CERNET2 IPv6-only Practice: Backbone, Servers, Clients and 4aaS

Xing Li, Congxiao Bao

2018-11-05

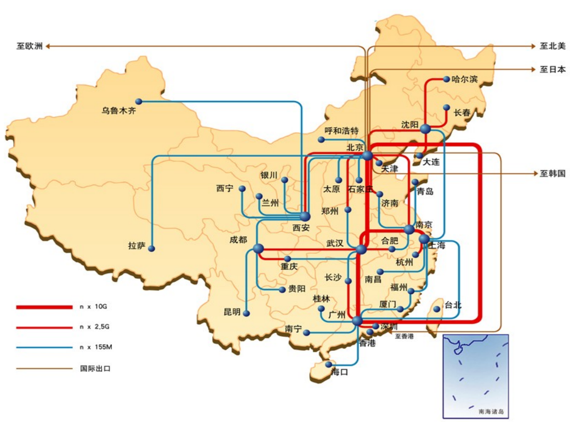
Outline

- A brief history
- Current practice
  - Backbone
  - IPv6-only servers
  - IPv6-only clients
  - 4aaS
- Remarks

A brief history

- CERNET
  - 1994, IPv4, 38 Pops, 2,000+ universities
- CERNET2
  - 2004, IPv6-only, 25 Pops, 1,000+ universities

CERNET (IPv4)



CERNET2 (IPv6-only)

CERNET2 design considerations

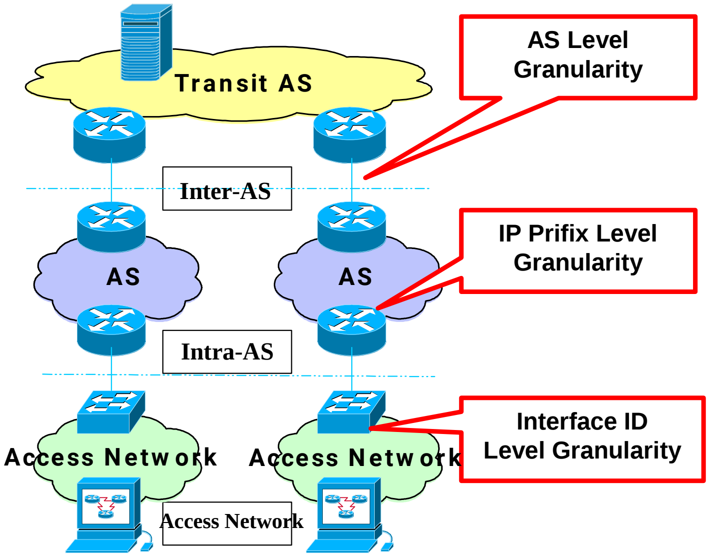
- Protocol selection
  - IPv6-only
- Promotion strategy
  - High performance and free
- Security
  - SAVA (IETF savi)
- Transition
  - IPv4 over IPv6 (IETF softwire)
  - Translation (IETF behave, softwire, v6ops)

Promotion strategy

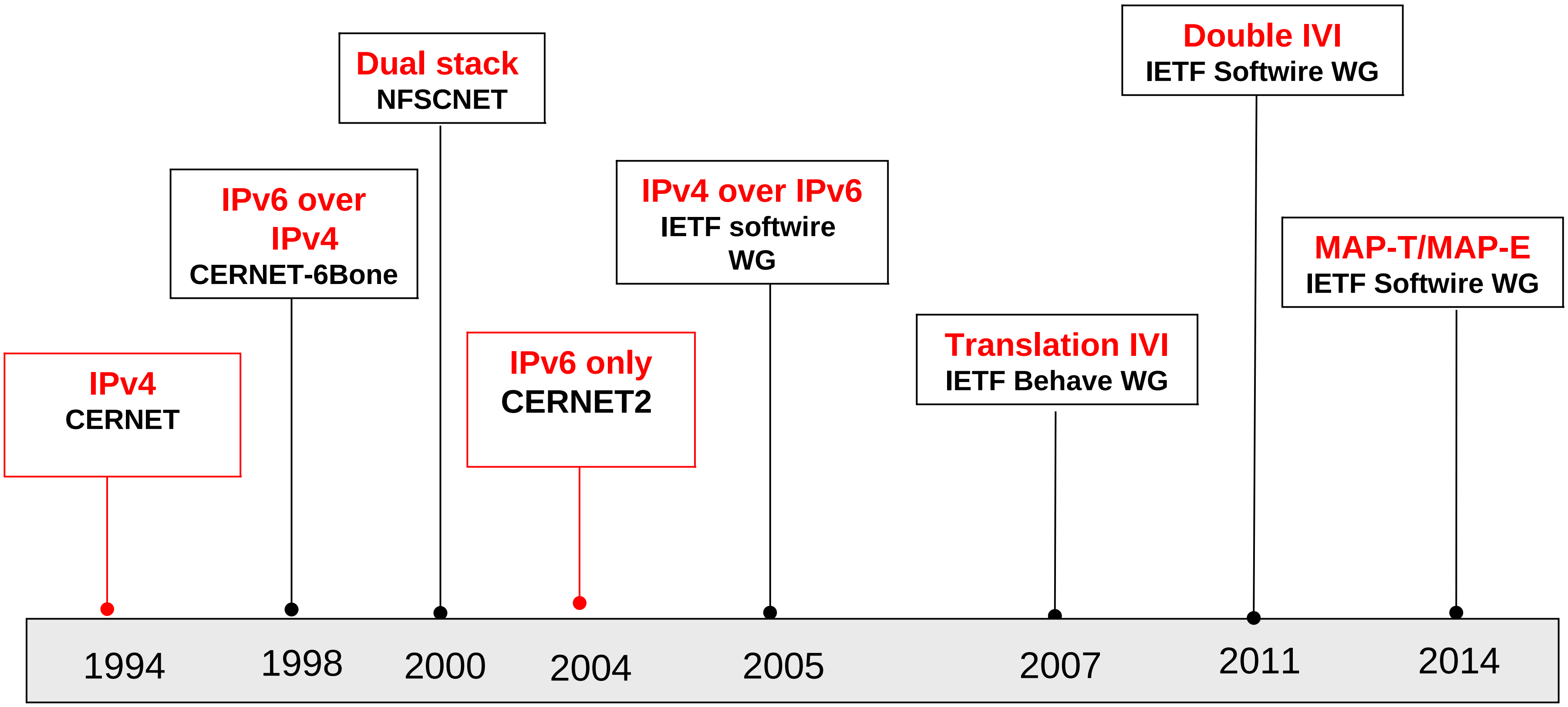
- Plan
  - Performance
    - CERNET (IPv4) is congested
    - CERNET2 (IPv6) is light loaded
  - Charging
    - CERNET (IPv4) is not free.
    - CERNET2 (IPv6) is free
  - Requirements
    - The Users need to run their applications using IPv6
- Reality
  - The users need to communicate with the IPv4 Internet, even the network is somehow congested and not free.

