UKERNA
IPv6 Hands-on Workshop

Lab 4: **Tunneling and BGP**

UKERNA, Lancaster University
and University of Southampton, 2006

Lab Overview

- We will now assume that no groups have external IPv6 connectivity
- Each group is now a separate campus with only native IPv4 connectivity to JANET
- IPv6 connectivity will be gained through the use of a tunnel.
  - e.g. Your campus to the JANET IPv6 Experimental Service
- We will use GRE tunnels.

- Later we will establish a BGP peering via the tunnel for both IPv6 unicast and multicast.
  - Note that the BGP peering isn't strictly necessary. In many cases a static default route would be fine.

For this lab we have removed IPv6 connectivity on the up interface of your router, meaning that IS-IS has also been removed.

At this stage only the downstream interfaces have IPv6 connectivity and your hosts should still be able to access each other.

Tunneling parameters

- For these exercises the local tunnel end-point (tunnel source) will be the IPv4 address of your upstream interface.
  - This is because we have preconfigured the remote side of the tunnels using these addresses.
- The remote tunnel end-point (tunnel destination) is **148.88.147.220**
  - This the loopback address of the core router
- You also have to specify the IPv6 address to use on the tunnel interface. The core router will be using **2001:630:81:4a0::X:1/112**, where **X** is the group digit (1-6).
  - We are using prefix length 112 for the tunnels here. In general you might use anything in the range of 64-126; we suggest a /64 is just fine.

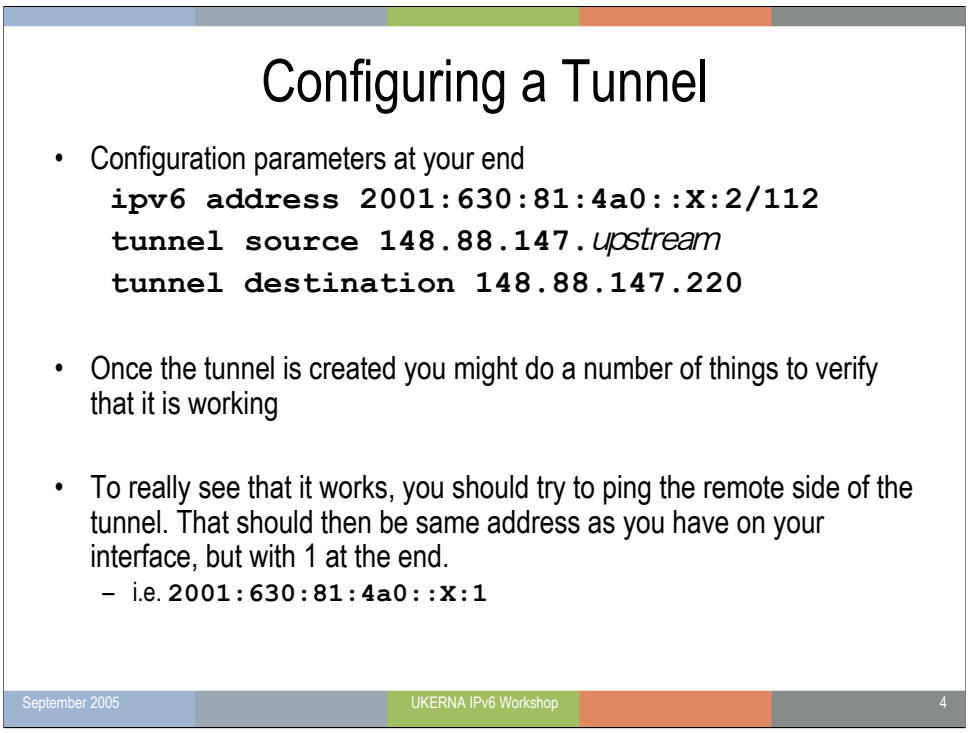
CISCO

The name of the tunnel interface must be "**tunnel"** followed by a number

If you may be doing multicast we suggest not starting from 0 (this is because IOS automatically creates tunnel interfaces for IPv6 PIM-SM multicast Rendezvous Points (RPs) starting from 0)

JUNIPER

In JUNOS GRE is configured by using a gr interface. On hardware based Junipers a tunnel PIC is required.



