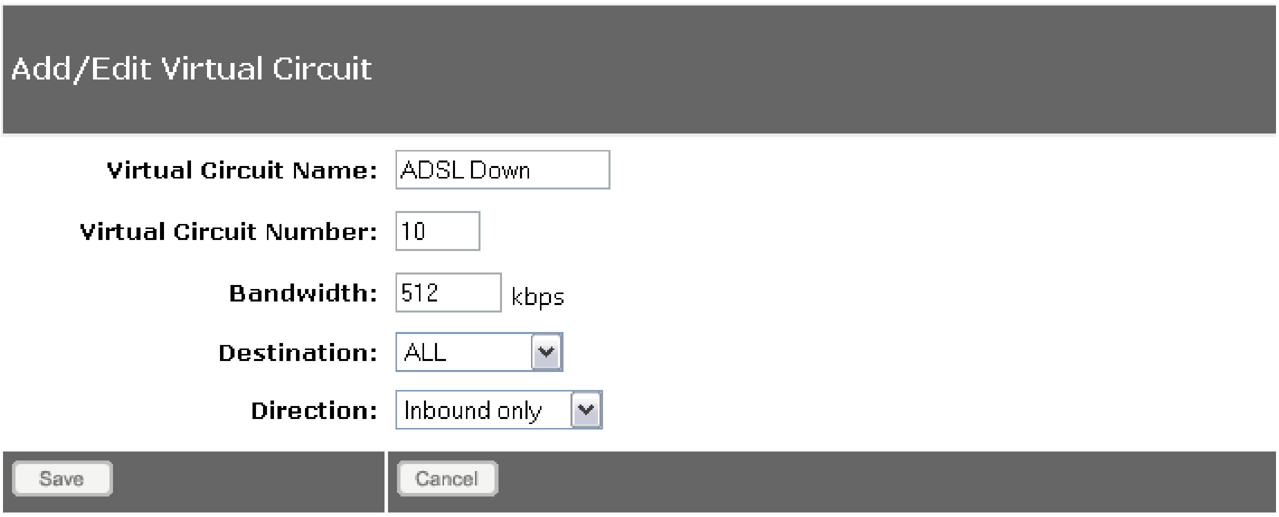


Figure 4a: Setting up the downstream ADSL virtual circuit.

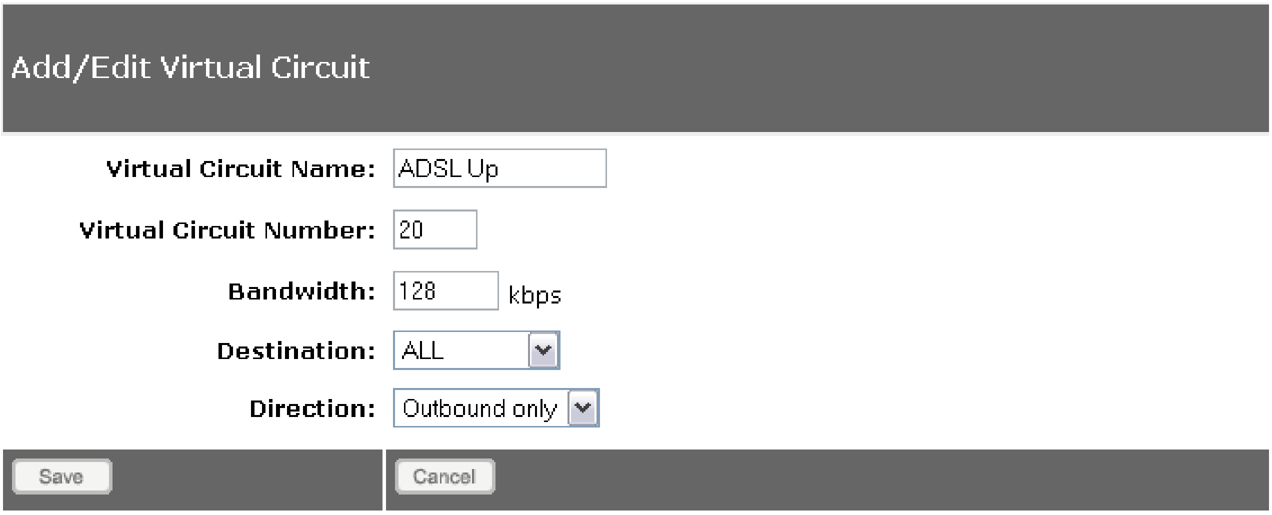


Figure 4b: Setting up the upstream ADSL virtual circuit.