Security

- SAVA/SAVI
  - RFC5210: A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience
  - RFC6620: FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses
  - RFC7219: SEcure Neighbor Discovery (SEND) Source Address Validation Improvement (SAVI)
  - RFC7219: Source Address Validation Improvement (SAVI) Solution for DHCP
  - RFC8074: Source Address Validation Improvement (SAVI) for Mixed Address Assignment Methods Scenario



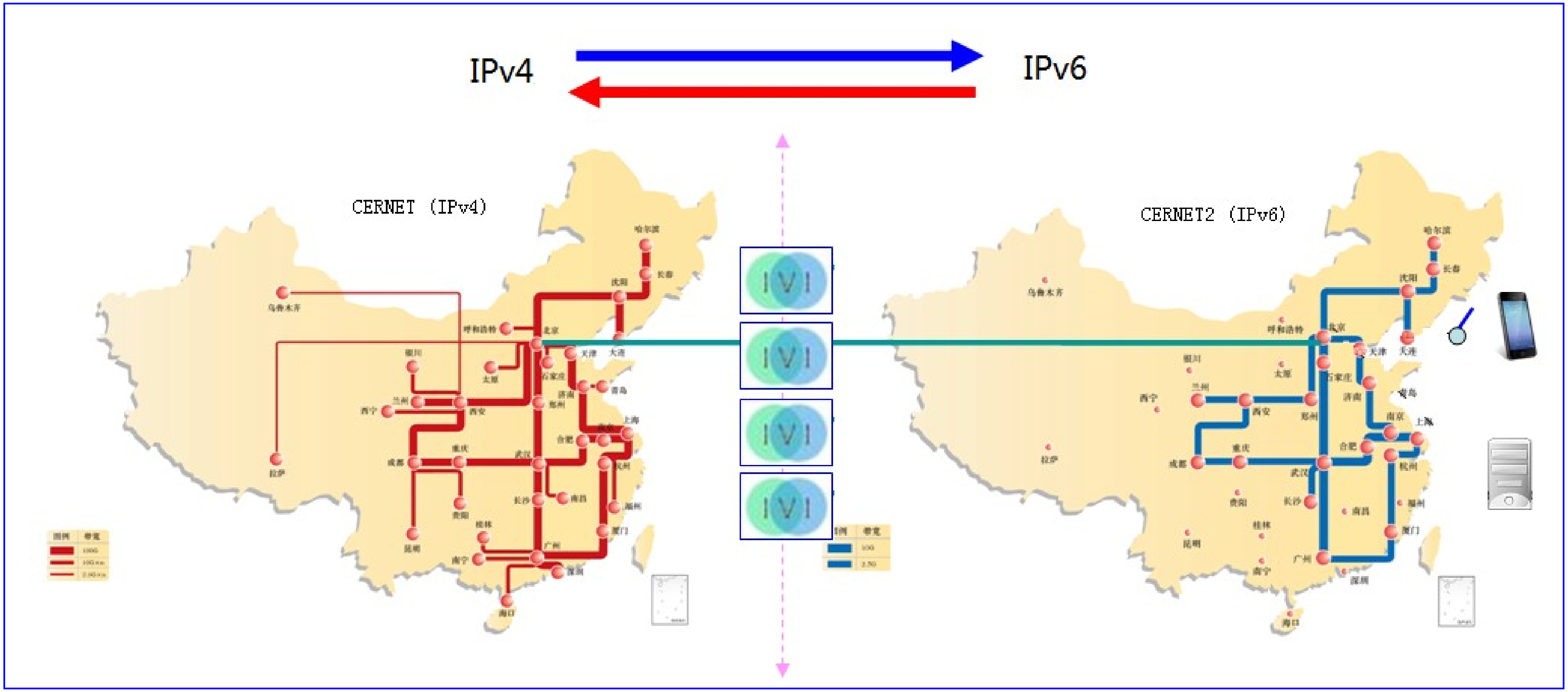
Transition


Dual stack
NFSCNET
Double IVI
IETF Softwire WG
IPv6 over IPv4
CERNET-6Bone
IPv4 over IPv6
IETF softwire WG
MAP-T/MAP-E
IETF Softwire WG
Translation IVI
IETF Behave WG
IPv4
CERNET
IPv6 only
CERNET2
1994
1998
2000
2004
2005
2007
2011
2014

