IPv6: why, how, where are we

Jun-ichiro itojun Hagino, Ph.D
iijlab/KAME/WIDE/IAB
itojun@iijlab.net

Outline

- WHAT is IPv6?
- WHY do we need to transition to IPv6?

- Protocol technical details

- HOW should we transition to IPv6?
- WHERE ARE WE now in the transition process?
- What is blocking deployment today? How to boost deployment?

WHAT is IPv6?

- Re-engineer the Internet Protocol (IPv4) and expand address space
  - 32bit -> 128bit

- Make recent technologies mandatory and make them available everywhere
  - IPv4: designed in 1970s
  - IPv6: designed in 1990s
  - auto configuration, multicast, security

- Basically a software technology, you do not (should not) need to buy a new router

- We’ll cover technical details later

WHY do we need to transition to IPv6? (1)

- Every communication technologies are becoming IP-based

- IP is the cheapest and easiest communication technology for computers/devices/censors
  - used to be: setup a dedicated copper phone line for propane gas telemetry
  - from now on: connect propane gas gauge to home IP network

  - used to be: design ultra-expensive telephone switch for cellphones
  - from now on: make cellphones wireless VoIP terminals, and use IP routers (cheaper!) to carry data

  - used to be: home appliances share a single phone line (complicated, errors)
  - from now on: home appliances use IP, IP naturally multiplexes the data

WHY do we need to transition to IPv6? (2)

- Why 32bit is not enough?
  - Theoretical upper limit for IPv4-connected computers: 430 million
  - Actually, we cannot assign addresses that efficiently, so upper limit would be a couple of millions

  - World wide population: 600 million
  - "Computing devices" per capita: expect 100 or more per person

  - Due to IPv4 address space shortage, many new technologies are unable to use IP
  - example: would like to deploy VoIP wireless phones, IPv4 address is not that plenty

- With 128bit... theoretical limit: 3.4x10^38

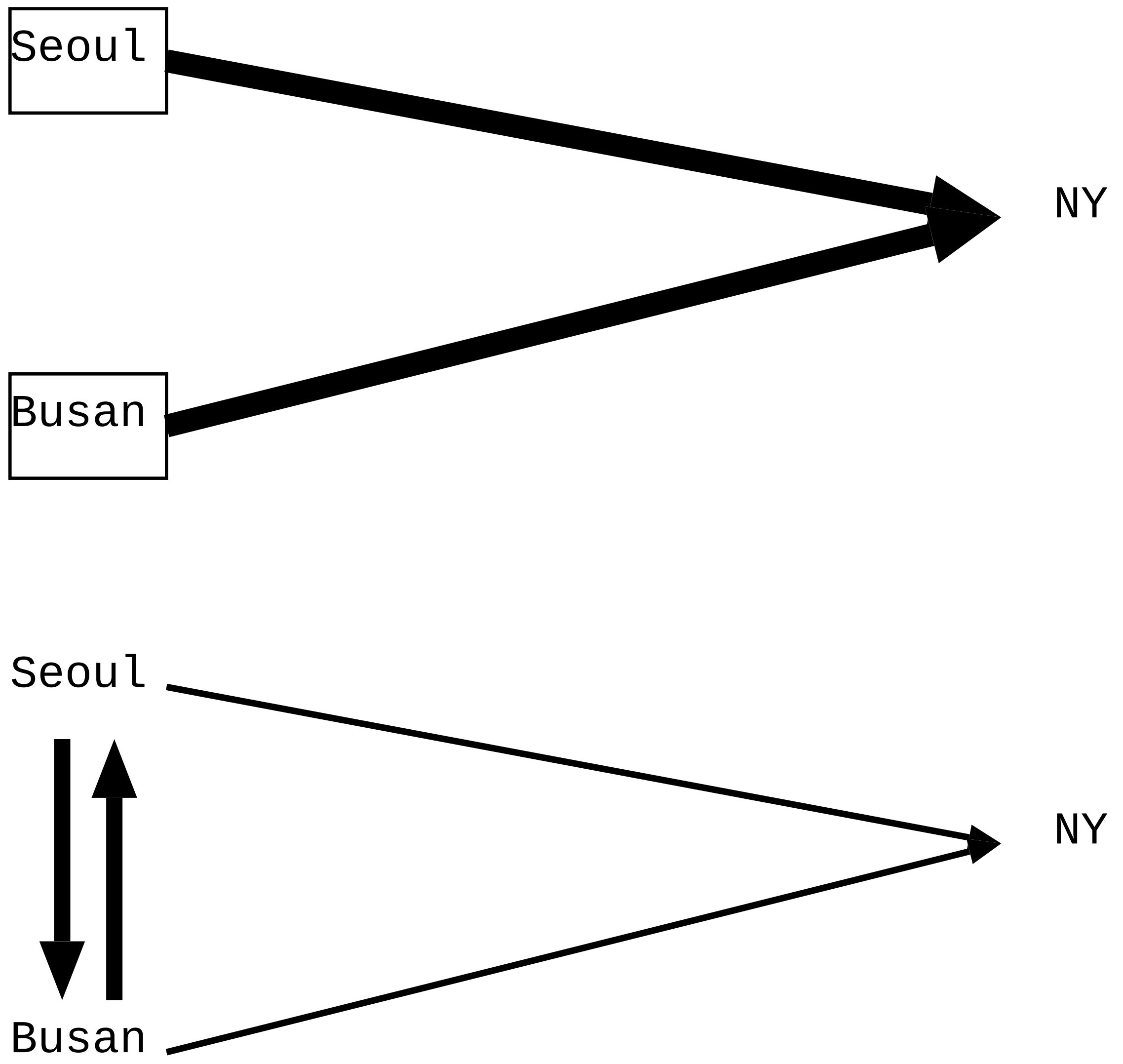
WHY do we need to transition to IPv6? (3)

