Installation and Deployment Guide: AutoBAHN System

Last Updated:	05-12-08
Activity:	JRA3
Work Item	WI-03
Dissemination Level	PU (Public)
Document Code	GN2-08-251

Authors:	Jacek Lukasik (PSNC), Ophelia Neofytou (GRNET-UoC), Afrodite Sevasti (GRNET), Stella-Maria Thomas (DANTE), Sue Tyley (DANTE)

Table of Contents

Table of Figures

1 Introduction

The GÉANT2 Automated Bandwidth Allocation across Heterogeneous Networks (AutoBAHN) system provides a user-friendly interface for instantiating dynamic circuits over global research and education (R&E) network infrastructures. As at November 2008, AutoBAHN has pre-production status; it will be offered as a full production service in 2009.

Researchers today in fields such as astronomy, physics and the environment often need dedicated channels to transport large volumes of data between varying locations at high rates with guaranteed levels of service. Internet Protocol (IP) networks provide always-on services for data transfer but cannot guarantee quality or resources for bulk transfers with time constraints. This is due to heterogeneity along end-to-end paths, traffic multiplexing, and open access to a large user base. On the other hand, fixed circuits interconnecting end-points participating in demanding research applications are costly and often result in under-utilisation of e-Infrastructures.

A dynamic circuit service addresses the limitations of IP networks and fixed circuits for certain use cases by isolating resources over existing infrastructures, reserving them, and providing quantity and quality guarantees at the level required for the time period required between the end-points involved. As soon as a circuit's resources are no longer necessary, they are released for another potential transfer between different end-points utilising the same resources.

This document describes the prerequisites and sets out the instructions to follow in order to install and deploy the AutoBAHN system over a network domain.

For a technical overview of the AutoBAHN system, please refer to the AutoBAHN public web page http://www.geant2.net/server/show/nav.2240.

This document covers the following topics:

- System requirements.
- Data plane configuration.
- Installation.
- Running AutoBAHN.

Please note that this document is continuously updated. The latest version is available from the AutoBAHN website. For any questions or open issues, contact: gn2-jra2@dante.org.uk.

2 System Requirements

This section outlines the system and configuration requirements for the server to host the AutoBAHN system. It covers the following topics:

- Hardware.
- Operating system.
- Required software.
- Optional software.
- Default ports.

2.1 Hardware

Configurable Element	Minimum	Recommended
Central Processing Unit (CPU)	500 megahertz (MHz)	1 gigahertz (GHz)
MEM	512 megabytes (MB)	1 gigabyte (GB)
Disk space	50 MB	500 MB (for long-term logs)
Network Information Centre (NIC)	N/A	1 Fast Ethernet NIC

Table 2.1: Hardware – minimum and recommended specifications

2.2 Operating System

AutoBAHN is certified to run on any Operating System (OS) that supports Java.

Linux is recommended (AutoBAHN has been tested on Fedora, Debian, Suse, Ubuntu and Windows XP).

2.3 Required Software

The following software is required:

- Java 1.5 or higher (1.6 recommended).
- Jetty 6.x or higher, or Tomcat 5.x or higher.
- PostgreSQL 8.x or higher. (Any other SQL Relational Database Management System (RDBMS) can be used, so long as it is supported by Hibernate).
- Quagga 0.99.6 or higher.

2.4 Optional Software

The following software is optional:

- Apache Ant 1.6 or higher. Can be used as an alternative to a .war file to build and install the Inter-Domain Manager (IDM) module of AutoBAHN.