CISCO

The tunnel setup for cisco requires the parameters from above entered under a tunnel interface:

```
interface Tunnel100
    ipv6 address 2001:630:81:4a0::X:2/112
    tunnel source 148.88.147.upstream
    tunnel destination 148.88.147.220
```

Here we also have to define the tunnel type (in this case IPv6 in IPv4):

```
    tunnel mode ipv6ip
```

To see that it works try:

```
show ipv6 interfaces
```

JUNIPER

In Juniper you can either use a GRE encapsulated interface (gr-1/0/0 etc) or an IP-IP interface. (ip-1/0/0 etc)

In the case of this lab we are going to use a GRE encapsulated interface (gr)

```
From [edit interfaces gr-0/0/0 unit 0]
    set tunnel source 148.88.147.upstream
    set tunnel destination 148.88.147.220
    set family inet6 address 2001:630:81:4a0::X:2/112
```

The type of tunnel is set automatically by the contents of the source and destination fields. To see if it is working try:

```
show route table inet6
```

Tunnel verification 2

- To test the tunnel you may also ping the all nodes multicast address **ff02::1** over the tunnel interface.
- You should then get replies from the link-local address of the other tunnel end-point
- Note that for protocol-41 tunnels:
  - IOS derives link-local address from tunnel end-point address
    - e.g. 152.78.108.2 has a link-local address of **FE80::984E:6C02**
  - JUNOS uses physical interfaces which have hardware addresses
    - e.g. Our gr-1/2/0 on the head router (148.88.147.220) has link-local of fe80::2a0:a5ff:fe56:fc5
- We can now reach the other side of the tunnel, but we can't go beyond without adding some routing. We will use BGP for this.

CISCO

To do this, type just "**`ping`**" and press enter.

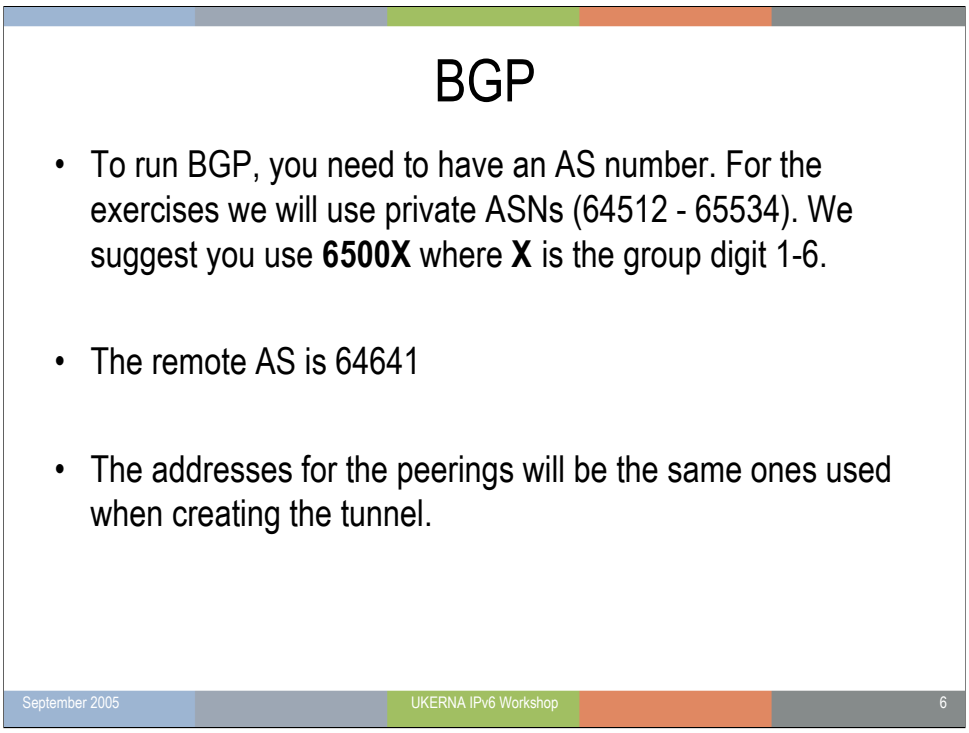
Then specify **`ipv6`** and target address **ff02::1**.

