Migrating NITK Surathkal Campus Network to IPv6: Challenges and Opportunities

Mohit P. Tahiliani

Department of Computer Science and Engineering

Centre for Open-source Software and Hardware (COSH)

National Institute of Technology Karnataka, Surathkal

Mangalore, Dakshina Kannada, India

Credits: Akshay Revankar and Sushanth S. Rao

Observation #1: Router Advertisements (RAs)

1. Even when ‘accept_ra = 0’, RAs are accepted and routes get installed in kernel
    - `systemd-network` maintains the routing table in kernel. The default configuration in `systemd-network` is to ‘accept’ RAs.
    - Setting ‘accept_ra = 0’ via ‘sysctl’ is not sufficient. One must disable RA acceptance in `netplan` / `systemd-network`.
    - The issue was identified when the monitoring tool and the reverse proxy running in the host randomly stopped working on IPv6.
    - The root cause of the issue was identified by verifying the address configuration (`iproute2`, `systemd-network`) and later by verifying the routing table using `iproute2`
    - Packet captures (using `tshark`) confirmed the router advertisements and its configuration.
        - fe80::3640:b5ff:fed2:e2b8 → ff02::2 ICMPv6 70 Router Solicitation from 34:40:b5:d2:e2:b8
        - fe80::8296:21ff:fee0:301 → ff02::1 ICMPv6 86 Router Advertisement from 80:96:21:df:03:01

Observation #1: Router Advertisements (RAs)

2. Accept RAs from known sources only
    - Any system in the network can run RA daemon and broadcast RAs
    - One device that accepted RAs from multiple sources (known and unknown) lost connectivity
    - Setting up ACLs appropriately helped resolve the issue
    - Two questions that required attention:
        - How did the device end up selecting the route only for specific hosts/addresses?
            - The answer was identified by using `bpftrace`. Based on the packet hash, the kernel decides any one when having equal cost multiple routes (default behavior).
        - Why did `systemd-network` fail to identify the packet losses on that route and did not attempt to failback to another route (given that the other route existed)?
            - By default kernel only polls the status of nexthop/gateway by pinging it and since it was alive packets were routed to the said destination.

Observation #2: Support for Dual Stack

1. Captive portal
    - Network authentication at NITK is a part of the solution provided by the firewall vendor
    - Firewall did not support dual stack functionality; didn't work with IPv6
    - If the device is authenticated on IPv4, the firewall does not detect and store the corresponding IPv6 addresses associated with that device.
    - Disabling the captive portal was the only option; 802.1x authentication has been introduced for WiFi.
2. Kubernetes
    - Moodle deployment at NITK uses a Kubernetes Cluster
    - IPv6 was enabled on a few VLANs as a part of the gradual deployment process
    - Kubernetes stack picked up an IPv6 address and internally enabled IPv6 for its hosts
    - The cluster lost connectivity because dual stack functionality wasn't fully supported in versions < 1.21
    - Temporarily solved by performing a manual recovery; upgraded the Kubernetes version subsequently

Observation #3: Address assignment

1. DHCPv6 on tagged VLANs
    - Clients listen to ‘all the tagged VLANs’ on an interface to obtain information from DHCPv6
    - Obtain addresses from all the tagged VLANs; confirmed on Windows and Linux clients
    - *It’s a feature and not a bug* in Windows; this feature is in use with SMB file shares
    - Fix for Linux was supposedly pushed in versions > 5.15; issue still persists in versions > 6
    - More efforts are required to resolve this issue in Linux; untagged all the VLANs for now
2. Privacy extensions
    - Clients may use privacy extensions when SLAAC is enabled, where additional short-lived random addresses are generated
    - We saw temporary dynamic addresses in testbed, which were due to IPv6 privacy extensions in use
    - Made it difficult to analyze the network traffic
    - Temporary fix: used DHCPv6 with RAs (for gateway to be known) and disabled ‘AdvAutonomous’ flag

