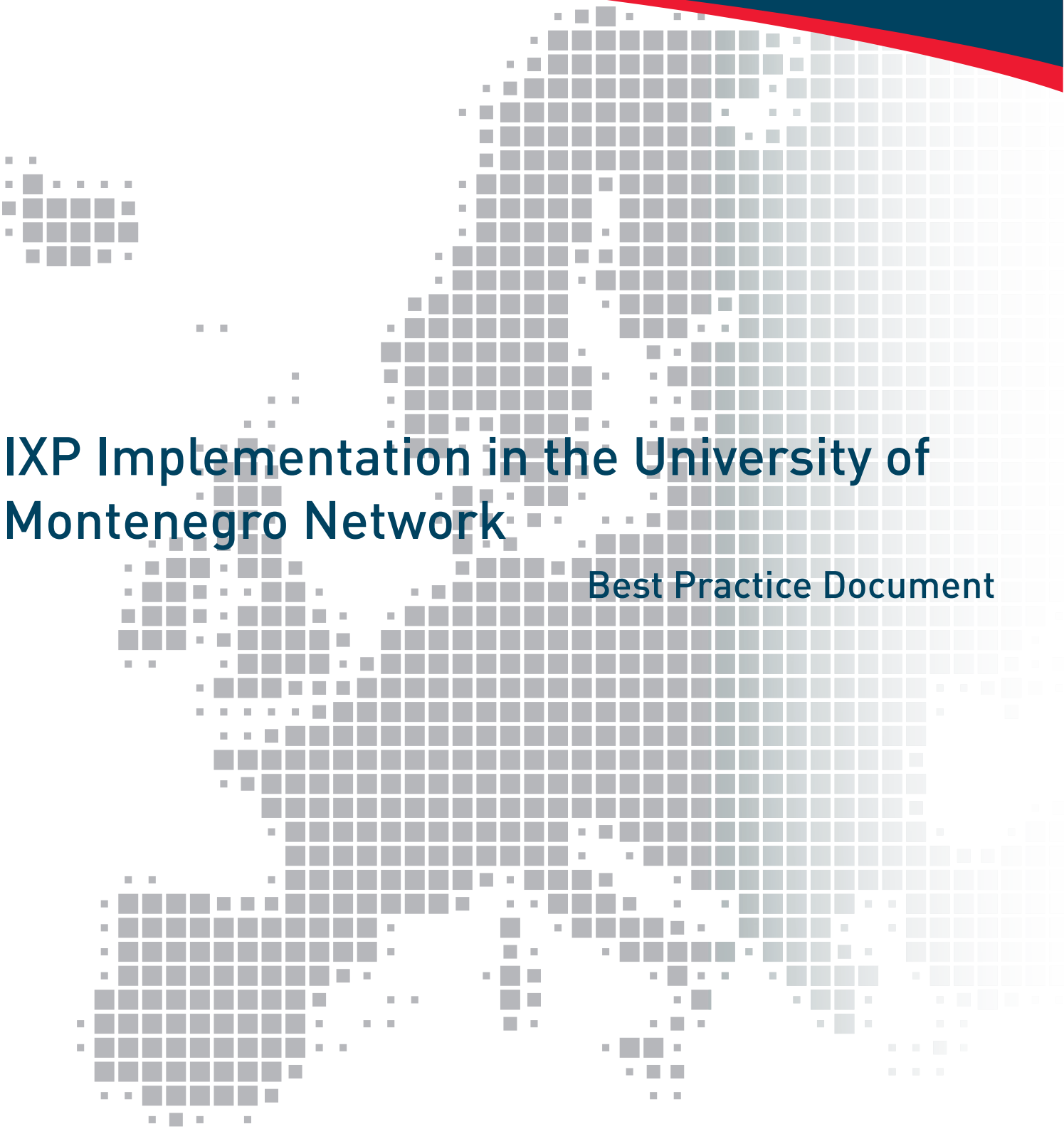

IXP Implementation in the University of Montenegro Network

Best Practice Document

Produced by the MREN-led Campus Networking working group

Authors: Vladimir Gazivoda (MREN), Božo Krstajić (MREN), Dragiša Krstajić (MREN)

April 2016

Document No.: GN4P1-NA3-T2-MREN005
Version / date: March 2016
Original language: Montenegrin
Original title: “Implementacija IXP-a u Akademskoj mreži Univerziteta Crne Gore“
Original version / date: Version 1 / 07 March 2016
Contact: Vladimir Gazivoda, vladg@ac.me; Božo Krstajić, bozok@ac.me; Dragiša Krstajić, dragisak@ac.me

MREN is responsible for the contents of this document. The document was developed by the MREN-led working group on campus networking, with the purpose of implementing joint activities on the development and dissemination of documents encompassing technical guidelines and recommendations for network services in higher education and research institutions in Montenegro.

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

Table of Contents

Table of Figures

Tables

Summary

The document summarises necessary steps carried out for establishing the *Internet eXchange Point,* hereinafter referred to as IXP in Montenegro, from planning and design, through realisation, including fulfilment of requirements of the relevant bodies, configuring devices, and necessary software.

1 Introduction

IXP enables autonomous systems exchange through peering. Peering is known as a method of exchange between two networks free of charge, although in some cases there are peering contracts including fees. Public peering is established when three or more clients decide to connect their networks through one point. This public intersonnection point is called an Internet eXchange Point - IXP. The part with which the autonomous system, hereinafter referred to as the AS, will establish peering using an IXP depends on the IXP peering rule. Certain IXPs enable peering between all participants, while others leave the option for the participants to decide with which client they will peer. Establishing peering relation with other ISPs (or content provider) is usually due to reduction of interconnection costs. Generally speaking, an IXP does not interfere with relations between participants.