Next use the proposed values, but output interface should be **`tunnel100`** or whatever you called your interface

JUNIPER

Much like the normal ping syntax JUNOS accepts an interface as input to the ping command:

"ping ff02::1 interface gr-0/0/0"



CISCO

```
router bgp 6500X
    no bgp default ipv4-unicast
    neighbor 2001:630:81:4a0::X:1 remote-as 64641
    address-family ipv6 unicast
    neighbor 2001:630:81:4a0::X:1 activate
```

JUNIPER

To connect two different ASNs:

```
From [edit routing-options]
    set autonomous-system 6500X

From [edit protocols bgp group session-to-AS64641]
    set type external
    set peer-as 64641
    set neighbor 2001:630:81:4a0::X:1
    set description “BGP Tunnel to the M7i”
```

Verifying peering

- Once you have configured your peering, try viewing its status.
- You should see that you have a peering to one neighbour, how long it has been up for and that you are receiving at least one prefix from them
  - It might take a few seconds for this to happen
- You might see that state is **active.** That means that the router is attempting to set up a peering to the neighbour but it has not succeeded yet
  - If this is still the case after short while, check that your config is correct

CISCO

show bgp ipv6 unicast neighbors 2001:630:D0:8000::X:1 routes

JUNIPER

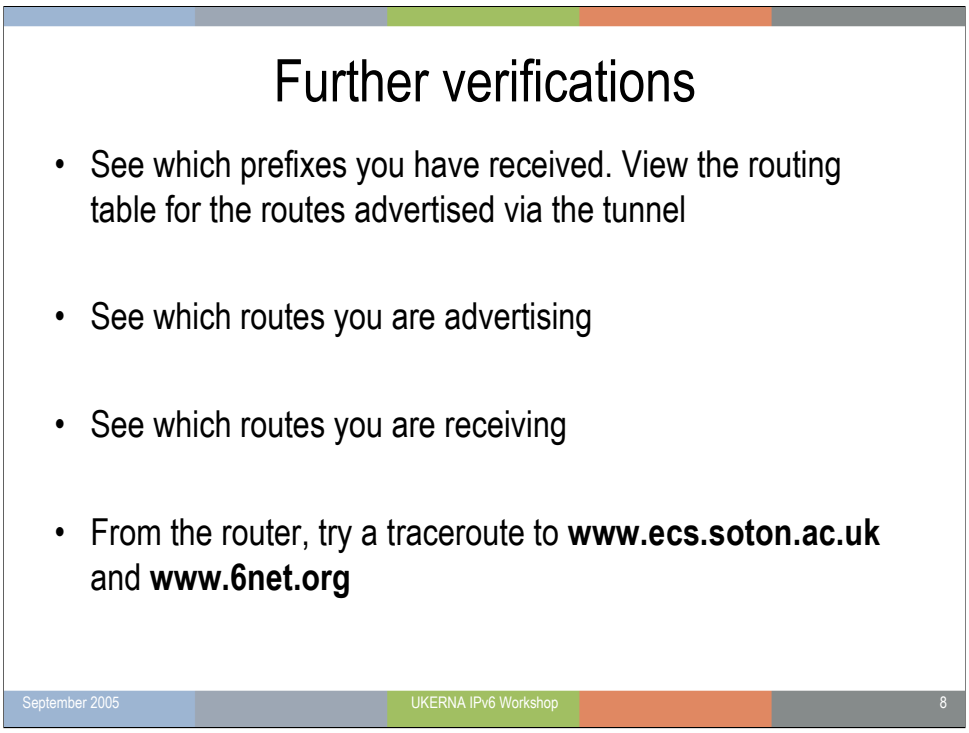
After you have configured your peer check that you are connected:

show bgp neighbor

(we are looking for “State: Established”)