- NAT broke bi-directional communication property of IP
  - Client (home/office) to server (google/yahoo), we can communicate
  - Opposite is impossible with NAT
  - Traffic goes to central servers, and we need giant server to handle the traffic
    - Seoul - Busan traffic goes through New York, because Seoul - Busan direct communication is not possible

- IP is bi-directional by nature - we need to get the property back by destroying NATs
  - With IPv6, addresses are plenty and you never need NAT
  - (note: NAT does NOT improve security)

  - Clients communicate with each other for high-volume traffic (like voice/video)
    - Seoul - Busan direct communication
  - Server would introduce peers, handle small amount of control traffic, and other data (like game status)
    - Server can be built cheap, traffic to server will be reduced drastically

WHY do we need to transition to IPv6? (4)



IPv6 and IPv4 - what is the difference?

- IPv6 is "IP with bigger address space" - basically, it is the same as IPv4
  - "Bigger address space" makes a huge difference
- Base spec
  - 128bit address, simplified base header, extensible header structure
  - Will be able to adapt to future application/technologies
- Routing - OSPFv3, RIPng, BGP4+
- QoS - diffserv, RSVP
  - (discussed separately from IPv6 itself. with IPv6 it is easier to perform diffserv/RSVP than IPv4)
- Mobility - mobile-ip6
- Security - IPsec (discussed separately from IPv6 itself)
  - To state "IPv6 complete conformance" you need to implement IPsec
- Autoconfiguration
  - stateless autoconf, DHCPv6
- Multicast - PIM, MLD (IGMP for IPv6)
- application - HTTP, FTP, whatever you have been doing on IPv4
  - IPv6 deployment has to reach certain critical-mass before "killer application" appears
  - Application that utilizes huge IPv6 address space?

IPv6 features

- Can coexist with IPv4 network
  - There’s no flag day, you can use existing devices
  - "Dual stack" - devices that can use IPv4/IPv6 at the same time
- Only needs software update in most cases
  - You do not need to buy new routers
- Vast amount of address space
  - 2^32 or 2^13 route entries in ISP
  - 2^64 nodes on a subnet
  - 2^16 subnets to a site
- Aggregation friendly
  - Core routes are limited to 8192 (2^13) by addressing architecture
  - No portable address available
- Designed for less operation cost in routers
  - Mandates PMTUD, no fragmentation in intermediate routers
- Autoconfiguration works beautiful
  - No server necessary, no wacky state management
  - Autoconfig available everywhere: IPv4 autoconfig required DHCP server