2.5 Default Ports

The following default ports are used (these can be reconfigured):

Default Port	Used For…
8443	Inter-domain: Inter-Domain Manager (IDM)-IDM
8443	Intra-domain: IDM-Domain Manager (DM) – as for inter-domain
5432	Database access within the domain
25	Simple Mail Transfer Protocol (SMTP) for sending mails with notifications (optional)
2607	Intra-domain: IDM-Quagga Application Program Interface (API) access
4000	Inter-domain sending Large Hadron Collider (LHC) Software Applications (LSA)
4001	Inter-domain: receiving LSA

Table 2.2: Default port configuration

3 Data Plane Configuration

This section provides examples of the minimum configuration required to enable a domain to participate in the AutoBAHN service cloud. The cases presented are:

- Case A: End-host at the domain's ingress Point-of-Presence from GÉANT.
- Case B: End-host connected to any Point-of-Presence in the domain.

3.1 Case A: End-Host at the Domain's Ingress PoP from GÉANT2

In Case A, a host is connected on the Point of Presence (PoP) where the domain peers with the AutoBAHN cloud. The example is taken from GRNET (see **Figure 3.1**).

The configuration is minimal. One end-host with two 1 GE NICs is located in the GÉANT2 PoP in Greece, co-located with a GRNET PoP. One of the NICs is connected to the public Internet. The other terminates the AutoBAHN circuit on the host.

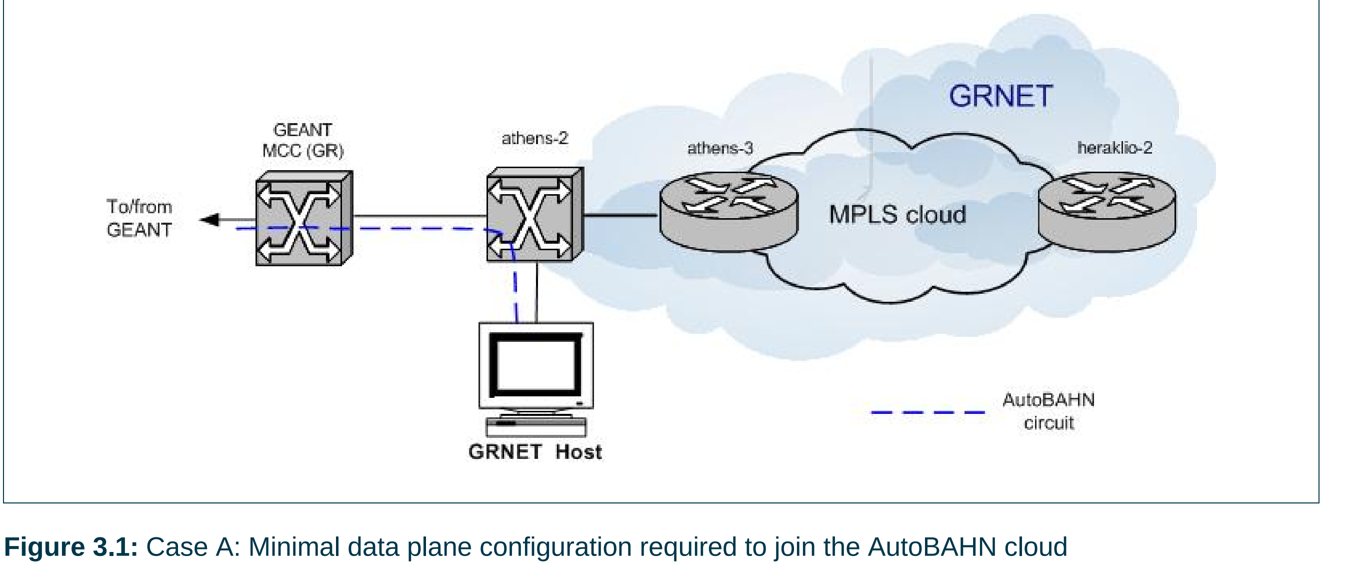

Figure 3.1: Case A: Minimal data plane configuration required to join the AutoBAHN cloud

The AutoBAHN circuit arrives at GRNET through one of the GE client interfaces on the GÉANT2 PoP Metro Core Connect (MCC) in Athens. The circuit is received at a Layer 2 (L2) switch on the border of GRNET and then switched directly to the host. In this case, the GRNET part of the AutoBAHN circuit data plane comprises a single L2 switch.