Observation #4: Routing

1. Static IPv6 routes on Core Switch
    - If and when the L3 managed Core Switch in our Data Centre reboots, the static IPv6 routes established for incoming and outgoing traffic do not work
    - Manually removing the routes and adding them again solves the problem
    - The suspected cause is neighbour solicitation is not being sent by the switch to the router
        - Difficult to reproduce the above behavior or even take a packet capture due to an old firmware
2. Re-establishment of connectivity with IPv6 takes more time
    - If an IPv6 router goes down, the re-establishment of IPv6 network connectivity takes quite some time.
    - It takes several minutes for the network connectivity to re-establish; perhaps due to RAs and NDP.
    - Overall network downtime is less if dual stack is enabled; not the case if the stack is IPv6-only
    - It is not clear whether this is a bug or an expected outcome:
        - Writing eBPF scripts (ideal) or using `bpftrace` will help in understanding Linux specific behavior in the above case (currently work in progress).

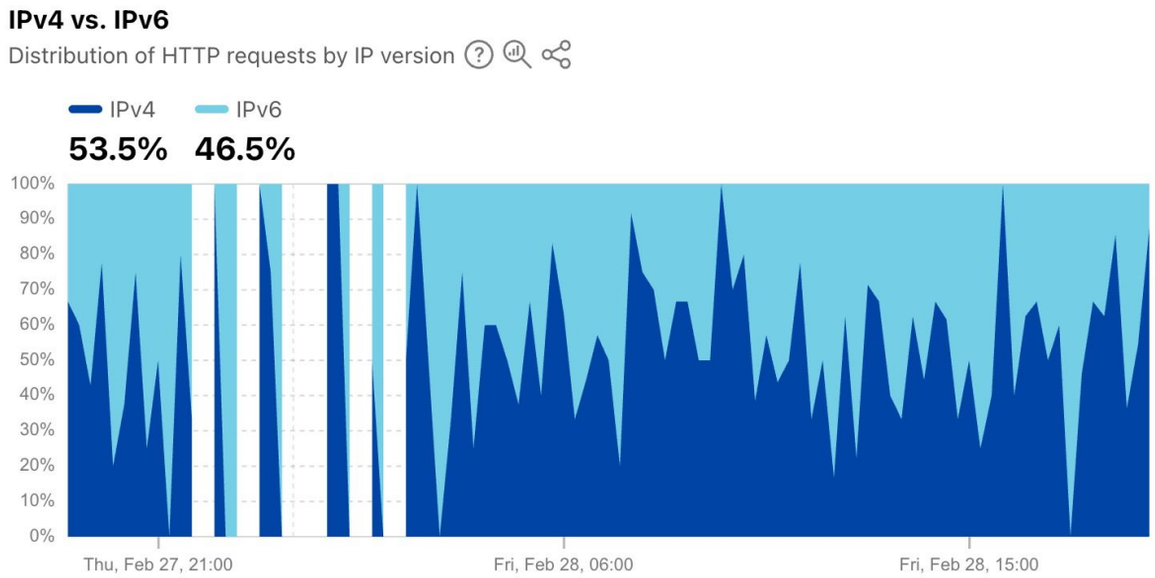
Observation #5: Miscellaneous

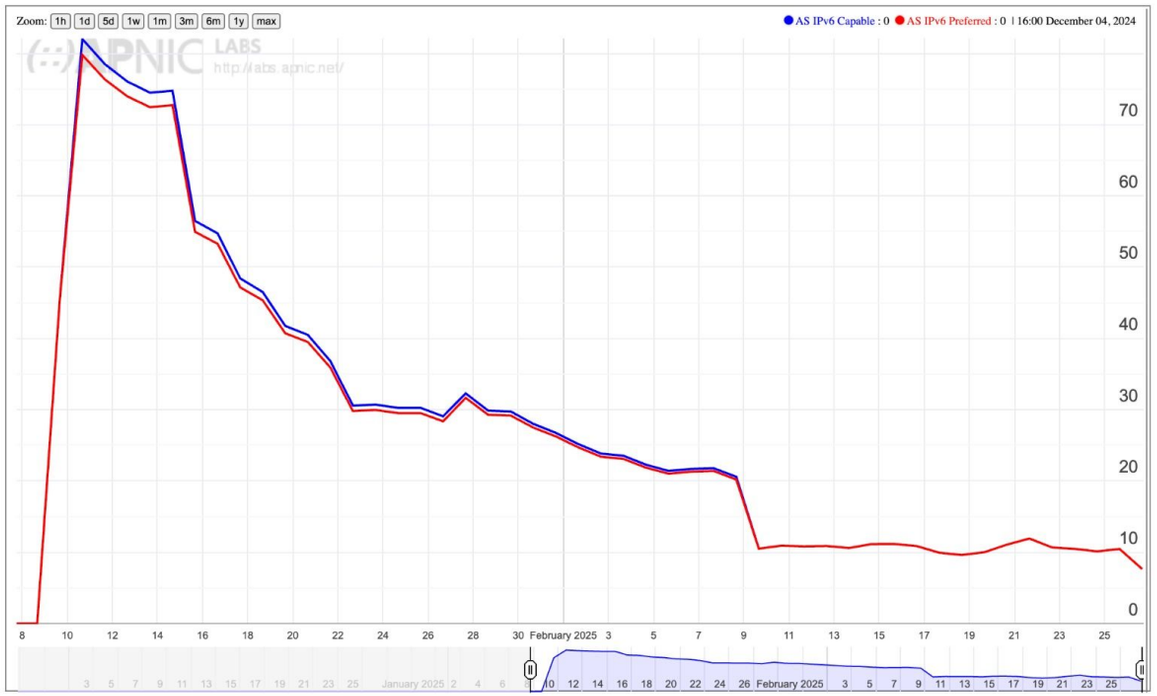
1. Noticed the following during the IPv6 deployment at NITK:
    - Our vendor specific limitations
        - Firewalling/web filtering support for IPv6 was not available
        - Support for monitoring BGPv6 was not available
            - Our BGP router runs on VyOS. VyOS did not support MIB for BGP4v2 due to lack of support from underlying FRR.
            - Workaround: used bgp.tools which showed the status of BGP routes for both v4 and v6.
        - IPv6 and MAC binding was not possible
            - Certain TP-Link switches were not registering IPv6 based MAC binding. In spite of multiple trials, we could not get this working.