IPv6 INCORRECT rumors

- Fixes all Internet problems
  - How? :-P
- Noone uses it
  - There are so many users in Japan (ISP IPv6 service is available as well)
  - Asia and Europe are more active, US seems to be more optimistic about IPv4 address alloc
- QoS problem is solved
  - Fields are reserved for QoS
  - QoS technology is still in its infancy, and needs more time to be stabilized
- Security problem is solved
  - IPsec is mandatory
  - NAT will be gone, so IPsec can be used everywhere
  - However, IPsec itself has very difficult issue
- IPv6 is not necessary since we have NAT
  - BIG NO!
- Need hardware upgrade
  - In most cases, no. Software upgrade is enough

IPv4 header

- Many fields
  - TOS: left unused in most cases
  - IP header checksum: high cost to maintain, cost/benefit?
  - IPv4 options: not really used
- Usually 20 bytes (8 bytes are for src/dst)

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                               .               .
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 header

- Much simpler than IPv4 header
  - No IPv6 header checksum: saves routers' labor
  - Most of extensions goes to extension header chain
- Always 40 bytes (32 bytes are for src/dst)
  - Fixed size = easier to make hardware forwarding engine

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

How to write IPv6 addresses

- 16 groups of 4-digit hexadecial
  - **`3ffe:0507:0000:0001:0200:86ff:fe05:80fa`**
  - **`0000:0000:0000:0000:0000:0000:0000:0001`**
- You can omit starting "0" in each of the groups
  - **`3ffe:507:0:1:200:86ff:fe05:80fa`**
  - **`0:0:0:0:0:0:0:1`**
- In only one place, you can use "::" to denote continuous "0"
  - **`3ffe:507::1:200:86ff:fe05:80fa`**
  - **`::1`**

- Usually you do not write numeric IPv6 addresses, use DNS names

Upper layer protocols

- Upper layer
  - ICMPv6: ICMP for IPv6
    - Very similar
  - TCP, UDP and other L4 protocols: same as IPv4
    - Only pseudo-header checksum is different
  - Protocols on TCP/UDP and other application protocols: same
    - HTTP, FTP will work just like it was on IPv4
    - If we embed IP address into upper-layer protocol message, we need to modify it (FTP)

Modifications to upper layer protocols

- FTP
  - PORT, PASV passes IPv4 address, and does not support other types
  - EPSV, EPRT command - RFC2428
- HTTP
  - host:port form is ambiguous if host is an IPv6 address
  - [host]:port notation - RFC2732

IPv4 packet example

- TCP packet toward port 22, in IPv4

```
16:09:25.113204 202.232.15.102.63472 > 202.232.15.98.22: P 20:40(20) ack 21
win 17520 <nop,nop,timestamp 245605 38398> (ttl 64, id 13795)
                         4500 0048 35e3 0000 4006 9034 cae8 0f66
                         cae8 0f62 f7f0 0016 8520 7612 1672 89e4
                         8018 4470 4060 0000 0101 080a 0003 bf65
                         0000 95fe 0000 000a 0eef be30 a72a 41f1
                         7be1 2f9b 3ccd b5b0
16:09:25.113364 202.232.15.102.63472 > 202.232.15.98.22: P 20:40(20) ack 21
win 17520 <nop,nop,timestamp 245605 38398> (ttl 64, id 13795)
                         4500 0048 35e3 0000 4006 9034 cae8 0f66
                         cae8 0f62 f7f0 0016 8520 7612 1672 89e4
                         8018 4470 4060 0000 0101 080a 0003 bf65
                         0000 95fe 0000 000a 0eef be30 a72a 41f1
                         7be1 2f9b 3ccd b5b0
```