The above setup can also be achieved using the CLI (command line interface) "conf" command. This is useful if a large number of devices need to be configured (using copy and paste). The following text shows the input required to perform this setup using the "conf" command (refer to the "Exinda Optimizer CLI Reference Guide" for further information).

```
optimizer
  circuit 10 "ADSL" 512 128
    vcircuit 10 "ADSL Down" 512 "ALL" inbound
    vcircuit 20 "ADSL Up" 128 "ALL" outbound
```

The resulting configuration is as shown in Figure 5, below.

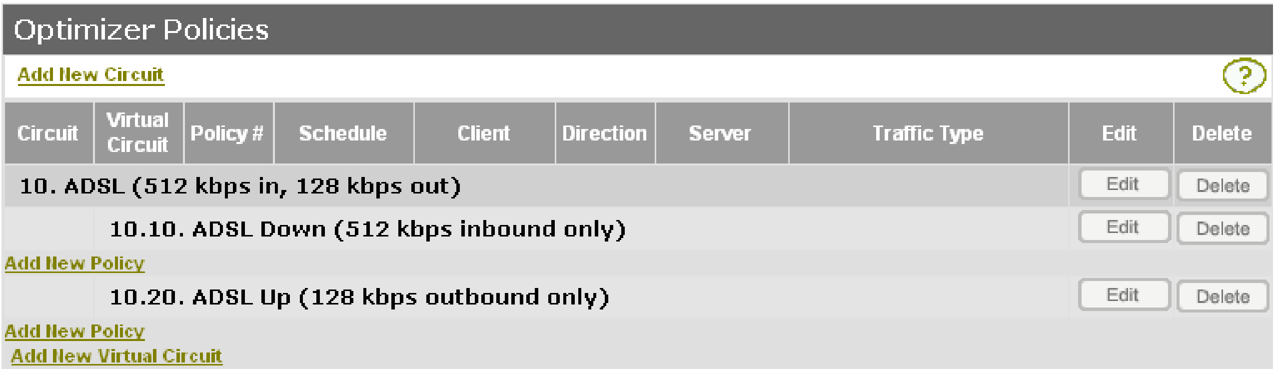


Figure 5: View of the circuits and virtual circuits after the initial configuration.

This example so far, only shows how to setup an asynchronous (ADSL) link; however, for synchronous links (such as ISDN) only one virtual circuit is required since the upstream and downstream bandwidth is the same. See Figure 6, below.

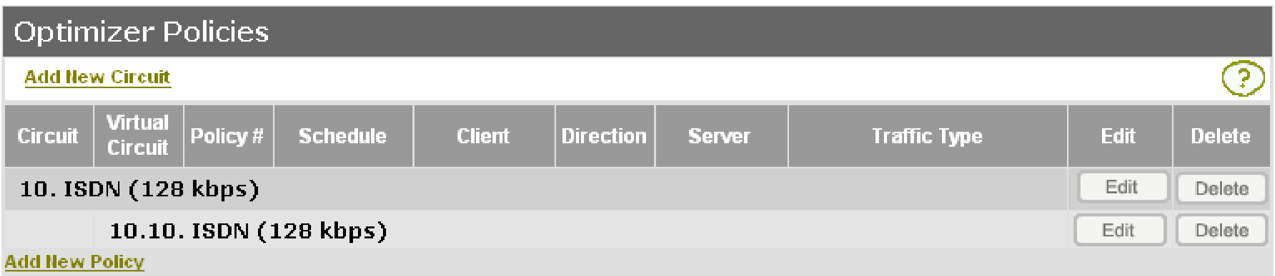


Figure 6: View of the policies screen if only one virtual circuit is required.

Once the virtual circuits have been established, optimizer policies can then be configured. A set of policies is required for each virtual circuit. Figure 7 shows an example policy set for the inbound and outbound virtual circuits.