Establishing IXP brings short-term and long-term benefits, among which the most significant ones are as follows:

- Delay reduction, since all domestic traffic will avoid international hops;
- Decreased costs of international transit;
- Increased autonomy for local communications;
- Development and growth of the local Internet ecosystem;
- Services, above all e-Government services, become safer and more feasible; and
- International content providers can build network infrastructure in the state in order to increase user database.

Finally, developed local IXPs can become hubs for regional exchange, where the ISPs from the neighbouring countries would exchange traffic.

2 Preconditions for Establishing IXPs

Before establishing an IXP, we need the support of the community and of the parties interested in participating in the project of establishment of the Internet Exchange Point. This mostly refers to the agency for e-communications, as well as operators and other networks with their own AS. Having the initially agreed framework of IXP operations, it is necessary to make a legal framework for IXP operations i.e. legal establishment of an IXP, with all the accompanying documents which would present the principle of IXP operations to operators in a transparent manner, including the contract proposal with IXP members. The following step is preparation of a plan for IXP development, which would encompass all parts of establishment and operations of IXPs, technical principles, as well as the commercial aspect. We have to take into consideration the possibility to bring cables to an IXP by the operator. It is therefore recommend to select a location which is easily accessible for the introduction of new links and possibly already established links towards the operator. For operative functioning of the IXP it is necessary to ask RIPE for the set of Ipv4 and Ipv6 addresses for the purposes of IXP peering, as well as the set of addresses for public services.

The IXP equipment should be placed in the room specially allocated for that purpose which has all the characteristics of the modern communication centre (system room or data centre), with all necessary infrastructure preconditions to be stated in the following chapters, certified by ISO standards for management (9001 – quality, 14001 – enviroment, 27001 - information security, and 50001 - energy) or to the maximum possible extent harmonised with the mentioned standards.
It is necessary to provide needed conditions for storing the equipment, heating, cooling, and ventilation as well as electrical power supply through voltage protection and aggregate pover supply.

System topology and device description is given in the following chapters.

3 IXP Architecture

The figure presents the IXP architecture realised at the University of Montenegro.

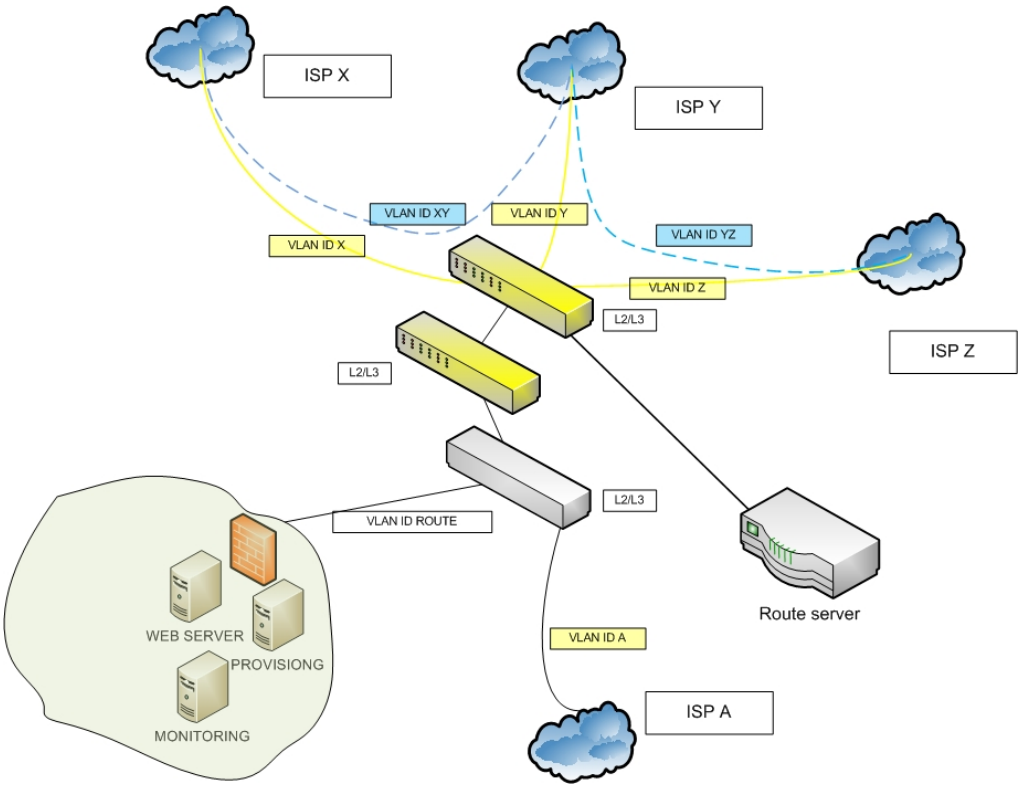

Figure 1: IXP Architecture realised at the University of Montenegro

Technical solution is realised with Layer 2/3 switches to which operators-providers are connected. From the hardware point, switches have to be redundant and enable continuous work in case of failure of one of the devices. After connection, each operater-provider is assigned an IP address from the unique set of IXPs and defined unique VLAN ID "isolating" his port from other ports. Besides switches, which play the role of connection point, i.e. aggregation of links from operator-prvider, route server which operators-providers use to establish BGP session is needed.

The basic service to be provided by the IXP is peering between operators-providers and exchange of national internet traffic.

Besides the basic service, IXPs enable additional network services to clients, such as: hosting providers, DNS operators, cloud providers, etc. Since an IXP would contain hosts with the possibility of implemenatation of various servers, additional services could be established and offered to clients (automatic configuring of equipment in accordance with the terms of contract with operator that is connected to IXP, provisioning service, etc.).