3.2 Case B: End-Host Connected to Any PoP in the Domain

In Case B, the host is connected on any PoP of the domain other than the peering PoP with GÉANT2. The example is again taken from GRNET (see **Figure 3.2**).

The end-host has two 1 GE NICs, one of which is connected to the public Internet. The other terminates the AutoBAHN circuit on the host.

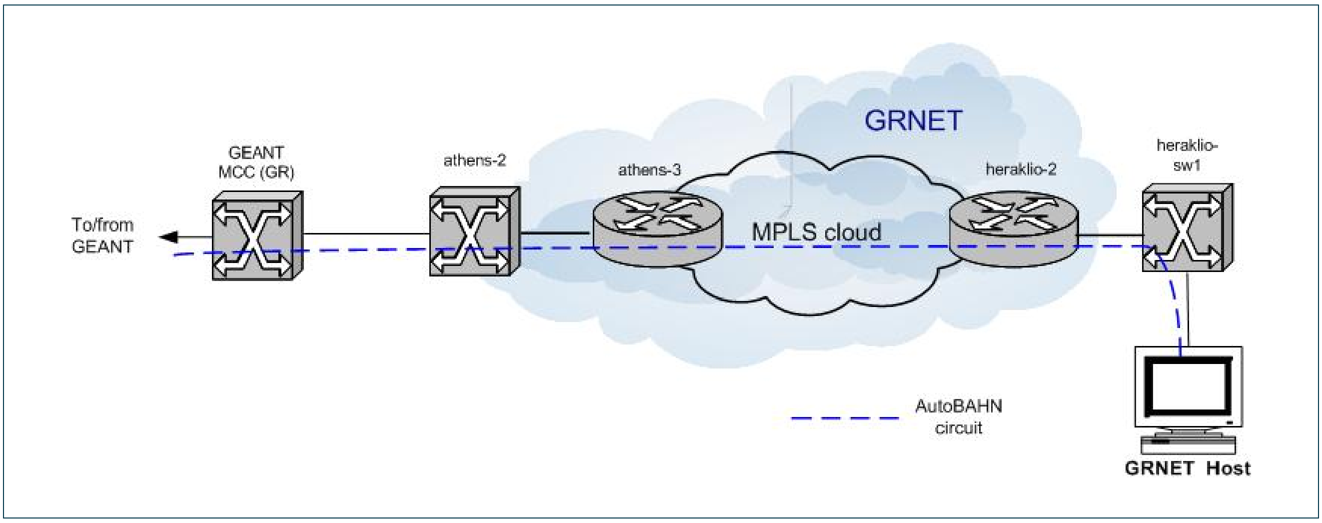

Figure 3.2: Case B: Minimal data plane configuration required to join the AutoBAHN cloud

GRNET implements AutoBAHN circuits within its core using L2 Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). The AutoBAHN circuit arrives at GRNET through one of the GE client interfaces on the GÉANT2 PoP MCC in Athens. The circuit is switched through an L2 switch to the border router of the GRNET MPLS cloud in Athens and then through an L2 MPLS VPN to the border router of the GRNET MPLS cloud in Heraklion (Crete). The circuit is then switched through another L2 switch to the end-host.

In **Figure 3.2** the host is assumed to reside in a PoP of the domain. If this is not feasible (for example, the end-host is located in a campus, lab, or similar that does not belong to the domain), then the AutoBAHN circuit should reach the end-host using a fixed or virtual L2 circuit. The detailed implementation depends on the existing technologies at the last mile.

The AutoBAHN team is investigating possible solutions to the last-mile problem but at present manual configuration is required.

4 Installation

This section describes how to install the following software:

- PostgreSQL.
- Quagga, including how to configure Generic Routing Encapsulation tunnels.
- Ant.
- AutoBAHN system.

