

- [Twitter](#)
- [Facebook](#)
- [Google+](#)
- [LinkedIn](#)
- [Pinterest](#)
- [Reddit](#)
- [Email](#)

# Sixxs AICCU tunnel with Mikrotik routerOS

## Introduction

Despite that many Internet providers are now providing ipv6 addresses, the implementation done by providers are not always satisfying. Examples includes:

- Only one single routed address is provided
- Dynamically allocated ipv6 that change every x days like what is done with ipv4
- No ipv6 provided if using the modem in bridge mode behind your own router (my case)

So, as long as they still exists, ipv6 tunnel brokers can be very useful services.

There are mainly two well-known tunnel brokers:

- [Hurricane Electric Tunnel Broker](#)
- [Sixxs](#)

If you have a static ipv4 address, you will issue no problem setting it up on routerOS but if, like me, you have a dynamic ipv4 address, it's a bit more complicated.

For Hurricane Electric, you can search in google, there are many tutorial available as HE provide a simple http api to update your ipv4.

I used hurricane electric for a while but I had two mains problems with it:

- I'm located in Belgium and there are no POPs available in my country, so the latency was a bit high
- The ipv6 subnet provided by hurricane electric is geolocated by many services as being in United States, for some services this is a problem (some services are reported as unavailable in my country, search for local shops in google give results located in other countries, ...)

For Sixxs, this is a bit more complicated since sixxs require a special client called aiccu to perform the update. Aiccu is not available in routerOS. This is the point of this tutorial.

The solution was inspired by a topic on mikrotik forum [Mikrotik with dynamic outside IP address](#). The referred blog post is not more available but fortunately the archive.org way back machine saved it.<sup>1)</sup>

The problem was that this article is old and I cannot achieve to make it work (especially the routing part), so I will share with you what I did to make it work.

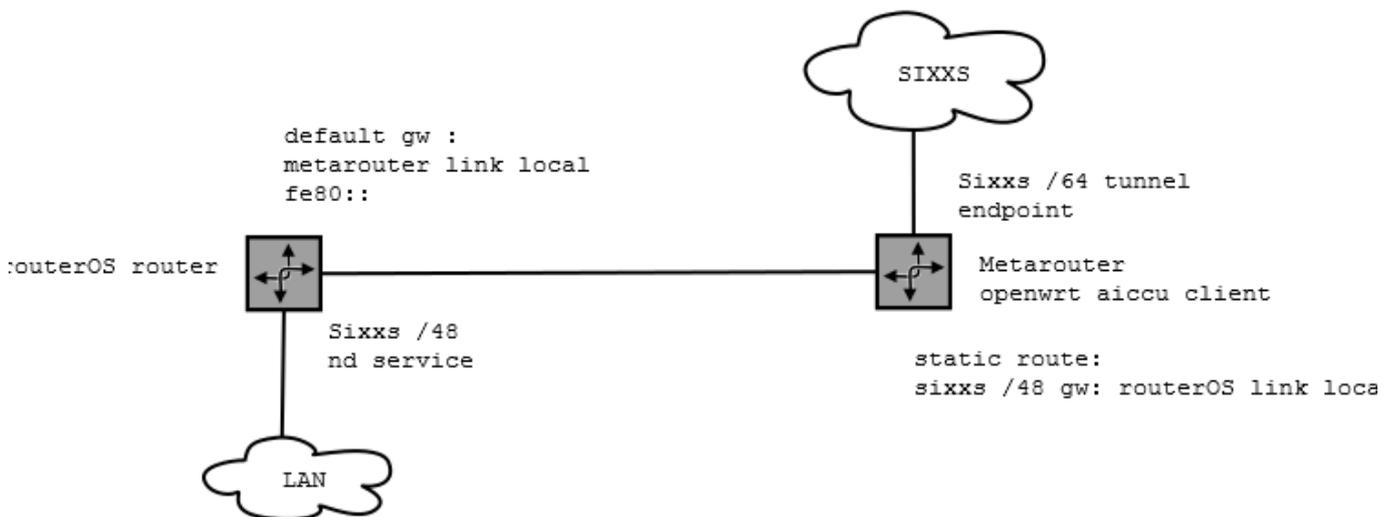
The idea is to setup a [Metarouter](#) in routerOS to run an openwrt router. The openwrt will handle aiccu and route ipv6 traffic through the native routerOS router. The /48 subnet allocated by sixxs will then be distributed using nd service by routerOS.