Optimizer Policies

Add New Circuit

Circuit	Virtual Circuit	Policy #	Schedule	Client	Direction	Server	Traffic Type	Edit	Delete
10. ADSL (512 kbps in, 128 kbps out)								Edit	Delete
	10.10. ADSL Down (512 kbps inbound only)							Edit	Delete
		10	ALWAYS	Fast_In (300 - 512 kbps, priority 1)				Edit	Delete
				ALL	◀ ▶	ALL	https (tcp, port 443)		
				ALL	◀ ▶	ALL	https (tcp, port 443)		
		20	ALWAYS	Slow_In (50 - 512 kbps, priority 10)				Edit	Delete
				ALL	◀ ▶	ALL	pop3 (tcp, port 110)		
				ALL	◀ ▶	ALL	smtp (tcp, port 25)		
		30	ALWAYS	General_In (162 - 512 kbps, priority 5)				Edit	Delete
				ALL	◀ ▶	ALL	All traffic		
Add New Policy									
	10.20. ADSL Up (128 kbps outbound only)							Edit	Delete
		10	ALWAYS	Fast_Out (60 - 128 kbps, priority 1)				Edit	Delete
				ALL	◀ ▶	ALL	http (tcp, port 80)		
				ALL	◀ ▶	ALL	https (tcp, port 443)		
		20	ALWAYS	Slow_Out (20 - 128 kbps, priority 10)				Edit	Delete
				ALL	◀ ▶	ALL	pop3 (tcp, port 110)		
				ALL	◀ ▶	ALL	smtp (tcp, port 25)		
		30	ALWAYS	General_Out (48 - 128 kbps, priority 5)				Edit	Delete
				ALL	◀ ▶	ALL	All traffic		
Add New Policy									
Add New Virtual Circuit									

Figure 7: View of the virtual circuits after policy configuration.

Note that the bandwidth allocations for the inbound and outbound policies are different since 512kbps of downstream bandwidth is available but only 128kbps of upstream bandwidth is available. The following figure shows the bandwidth allocation for each virtual circuit.

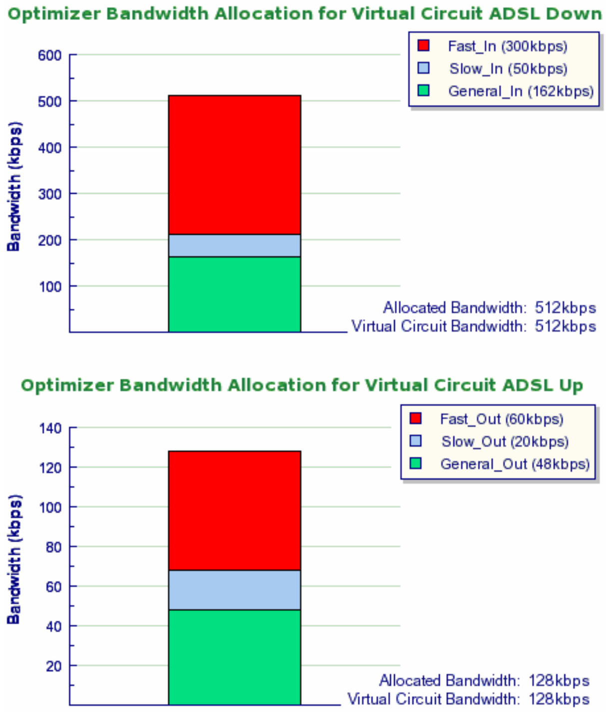


Figure 8: Outbound and inbound virtual circuit bandwidth allocation graphs.