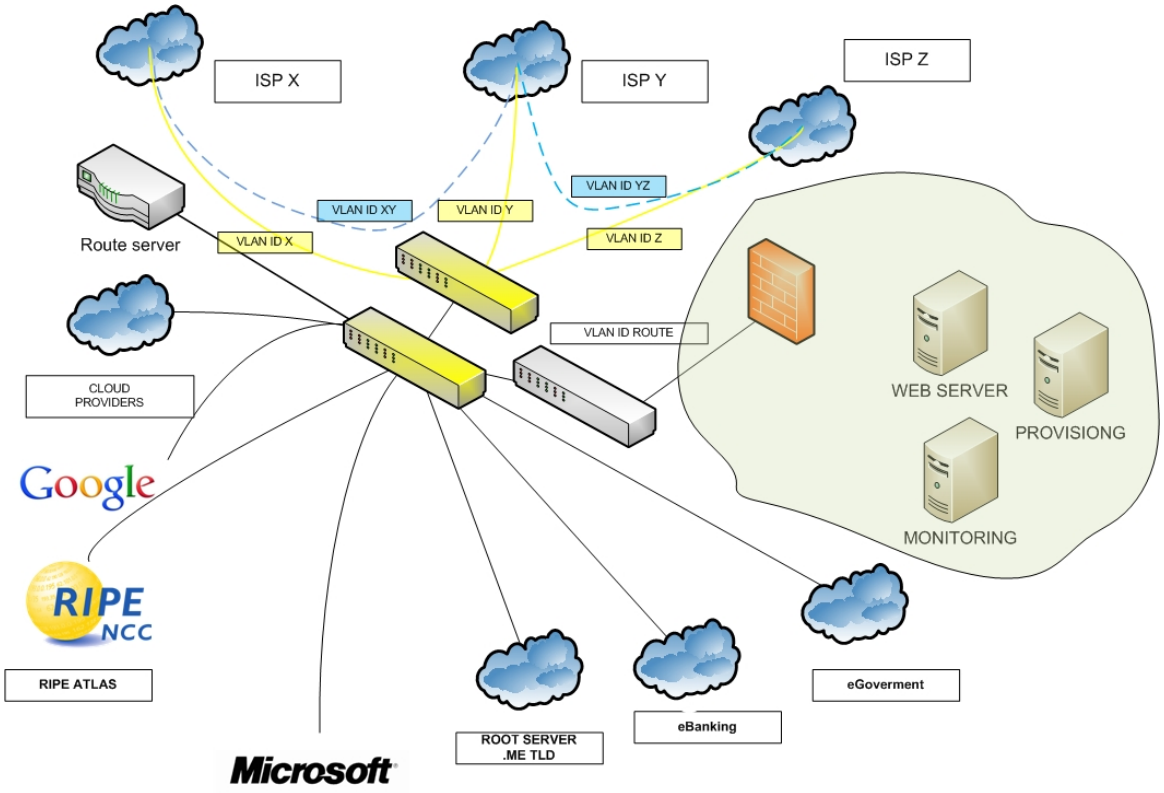

Figure 2: IXP architecture realised at the University of Montenegro

3.1 Hardware Equipment

The ICT equipment which can be used for realisation of the above mentioned recommended solution has to provide services to operators-providers in accordance with the contract signed with the IXP operator. Access equipment has to be fully redundant and scalable, while monitoring and management equipment has to be as redundant as possible. Use of "carrier grade" equipment is recommended.

Following the analysis of the internet traffic status in Montenegro, evaluation of share of national internet traffic, number of existing operators and proposed solution, the following equipment should be provided in the initial stage:

- Two (2) modular Ethernet L3 switches with 24 heterogenous ports, with possibility of work in redundant regime and chain connection on data bus level;
- One (1) Ethernet switch with 24 Ethernet T-ports and possibility of chain connection with modular switches;
- Necessary modules for connecting individual users, with authomatic adjustment of access speed 10/100/1000 Mbps. Access media is couple of single-mode optic fibres, with lasers for distance to 10 (or 15) km;
- Router with adequate software support for working in the route-server regime;
- Firewall for increased safety of public and local IXP servers;

- Computer which will host server for public web presentation, monitoring, management, and other user services.

3.1.1 Switches

In accordance with the proposed L3 approach, modular switches for connecting optic links and switches for support for copper RJ45 ports for connecting Ethernet copper links are used for realisation of link aggregation by the provider. Minimum of technical and software requirements to be supported by switches are given further on.
For the IXP basic needs, ports of aggregation switch are connected to the same peering VLAN. Depending on the needs of clients and further development of IXP, ports can be connected to several VLANs, in order to enable selecting traffic exchange between peers.

Name/description of characteristics	**Support**
Protocols/standards	SNPM, RMON, Sflow/Netflow, Syslog, QoS, MAC bridges, MSTP, RSTP, VLAN
Switching fabric	130Gbps/96Gbps (T based)
Forwarding	100 Mbps/70Mbps (T based)
Stacking bandwith	48Gbps

Table 1: Switch characteristics

Name	**Quantity**
Switch, 24 SFP port, stack module	2
Switch 24RJ port, stack module	2

Table 2: Switch characteristics

3.1.2 Firewall

Firewall which should provide security of server segment has to support stateful inspection mode and Intrusion Prevention System. Besides, it has to support site-to-site and client VPN connections. The following table gives detailed specification of firewall characteristics.

Name/description of characteristics	Support
Stateful Inspection throughput (max)	1Gbps
Stateful Inspection throughput Multi-Protocol	500Mbps
Concurent Sessions	100,000
Conection per second	10,000
VLANs	50
Intrusion Prevention	Yes
Firewall + IPS throughput	250Mbps
Site-to-Site VPN	Yes
Ipsec client VPN user Session	Yes
SSL VPN user session	Yes
RAM	4GB
Portovi 10/100/1000	Yes (min 4)
IPv6	Yes

Table 3: Firewall characteristics

3.1.3 Servers

Computer subsystem of IXP consists of 3 computers-hosts, with Linux operative system and software solutions for monitoring and management of the IXP. Besides that, it is necessary to have a Web server as well. Hardware requirements of all servers are harmonised: processor Intel Xeon QuadCore, 8GB RAM, 2 * 1TB HDD.