Observation #5: Miscellaneous

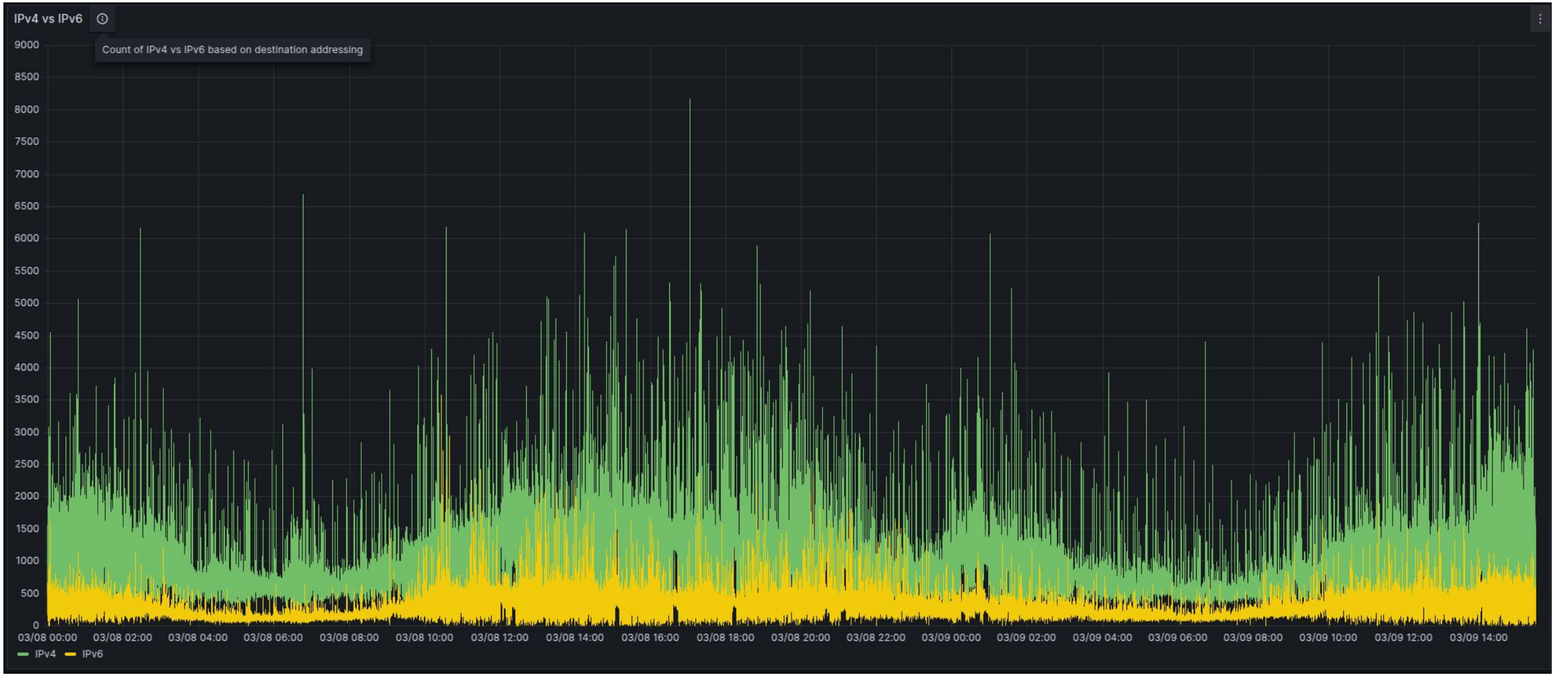
1. Noticed the following during the IPv6 deployment at NITK:
    - Lack of support for IPv6 in some network debugging and monitoring tools.
        - Tools such as hping, etc do not support IPv6 yet.
    - Lack support for DHCPv6 in Android
        - We need to rely on RDNSS which means we cannot use switches that don't support this RFC.
            - The work around is to use a separate router which advertises DNS configuration and nothing more.
        - A large number of people cannot go IPv6-only without it
    - No support for IPv6 on important proxies (e.g., some federations proxying 'eduroam')

