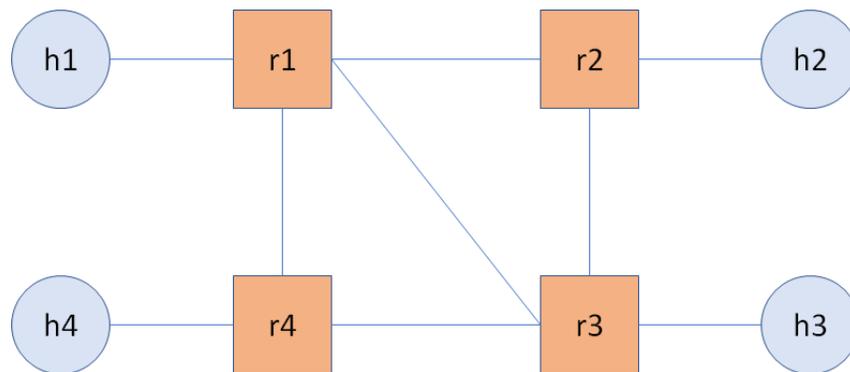


## 1. Prerequisites

You need to install `ipmininet` emulation tool. For more details, check [this](#) web page. It is recommended to use the provided virtual machine through Vagrant. At a high level, `ipmininet` enables creating full functional IP networks inside a single machine. You should get familiar with the [documentation](#) to complete this lab.

## 2. Lab Description

In this lab, you will build a small IP network representing a single AS. The network is depicted in the following figure, and it consists of four routers and four hosts. You will create the hosts, links and routers. You will also install the required forwarding tables. A startup code, called `network.py`, is provided to show a typical usage of `ipmininet` APIs. You will need to modify this file to build the final network.



The network has the following subnets:

Network segment	Subnet	Network segment	Subnet
<b>h1-r1</b>	10.11.0.0/24	<b>r1-r2</b>	10.12.0.0/24
<b>h2-r2</b>	10.22.0.0/24	<b>r1-r3</b>	10.13.0.0/24
<b>h3-r3</b>	10.33.0.0/24	<b>r1-r4</b>	10.14.0.0/24
<b>h4-r4</b>	10.44.0.0/24	<b>r2-r3</b>	10.23.0.0/24

		<b>r3-r4</b>	<b>10.34.0.0/24</b>
--	--	--------------	---------------------

In this lab, you will not rely on routing protocols (e.g., OSPF) to build the forwarding tables. Instead, you will populate the four routers with *static routes* to forward packets on the shortest path for every host pair in the network. We assume that all links have the same cost. For instance, a packet from h1 to h3 should go through r1 → r3 and not r1 → r2 → r3. To build these tables, you need to know the **IP address of every interface** in the network. For the purpose of this lab, every entry in the forwarding table should be in the format: (Destination Prefix → Next Hop).

*In the future*, we may build multiple ASs and run BGP and OSPF to populate the forwarding tables.

First, run the following commands and include screenshots of their outputs in the report:

```
pingall
h1 ping h2
h1 ping h3
h3 ping h4
h1 traceroute h2
h1 traceroute h3
h2 traceroute h4
```

Second, show the routing tables for r1 and r3. You can do so by running the following command:

```
<router> ip -f inet route
```

Then, take down the network segment r1-r3 (i.e., comment out the line of code that creates that subnet). Identify the impacted routes, and run the following commands (and show their outputs):

```
h1 ping h3
h3 ping h4
h1 traceroute h2
h1 traceroute h3
h2 traceroute h4
```

**Note:** The provided code and lab description are based on `ipmininet v0.9`. If you plan to use Vagrant, you can find the Vagrant box of `ipmininet v0.9` [here](#).

### 3. Submission

You need to submit:

- (1) The source code that you developed to create the network.
- (2) A detailed lab report.

The files should be compressed in a single (.zip) archive. The code should run without any errors.