## Requirement

It's assumed that you have a **working ipv4 setup** on routerOS with NAT allowing any client to access the internet. The metarouter openwrt image will require internet access in order to download packages and to run the tunnel.

I successfully tested this on routerOS 6.30 on a 951G-2HnD.

## Diagram



## Finding a suitable OpenWrt image for metarouter

The whole problem is to find an openwrt image that cover the following requirements:

- metarouter patch must be applied <sup>2)</sup>
- ipv6 must be supported
- a repository with aiccu must be available (or aiccu must be installed)

One solution can be to build your own image <sup>3)</sup>, or to find a pre-built image which match all requirements.

I tried some images found on the internet and finally I got one that work well. It's available from: <http://openwrt.wk.cz/kamikaze/>. It's a old version (kamikaze) but it work well for the limited use.

# Setup metarouter on RouterOS

## Create a bridge interface for metarouter

In order to communicate with the metarouter openwrt, we need a bridge interface. This will be the interface on which the routerOS will see the metarouter. This is the virtual cable connecting the two routers.

```
/interface bridge add admin-mac=02:00:00:00:00:42 auto-mac=no name=ipv6gw_br
```

Explanation:

We create a new bridge interface, with a fixed mac address being 02:00:00:00:00:42. The auto-mac=no flag mean that the bridge will not use mac addresses from interfaces connected to the bridge as bridge mac. This is important for routing part as we will use ipv6 link-local addresses as gateway and these addresses are derived from the interface mac address.

## Create the metarouter and its interface

In order to create the metarouter, you must upload the image (tar.gz) to your routerOS (using ftp or winbox). Then, run the following command:

```
/metarouter import-image file-name=openwrt-mr-mips-rootfs-18961.tar.gz
```

After a while the image will be imported. You now need to change some parameters of the metarouter:

```
/metarouter set 0 memory-size=24  
/metarouter set 0 name=ipv6gw
```

The name to be a bit more meaningful than mr1 and the memory size to 24MiB.

You then need to create an interface for the metarouter and include it in the bridge we created earlier. By doing this, the interface will automatically be added to the bridge when the router is started.

The metarouter openwrt console can be accessed by using:

```
/metarouter 0 console
```

To exit the console, type <C>-a q.

```
/metarouter interface add dynamic-bridge=ipv6gw_br dynamic-mac-  
address=02:25:61:EF:AC:33 type=dynamic virtual-machine=ipv6gw vm-mac-  
address=02:4C:E4:16:D1:76
```

Explanation:

- `dynamic-bridge` is the name of the bridge as created before, the interface will join this bridge automatically
- `dynamic-mac-address` is the mac address shown in routeros for the dynamically create interface
- `virtual-machine` is the name of the metarouter as defined before
- `type=dynamic` mean the interface is created when the metarouter is launched
- `vm-mac-address` is the mac address of the interface seen by metarouter, it will be the mac of eth0 interface.

## Setup ipv4 connectivity between routers

Next we need to setup ipv4 connectivity between routerOS and the metarouter. This is needed in order to grant internet access to the openwrt metarouter.

