# static-routing

<table>
<thead>
<tr>
<th>Version</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini</td>
</tr>
<tr>
<td><strong>E-mail</strong></td>
<td><a href="mailto:contact@netkit.org">contact@netkit.org</a></td>
</tr>
<tr>
<td><strong>Web</strong></td>
<td><a href="http://www.netkit.org/">http://www.netkit.org/</a></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>an example of configuration of static routes</td>
</tr>
</tbody>
</table>
copyright notice

- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material “as is”, with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.
step 1 – network topology
high level view
step 1 – network topology configuration details

- **Collision Domain A**
  - **pc1**
  - **r1**
  - **eth0**
  - **195.11.14.0/24**
  - **.5**

- **Collision Domain B**
  - **r1**
  - **eth1**
  - **100.0.0.8/30**
  - **.1**
  - **.9**

- **Collision Domain C**
  - **pc2**
  - **r2**
  - **eth0**
  - **200.1.1.0/24**
  - **.7**
  - **.1**
step 2 – the lab

- lab directory hierarchy
  - lab.conf
  - pc1/
  - pc1.startup
  - pc2/
  - pc2.startup
  - r1/
  - r1.startup
  - r2/
  - r2.startup
step 2 – the lab

lab.conf

r1[0]="A"
r1[1]="B"
r2[0]="C"
r2[1]="B"
pc1[0]="A"
pc2[0]="C"

pc1.startup
ifconfig eth0 195.11.14.5 netmask 255.255.255.0 broadcast 195.11.14.255 up
#route add default gw 195.11.14.1 dev eth0

pc2.startup
ifconfig eth0 200.1.1.7 netmask 255.255.255.0 broadcast 200.1.1.255 up
#route add default gw 200.1.1.1 dev eth0

the routing table entries will be added manually
step 2 – the lab

---

**r1.startup**

```plaintext
ifconfig eth0 195.11.14.1 netmask 255.255.255.0 broadcast 195.11.14.255 up
ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
#route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
```

**r2.startup**

```plaintext
ifconfig eth0 200.1.1.1 netmask 255.255.255.0 broadcast 200.1.1.255 up
ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
#route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
```

---

The routing table entries will be added manually.
step 3 – testing connectivity

pc1:~# ping 195.11.14.1
PING 195.11.14.1 (195.11.14.1) 56(84) bytes of data.
64 bytes from 195.11.14.1: icmp_seq=1 ttl=64 time=3.17 ms
64 bytes from 195.11.14.1: icmp_seq=2 ttl=64 time=0.371 ms
64 bytes from 195.11.14.1: icmp_seq=3 ttl=64 time=0.308 ms

--- 195.11.14.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2019ms
rtt min/avg/max/mdev = 0.308/1.285/3.176/1.337 ms

interfaces on the same domain can reach each other
step 3 – testing connectivity

interfaces on different domains cannot be reached

Can you tell why?
step 3 – inspecting routing tables

- both routers and pcs don’t know how to reach networks that are not directly connected to them

**pc1**

```
pc1:~# route
Kernel IP routing table
Destination    Gateway    Genmask     Flags Metric Ref  Use Iface
195.11.14.0    *          255.255.255.0 U     0   0      0  eth0
```

**r1**

```
r1:~# route
Kernel IP routing table
Destination    Gateway    Genmask     Flags Metric Ref  Use Iface
100.0.0.8       *          255.255.255.252 U    0    0      0  eth1
195.11.14.0     *          255.255.255.0    U    0    0      0  eth0
```

- directly connected networks are automatically inserted into the routing table when the corresponding interface is brought up
- this is a common behavior of all ip devices (even real-world routers!)
step 4 – default routes on pcs

- to fix the problem we could specify the default route on the pcs: “through this gateway (ip number) you can reach all the other networks”

For pc1:

```
$ route add default gw 195.11.14.1
$ route
```

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
195.11.14.0 * 255.255.255.0 U 0 0 0 eth0
default 195.11.14.1 0.0.0.0 UG 0 0 0 eth0
```

For pc2:

```
$ route add default gw 200.1.1.1
$ route
```

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
200.1.1.0 * 255.255.255.0 U 0 0 0 eth0
default 200.1.1.1 0.0.0.0 UG 0 0 0 eth0
```
step 4 – default routes on pcs: test

pc1:~# ping 100.0.0.9
PING 100.0.0.9 (100.0.0.9) 56(84) bytes of data.
64 bytes from 100.0.0.9: icmp_seq=1 ttl=64 time=0.451 ms
64 bytes from 100.0.0.9: icmp_seq=2 ttl=64 time=0.299 ms
64 bytes from 100.0.0.9: icmp_seq=3 ttl=64 time=0.320 ms