Related RFCs

- Translation
  - RFC6052: IPv6 Addressing of IPv4/IPv6 Translators
  - RFC6145/RFC7915: IP/ICMP Translation Algorithm
  - RFC6146: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers
  - RFC6147: DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers
  - RFC6219: The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition
  - RFC7050: Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis
- 4aaS
  - RFC6333: Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion
  - RFC6877: 464XLAT: Combination of Stateful and Stateless Translation
  - RFC7596: Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture
  - RFC7597: Mapping of Address and Port with Encapsulation (MAP-E)
  - RFC7599: Mapping of Address and Port using Translation (MAP-T)

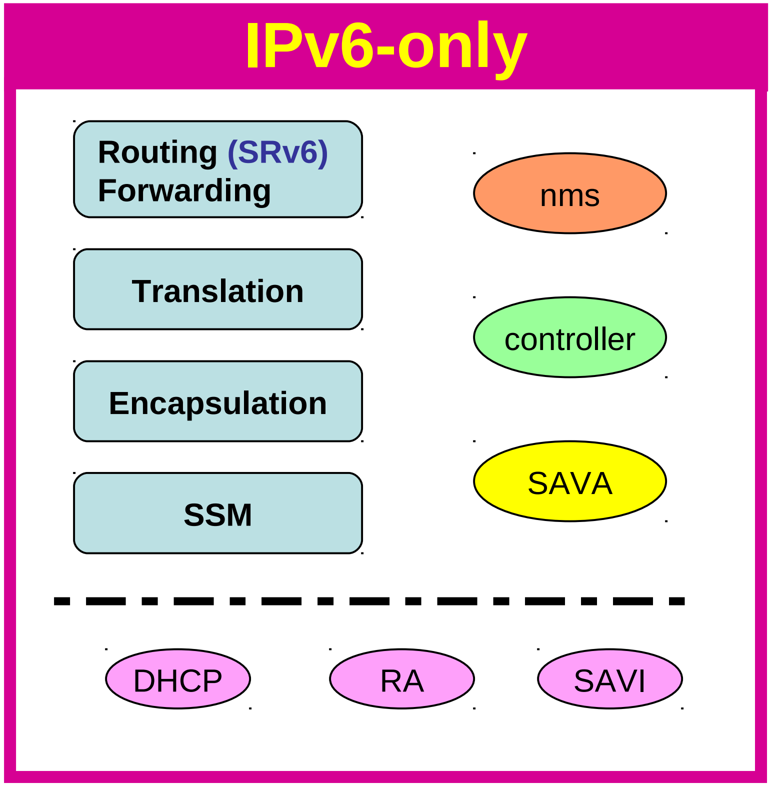
Current practice



- Multiple **stateless translators** are deployed between CERNET (IPv4) and CERNET2 (IPv6), same for the IPv6-only servers, IPv6-only clients and 4aaS
- **IPv6-only servers** are distributed in different universities
- **IPv6-only clients** or **4aaS clients** are distributed in different universities via **2nd translators**

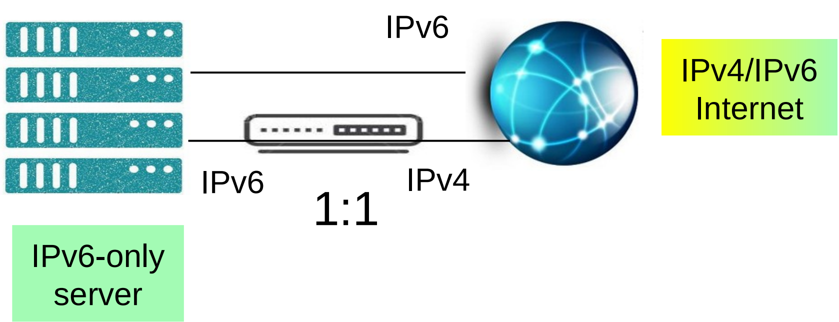
IPv6-only backbone

- Address block
  - 2001:da8::/32
  - Some campuses have their own PI address blocks
- Link speed
  - 2.5G, 10G, 100G
- IGP
  - OSPFv3
- BGP
  - iBGP, eBGP
  - AS23910
- Multicast
  - SSM (source prefix and static join in the edge)