IPv6 packet example

- TCP packet toward port 22, in IPv6

```
15:43:17.093542 3ffe:507:0:1:200:86ff:fe05:80fa.49226 >
3ffe:507:0:1:260:97ff:fe07:69ea.22: . ack 1093 win 17519 <nop,nop,timestamp
242469 35276> [flowlabel 0x62073] (len 32, hlim 64)
                         6006 2073 0020 0640 3ffe 0507 0000 0001
                         0200 86ff fe05 80fa 3ffe 0507 0000 0001
                         0260 97ff fe07 69ea c04a 0016 3ee4 92b2
                         29d4 2455 8010 446f 80da 0000 0101 080a
                         0003 b325 0000 89cc
15:43:17.093648 3ffe:507:0:1:200:86ff:fe05:80fa.49226 >
3ffe:507:0:1:260:97ff:fe07:69ea.22: . ack 1093 win 17519 <nop,nop,timestamp
242469 35276> [flowlabel 0x62073] (len 32, hlim 64)
                         6006 2073 0020 0640 3ffe 0507 0000 0001
                         0200 86ff fe05 80fa 3ffe 0507 0000 0001
                         0260 97ff fe07 69ea c04a 0016 3ee4 92b2
                         29d4 2455 8010 446f 80da 0000 0101 080a
                         0003 b325 0000 89cc
```

IPv6 extension headers

- Hop-by-Hop Options
  - Routers will look at it
- Routing
  - Source routes
- Fragment
  - Fragmentation, ONLY at originating node
- Destination Options
  - Pass info to final destination
- IPsec (AH, ESP)

- Header chain
  - Header parsing code becomes somewhat different from IPv4

```
IPv6(nxt=TCP) TCP payload
IPv6(nxt=routing) Routing(nxt=TCP) TCP payload
IPv6(nxt=fragment) Fragment(nxt=TCP) TCP payload
```

IPv6 extension headers example (1)

- IPv6 echo request/reply, without AH

```
16:14:31.750102 ::1 > ::1: icmp6: echo request (len 16, hlim 64)
                         6000 0000 0010 3a40 0000 0000 0000 0000
                         0000 0000 0000 0001 0000 0000 0000 0000
                         0000 0000 0000 0001 8000 7f1c d802 1000
                         57ea e137 d471 0b00
16:14:31.750155 ::1 > ::1: icmp6: echo reply (len 16, hlim 64)
                         6000 0000 0010 3a40 0000 0000 0000 0000
                         0000 0000 0000 0001 0000 0000 0000 0000
                         0000 0000 0000 0001 8100 7e1c d802 1000
                         57ea e137 d471 0b00
```

IPv6 extension headers example (2)

- IPv6 echo request/reply, with AH

```
16:14:57.060181 ::1 > ::1: AH(spi=9999,sumlen=16,seq=0x7): icmp6: echo
request (len 40, hlim 64)
                         6000 0000 0028 3340 0000 0000 0000 0000
                         0000 0000 0000 0001 0000 0000 0000 0000
                         0000 0000 0000 0001 3a04 0000 0000 270f
                         0000 0007 5a3c 2d0d efbf 3370 d029 1ed5
                         8000 c3a3 dd02 0300 71ea e137 88ea 0000
16:14:57.060299 ::1 > ::1: icmp6: echo reply (len 16, hlim 64)
                         6000 0000 0010 3a40 0000 0000 0000 0000
                         0000 0000 0000 0001 0000 0000 0000 0000
                         0000 0000 0000 0001 8100 c2a3 dd02 0300
                         71ea e137 88ea 0000
```

Lower layer

- Ethernet protocol type: different from IPv4
  - IPv4: 0x0800
  - IPv6: 0x86dd

- Hardware address resolution
  - IPv4: medium dependent, ARP for ethernet
    - Simple protocol
  - IPv6: medium independent, NDP on ICMPv6
    - Manages more state than ARP, more complex
    - Implements DAD, Duplicated Address Detection

- Multicast address mappings
  - Take lowermost 4 bytes from IPv6 address, and embed to MAC addr
  - 33:33:xx:yy:zz:uu
  - No use of broadcast
    - Link-layer broadcast will wake everyone up on the medium

Ethernet encapsulation example (1)

