

How to configure basic IPv6 interoperability between an Allied Telesyn AT-AR450S and Red Hat Linux

Introduction

IPv6 is enabled on Allied Telesyn routers via a special feature licence. To obtain a special feature licence contact your nearest Allied Telesyn authorised distributor or reseller.

Allied Telesyn routers and switches support the IPv6 protocol. For a detailed overview of the IPv6 protocol, please consult the IPv6 chapter in your product's software reference.

We recommend that you have the latest software release running on your router or switch, as well as the latest patch release.

Whenever this document refers to routers, the same principles can also be applied to the Allied Telesyn range of switches.

Software Used

In the following examples, an AR450 running 54-252 with patch-01, was configured against a Red Hat™ Linux™ client running Red Hat 8.0, Kernel 2.4.18-14. The PC has one network card installed.

We suggest you make sure you are running an IPv6-enabled kernel.

It is assumed that you have a working knowledge of IPv6 and Red Hat Linux.

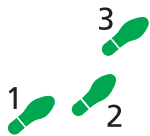
Some basic tools that function between Allied Telesyn and Linux IPv6-enabled devices are:

- PING6
- TCPDUMP
- TRACEROUTE6
- TELNET

Configuration Basics

When an interface is created on the AR450S, an IPv6 link local address is added to the interface. Devices on the same link can then PING each other.

i IPv6 link local addresses are assigned to interfaces by default on RED HAT 8, and when IPv6 is enabled on other Linux distributions.



I. Configure the AR450

```
enable ipv6
create ipv6 int=eth0
sh ipv6 int=eth0
```

Figure 1: Example output from the SHOW IPV6 INTERFACE command on the AR450 router

```
IPV6 Interface Configuration
-----
Interface ..... eth0
Ipv6 Interface Index ..... 1
Link-layer address ..... 00-00-cd-05-01-99
Link-layer state ..... Up
EUI-64 Interface Identifier ..... 0200CDFFFE050199
IPSec ..... No
True MTU/Link MTU ..... 1500/1500
Multicast status ..... Enabled
Send Router Advertizements ? ..... No
Ipv6 Interface Addresses :
  Int  Addresses                                PLen  Decrement
  Type  Scope  State      Enabled  Valid  Preferred  Publish
-----
  0     fe80::0200:cdff:fe05:0199                    /64    No
  unicast link preferred Yes      infinite infinite      No
```

2. Configure the Linux PC

```
NETWORKING_IPV6=yes
IPV6INIT=yes
```

The commands above need to be added in `/etc/sysconfig/network`. After the change, reboot your computer, or restart the network.

```
# ifconfig eth0
eth0  Link encap:Ethernet HWaddr 00:C0:4F:15:07:EC
      inet addr:10.0.0.2 Bcast:10.0.0.255
        Mask:255.255.255.0
      inet6 addr: fe80::2c0:4fff:fe15:7ec/10 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:10 errors:0 dropped:0 overruns:0 frame:0
      TX packets:18 errors:0 dropped:0 overruns:0
        carrier:0
      collisions:0 txqueuelen:100
      RX bytes:860 (860.0 b) TX bytes:1388 (1.3 Kb)
      Interrupt:11 Base address:0xdc00
```

When Pinging a link local address and interface must be specified as multiple interfaces can have the same link local address.

3. Check the connectivity from the AR450 to the Linux PC

```
ping fe80::2c0:4fff:fe15:7ec%eth0 num=2
Echo reply 1 from fe80::02c0:4fff:fe15:07ec
  time delay 0 ms
Echo reply 2 from fe80::02c0:4fff:fe15:07ec
  time delay 0 ms
```

4. Check the connectivity from the Linux PC to the AR450

```
# ping6 -I eth0 fe80::0200:cdff:fe05:0199
PING fe80::0200:cdff:fe05:0199 (fe80::200:cdff:fe05:199)
  from fe80::2c0:4fff:fe15:7ec eth0: 56 data bytes
64 bytes from fe80::200:cdff:fe05:199: icmp_seq=1 ttl=64
  time=0.170 ms
64 bytes from fe80::200:cdff:fe05:199: icmp_seq=2 ttl=64
  time=0.336 ms
64 bytes from fe80::200:cdff:fe05:199: icmp_seq=3 ttl=64
  time=0.324 ms
```

5. Assign an IPv6 Address to the AR450

```
add ipv6 int=eth0 ip=3ffe::1/64
Info (1066267): interface successfully added
sh ipv6 int=eth0
```

Figure 2: Example output from the SHOW IPV6 INTERFACE command on the AR450

```

IPV6 Interface Configuration
-----
Interface ..... eth0
Ipv6 Interface Index ..... 1
Link-layer address ..... 00-00-cd-05-01-99
Link-layer state ..... Up
EUI-64 Interface Identifier ..... 0200CDFFFE050199
IPSec ..... No
True MTU/Link MTU ..... 1500/1500
Multicast status ..... Enabled
Send Router Advertizements ? ..... No
Ipv6 Interface Addresses :
  Int  Addresses
  Type  Scope  State      Enabled  Valid      PLen      Decrement      Publish
-----
  0     fe80::0200:cdff:fe05:0199
      unicast link preferred Yes      infinite infinite      No
  1     3ffe::0001 /64
      unicast global preferred Yes      2592000 604800      No