--- 100.0.0.9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.299/0.356/0.451/0.070 ms

pc1:~#  

the “backbone interface” of r1 is reachable
step 4 – default routes on pcs: test

pc1:

```
# ping 100.0.0.10
PING 100.0.0.10 (100.0.0.10) 56(84) bytes of data.
--- 100.0.0.10 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6105ms
```

pc1:

```
#?
```

pc1:

```
```

interfaces on r2 seem unreachable! can you tell why?
step 4 – let’s inspect the network

- do echo request packets reach r2?
- let’s check...
  - while pinging from pc1 100.0.0.10 sniff on interface eth1 of r2

```
r2:~# tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
16:06:58.977 851 arp who-has 100.0.0.10 tell 100.0.0.9
16:06:59.088 906 arp reply 100.0.0.10 is-at fe:fd:64:00:00:0a
16:06:59.899 909 IP 195.11.14.5 > 100.0.0.10: icmp 64: echo request seq 1
16:07:01.001 888 IP 195.11.14.5 > 100.0.0.10: icmp 64: echo request seq 2
16:07:01.001 888 IP 195.11.14.5 > 100.0.0.10: icmp 64: echo request seq 3
```

5 packets captured
5 packets received by filter
0 packets dropped by kernel

echo requests are arriving!
step 4 — r2’s routing table

- pc1’s address is 195.11.14.5
- r2 does not know how to reach such an address.
- echo requests arrive to r2 but r2 does not know where echo replies should be forwarded!
- somebody should teach r2 how to reach pc1
- we may insert a static route into the routing table of r2
step 5 – configuring a static route

```
r2:~# route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
```

- network 195.11.14.0...
- ...with netmask 255.255.255.0...
- ...is reachable via 100.0.0.9...
- ...on interface eth1

```
r2:~# route
Kernel IP routing table
Destination     Gateway          Genmask    FlagsMetric   Ref   Use Iface
100.0.0.8        *               255.255.255.252 U      0     0   0 eth1
200.1.1.0        *               255.255.255.0     U      0     0   0 eth0
195.11.14.0      100.0.0.9       255.255.255.0     UG     0     0   0 eth1
```

netkit – [ lab: static routing ]  
last update: Apr 2007
step 5 – configuring a static route

- a similar configuration should be deployed on r1

```
r1:~# route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
r1:~# route
```

Kernel IP routing table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Genmask</th>
<th>Flags</th>
<th>Metric</th>
<th>Ref</th>
<th>Use</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0.0.8</td>
<td>*</td>
<td>255.255.255.252</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth1</td>
</tr>
<tr>
<td>200.1.1.0</td>
<td>100.0.0.10</td>
<td>255.255.255.0</td>
<td>UG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth1</td>
</tr>
<tr>
<td>195.11.14.0</td>
<td>*</td>
<td>255.255.255.0</td>
<td>U</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>eth0</td>
</tr>
</tbody>
</table>
step 5 – testing static routes

the pcs can reach each other

pc1:

```
pc1:~# ping 200.1.1.7
PING 200.1.1.7 (200.1.1.7) 56(84) bytes of data.
64 bytes from 200.1.1.7: icmp_seq=1 ttl=62 time=111 ms
64 bytes from 200.1.1.7: icmp_seq=2 ttl=62 time=1.05 ms
64 bytes from 200.1.1.7: icmp_seq=3 ttl=62 time=0.820 ms

--- 200.1.1.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.820/37.779/111.467/52.105 ms
```

pc2:

```
pc2:~# ping 195.11.14.5
PING 195.11.14.5 (195.11.14.5) 56(84) bytes of data.
64 bytes from 195.11.14.5: icmp_seq=1 ttl=62 time=0.954 ms
64 bytes from 195.11.14.5: icmp_seq=2 ttl=62 time=0.947 ms
64 bytes from 195.11.14.5: icmp_seq=3 ttl=62 time=1.27 ms

--- 195.11.14.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.947/1.057/1.271/0.153 ms
```
proposed exercises

- The default route can be statically configured by using:

  ```
  route add default gw 195.11.14.1 dev eth0
  ```

- Can you give a command to configure a static route that is equivalent to the default route?

  ```
  route add -net __ netmask __ gw __ dev __
  ```
proposed exercises

- not all the routing tables contain a default route
- the network of this lab is so simple that routers $r_1$ and $r_2$ can be also configured to exclusively use default routes
- try such a configuration and test it