4.1 Installing PostgreSQL

To install PostgreSQL, download the latest release from http://www.postgresql.org/ and follow the installation instructions from the documentation.

4.2 Installing Quagga

Quagga is used as the routing daemon for the AutoBAHN system to exchange topology information. One instance of Quagga must be operating on each AutoBAHN server.

To install Quagga, take the following steps:

1. Download the latest tarball from http://www.quagga.net/, and unpack it to a convenient location.
2. Configure and compile it following the installation instructions.
3. Insert the following lines into `/etc/services`, if they are not already present:

```
zebrasrv 2600/tcp # zebra service
zebra 2601/tcp # zebra vty
ripd 2602/tcp # RIPd vty
ripngd 2603/tcp # RIPngd vty
ospfd 2604/tcp # OSPFd vty
bgpd 2605/tcp # BGPd vty
```

```
ospf6d 2606/tcp # OSPF6d vty
ospfapi 2607/tcp # ospfapi
isisd 2608/tcp # ISISd vty
```

4.2.1 Configuring Generic Routing Encapsulation Tunnels

Generic Routing Encapsulation (GRE) tunnels are needed for control plane communication, that is, to implement communication channels between the different AutoBAHN servers in the domains participating in the AutoBAHN service cloud.

For a Linux OS, you need the `ip_gre.o` kernel module to configure the tunnels.

The following guidelines for configuring GRE tunnels are based on the configuration used in the demonstration of AutoBAHN run during the 5th GÉANT2 Technical Workshop (Rome, Italy, January 2008), where four AutoBAHN servers were deployed for the GRNET, HEAnet, GÉANT2 and PIONIER domains (see **Figure 4.1**).