The following text shows the input required to achieve this setup using the Exinda CLI (Command Line Interface) “conf” command.

```
optimizer
  circuit 10 "ADSL" 512 128
    vcircuit 10 "ADSL Down" 512 "ALL" inbound
      policy 10 "Fast_In"
        schedule 0
        minbw 300
        maxbw ALL
        priority 1
        action OPTIMIZE
        filter ALL any ALL "https (tcp, port 443)"
        filter ALL any ALL "https (tcp, port 443)"

      policy 20 "Slow_In"
        schedule 0
        minbw 50
        maxbw ALL
        priority 10
        action OPTIMIZE
        filter ALL any ALL "smtp (tcp, port 25)"
        filter ALL any ALL "pop3 (tcp, port 110)"

      policy 30 "General_In"
        schedule 0
        minbw 162
        maxbw ALL
        priority 5
        action OPTIMIZE
        filter ALL any ALL "All traffic"

    vcircuit 20 "ADSL Up" 128 "ALL" outbound
      policy 10 "Fast_Out"
        schedule 0
        minbw 60
        maxbw ALL
        priority 1
        action OPTIMIZE
        filter ALL any ALL "http (tcp, port 80)"
        filter ALL any ALL "https (tcp, port 443)"

      policy 20 "Slow_Out"
        schedule 0
        minbw 20
        maxbw ALL
        priority 10
        action OPTIMIZE
        filter ALL any ALL "pop3 (tcp, port 110)"
        filter ALL any ALL "smtp (tcp, port 25)"

      policy 30 "General_Out"
        schedule 0
        minbw 48
        maxbw ALL
        priority 5
        action OPTIMIZE
        filter ALL any ALL "All traffic"
```

Each policy set (within a virtual circuit) finishes with a “General” policy. This policy is used to “catch” any traffic that has not been matched by any previous policies and provides a way of controlling any other or unknown traffic on the network.

It should be noted that Exinda Optimizer also uses a fair bandwidth allocation system. For example, if a particular policy limits http traffic to a certain bandwidth, and there are many simultaneous http connections, then each http connection will receive a fair (proportional) allocation of bandwidth (i.e. no one http connection will hog the bandwidth).

For further information regarding policies, please refer to the Exinda Optimizer policy guide entitled “Configuring Policies and Filter Rules in the Exinda Optimizer”.

An Example: Configuring Complex Networks

Circuits and virtual circuits allow much greater control of traffic on more complex networks. Consider the network diagram shown in Figure 9, below.

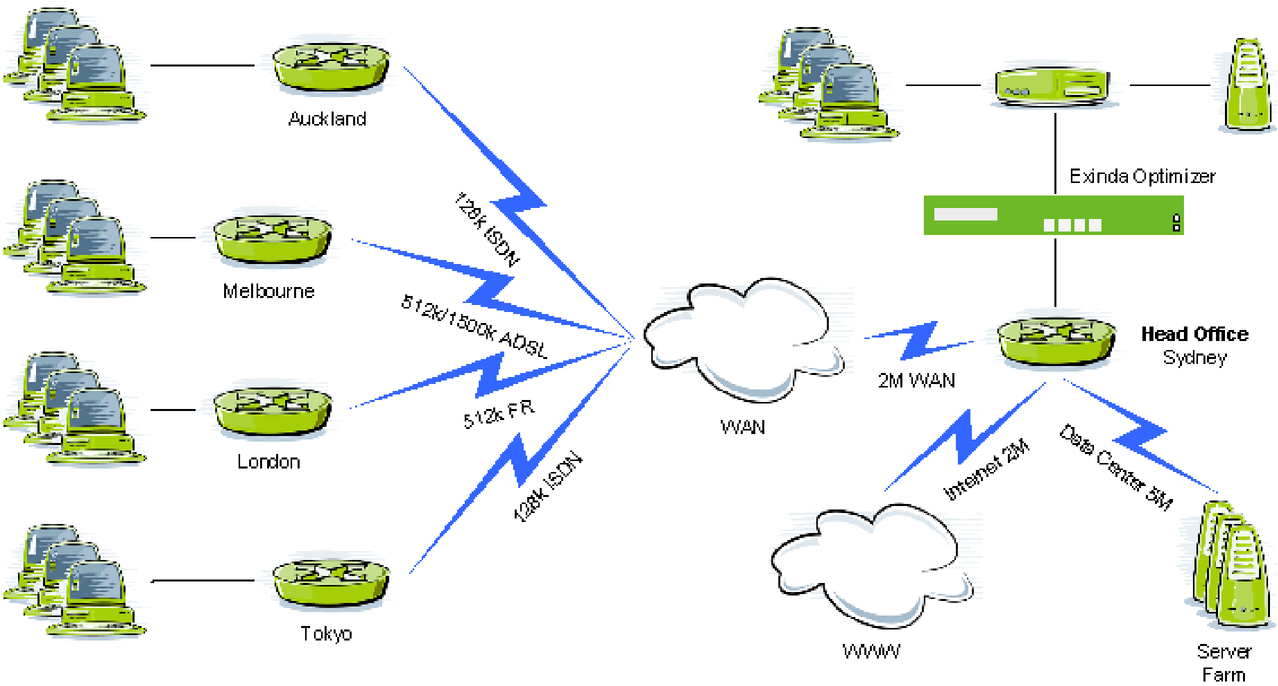
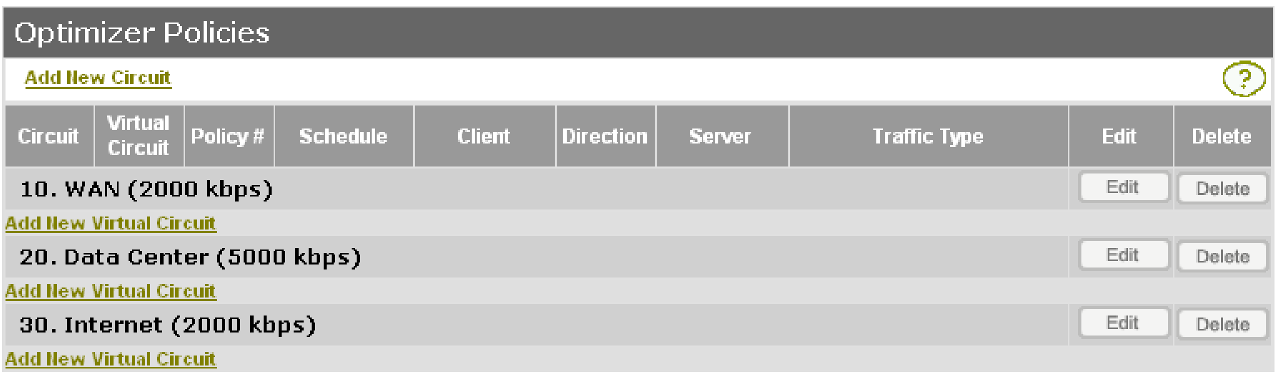


