ICN – Examination date: 13-02-2014 – “Knot”

Available time: <u>150 minutes</u>.

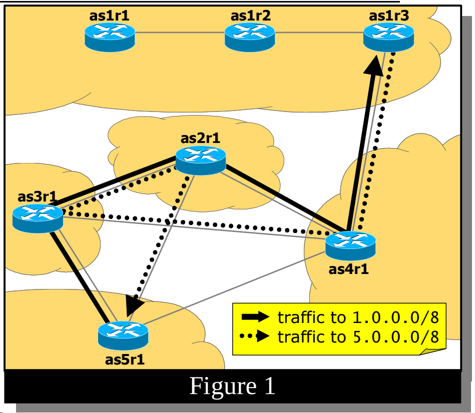
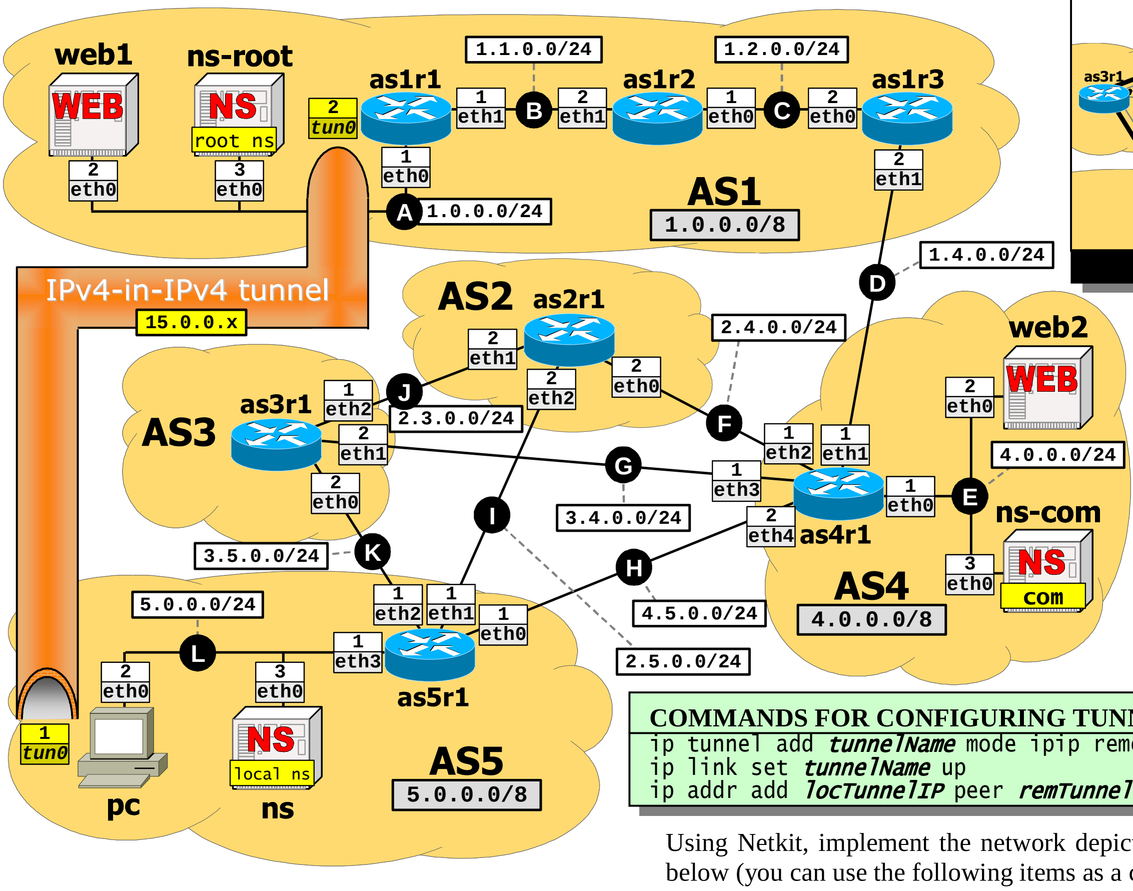


Figure 1



COMMANDS FOR CONFIGURING TUNNELS

```
ip tunnel add tunnelName mode ipip remote remIP local locIP ttl 20
ip link set tunnelName up
ip addr add locTunnelIP peer remTunnelIP dev tunnelName
```

Using Netkit, implement the network depicted in the figure and described below (you can use the following items as a checklist).

□ Remember to configure a default route on the end hosts.

□ Routing between different ASes is implemented by using BGP.

- □ None of the routers announce the default route (`0.0.0.0/0`) or apply filters.
- □ All peering LANs are announced in BGP. `AS1`, `AS4`, and `AS5` also announce their own subnet (in gray).
- □ `as2r1`, `as3r1`, `as4r1`, and `as5r1` apply **<u>preferences</u>** to direct traffic between `AS1` and `AS5` as indicated in Figure 1.

□ Internal routing within `AS1` is implemented by using OSPF. All router interfaces belong to area `0.0.0.0` (backbone).

□ `web1` and `web2` are web servers running Apache. They just serve a default web page.

□ `ns` is the local name server within `AS5`; `ns-root` is the root name server; `ns-com` is the authority for `com`. The only requested name is `server.com`, which resolves to both web servers, implementing a DNS-based load balancing policy.

□ There is an IPv4-in-IPv4 tunnel between `pc` and `as1r1`.

- □ The tunnel is used **<u>exclusively</u>** for sending traffic **<u>from</u>** `pc` **<u>to</u>** `web1` (and is not used in the opposite direction).
- □ In-tunnel routing is implemented statically (in addition to any other existing routing mechanisms).

<u>Goals</u>:

- All routers must be able to reach any IP addresses in the network.
- Traffic from `AS5` to `AS1` and from `AS1` to `AS5` must **<u>preferably</u>** flow along the paths in Figure 1.
- `pc` must be able to access the web page served by `server.com` using the `links` browser.
- `pc` must communicate with `web1` by using the tunnel.