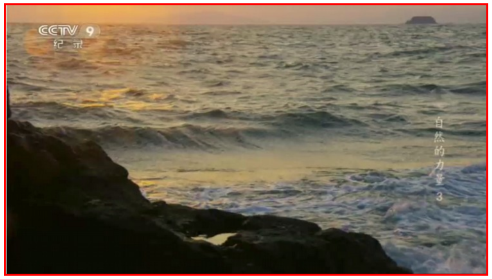
IPv6-only servers

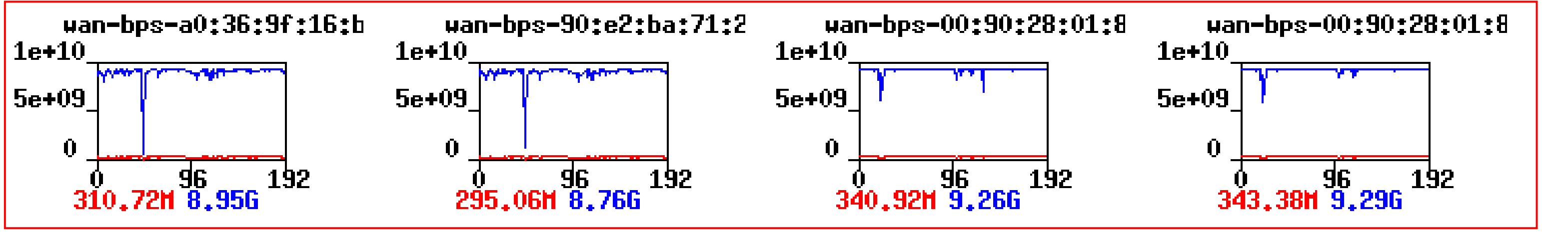
http://ivi.bupt.edu.cn



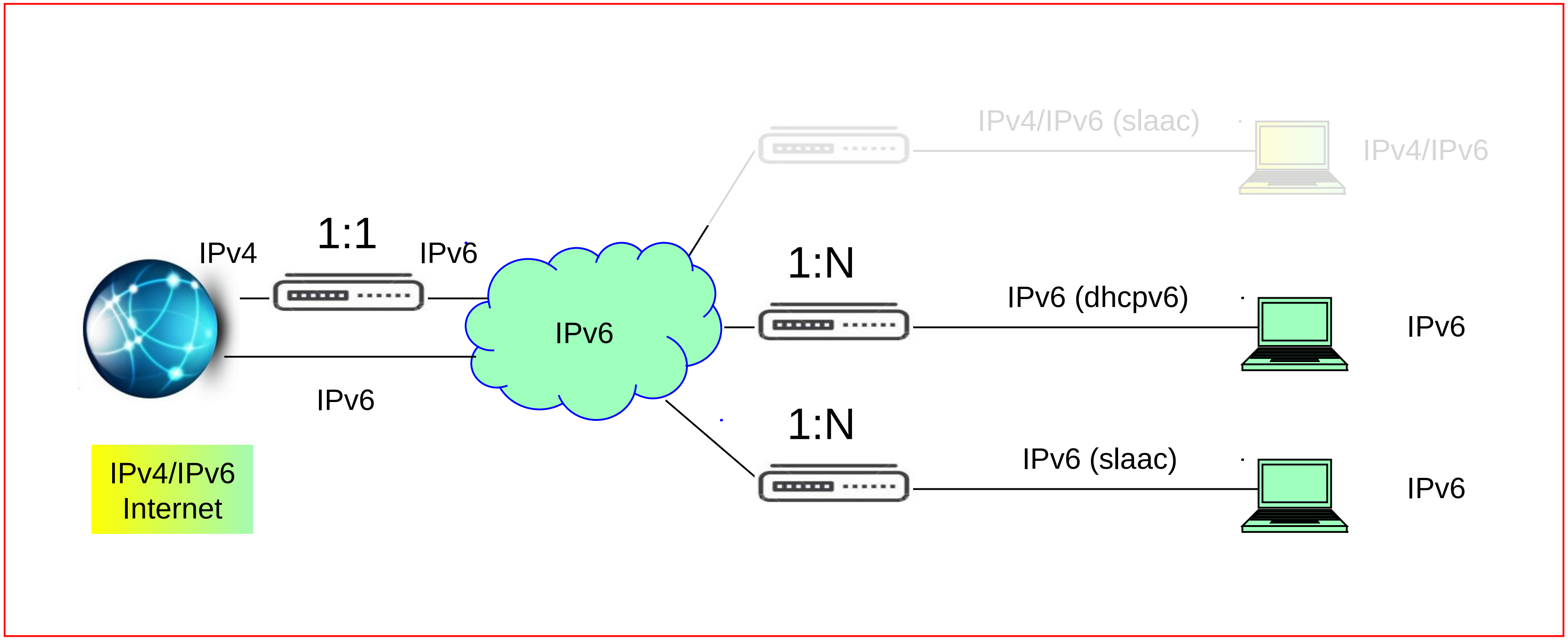
2001:da8:ff3a:c883:600::