Figure 9: An example complex WAN network diagram.

Figure 9, above, shows an Exinda Optimizer placed between the LAN segment and a router at the Sydney head office. There are three physical links connecting the head office, viz

- A 2Mbps WAN connection to multiple branch offices
- A 5Mbps connection to servers in a data center
- A 2Mbps Internet connection

These three physical circuits should form three circuits within the Exinda Optimizer as shown in Figure 10, below.



Optimizer Policies

Add New Circuit

Circuit	Virtual Circuit	Policy #	Schedule	Client	Direction	Server	Traffic Type	Edit	Delete
10. WAN (2000 kbps)								Edit	Delete
Add New Virtual Circuit									
20. Data Center (5000 kbps)								Edit	Delete
Add New Virtual Circuit									
30. Internet (2000 kbps)								Edit	Delete
Add New Virtual Circuit									

Figure 10: Setup of circuits for the network pictured in Figure 9.

The Internet circuit has been entered last because it is the only link that does not have a specific destination. It therefore, acts as a "catch all" type circuit and must appear last given that the Exinda Optimizer is a hierarchical, policy-based system.

Virtual circuits can now be setup for each of the circuits. The virtual circuits for the Data Center and Internet links are quite straight forward to configure as they should simply be configured with the same parameters as their respective circuit configuration (see Figure 12).

The virtual circuits for the WAN link, however, are a little more complex. These need to be configured with the parameters of the link at the other end of the connection in mind, viz

- Auckland – 128kbps ISDN
- Melbourne – 512kbps upstream, 1500kbps downstream ADSL
- London – 512kbps Frame Relay
- Tokyo – 128kbps ISDN

Each of these connections will form a virtual circuit within the WAN circuit. Furthermore, the Melbourne link will require two virtual circuits, one for upstream and one for downstream. Note that the *upstream bandwidth* (outbound) from Melbourne to Sydney (512kbps) must be interpreted as *downstream bandwidth* (inbound) from the perspective of the Exinda Optimizer in the Sydney head office. The same also applies for the reverse direction.