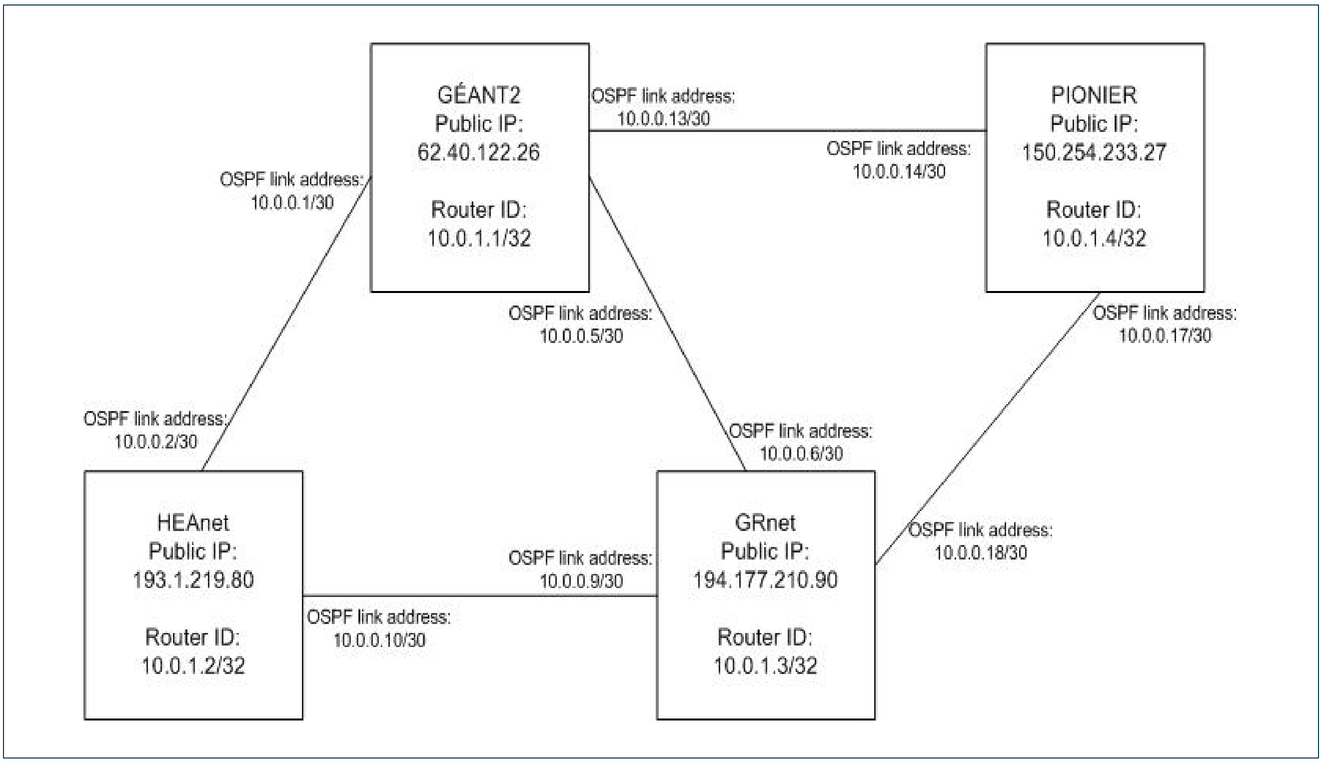

Figure 4.1: Example of control plane tunnel configurations among four AutoBAHN-enabled domains

The guidelines provided here refer to the tunnel between the GRNET and GÉANT2 AutoBAHN hosts. The IP address of the local end of tunnel is 194.177.210.90 (the IP address of the GRNET AutoBAHN host) while the IP address of the remote end of the tunnel is 62.40.122.26 (the IP address of the GÉANT2 AutoBAHN host).

The guidelines assume that the following values have been agreed among the domains:

Attribute	Assumed Value
Private addressing for Open Shortest Path First (OSPF)	OSPF area = 100 network 10.0.0.0/8 area 0.0.0.100
HOST OSPF loopback address (router ID)	GRNET 10.0.1.3/32
Address for GRE tunnel (OSPF link addresses)	Remote to Local: 10.0.0.5/30 <----> 10.0.0.6/30
Tunnel interface name	gre1

Table 4.1: Assumptions made in GRE tunnel configuration

To configure the tunnel above, take the following steps:

1. Execute the following commands (where `gre1` is the tunnel interface name):

```
#ip tunnel add gre1 mode gre remote 62.40.122.26 local 194.177.210.90 ttl 255
#ip link set gre1 up multicast on
#ip addr add 10.0.0.6/30 brd + dev gre1
```

2. Edit the `ospfd.conf` configuration file so that it contains the following:

```
!
! Zebra configuration saved from vty
router ospf
  ospf router-id 10.0.1.3
  network 10.0.0.0/8 area 0.0.0.100
  capability opaque
!
password XXXX
log file /var/log/quagga/ospfd.log
!
```

Notes:
The password is required to telnet to the ospfd (`#telnet localhost ospfd`, or see the documentation on the Quagga site for more information).
The log file is not mandatory, but may be useful for debugging.

3. Start the zebra and ospf daemons:

```
#zebra –d
```

```
#ospfd -d -a -u <username>
```

4.3 Installing Ant

Apache Ant (version 1.6 or higher) can be used as an alternative to a .war file to build and install the IDM module of AutoBAHN.

To install Ant, if it is being used, download from http://ant.apache.org/. Ensure that the `JAVA_HOME` variable is specified according to the Ant installation instructions.

4.4 Installing AutoBAHN

The stages / procedures in the process for installing an operational AutoBAHN instance are:

1. Extract Java binary distribution of AutoBAHN.
2. Create an empty SQL database.
3. Populate the database with domain-specific topology.
4. Edit configuration files in `etc.` directory.

Each of these is described below.

4.4.1 Extracting Binary Distribution

Obtain the Java binary distribution of the AutoBAHN system (by submitting a request to: gn2-jra3@dante.org.uk) and extract it to the chosen destination.