Cloudflare and APNIC Measurements



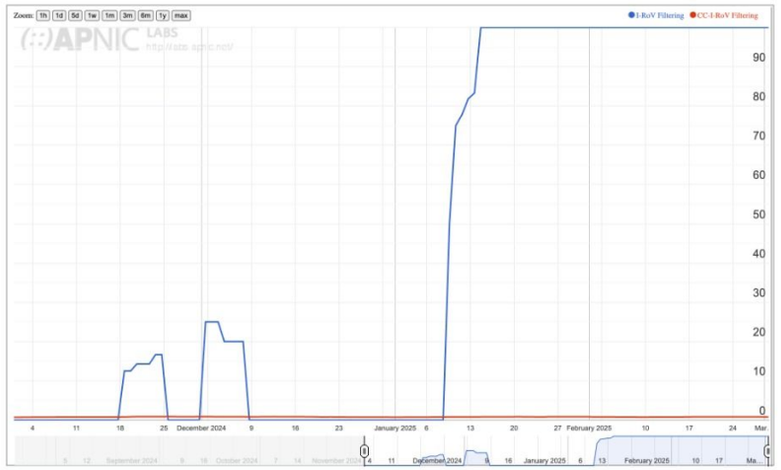


Local Measurements



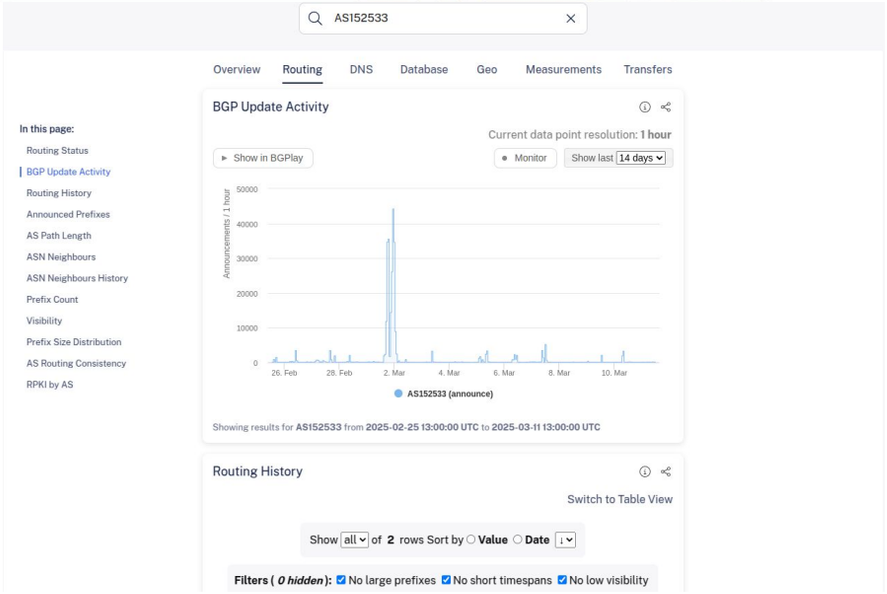
RIPE and APNIC Measurements

RPKI I-ROV Per-Country filtering for AS152533: NITK-AS-IN NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, India (IN)



7 Average Interval (days)
Hide country ASN List
Hide Country Totals
Show V4/V6 RPKI filtering
Redraw

Download JSON Time Series Data for AS152533 in Code IN



Traceroute for facebook.com (v6)

Point Selected: 2025-03-01 14:00:00 (UTC)

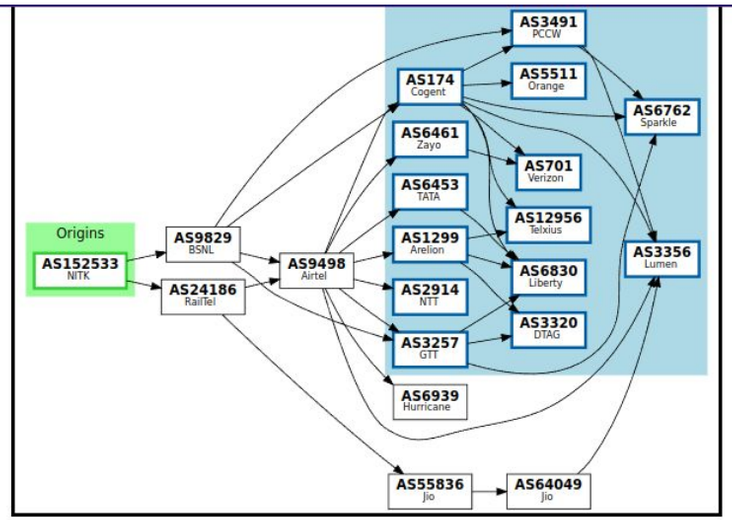
Traceroute Data at 2025-03-01 13:47:26 (UTC)