3.2 Services

Primary services to be provided by the Internet Exchange Point are the connecting of IXP members, exchange of routes and routing traffic, traffic management by members, as well as traffic monitoring. From the additional services, the following are necessary: web portal for overview and management of services, mail server for IXP needs, looking glass service, route collector, AS112 service. The following chapters will describe some of the basic services, their installation, and configuration.

3.2.1 IXP Manager

For the purposes of implementation of the first Internet Exchange Point in Montenegro (MIXP), the role of IXP Manager is very important. It is a tool that has to be installed on certain servers in the architecture of the IXP itself. Hardware-related, that server is connected to a certain switch, which is through its ports connected to border devices of autonomous systems, i.e. ISPs, which are connected to the IXP through it.

Administrator can add new autonomous systems through IXP Manager, add new users and define their rights, supervise network traffic, etc. Traffic supervision is carried out by later installation of MRTG traffic software, based on which the information on exchanged traffic between the IXP and autonomous systems are "gathered", as well as between autonomous systems themselves. All this enables administrator to simply create different reports concerning exchange of network traffic.

3.2.2 Installation requirements

One of the basic requirements when installing IXP Manager on specific servers concerns the operative system on that server. It is good to know that IXP Manager should simply be observed as one PHP application requiring the following:

- **Operative system**: Linux host. Therefore, the server on which the IXP Manager is installed has to work under Linux OS since IXP Manager still does not have support for initiating it on another platform, e.g. Windows OS.
- **Database**: MySQL version 5.5.4 or later, as MariaDB version 10, or later.
- Apache server
- PHP version 5.4 – IXP Manager will not work with older versions of PHP.
- Memcached – optional, but recommended to be installed.

In order to successfully install IXP Manager, the following requirements have to be fulfilled:

- Text editor such as i, nano, joe, etc., has to be installed.
- Install **git** (distributed inspection control system) – e.g. *apt-get install git*;
- Install **subversion** (advanced system control version) – e.g. *apt-get install subversion*;
- Install **memcache** and **SNMP** upgrade for PHP5 – e.g. *apt-get install php5-memcache* or *apt-get install php5-snmp*;
- Install **php-apc** – APC (*Alternative PHP Cache*) – e.g. *apt-get install php-apc*;
- Install: **php-mbstring**, **php-gettext**, **php-icovn**, **php-ctype**, **php-json**, **php-simplexml** with the note that these modules may already be installed in the PHP core.
- Also, it is necessary to have PHP PEAR library (**php-PEAR**), as well as the PHP database library, for the selected database (e.g. **php-pdo_mysql**).

It is very important to mention that certain parts of IXP Manager use perl, i.e. they are perl-dependent. Perl version 5.10.1 or older may be necessary for installation of certain CPAN (*Comprehensive Perl Archive Network*) modules. All perl dependencies can be checked by performing the command *check-perl-dependencies.pl*, within *tools/runtime* directory.

3.2.3 Installation of third party libraries

IXP Manager requires a large number of third party libraries. Some of these libraries (especially Zend, Smarty, Doctrine, etc.) are often available as packages within Linux distributions. While there is possibility to use these versions of packages and link for them from *library/* directory, they are not recommended since they can be updated in discrepancy with the IXP Manager. Because of this problem during installation of libraries, i.e. certain versions of packages, good practice is to use the following installation method: all libraries are included as part of **git** sub-module and can be installed using these commands:

```
cd /usr/local/ixp
git submodule init
git submodule update
```

Note: All directories to be used further on in the installation procedure without specified path are under the directory /usr/local/ixp. Also, the complete installation process of IXP Manager is performed from this directory.

This set of commands contributes to simple taking over and installation of all required libraries under directory library/.

After installation of package, installation of Doctrine ORM 2.3 can start, and it can be installed using **php-PEAR** library and the following set of commands:

```
pear channel-discover pear.symfony.com
pear channel-discover pear.doctrine-project.org
pear install doctrine/DoctrineORM
```

After installing Doctrine, it is necessary to add one soft (symbolic) in, which would point to the Doctrine directory. Within the Ubuntu server, Doctrine is located at /usr/share/php, which means that the symbolic link will be created in the following manner:

```
cd /usr/share/php/Doctrine
ln -s ../Symfony
```

3.2.4 Installation of Perl libraries

While for the large number of standalone IXP Managers, the migration of perl scripts into PHP form is carried out, there are also cases where it is not necessary to perform such type of file migration. In order not to perform the stated script migration, it is necessary only to install perl libraries, as mentioned before. After carrying out simple check of perl dependencies using command *check-perl-dependencies.pl*, within *tools/runtime* directory, installation of IXP Manager for perl library should be performed using the following commands:

```
cd /usr/local/ixp/tools/perl-lib/IXPManager
perl Makefile.PL
make install
Configuration file has to be copied and changed, so that the data referring to the database is changed.
cp /usr/local/ixp/tools/perl-lib/IXPManager/ixpmanager.conf.dist /usr/local/etc/ixpmanager.conf
vi /usr/local/etc/ixpmanager.conf              # change of database setup
```

The following figure presents one of the examples of configuration file after change of settings concerning the database.

```
<sql>
        dbase_type      = mysql
        dbase_database  = ixp
        dbase_username  = root
        dbase_password  = mgmtmixp
        dbase_hostname  = localhost
        #dbase_portname = /tmp/mysql.sock
</sql>

<ixp>
        rs_asn = 200608
        rs_peval_bird = /usr/local/bin/peval-output-to-csv.sh
        rs_identity = 1
        sflow_rrdcached =
        sflowtool =
        sflowtool_opts =
        sflow_rrddir =
#       debug = 1
</ixp>
```

Figure 3: Setting database

In case IXP Manager Perl library has to be installed on some other server without installed IXP Manager (e.g. Route Server, Route Collector, etc.), there is a method of very simple creation of library distribution tar file, by performing the following commands:

```
cd /usr/local/ixp/tools/perl-lib/IXPManager
make dist
```

Thus the created file can be copied independently from IXP Manager.