58.200.131.6



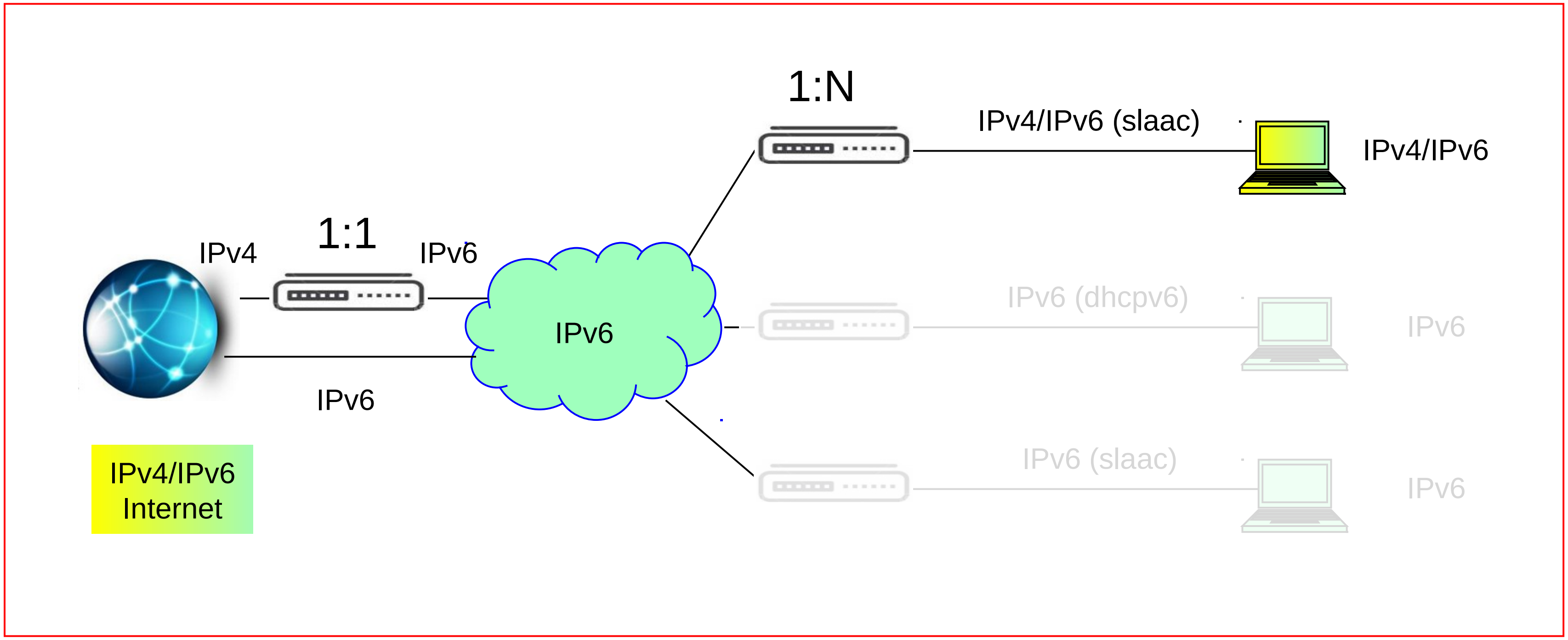


IPv6-only clients



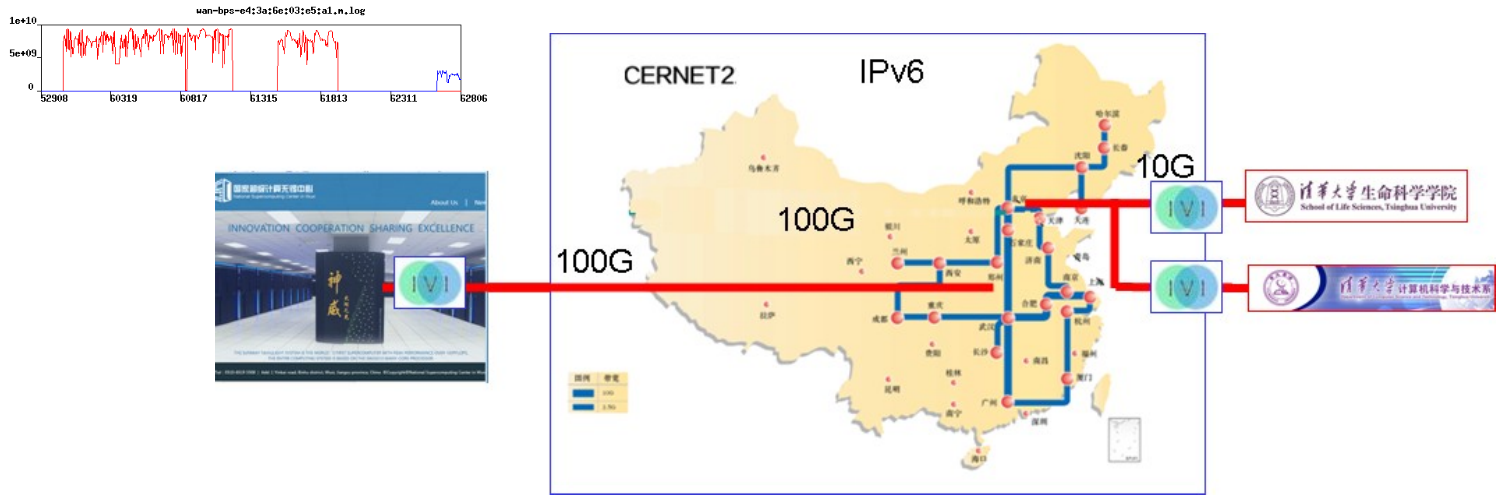
- **Partial function (supports DNS-only)**
  - Any OS which supports dhcpv6 stateful or
  - Any OS which support IPv6-only and RDNSS
- **Full function (supports DNS and IPv4 address literal)**
  - Any OS which supports RFC7050, etc

4aaS (dIVI, MAP-T)