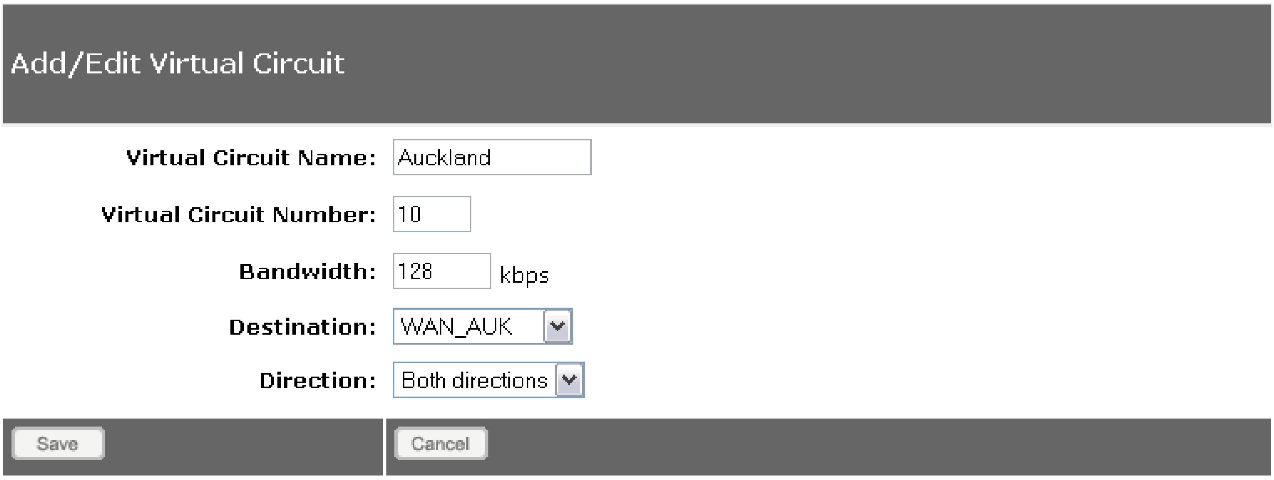


Figure 11: Setting up a virtual circuit for the network pictured in Figure 9.

For the destination field, simply select a network object. Network objects are groups of one or more subnets and can be configured by clicking the link on the left-hand side menu.