3.2.5 Creating database

IXP Manager requires using MySQL database. It is very important to note that Doctrine ORM 2.3 which has been installed earlier on, uses DBAL (*Database Abstraction Layer*) which means that it supports work with different databases, where each of those bases which Doctrine supports[1], shall function in the right manner. However, since IXP Manager supports certain versions of MySQL database, the best thing is to use that very type of database. The following set of commands is used as indicator for creating simple databases and users, as well as defining rights within IXP Manager.

```
$ mysql -u root -p
mysql> CREATE DATABASE `ixp` CHARACTER SET = 'utf8mb4' COLLATE ='utf8mb4_unicode_ci';
mysql> GRANT ALL ON `ixp`.* TO `ixp`@`127.0.0.1` IDENTIFIED BY 'password';
mysql> GRANT ALL ON `ixp`.* TO `ixp`@`localhost` IDENTIFIED BY 'password';
mysql> FLUSH PRIVILEGES;
```

[1] http://www.doctrine-project.org/projects/dbal.html

If there is an error related to part referring to CHARACTER SET, i.e. with 'utf8mb4', it is possible that an older version of MySQL database was run on server, i.e. version 5.5.3, and it needs an upgrade. Alternative way for solving the possible problem is performing the following command when creating database:

```
mysql> CREATE DATABASE `ixp` CHARACTER SET = 'utf8' COLLATE ='utf8_unicode_ci';
```

3.2.6 Configuration of IXP Manager

Configuration of IXP Manager is certainly the riskiest step in installation of this product. The administrator is expected to edit file application.ini.dist on the location /usr/local/ixp/application/configs/. The first thing which needs to be done is to copy file application.ini.dist to the same location, named application.ini, in order not to lose important data from the file application.ini.dist:

```
cd /usr/local/ixp
cp ./application/configs/application.ini.dist ./application/configs/application.ini
```

After this step, it is necessary to edit the copy, i.e. application.ini file, so that the data from the file referring to (nano ./application/configs/application.ini):

- Doctrine2 settings
- Logger settings
- SMTP relay host
- Organisational details

is adjusted to the needs of the relevant IXP Manager.

Note: the file contains value $APPLICATION_PATH, which represent place within the system where git sub-modules are located, i.e. it should be replaced by path /usr/local/ixp.

3.2.7 Setting database

All database settings are reduced to setting application environment and creating scheme, i.e. table using the Doctrine library.

Many tools within bin/ directory (/usr/local/ixp/bin) use utils.inc script to "read" application environment (APPLICATION_ENV) from public/.htaccess (which does not exist by default, there is.htaccess.dist). This very environment should correspond to certain sections within file application/configs/application.ini (the role of which has been explained in the previous part) depending on which sections of this file administrator wishes to use.

This public document (public/.htaccess) is very significant from the aspect of virtual configuration of Apache server, and has to be created as copy of.htaccess.dist file, so that the script utils.inc could read its content:

```
cp public/.htaccess.dist    public/.htaccess
```

Upon performing this command, document can be arranged depending on administrator's needs.

If all the settings have been performed without problems, tabular scheme (i.e. database) can be created, presuming that the command is performed within the /usr/local/ixp/bin/ directory.

```
./doctrine2-cli.php          orm:schema-tool:create
```

If everything is fine, upon performing this command, the screen will present the following message:

```
ATTENTION: This operation should not be executed in a production environment.

Creating database schema...
Database schema created successfully!
```

Note: It is very significant to note that when creating scheme, certain system errors may occur, and one of the most common errors relates to Memcache. At the beginning of the chapter, among other things, it is said that memcache upgrade for PHP5 has to be installed. If memcache is not installed, when attempting to create scheme, the screen will present the following error message:

```
# ./doctrine2-cli.php orm:schema-tool:create
PHP Fatal error: Class 'Memcache' not found in /usr/local/ixp/library/OSS-Framework.git/OSS/Resource/Doctrine2cache.php on line 101
```

Solution to this problem is very simple and concerns installation of memcache:

```
apt-get install php5-memcache
```

Note: Another common mistake is the so called “PHP Parse error: syntax error…” When attempting to create scheme, the screen will present the following message:

```
# ./doctrine2-cli.php orm:schema-tool:create
PHP Parse error: syntax error, unexpected '[' in /usr/local/ixp/library/OSS-Framework.git/OSS/Resource/Doctrine2.php on line 88
```

The essence of the previous problem is that PHP which has been installed in the system is not from version 5.4 or older (as noted in the part “Installation requirements”). In order to fulfil the system need for PHP 5.4 version, it is necessary to carry out the following set of commands (Ubuntu 12.04):

```
apt-get install python-software-properties
add-apt-repository ppa:ondrej/php5-oldstable
apt-get update
apt-get upgrade
apt-get dist-upgrade
```

In order for the database to function correctly, it is necessary to create SQL Views, which can be done in the following manner:

```
mysql -u [user_name] -ppassword [db_name] < tools/sql/views.sql
```

After this, it can be said that database settings are complete so the Apache setting can start.