The following command shows further useful information about the connection.

show bgp summary



CISCO

To see what you are advertising, do

show bgp ipv6 unicast neighbors 2001:630:71:4a0::X:1 advertised-routes

You should see that nothing is being advertised. To see the BGP routes you received:

show bgp ipv6 unicast

To see that you got BGP routes in the routing table together with your other routes

show ipv6 route

(look for B prefix)

JUNIPER

To see what you are advertising:

show route advertising-protocol bgp 2001:630:81:4a0::X:1

(you should not see any yet)

To see the BGP routes you received:

show route protocol bgp

To view all your routes:

show route

Advertising a prefix

- Now it's your turn to advertise something. Each group has a /60 prefix **2001:630:81:04X::/60** which needs advertising
- IOS and JUNOS have different procedures for doing this
  - In IOS we have to announce which prefix we have manually
  - In JUNOS we can reuse the two policies from the IS-IS exercise
- Once done you use the command from the previous slide to check which prefixes you are advertising. Also check which prefixes you are now receiving
  - Unless you are ahead of the other groups, you should see prefixes that the other groups are advertising
- Please don't proceed any further after advertising your /60 until you can see the prefixes from several of the other groups

CISCO

As a first step, we create a static null route for the /60 prefix. This means that packets sent to an address inside your prefix for which there is no more specific route will be dropped, which is what we want.

To do this

```
ipv6 route 2001:630:D0:04X0::/60 null 0
```

Next, for this to be announced, add "network 2001:630:81:04X0::/60" in the "address-family ipv6 unicast" section.

JUNIPER

From the earlier lab on IS-IS routing we have our two policies; redistribute-connected and v6-aggregation an we can use these here in the same manor:

```
From [edit protocols bgp group session-to-AS64641]
    set export v6-aggregation
    set export redistribute-connected
```

NOTE: The order matters as policies are run one after another.

Check that you are now advertising your routes:

```
show route advertising-protocol bgp 2001:630:81:4a0::X:1
```

Prefix-list filtering

- We will now try to use prefix-lists to filter out all but the **2001:630:81:04X0::/60** prefix
- This can be done in several ways. You might define one or both of the below
  - Block any prefixes longer than /60
  - Allow only the specific /60 prefix
- See how the two methods differ

CISCO

Method 1 - Blocking longer than /60:

```
ipv6 prefix-list denygt60 permit ::/0 le 60
ipv6 prefix-list denygt60 deny ::/0 le 128
```

Method 2 - Allowing only the specific /60 prefix:

```
ipv6 prefix-list allow60 permit 2001:630:81:04X0::/60
ipv6 prefix-list allow60 deny ::/0 le 128
```