- **Full function (DNS and IPv4 address)**
  - Any OS which supports IPv6

4aaS for high-performance VPN

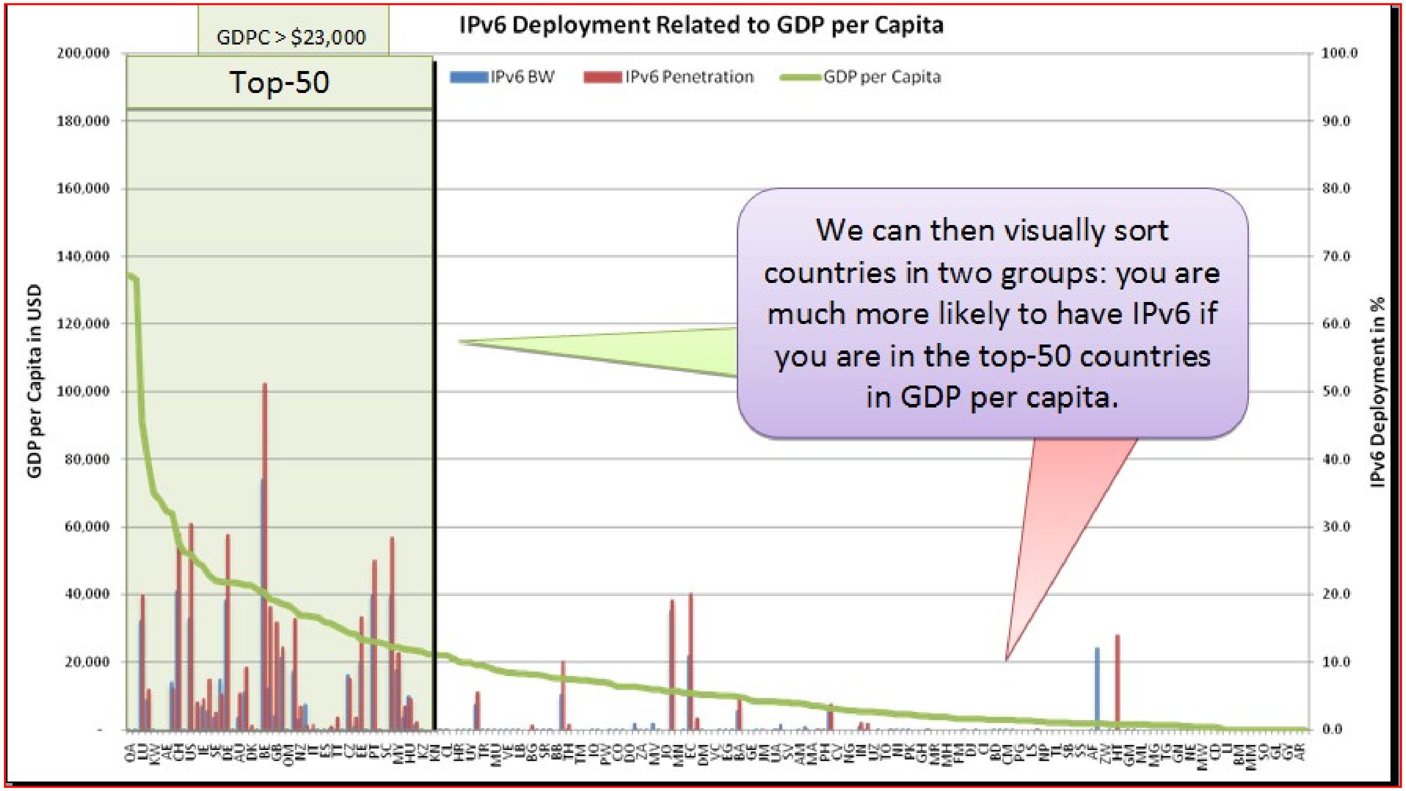


- The network can by fine-tuned using IPv6 prefixes
  - More flexible and controllable than IPv4 BGP
  - More cost effective than MPLS
- The network O&M is very simple
  - End to end address transparency, no en/decapsulation is required
- No need to upgrade the application at this stage
  - The application can still be IPv4
- The different charging model can be applied via difference IPv6 prefix
  - 25Gbyte/user/month → global Internet access
  - Unlimited → limited domain (Wuxi) IPv6 prefix

Discussion

- Encapsulation vs translation (4aaS)
  - One world, one Internet and IPv6-only
  - acl and TE control without decapsulation
- Stateless vs stateful
  - End-to-end address transparency
  - Scalability
- RFC6052 vs other mapping algorithms
  - Easy indentify the addresses in another address family
- Same IPv6 prefix for source/destination vs different IPv6 prefixes
  - Optimal routing
  - Single or multiple administration domain (s)
- DHCPv6 vs SLAAC
  - Support popular OS (Andriod)

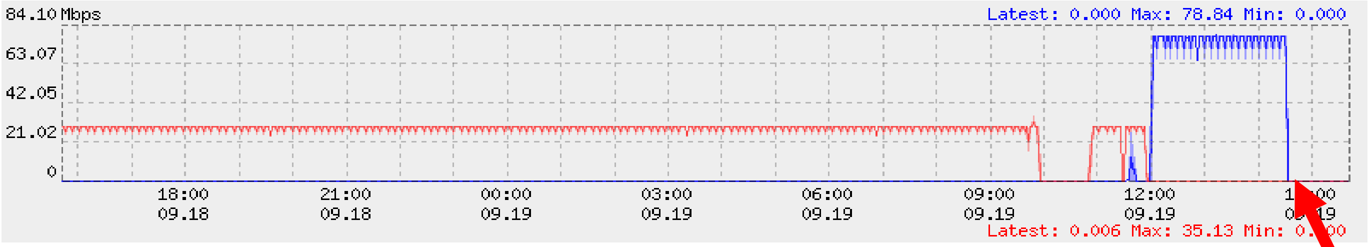
One world, one Internet and IPv6-only



- The above data shows that the IPv4/IPv6 translators MUST deployed somewhere for the IPv6-only network in order to keep ONE Internet in the next decade or decades.