- IPv4 broadcast icmp echo request and icmp echo reply

```
15:46:30.651062 0:0:86:5:80:fa ff:ff:ff:ff:ff:ff 0800 98: 202.232.15.102 >
202.232.15.103: icmp: echo request (ttl 255, id 9961)
                         4500 0054 26e9 0000 ff01 e021 cae8 0f66
                         cae8 0f67 0800 b14d 02a5 0000 c6e3 e137
                         a7ee 0900 0809 0a0b 0c0d 0e0f 1011 1213
                         1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
                         2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
                         3435 3637
15:46:30.651377 0:0:f8:5:69:42 0:0:86:5:80:fa 0800 98: 202.232.15.100 >
202.232.15.102: icmp: echo reply (ttl 255, id 2369)
                         4500 0054 0941 0000 ff01 fdcc cae8 0f64
                         cae8 0f66 0000 b94d 02a5 0000 c6e3 e137
                         a7ee 0900 0809 0a0b 0c0d 0e0f 1011 1213
                         1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
                         2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
                         3435 3637
```

Ethernet encapsulation example (2)

- IPv6 multicast icmp6 echo request and icmp6 echo reply

```
15:44:52.080190 0:0:86:5:80:fa 33:33:0:0:0:1 86dd 70:
fe80::200:86ff:fe05:80fa > ff02::1: icmp6: echo request (len 16, hlim 64)
                         6000 0000 0010 3a40 fe80 0000 0000 0000
                         0200 86ff fe05 80fa ff02 0000 0000 0000
                         0000 0000 0000 0001 8000 ebd9 9c02 0100
                         64e3 e137 aa38 0100
15:44:52.080480 0:8:c7:a4:9c:d 0:0:86:5:80:fa 86dd 70:
fe80::208:c7ff:fea4:9c0d > fe80::200:86ff:fe05:80fa: icmp6: echo reply (len
16, hlim 64)
                         6000 0000 0010 3a40 fe80 0000 0000 0000
                         0208 c7ff fea4 9c0d fe80 0000 0000 0000
                         0200 86ff fe05 80fa 8100 86a2 9c02 0100
                         64e3 e137 aa38 0100
```

IPv6 addressing architecture (1)

- Global address: 2000::/3 (RFC3513/3587)
  - Hierarchy design to encourage route aggregation
  - Interface ID to help autoconfiguration (later)
  - Site boundary is fixed to /48 to encourage ISP change and site-local routing
  - A site can always have 2^16 subnets
- Current RIR practice
  - Large ISP gets /32 (up to /19) from RIR (like 2001:200::/32)
  - Smaller ISP gets /40 suballocation from upstream ISP
  - End site (universities, companies) gets /48

```
| 3|        45          |   16   |          64 bits                 |
+--+--------------------+--------+--------------------------------+
|FP|                    | subnet |         Interface ID           |
|  |                    |  number|                                |
+--+--------------------+--------+--------------------------------+
<--><------------------><------->
      managed by RIR   site subnet
```

IPv6 addressing architecture (2)

- Link-local address: fe80::/10
  - fe80 + Interface ID
  - used for bootstrap (autoconfig), routing, and neighbor discovery (ARP)

```
| 10 |          54 bits          |           64 bits                |
+----+-------------------------+---------------------------------+
|FE8 |          0              |          Interface ID           |
|    |                         |                                 |
+----+-------------------------+---------------------------------+
```

- Multicast address: ffxx::/16
  - "xx" indicates multicast address types, scope (control propagation)

```
| 16 |          48 bits          |           64 bits                |
+----+-------------------------+---------------------------------+
|FFXX|          0              |          Interface ID           |
|    |                         |                                 |
+----+-------------------------+---------------------------------+
```

IPv6 autoconfiguration

- Stateless address autoconfiguration
  - No resource management thanks to address architecture
  - Routers advertise info about the subnet
  - Hosts receive the information and configures itself

- DHCPv6
  - Stateful, needs server

- Stateless address autoconfiguration is much easier, and will be available everywhere