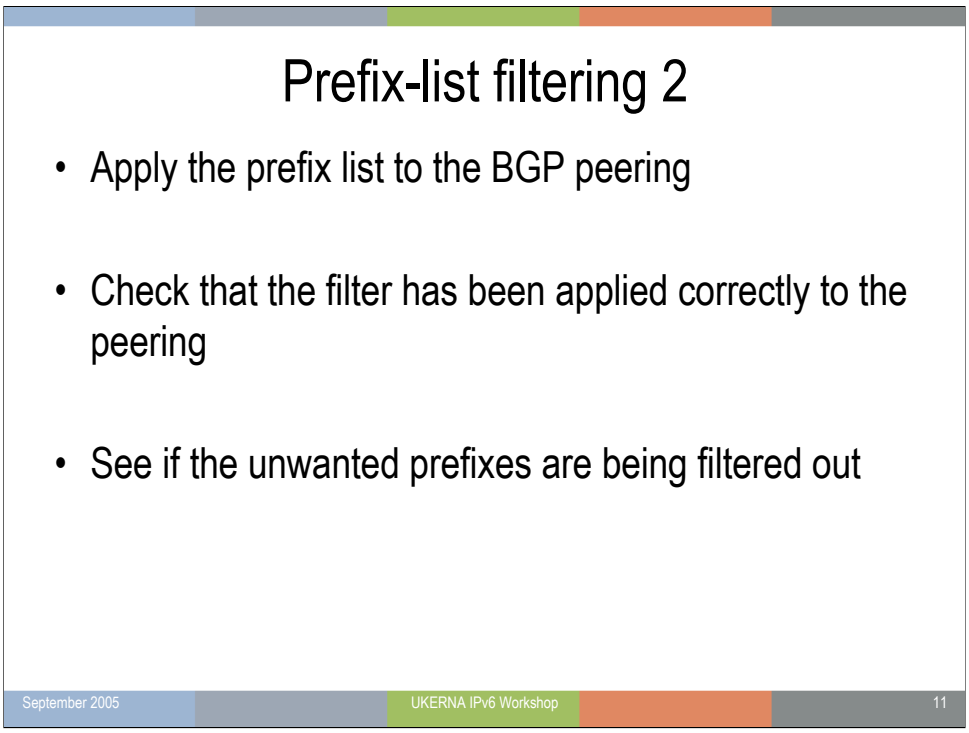
To check what these would allow, you might try:

```
show bgp ipv6 unicast prefix-list denygt60
show bgp ipv6 unicast prefix-list allow60
```

JUNIPER

Allow the /60 prefix only:

```
From [edit policy-options policy-statement ipv6-bgp]
    set term our-prefix from family inet6
    set term our-prefix from route-filter 2001:630:81:04X0::/60 exact
    set term our-prefix then accept
    set term reject-others from family inet6
    set term reject-others then reject
```



CISCO

To apply one of the prefix-lists, from "address-family ipv6 unicast" in the BGP config run:

neighbor 2001:630:81:4a0::X:1 prefix-list denygt60 in

Next type

clear bgp ipv6 unicast 2001:630:81:4a0::X:1

This will cause the peering to be reset, and prefix-list will be in use when the peering is reestablished. We will show you a less brutal way to do this on next slide. To check that the prefix list is in fact in use, try the following commands:

show bgp ipv6 uni neigh 2001:630:81:4a0::X:1 routes

show ipv6 routes

JUNIPER

Now export the filter in the bgp session:

From [edit protocols bgp group session-to-AS64641]

insert export ipv6-bgp before v6-aggregation

To see the routes you are now advertising:

show route advertising-protocol bgp 2001:630:81:4a0::X:1

To see the aggregated routes from the other workgroups:

show route protocol bgp

Soft-reconfiguration (IOS only)

- By enabling soft-reconfiguration you can add prefix-lists etc without resetting the peering
- Add the following under **`address-family ipv6 unicast`**

```
neigh 2001:630:81:4a0::X:1 soft-reconfiguration inbound
```

- You can then do e.g. **`clear bgp ipv6 unicast * soft`** to reconfigure without resetting the sessions and temporarily losing the routes
- After adding soft-reconfiguration, compare
  - `sh bgp ipv6 uni neigh 2001:630:81:4a0::X:1 received-routes`
  - `sh bgp ipv6 uni neigh 2001:630:81:4a0::X:1 routes`
  - The former shows what we receive, while the latter shows what we actually use

CISCO

Follow the steps laid out in the slide above.

JUNIPER

No configuration required - we're done!

Summary

- GRE Tunnels
- BGP

- Next: Multicast - if time permits :)