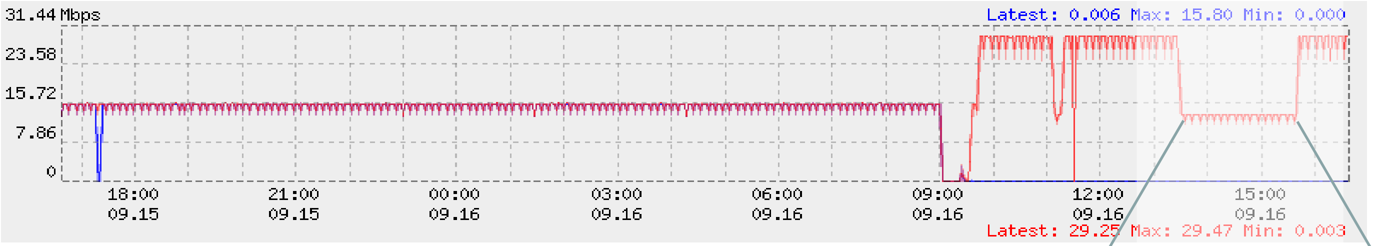
acl and TE control without decapsulation

2001:da8:ff3a:c8fb:300:: -> 202.38.101.26 70M (blue)



IPv6 null route

202.38.101.26 -> 2001:da8:ff3a:cbfb:200:: (w/ rate-limit)



Beginning of rate-limit

End of rate-limit

202.38.101.26 -> 2001:da8:ff3a:cbfb:300:: (w/o rate-limit)

End-to-end address transparency

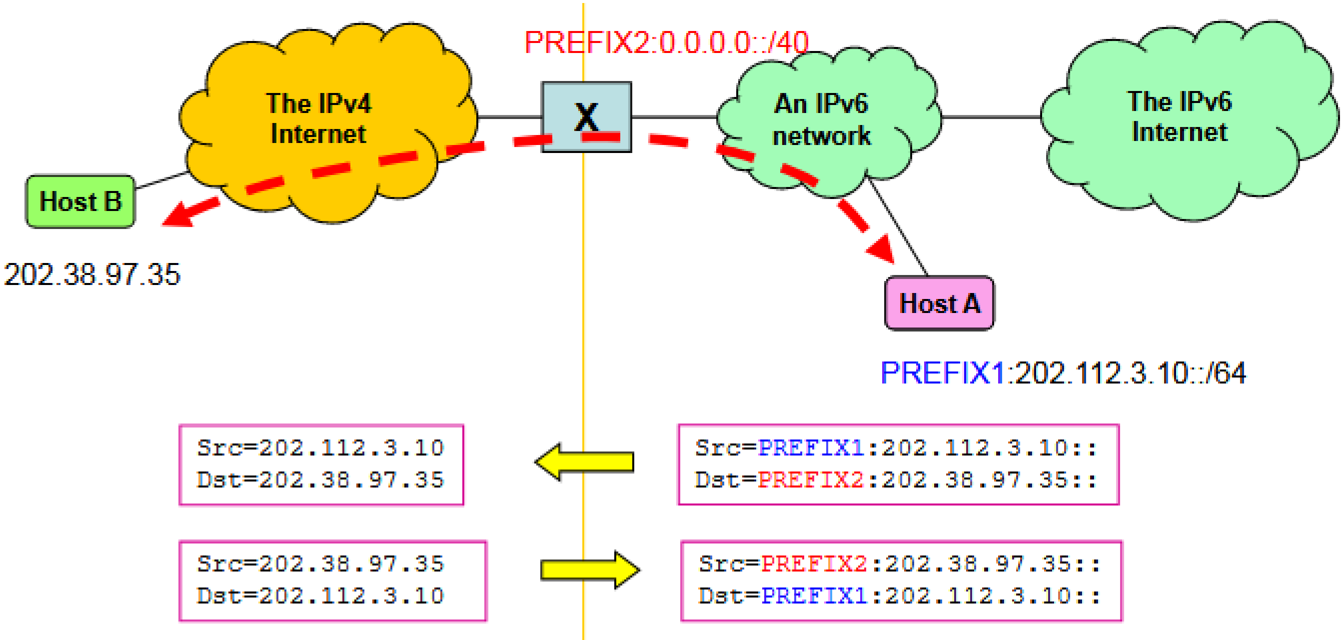
PL	0--------------------32----40----48----56----	64	----72----80----88---96---104---112---120		
32	Prefix / v4 (32)	u	Suffix	zero	
40	Prefix / v4 (24)	u	(8)	Suffix	zero
48	Prefix / v4 (16)	u	v4 (16)	Suffix	zero
56	Prefix / (8)	u	v4 (24)	Suffix	zero
64	Prefix	u	v4 (32)	Suffix	zero