IPv6 autoconfig details

- End host transmits router solicitation
  - "I want to know the configuration for the subnet"
- Router transmits router advertisement
  - "Okay, here’s information on subnet"
- End host configures itself

```
15:40:16.590444 fe80::200:86ff:fe05:80fa > ff02::2: icmp6: router
solicitation (src lladdr: 0:0:86:5:80:fa) (len 16, hlim 255)
                         6000 0000 0010 3aff fe80 0000 0000 0000
                         0200 86ff fe05 80fa ff02 0000 0000 0000
                         0000 0000 0000 0002 8500 6d2e 0000 0000
                         0101 0000 8605 80fa
15:40:16.819647 fe80::260:97ff:fe07:69ea > ff02::1: icmp6: router
advertisement (chlim=64, router_ltime=1800, reachable_time=30000,
retrans_time=1000) (src lladdr: 0:60:97:7:69:ea) (mtu: mtu=1500) (prefix
info: LA valid_ltime=3600000, preffered_ltime=3600000,
prefix=3ffe:507:0:1::/64) (len 64, hlim 255)
                         6000 0000 0040 3aff fe80 0000 0000 0000
                         0260 97ff fe07 69ea ff02 0000 0000 0000
                         0000 0000 0000 0001 8600 4625 4000 0708
                         0000 7530 0000 03e8 0101 0060 9707 69ea
                         0501 0000 0000 05dc 0304 40c0 0036 ee80
                         0036 ee80 0000 0000 3ffe 0507 0000 0001
                         0000 0000 0000 0000
```

IPv6 and DNS

- IPv6 in DNS payload
  - AAAA record: similar to A record
  - PTR record: use "ip6.arpa." tree
    - "ip6.int" tree has been used, transition ongoing

```
turmeric.itojun.org. IN AAAA 3ffe:507:0:1:200:86ff:fe05:80fa
a.f.0.8.5.0.e.f.f.f.6.8.0.0.2.0.1.0.0.0.0.0.0.0.7.0.5.0.e.f.f.3.ip6.arpa.
         IN PTR turmeric.itojun.org.
```

- IPv6 in DNS transport
  - old BIND8 supports queries over IPv4 TCP/UDP only
  - recent BIND8 do suppport queries over IPv6 TCP/UDP
  - BIND9 supports queries over IPv6 TCP/UDP as well

Operating IPv6 network (1)

- Very similar to IPv4 network operation
  - Prepare IPv6-ready software
    - Router, Host
  - Upstream provider assigns IPv6 address space
  - Setup IPv6 addresses to routers
  - Hosts will be autoconfigured by info from routers

- Would like to use IPv4 resources as well... (like google)
  - dual stack (IPv4/v6 dual-capable router/host/network)
  - To communicate with IPv4 peer, use IPv4
  - To communicate with IPv6 peer, use IPv6

- My home and my university are IPv6 ready, but there’s IPv4-only network in the middle
  - Tunnelling
  - Encapsulate IPv6 packet into IPv4 packet and deliver it to peer

- IPv6-only network is tough, avoid it at all costs

Operating IPv6 network (2)