3.2.8 Setting access rights to Web resources of IXP Manager

Above all, rewrite has to be enabled on the Apache server:

```
a2enmod rewrite
/etc/init.d/apache2 restart
```

After that, it is necessary to adjust access rights. The Apache server requires that users which are members of owner group have reading rights for all directories and files under /usr/local/ixp. Also, users have to be given inscription right for all under /usr/local/ixp/var (since www directory is also under that, and owner group will be www-data, so that Web users are enabled access to IXP resources). This means that it is necessary to change owner of all directories under /usr/local/ixp, and assign the ownership right to group www-data, while users, members of the owner group should have reading and execution right of all under /usr/local/ixp, as well as entry right into all under /usr/local/ixp/var.

```
chown –R www-data /usr/local/ixp
chmod –R u+rX /usr/local/ixp
chmod –R u+w /usr/local/ixp/var
```

After these settings, Web browser can be accessed, and after entering http://hostname/ixp (where the hostname is the IP address of IXP Manager server), the page for logging in the IXP Manager will open. However, these settings are not complete, since the following should be done, among other things:

- Open user accounts on Web page of IXP Manager, and add data about initial ISP for the beginning, which wants to connect to IXP;
- Physically connect server where IXP Manager is installed to Layer3 switch;
- Configure switch and connect end network devices of autonomous systems to its ports, i.e. ISPS wishing to access IXP;
- Install software for monitoring network traffic (in this case MRTG traffic), so that system administrator can have the best possibility to oversee traffic and make report on exchanged traffic.

When starting IXP Manager in Web browser, a problem will occur, since there is no autonomous systems or user or administrator accounts. Because of that, data about the IXP should be initially set, as well as the data related to the first autonomous system to be connected to IXP. /usr/local/ixp/bin/ should be entered and script fixtures.php.dist has to be edited.

```
cd bin
cp fixtures.php.dist fixtures.php
vi fixtures.php
```

After editing the script according to certain needs, it is necessary to find the sector titled “MODIFY YOUR FIXTURES HERE”, after which the initial IXP object is created by editing this part of document.

```
### MODIFY YOUR FIXTURES HERE
###
###

### First you need to create an initial IXP object for your IXP

$ixp = new \Entities\IXP;
$ixp->setName( "Montenegro Internet Exchange Point" );
$ixp->setShortname( "MIXP" );
$ixp->setAddress1( "Univerzitet Crne Gore" );
$ixp->setAddress2( "Centar Informacionog Sistema" );
$ixp->setAddress3( "Podgorica" );
$ixp->setAddress4( "81000" );
$ixp->setCountry( 'ME' );

$em->persist( $ixp );
```

Figure 4: Example of creating initial IXP object

In order to grant right of access to these resources (reading) to Web users, it is necessary to do as follows (provided we are in /usr/local/ixp/bin):

```
chown www-data fixtures.php
sudo -u www-data ./fixtures.php
```

After this, it is only necessary to run the script:

```
./fixtures.php
```

3.2.9 Adding physical infrastructure within IXP Manager

Upon performed installation of IXP Manager, it is possible to add physical objects through its Web page, which is performed by administrator. Complete physical infrastructure of IXP Manager presented through Web page can be added and changed in the left vertical menu, under **IXP ADMIN ACTIONS**.

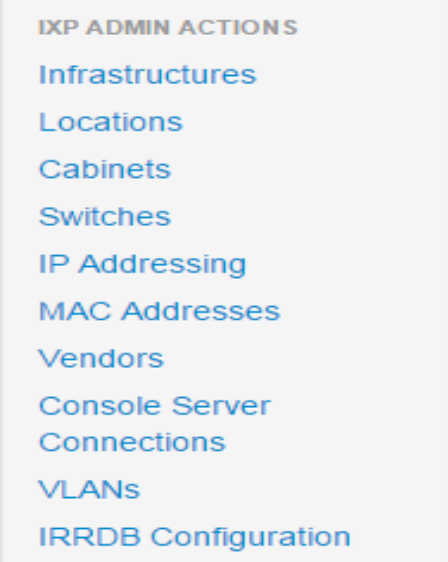

Figure 5: Menu for setting physical infrastructure

As it can be seen from the figure, it is possible to physically set and add:

- Locations – data centres, i.e. IXP access points;
- Cabinets – rack cabinets in data centres;
- Switches and switch ports – This defines switch or several of them representing "network centre" of IXP, as well as defining number of switches and their ports to which autonomous systems are connected. Ports have to be correctly configured, due to traffic exchange and its graphic presentation within IXP Manager (on web page);
- IP Addressing – this part defines scope of IPv4 or IPv6 addresses available to IXP.

Also, administrator duties within Web page concern adding new customers (autonomous systems) and clients. The IXP Manager administrator has to define access rights for all of them, by defining the type of customer and client. Customer based on type can be: FULL, ASSOCIATE, INTERNAL, PROBONO, while the users can be:

- AUTH_CUSTUSER (standard client of an autonomous system, which is an IXP member with the access right for the portal),
- AUTH_CUSTADMIN (administrator user of certain system. This user can access portal, add, edit, or delete users with status AUTH_CUSTUSER, but are solely members of his autonomous system),
- AUTH_SUPERUSER (such user can be only the IXP administrator. It has full right to supervise all customers and their clients, can add, edit, or delete all users).

3.2.10 Configuring MRTG

Besides defining physical infrastructure, adding clients and defining their authorisations within portal, the IXP Manager administrator also has the task to monitor network traffic, which has been mentioned several times so far. For the purpose of monitoring network traffic, it is necessary to configure MRTG (*Multi Router Traffic Grapher*) on the server where IXP Manager is installed.