4.4.2 Creating an SQL Database

Create an empty SQL database and execute: `sql/create_db.sql`.

4.4.3 Populating the Database with Domain-Specific Topology

The domain topology (including links for interconnecting with other domains) should be stored in the AutoBAHN system's database.

To do this, take the following steps.

1. Manually edit the system .sql file or use the AutoBAHN topology builder tool to create a domain-specific topology. The topology builder tool is available upon request from: gn2-jra3@dante.org.uk, and is complemented by a user guide.

For more information about creating a topology, see sections 4.4.3.1 Sample Topology Files, 4.4.3.2 Hiding Topology Information and 4.4.3.3 Tips for Creating a Topology.

2. After generating a topology using the topology builder tool, choose `Export topology` and save the file as, for example, `/etc/topology.xml`.
3. In the `etc/dm.properties` file, change the value of the `topology.file` property to `etc/topology.xml`.

   Result: When the DM module of the AutoBAHN system is initiated, the contents of the topology database will be updated from the latest topology file.

4.4.3.1 *Sample Topology Files*

A sample topology .sql file can be found at http://www.geant2.net/upload/sql/sample_topo_dom1.sql. It describes a hypothetical domain DOM1, for which the AutoBAHN host's IDM instance is accessible from http://150.254.160.216:8080/autobahn/interdomain.

DOM1 is connected to domain DOM2 (for which the AutoBAHN host's IDM instance is accessible from http://150.254.160.216:8081/autobahn/interdomain) by two inter-domain links. DOM1 is also connected with a client (end-user) domain through one link.

The DOM1 topology includes:

- 4 nodes: 2 DOM1 nodes, 1 client node, 1 DOM2 node.
- 8 generic_interfaces (ports) : 6 DOM1 ports, 1 client port, 2 DOM2 ports.
- 4 generic_links (links).

The .sql topology of DOM2 can be found at http://www.geant2.net/upload/sql/sample_topo_dom2.sql.

4.4.3.2 *Hiding Topology Information*

A domain administrator may not want to announce internal topology details such as port or node identifiers to other domains. AutoBAHN is based on the concept of mapping internal port/node identifiers and addresses to generalized and public identifiers/labels. For example (in the topology examples above): DOM1 is connected to DOM2 with 2 links, both ending in a port of the DOM2 domain. DOM2 administrators don't want to expose their actual identifiers of ports/nodes used to inter-connect with adjacent domains. DOM2 administrators can assign identifiers such as `DOM2-port-x` for ports and `DOM2-node-x` for nodes. Such mappings should be placed in the `etc/public_ids.properties` file (see examples in distribution pack).

As a result, in the domain specific topology of DOM1, the references to DOM2 ports are made through the `DOM2-port-x` identifiers. Obviously DOM1 keeps its own mappings for ports and nodes of adjacent domains.

4.4.3.3 *Tips for Creating a Topology*

Here are a few tips for creating a topology:

- Links to other domains (as well as to client ports) should be directed towards the other domain (links should leave the current domain).
- Each border interface should have the `domainId` property set according to the domain it belongs to – it's the address of the AutoBAHN host's IDM instance deployed over the domain to which the port belongs.
- Domain interfaces should have their `domainId` field empty.
- Client ports should have the `clientPort` property set to `TRUE`.

4.5 Editing AutoBAHN Configuration Files

For simplicity, the AutoBAHN software is available as ready-to-use binary packs. The AutoBAHN team endeavours to keep these files up to date. We believe this is much easier for users than accessing Subversion (SVN) repositories and building distributions.

The AutoBAHN configuration files can be found in the `etc.` directory of the distribution pack. The most important properties that should be set to run the AutoBAHN application properly are shown in **Table 4.2** and **Table 4.3**, which cover the `dm.properties` and `idm.properties` files respectively.