Prefix Suffix

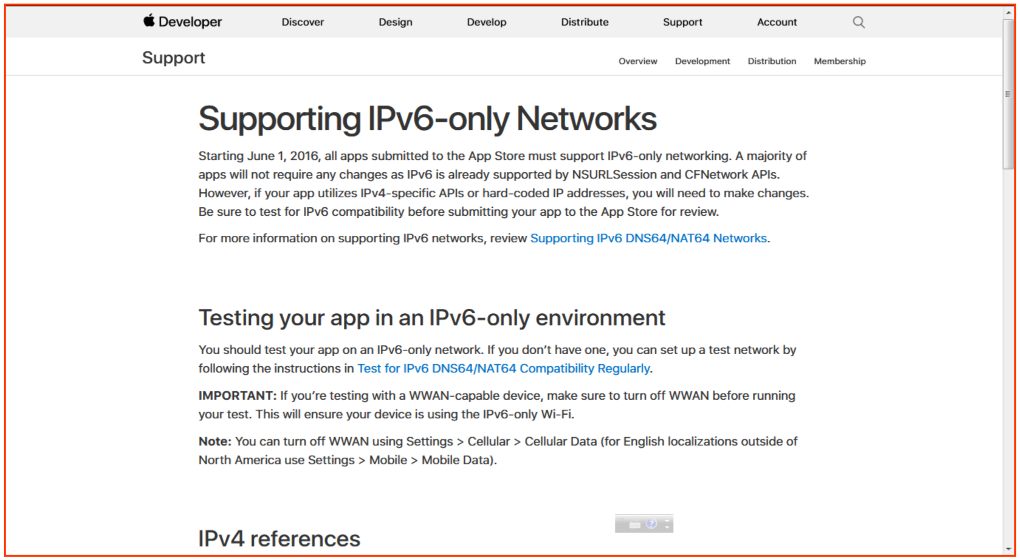
`IVI 220.247.152.0/24 2001:df9:da00::/40`

	0	40	72	
more specific	2001:df9:da00::	dc:f798:xx00	0	IPv4-translatable address
less specific	2001:df9:da00::	x.x.x.x	0	IPv4-converted address

Optimal routing



IPv6-only support, full function



Developer | Discover | Design | Develop | Distribute | Support | Account

Support | Overview | Development | Distribution | Membership

Supporting IPv6-only Networks

Starting June 1, 2016, all apps submitted to the App Store must support IPv6-only networking. A majority of apps will not require any changes as IPv6 is already supported by NSURLSession and CFNetwork APIs. However, if your app utilizes IPv4-specific APIs or hard-coded IP addresses, you will need to make changes. Be sure to test for IPv6 compatibility before submitting your app to the App Store for review.

For more information on supporting IPv6 networks, review Supporting IPv6 DNS64/NAT64 Networks.

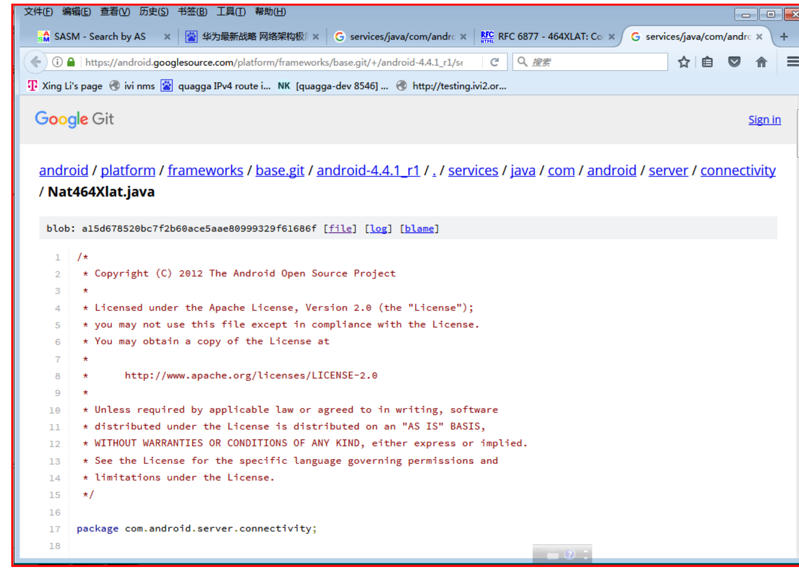
Testing your app in an IPv6-only environment

You should test your app on an IPv6-only network. If you don't have one, you can set up a test network by following the instructions in Test for IPv6 DNS64/NAT64 Compatibility Regularly.

IMPORTANT: If you're testing with a WWAN-capable device, make sure to turn off WWAN before running your test. This will ensure your device is using the IPv6-only Wi-Fi.

Note: You can turn off WWAN using Settings > Cellular > Cellular Data (for English localizations outside of North America use Settings > Mobile > Mobile Data).

IPv4 references



Google Git

android / platform / frameworks / base.git / android-4.4.1_r1 / . / services / java / com / android / server / connectivity / Nat464Xlat.java

blob: a15d678520bc7f2b60ace5aae80999329f61686f [file] [log] [blame]

```
/*
 * Copyright (C) 2012 The Android Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package com.android.server.connectivity;
```

NAT64

- **iOS (9.2+) and MacOS (10.13+)**
  - RFC7050, etc
  - dhcpv6 stateful or
  - slaac

IVI version of NAT64

464xlat

- **Andriod (6.0+), Win10 (1703+), Linux**
  - RFC6877, RFC7050, etc
  - slaac only

Remarks

- IPv6-only (with single translation) if you can
- 4aaS (with double translation or encapsulation) if you should
- Dual stack if you must

4aaS → IPv6-only (with single translation) → IPv6-only

IPv6 transition