Hop	ASN	IP Address	Reverse DNS	RTT 1	RTT 2	RTT 3
1	152533	2400:4f20:80:c00::1	core-gw-datacenter.v6.nitk.ac.in	4.217 ms	4.148 ms	*
2	152533	2400:4f20:80:1::2	gw2-core.v6.nitk.ac.in	0.345 ms	0.283 ms	0.289 ms
3	152533	2400:4f20:80:3::1	bgp1.v6.nitk.ac.in	0.514 ms	0.363 ms	0.358 ms
4	9829	2001:4490:dffc:c04::2		12.987 ms	1.859 ms	1.782 ms
5	9829	2001:4490:efff:1::4:a5		17.862 ms	16.333 ms	16.645 ms
6	32934	2620:0:1cff:dead:beee::870	ae18.pr01.maa2.tfbnw.net	42 ms	22.188 ms	21.205 ms
7	32934	2620:0:1cff:dead:bef0::77b	po101.psw01.maa2.tfbnw.net	16.584 ms	17.106 ms	16.426 ms
8	32934	2a03:2880:f037:ffff::e9	po5.msw1am.01.maa2.tfbnw.net	16.646 ms	16.483 ms	16.392 ms
9	32934	2a03:2880:f137:83:face:b00c:0:25de	edge-star-mini6-shv-01-maa2.facebook.com	17.039 ms	16.75 ms	17.46 ms

Traceroute Data at 2025-03-01 14:02:21 (UTC)

Hop	ASN	IP Address	Reverse DNS	RTT 1	RTT 2	RTT 3
1	152533	2400:4f20:80:c00::1	core-gw-datacenter.v6.nitk.ac.in	3.978 ms	4.637 ms	4.299 ms
2	152533	2400:4f20:80:1::2	gw2-core.v6.nitk.ac.in	0.336 ms	0.268 ms	0.22 ms
3	152533	2400:4f20:80:3::1	bgp1.v6.nitk.ac.in	0.478 ms	0.459 ms	0.378 ms
4	24186	2401:b200:64::1		15.53 ms	15.499 ms	15.474 ms
5	24186	2401:b200:10::c1		15.981 ms	17.968 ms	15.862 ms

bgp.tools



▶ Chart Display Options

How are upstreams and downstreams calculated?

Upstreams

	ASN	Description	IPv4	IPv6
	AS24186	RailTel Corporation of India Ltd	✔	✔
	AS9829	BSNL (Bharat Sanchar Nigam Ltd)	✔	✔

Peers

	ASN	Description	IPv4	IPv6
	AS24186	RailTel Corporation of India Ltd	✔	✔
	AS9829	BSNL (Bharat Sanchar Nigam Ltd)	✔	✔

View | Looking Glass

National Institute of Technology Karnataka, Surathkal

AS Number **152533**

BGP | RIPE Atlas

Select BGP Session to query:

NITK-V4-01 [IPv4]

Input Prefix:

8.8.8.8

-

Query

```
8.8.8.0/24          unicast [NITK-V4-01 0000-00-00] * (?/-) [AS15169]
 Type: BGP
 BGP.as_path: 152533 9829 15169


8.0.0.0/12          unicast [NITK-V4-01 0000-00-00] * (?/-) [AS3356]
 Type: BGP
 BGP.as_path: 152533 9829 3491 3356
 BGP.community: (3356,0) [AS3356: APAC] [AS3356: Set BGP Local Pref to 100 (equal to
default)] [AS3356: Customer route] [AS3356: Singapore] [AS3356: SNG3 - Singapore3]
[AS3491: Asia peer] [AS3491: SIN03 Singapore peer] [AS3491: Peer]


8.0.0.0/9           unicast [NITK-V4-01 0000-00-00] * (?/-) [AS3356]
 Type: BGP
 BGP.as_path: 152533 9829 3491 3356
 BGP.community: [AS3491: Asia peer] [AS3491: SIN03 Singapore peer] [AS3491: Peer]
```

Funded by:

isif asia

Thank you!