First we add an ip address on the bridge, from routerOS side:

```
/ip address add address=192.168.1.254/24 interface=ipv6gw_br
network=192.168.1.0
```

Then on openWrt:

1. edit `/etc/config/network` and use the following:

```
config interface lan
    option ifname eth0
    option proto static
    option ipaddr 192.168.1.1
    option netmask 255.255.255.0
    option gateway 192.168.1.254
    option dns 192.168.1.254
```

2. edit `/etc/resolv.conf`: (It should be set by option dns but sometime it work, sometime not ...)

```
nameserver 192.168.1.254
```

3. restart networking : `/etc/init.d/network restart`
4. Try to ping routerOS

```
ping 192.168.1.254
```

5. Try to ping external domain - this will test both dns and default route settings

```
ping www.google.com
```

## Setup openwrt for sixxs/aiccu

## Install aiccu package

Firstly, edit `/etc/opkg.conf`. If using the image mentioned before, the content of the file should be:

```
src/gz snapshots http://openwrt.wk.cz/kamikaze/packages/mr-mips
dest root /
dest ram /tmp
lists_dir ext /var/opkg-lists
option overlay_root /jffs
```

Then update opkg list of available package:

```
opkg update
```

Then install aiccu and activate it

```
opkg install aiccu
/etc/init.d/aiccu enable
```

A small modification is required in aiccu init script in order to wait for ntp client to setup the correct date and time before running aiccu (aiccu will fail if the date/time is incorrect): edit `/etc/init.d/aiccu`:

```
root@metarouter:/etc/init.d# diff /tmp/aiccu aiccu
--- /tmp/aiccu  Wed Jan  6 15:28:16 2016
+++ aiccu      Mon Jan  4 22:48:43 2016
@@ -14,6 +14,9 @@
 }

 start() {
+   while [ `date +%Y` -eq 1970 ]; do
+       sleep 3
+   done
  config_load aiccu
  for cfgs_section in $cfgs_sections; do
      config_get username $cfgs_section username
```

## Configure ipv6 connectivity

Edit `/etc/config/aiccu` and adjust the settings to your sixxs account settings

```
config aiccu
    option username      '<SIXXS user>'
    option password     '<SIXXS password>'
    option protocol     'tic'
    option server       'tic.sixxs.net'
    option interface    'sixxs0'
    option tunnel_id    '<SIXXS tunnel id>'
```

```
option requiretls      '0'  
option defaultroute   '1'  
option nat             '1'  
option heartbeat      '1'
```

Edit `/etc/config/network` and add the following

```
config interface wan6  
    option ifname  sixxs0  
    option proto   static  
    option auto    1  
    option ip6addr <SIXXS ipv6 /64 network>::2  
    option send_rs 0
```

<SIXXS ipv6 /64 network>::2 is the field in Your IPv6 in sixxs tunnel information page.

We also need to configure OpenWRT to act as a router and to forward ipv6 packets from one network to another. This can be done in file `/etc/sysctl.conf`:

```
net.ipv6.conf.all.forwarding=1
```

Now reboot OpenWrt with `reboot` command and try to ping6 an external host to test your ipv6 connectivity:

```
ping6 www.kame.net
```

## Remove useless services (optional)

You can disable some services and remove useless packages (this will free-up some memory and improve security):

```
/etc/init.d/dropbear disable  
/etc/init.d/httpd disable  
opkg remove webif
```

## Setup the static route between our routers

Now that we have a working ipv6 tunnel setup on openwrt, we need to setup static routes between this metarouter instance and our host router running routerOS.

So we want that :

- From routerOS, all non-local ipv6 traffic is redirected through openwrt. This is equivalent to setting up openwrt as default gateway.
- From openwrt, we want that all traffic from the internet destined to our local subnet be redirected to routerOS. This is equivalent to setting up routerOS as static route for sixxs :: /48

subnet.

We need an IPv6 address to set as gateway in both situation. Fortunately IPv6 have a very nice functionality called local-link address. That mean that every interface where ipv6 is activated is automatically assigned an ip address derived from its mac address. This address is stable as soon the mac address don't change.