```

6. Assign an IPv6 Address to the Linux PC

```

/etc/sysconfig/network-scripts/ifcfg-eth0:
    IPV6ADDR=3ffe::2/64

```

Stop and Start the interface:

```

# /etc/sysconfig/network-scripts/ifdown eth0
# /etc/sysconfig/network-scripts/ifup eth0
# ifconfig eth0

eth0  Link encap:Ethernet HWaddr 00:C0:4F:15:07:EC
      inet addr:10.0.0.2 Bcast:10.0.0.255
          Mask:255.255.255.0
      inet6 addr: fe80::2c0:4fff:fe15:7ec/10 Scope:Link
      inet6 addr: 3ffe::2/64 Scope:Global
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:37 errors:0 dropped:0 overruns:0 frame:0
      TX packets:73 errors:0 dropped:0 overruns:0
          carrier:0
      collisions:0 txqueuelen:100
      RX bytes:3838 (3.7 Kb) TX bytes:6190 (6.0 Kb)
      Interrupt:11 Base address:0xdc00

```

7. Check the connectivity from the AR450 to the Linux PC

```

ping 3ffe::2

Echo reply 1 from 3ffe::0002 time delay 0 ms
Echo reply 2 from 3ffe::0002 time delay 0 ms
Echo reply 3 from 3ffe::0002 time delay 0 ms
Echo reply 4 from 3ffe::0002 time delay 0 ms
Echo reply 5 from 3ffe::0002 time delay 0 ms

```

8. Add another IPv6 address to ETH0 on the Linux PC

```
#ifconfig eth0 inet6 add 3ffe::3/64
eth0  Link encap:Ethernet HWaddr 00:C0:4F:15:07:EC
      inet addr:10.0.0.2 Bcast:10.0.0.255
        Mask:255.255.255.0
      inet6 addr: fe80::2c0:4fff:fe15:7ec/10 Scope:Link
      inet6 addr: 3ffe::2/64 Scope:Global
      inet6 addr: 3ffe::3/64 Scope:Global
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:59 errors:0 dropped:0 overruns:0 frame:0
      TX packets:107 errors:0 dropped:0 overruns:0
        carrier:0
      collisions:0 txqueuelen:100
      RX bytes:5778 (5.6 Kb) TX bytes:8827 (8.6 Kb)
      Interrupt:11 Base address:0xdc00
```

9. Add another IPv6 interface to VLAN1 on the AR450

```
create ipv6 int=vlan1
add ipv6 int=vlan1 ip=4ffe::0001/64
```

To reach the 4ffe::/64 network, the Linux client now has to route out through the gateway of 3ffe::1.

```
# /sbin/route -A inet6 add 4ffe::/64 gw 3ffe::1
# /sbin/route -A inet6 | grep -i eth0
```

Destination	Flags	Metric	Ref	Use	Iface	Next Hop
3ffe::/64						::
UA	256	2	0	eth0		
4ffe::/64						3ffe::1
UG	1	2	0	eth0		
fe80::/10						::
UA	256	0	0	eth0		
ff00::/8						::
UA	256	0	0	eth0		
::/0						::
UDA	256	0	0	eth0		

To check that the AR450 is set up correctly you can use the following command:

```
ping 4ffe:1
```

10. View the IPv6 neighbours on the AR450

```
ping 3ffe:2
ping 3ffe:3
sh ipv6 ndca
```

Figure 3: Example output from the SHOW IPV6 NDCACHE command on the AR450.

```

Ipv6 Neighbour Cache:
Ipv6 Address                               Link-layer address
Interface [port]      State                LastReachble      IsRouter
-----
3ffe::0003                                00-c0-4f-15-07-ec
eth0                                stale              0 msecs           no
3ffe::0002                                00-c0-4f-15-07-ec
eth0                                reachable         28300 msecs       no
fe80::02c0:4fff:fe15:07ec              00-c0-4f-15-07-ec
eth0                                reachable         37800 msecs       no
-----

```

II. View the IPv6 neighbours on the Linux PC

```

# /sbin/ip -6 neigh show
3ffe::1 dev eth0 lladdr 00:00:cd:05:01:99 router nud reachable

```

Basic troubleshooting

AR450S

- Check for valid feature licenses
- Check IP addresses
- SHOW IPV6 INT=<INTERFACE>
- SHOW IPV6 COUN
- ENABLE IPV6 DEBUG=ALL
- SHOW IP ROUTE
- Check cable connectivity

Red Hat Linux

- Make sure all packages are installed that are pre-requisites for IPv6
- Check to ensure that the IPv6 module is present
- ifconfig -a
- Check routes
- Check IP addresses