Above all, it is necessary to install some of the basic packages, so that MRTG can function:

```
apt-get install libconfig-general-perl libnetaddr-ip-perl mrtg
```

Also, it is necessary to create directory for saving all MRTG files:

```
mkdir -p /srv/mrtg
mkdir -p /srv/mrtg/members
```

After creating directory for saving all files related to MRTG (it will become workdir), the following two parameters need to be set in the application.ini script:

```
;; destination for all MRTG output (log files, pngs, etc.)
mrtg.conf.workdir = '/srv/mrtg'
;; destintation file for the MRTG configuration. If not set, the generator sends it to stdout
mrtg.conf.dstfile = '/etc/mrtg/mrtg.cfg'
```

Upon performing previous settings, it is necessary to generate MRTG configuration by performing the following command:

```
/usr/local/ixp/bin/ixptool.php -a statistics-cli.gen-mrtg-conf
```

In order to start MRTG configuration, the following set of commands has to be performed:

```
cp /usr/local/ixp/tools/runtime/mrtg/ubuntu-mrtg-initd /etc/init.d/mrtg
chmod +x /etc/init.d/mrtg
update-rc.d mrtg defaults
/etc/init.d/mrtg start
```

Performing this set of commands enables monitoring of network traffic related to the newly added autonomous system.

3.2.11 Bird route server

Bird is routing daemon for Linux, Free BSD and other Unix like platforms which supports:

- IPv4 and IPv6
- Multiple routing tables
- BGP
- RIP
- OSPF
- BFD
- Static routes
- IPv6 router voicing
- Inter-table protocol
- Command-line interface (`birdc')
- Route filtering environment
- Linux, FreeBSD, NetBSD, OpenBSD ports

Bird is very simple to install. It is possible through repository or source code.

```
apt-get install bird
```

Document [3] gives overview of bird configuration, however, for the IXP purposes, bird is used as bgp routing daemon for route exchange with operators.
Considering that IXP Manager supports bird as routing daemon, the scripts accompanying IXP Manager are used for the purpose of easier configuration and complete harmonisation with the IXP Manager.

In order to use IXP scripts, it is necessary to insert clients into IXP Manager (operators) containing the following data:

- `cid` - client ID;
- `cname` - Full client name;
- `cshortname` - Client short name;
- `autsys` - Client AS number;
- `peeringmacro` - Client appropriate peering macro.
- `vliid` - VLAN interface ID
- `fvliid` - formatted VLAN interface ID using `sprintf( "%04d" )`;
- `address` - IPv4/6 address
- `bgpmd5secret` - configured BGP MD5 secret (or false if not defined)
- `maxprefixes` - configured maximum prefix setting;
- `location_name` - full location name;
- `location_shortname` - location short name;
- `location_tag` - location tag.

Parts of this sequence are used by scripts for generating both global configuration and special configuration files linked to providers.
Possibilites of route servers which can be generated include:

- full prefix filtering based on IRRDB entries;
- full origin ASN filtering based on IRRDB entries;
- filtering of prefixes for IPv4 and IPv6;
- ensuring that there is no multihop;
- max prefix limit;
- multiple VLAN interfaces per client.

Configuration for the purposes of each IXP member will be created separately using configuration template.
Template for creating users is in the IXP subdirectory:
application/views/router-cli/server/bird
3 files are here:

- header.cfg - **optional** – If it is used, it is used only in the beginning. It can be used for defining standard configuration such as log-in, unspecific access lists, vty configurations…
- neighbor.cfg - **mandatory** – it is parsed and printed for each client with necessary parameters.
- footer.cfg - **optional** – If it is used, it is used only in the end.

BGP configurations are generated in the following manner:

```
${APPLICATION_PATH}/bin/ixptool.php -a router-cli.gen-server-conf -p vlanid=X,target=bird,proto=6 --config=/full/path/to/config.conf
```

Where:

- vlan=X - ID in the VLAN base for which the configuration is generated;
- proto=[4|6] – generating configuration for the given protocol;
- target=bird – target directory (in router-cli/server/) from which the template file neighbor.cfg is uploaded.

In our specific case, the following lines are used to generate main configuration file, and configuration files for clients (in this case the UoM Information System Centre).

```
/usr/local/ixp/bin/ixptool.php -a router-cli.gen-server-conf -p vlanid=1,target=bird,cust=cis --config=/usr/local/ixp/application/configs/route-server.conf > /usr/local/etc/mixpeers/10.conf
/usr/local/ixp/bin/ixptool.php -a router-cli.gen-server-conf -p vlanid=1,target=bird --config=/usr/local/ixp/application/configs/route-server.conf > /usr/local/etc/bird.conf
```

```
root@mixp-b:/usr/local/etc/mixpeers# less 10.conf
### AS40981 - UCG Centar informacionog sistema - VLAN Interface #2
table t_0002_as40981;
filter f_import_0002_as40981
prefix set allnet;
int set allas;
{
    if !(avoid_martians()) then
        reject;
    # Route servers peering with route servers will cause the universe
    # to collapse.  Recommend evasive manoeuvers.
    if (bgp_path.first != 40981 ) then
        reject;
    allas = [ 40981  ];
    if !(bgp_path.last ~ allas) then
        reject;
    allnet = [ 89.188.32.0/20  ];
    if ! (net ~ allnet) then
        reject;
    accept;
}
protocol pipe pp_0002_as40981 {
    description "Pipe for AS40981 - UCG Centar informacionog sistema - VLAN Interface 2";
    table master;
    mode transparent;
    peer table t_0002_as40981;
    import filter f_import_0002_as40981;
    export where ixp_community_filter(40981);
}
protocol bgp pb_0002_as40981 from tb_rsclient {
```

```
        description "RIB for AS40981 - UCG Centar informacionog sistema - VLAN Interface 2";
        neighbor 185.1.44.10 as 40981;
        route limit 20;
        table t_0002_as40981;
}

# Bird Route Server configuration generated by IXP Manager
# Do not edit this file, it will be overwritten. Please see:
# https://github.com/inex/IXP-Manager/wiki/Route-Server
# Generated: 2015-09-21 11:55:33
# For VLAN: Peering (Tag: 2, Database ID: 1)
log "/var/log/bird-rs.log" all;
log syslog all;
define routeserverasn = 200608;
define routeserveraddress = 185.1.44.1;
router id 185.1.44.1;
listen bgp address 185.1.44.1;
# ignore interface up/down events
protocol device { }
# This function excludes weird networks
#  rfc1918, class D, class E, too long and too short prefixes
function avoid_martians()
prefix set martians;
{
      martians = [
            10.0.0.0/8+,
            169.254.0.0/16+,
            172.16.0.0/12+,
            192.0.0.0/24+,
            192.0.2.0/24+,
            192.168.0.0/16+,
            198.18.0.0/15+,
            198.51.100.0/24+,
            203.0.113.0/24+,
            224.0.0.0/4+,
            240.0.0.0/4+,
            0.0.0.0/32-,
            0.0.0.0/0{25,32},
            0.0.0.0/0{0,7}
        ];
        # Avoid RFC1918 and similar networks
        if net ~ martians then
            return false;
      return true;
}
# Standard IXP community filter
##

function ixp_community_filter(int peerasn)
```

```
{
      if !(source = RTS_BGP) then
            return false;
      # default community filtering schema doesn't support ASN32, as there
      # are only 6 octets available for numbering.  We need
      # draft-raszuk-wide-bgp-communities to become reality.
      if peerasn > 65535 then
            return true;
      # Implement widely used community filtering schema.
      if (0, peerasn) ~ bgp_community then
            return false;
##      if (routeserverasn, peerasn) ~ bgp_community then
##            return true;
##      if (0, routeserverasn) ~ bgp_community then
##            return false;
      return true;
}
## Route Server client configuration
template bgp tb_rsclient {
      local as routeserverasn;
      source address routeserveraddress;
      import filter {
            ## Prevent BGP NEXT_HOP Hijacking
            if !( from = bgp_next_hop ) then
               reject "BGP neighbor address [", from, "] != next hop address [", bgp_next_hop, "]", ",
net:[", net, "], path:[", bgp_path, "]";
            accept;
      };
      export all;
      rs client;
};
include "/usr/local/etc/mixpeers/*";
```