You can see local link address with `/ipv6 address print` on routerOS and `ifconfig` on OpenWRT. It can also be calculated by knowing mac address by calculators found on the Web (or even manually) <sup>4)</sup>

In my example, using the mac addresses we defined for `ipv6gw_br` and for `/metarouter` interface:

Router	MAC	Address
routerOS	02:00:00:00:00:42	fe80::ff:fe00:42
OpenWRT	02:4C:E4:16:D1:76	fe80::4c:e4ff:fe16:d176

## Setup on routerOS

```
/ipv6 route distance=1 dst-address=2000::/3
gateway=fe80::4c:e4ff:fe16:d176%ipv6gw_br
```

Explanation:

- `distance=1`: Used to give some route priority over other, not very important here
- `dst-address=2000::/3`: This prefix correspond to all ipv6 allocated for internet use, this correspond to route everything public through the route
- `gateway=fe80::4c:e4ff:fe16:d176%ipv6gw_br`: The local link address of OpenWRT. Please note `%ipv6gw_br` suffix which specify on which interface the local link is. This is required for using local link addresses with routerOS.

## Setup on OpenWRT

Edit `/etc/config/network` and add:

```
config interface route6
    option interface lan
    option target <SIXXS ipv6 /48 subnet network>::/48
    option gateway fe80::4c:e4ff:fe16:d176
```

I had some trouble with this setup, so I doubled the configuration in `/etc/rc.local`: add:

```
ip -6 route add <SIXXS ipv6 /48 subnet network>::/48 via
fe80::4c:e4ff:fe16:d176 dev eth0
```

In both case we setup openwrt to route all request destined to `sixxs ::/48` through routerOS.

## Setup ipv6 on RouterOS

Now we have to setup routerOS to provide some ipv6 from our `::/48` to the lan and to have itself some ipv6 connectivity.

### Assign a static ipv6 to RouterOS lan interface

We need to take a /64 out of the /48 in order to assign it to our lan.

For example if the /48 is `2001:6f8:aaa::/48`, we can take `2001:6f8:aaa:1::/64`.

```
/ipv6 address add address=2001:6f8:aaa:1::1 interface=bridge-local  
advertise=yes
```

- `address=2001:6f8:aaa:1::1` is one ip out of /64
- `interface=bridge-local` is the bridge on which lan is connected
- `advertise=yes` mean that this /64 range will be used for stateless ipv6 configuration on the lan

Once the ip is configured you should be able to ping external ipv6 from your router:

```
/ping address=2620:0:ccc::2
```

### Configure stateless ipv6 configuration

```
/ipv6 nd set [ find default=yes ] interface=bridge-local
```

Simply say that we want stateless configuration on all stations connected behind bridge-local bridge.

### Configure firewall

 All your lan station will now have a public ipv6 address. You should really configure correctly ipv6 firewall to avoid anyone accessing your stations. But this is out of this article scope.

### Enjoy

Now you should have access to ipv6 network from your lan. To test it you can point your web browser to <http://www.test-ipv6.com>

## References

- [https://en.wikipedia.org/wiki/Link-local\\_address](https://en.wikipedia.org/wiki/Link-local_address)

- [http://wiki.mikrotik.com/wiki/Manual:Simple\\_Static\\_IPv6\\_Routing](http://wiki.mikrotik.com/wiki/Manual:Simple_Static_IPv6_Routing)

1)

<https://web.archive.org/web/20141230220319/http://blog.one-it.de/ipv6-via-sixxs-using-mikrotik-and-openwrt>

2) 3)

[http://wiki.mikrotik.com/wiki/Manual:Metarouter#Building\\_your\\_own\\_OpenWRT\\_image](http://wiki.mikrotik.com/wiki/Manual:Metarouter#Building_your_own_OpenWRT_image)

4)

<http://ben.akrin.com/?p=1347>

From:

<https://www.fievel.be/> - **Fievel's site**

Permanent link:

[https://www.fievel.be/doku.php?id=en:documentation:sixxs\\_aiccu\\_tunnel\\_with\\_mikrotik\\_routers](https://www.fievel.be/doku.php?id=en:documentation:sixxs_aiccu_tunnel_with_mikrotik_routers)

Last update: **2016-01-18 11:55**