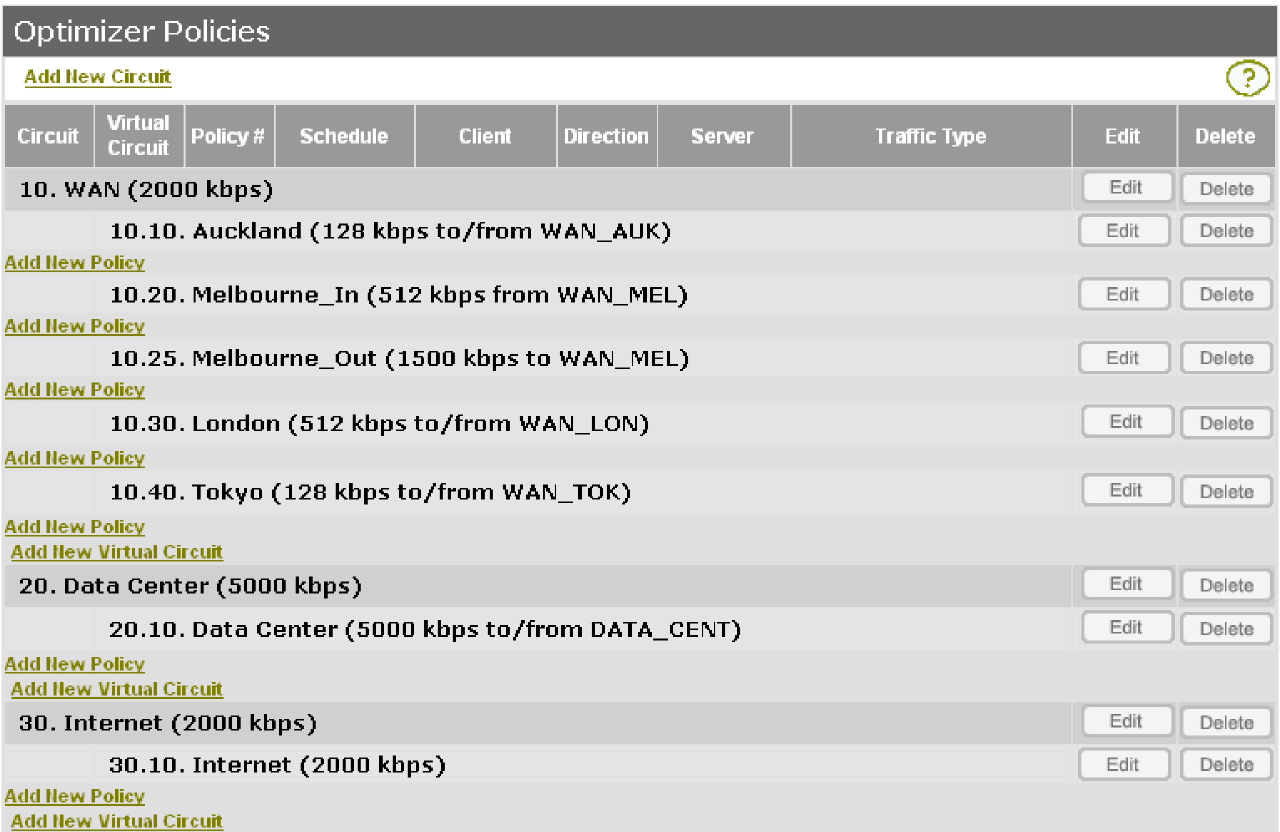


Figure 12: The completed setup of virtual circuits for the network pictured in Figure 9.

Each virtual circuit is now ready to have policies defined within it. The policies defined within a virtual circuit will only apply to that virtual circuit. This means that traffic between each location can be managed independently of the other locations.

It does not matter that the sum of the virtual circuit bandwidths exceed their respective circuit bandwidth because bandwidth is allocated proportionally within a circuit. For example, London may be the only site transferring data at any one point in time, so it uses it's full 512kbps allocation of the 2000kbps link. If all sites happen to be transferring data at any one point in time, they will all receive a proportion of the available 2000kbps. The only constraint is that an individual virtual circuit's bandwidth cannot exceed the bandwidth of it's respective circuit.

The following text shows the input required to achieve this setup using the "conf" command.

```
optimizer
  circuit 10 "WAN" 2000 2000
    vcircuit 10 "Auckland" 128 "WAN_AUK"
    vcircuit 20 "Melbourne_In" 512 "WAN_MEL" inbound
    vcircuit 25 "Melbourne_Out" 1500 "WAN_MEL" outbound
    vcircuit 30 "London" 512 "WAN_LON"
    vcircuit 40 "Tokyo" 128 "WAN_TOK"

  circuit 20 "Data Center" 5000 5000
    vcircuit 10 "Data Center" 5000 "DATA_CENT"

  circuit 30 "Internet" 2000 2000
    vcircuit 10 "Internet" 2000 "ALL"
```

Exinda Optimizer will only manage bandwidth for traffic that physically passes through it. To manage other traffic on the network in the above example (i.e. traffic between the remote offices and the data center), additional Exinda Optimizers would need to be placed at the remote office locations.

Placing Exinda Optimizers at the remote locations as well as at the head office has other benefits including

- The ability to monitor all traffic that passes through each remote office's link. This increases network visibility and assists with the troubleshooting and maintenance of individual remote office links.
- Tighter control of all traffic that passes through each remote office's link including the ability to optimize traffic to/from all destinations on your network from each remote office.

Recommended Reading

- "Configuring Policies and Filter Rules in the Exinda Optimizer"
- "Exinda Optimizer CLI Reference Guide"

For any further information, please contact Exinda Networks.

Exinda Networks Pty Ltd
Level 1, 235 Queen Street
Melbourne, VIC 3000, Australia
Phone: +61 (3) 9670 0714
Fax: +61 (3) 9670 0719
Email: info@exinda.com
Web: http://www.exinda.com

Recommended Reading

- "Configuring Policies and Filter Rules in the Exinda Optimizer"
- "Configuring Circuits and Virtual Circuits in the Exinda Optimizer"

Warranty

The information is this guide is supplied without warranty of any kind and is subject to change without notice. Exinda Networks will not be liable for any damages of any kind arising from the supply of this guide, regardless of the form of action, whether in contract, tort, strict liability or otherwise.

Trademarks