This configuration is complete for the purposes of connecting one client to IXP. As you can see, VLAN is set to client 1, routing daemon bird and CIS as client's name. Adequate configuration has to be generated for each provider and placed into separate file as defined. In the main configuration file include include "/usr/local/etc/mixpeers/*" defines that all files are read in this folder and added to configuration.

3.2.12 Route collector

Route collector is important tool for setting clients, diagnostics and measuring tools for IXP. As in the case of route servers, IXP Manager has scripts which can be used for generating configuration for route collector. The bird configuration template is located inn the directory application/views/router-cli/collector/bird/index.cfg. As well as in the previous case, it is necessary to fill in the following parameters in the IXP Manager.

- ‘cid` - client ID;
- `cname` - Full client name;
- `cshortname` - Client short name;
- `autsys` - Client AS number;
- `peeringmacro` - Client appropriate peering macro.
- `vliid` - VLAN interface ID
- `fvliid` - formatted VLAN interface ID using `sprintf( "%04d" )`;
- `address` - IPv4/6 address
- `bgpmd5secret` - configured BGP MD5 secret (or false if not defined)
- `maxprefixes` - configured maximum prefix setting;
- `‘cid` - client ID;
- `cname` - Full client name;
- `cshortname` - Client short name;
- `autsys` - Client AS number;
- `peeringmacro` - Client appropriate peering macro.
- `vliid` - VLAN interface ID
- `fvliid` - formatted VLAN interface ID using `sprintf( "%04d" )`;
- `address` - IPv4/6 address
- `bgpmd5secret` - configured BGP MD5 secret (or false if not defined)
- `maxprefixes` - configured maximum prefix setting;

By performing the following command, the necessary configuration for route collector is generated.

```
${APPLICATION_PATH}/bin/ixptool.php -a router-cli.gen-collector-conf -p vlanid=X --config=/full/path/to/config.conf
```

3.2.13 AS112 service

You can read more about the AS112 service in the relevant literature [6]. AS112 service setting is easy, considering that IXP Manager offered configuration generating script for this service as well.

If all necessary adjustments in the application.ini file have been performed, initiating the following script generates the following configuration file:

```
${APPLICATION_PATH}/bin/ixptool.php -a router-cli.gen-as112-conf -p vlanid=X
```

I.e. in case of realised IXP:

```
/usr/local/ixp/bin/ixptool.php -a router-cli.gen-as112-conf -p vlanid=1
```

4 Conclusion

This document gives overview of steps necessary for establishment of the IXP. Reference is made to the necessary introductory steps, where the community is consulted and necessary plans for further realisation are made. Surely, it is of particular importance to make sustainable system, which would ensure quality work, covering basic IXP operation costs as well as investment in infrastructure development. For the IXP realisation IXP Manager was used, considering that it proved to be excellent in the majority of existing IXPs. IXPs can, of course, be realised without using IXP Manager, with other infrastructure management and overview tools. The advantages of IXP Manager are evidently multiple, considering that it has generated a large part of necessary configuration with the already existing scripts.

Literature

[1] Case study on establishment of Montenegro IXP (MIXP),
http://www.mixp.me/

[2] Step by step installation how-to of the IXP Manager frontend using a Git clone,
https://github.com/inex/IXP-Manager/wiki/Installation

[3] The BIRD Internet Routing Daemon,
http://bird.network.cz/?get_doc&f=bird-3.html

[4] European Internet Exchange Association,
https://www.euro-ix.net/

[5] Croatian Internet eXchange,
http://www.cix.hr/

[6] AS112 Project,
https://www.as112.net/

Abbreviations

AS	Autonomous System
BFD	Bidirectional Forwarding Detection
BGP	Border Gateway Protocol
CIS	Centar Informacionog Sistema
DNS	Domain Name Server
HDD	Hard Disk Drive
IXP	Internet Exchange Point
L3	Layer 3
MIXP	Montenegro Internet Exchange Point
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
SNMP	Simple Network Management Protocol
VLAN	Virtual Local Area Network
VPN	Virtual Private Network