Properties	Description / Comment
id.nodes, id.ports, id.links	Ranges of private IPv4 addresses to be used as public identifiers (labels) to represent abstracted nodes, ports and links. For example: `id.nodes=10.10.0.0/24` means that nodes of the domain are represented by IPv4 private addresses as identifiers: 10.10.0.0, 10.10.0.1, 10.10.0.2 ... 10.10.0.255. Each domain has its own range of private IPv4 addresses to be used as topology element identifiers (contact the AutoBAHN team to obtain a range to be used).
public.ids.file	Path to the file that maps data plane topology element identifiers to their public identifiers (described above).
db.host, db.port, db.name, db.user, db.pass	To be changed according to the database access parameters.
idm.address	The AutoBAHN host's IDM instance address for the DM to report on certain reservation events (same host by default).

Table 4.2: Key dm.properties

Properties	Description / Comment
domain	Address of the AutoBAHN server. Change the domain name and port properly (should match the `domainId` value in the topology database).
latitude, longitude	Geographical co-ordinates of the AutoBAHN host's location. Needed by the AutoBAHN client portal to draw maps depicting available AutoBAHN systems.
ospf.use	Whether to use OSPF topology exchange mechanism.
ospf.address	Host address of the Quagga router.

Properties	Description / Comment
db.host, db.port, db.name, db.user, db.pass	To be changed according to the database access parameters.
dm.address	The AutoBAHN host's DM instance address (same host by default).

Table 4.3: Key idm.properties

The next stage is to initialize the AutoBAHN service. This is covered in Section 5.

5 Running AutoBAHN

This section describes how to initialize and stop the AutoBAHN service.

5.1 Initializing the AutoBAHN Service

This section describes how to initialize the AutoBAHN service and, optionally, how to prevent log messages from displaying in the console and how to change the default port.

To initialize the AutoBAHN service once the system has been installed, take the following steps.

1. In the distribution directory type: `./start.sh &`
   Result: If there are no errors in configuration or topology, the log messages of the initialization process are displayed on the console.
   Your system is bound to the following addresses:
   - IDM instance: http://(your-host):8080/autobahn/interdomain
   - DM instance: http://(your-host):8080/autobahn/idm2dm

5.1.1 Preventing Display of Log Messages

You may prevent log messages from displaying on the console by typing: `nohup ./start.sh &`

Result: AutoBAHN starts in the background and logs are kept in the `nohup.out` file.

5.1.2 Changing the Default Port

You may change the default 8080 port by editing the `etc/services.properties` file. Remember to change the other configuration files as well.

5.2 Shutting Down the Service

To shut down the service, take the following steps.

1. Telnet to the AutoBAHN instance (port 5000 is default):
   `telnet localhost 5000`
2. Then type:
   `Halt`

6 References

Apache Ant website	http://ant.apache.org/
Autobahn website	http://www.geant2.net/server/show/nav.756
PostgreSQL website	http://www.postgresql.org/
Quagga website	http://www.quagga.net/
sample_topo_dom1.sql	http://www.geant2.net/upload/sql/sample_topo_dom1.sql
sample_topo_dom2.sql	http://www.geant2.net/upload/sql/sample_topo_dom2.sql

7 Glossary

API	Application Program Interface
AutoBAHN	Automated Bandwidth Allocation across Heterogeneous Networks
BoD	Bandwidth on Demand
CPU	Central Processing Unit
DM	Domain Manager
GB	gigabyte
GHz	gigahertz
GRE	Generic Routing Encapsulation
IDM	Inter-Domain Manager
IP	Internet Protocol
L2	Layer 2
LHC	Large Hadron Collider
LSA	Large Hadron Collider Software Applications
MB	megabyte
MCC	Metro Core Connect
MHz	megahertz
MPLS	Multiprotocol Label Switching
NIC	Network Information Centre
OS	Operating System
OSPF	Open Shortest Path First
PoP	Point of Presence
RDBMS	Relational Database Management System
SMTP	Simple Mail Transfer Protocol
SVN	Subversion, a version control system
VPN	Virtual Private Network