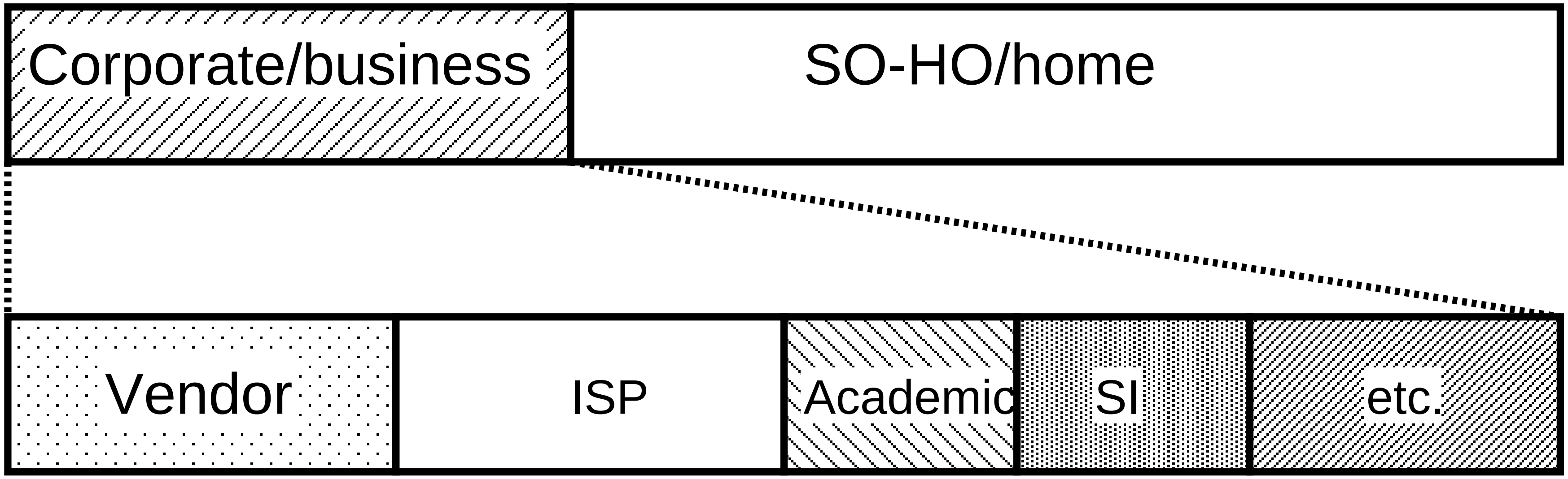
- IPv6 operation in ISP
  - Must not affect IPv4 SLA
  - Provide smooth transition plans/services to customers
- IPv6 backbone and IPv4 backbone
  - Mainain them separately (separate router, separate fiber)
    - More routers, more $$$ to fiber provider, no sideeffect to IPv4 operation
  - dual stack operation (same router, same fiber)
    - No additional routers/fiber bills, but IPv6 router behavior affects IPv4 operation
  - Tradeoff of many things: stability of vendor routers, ease of administration, SLA requirement, ...
- Connection to customers
  - DSL/fiber aggregation service by L2 provider
    - We do not own/manage/maintain routers that talks directly to customer routers
    - We have to poke aggregation provider to do IPv6
    - In the meantime, tunnelling to customer is possible and nice alternative
  - Leased lines
    - Upgrade routers on both ends to IPv6-capable and operate in dual stack
    - If you/customer cannot upgrade router, put another router and perform tunnelling
    - ATM virtual circuit, ethernet VLAN is sometimes useful

Deployment status (1)

- ARIN/RIPE/APNIC are getting massive amount of IPv6 address allocation requests
  - Maybe they are just "placeholders", but anyways it is a good sign
- Experimental/educational networks like Internet2, WIDE are using IPv6 very actively
- Many ISPs provide IPv6 connectivity and additional services
- Great number of free softwares are IPv6 ready
- Commercial softwares are incresingly becoming IPv6 ready
- IPv6 support is already there for desktop/server OSes
- Number of commercial devices with IPv6 support is increasing (router, firewall, network monitoring)

Deployment status (2)

- IIJ IPv6 customer profile
  - There are many personal users
  - Most of commercial customers are engineering/development
  - Link type to customer: majority is tunnel
    - NTT east/west terminates PPPoE session from customer
    - We have no control over those routers, so we cannot make them dual-stack



Things blocking deployment (1)

- IPv6 is a "communication protocol", therefore both end of communication has to support it
  - One-sided efforts (like user community upgrading to IPv6-ready OS) do not work
  - We need to convince the other side as well (like yahoo)
  - Online gaming is a good target?
    - Game software and servers are maintained by the same company
    - Game software CD/DVD includes OS (upgrade is easy)
- IPv6 over IPv4 tunnel that are not well-maintained/designed
  - IPv6-wise it is 1 hop, but the actual IPv4 packet could be travelling overseas (like trans-pacific)
    - IPv6 performance/throughput looks worse than IPv4
  - Try to avoid the use of tunnels
  - If a tunnel is needed, use the same topology as IPv4 for tunnels
    - Usually, it is a mistake to configure tunnels that runs over multiple IPv4 ASes
- Address space allocated but not maintained
  - Some people get some IPv6 address space, play with it, then leave it there
  - (In the case of company, the guy could have resigned already)
  - Source of bogus/non-optimal routing
  - If you operate IPv6, operate seriously

Things blocking deployment (2)

- Devices that do not implement protocols correctly
  - When AAAA record is queried, and there's A record but no AAAA record - DNS server should respond:
    - 0 AAAA record is there
  - not:
    - i don't know such domain name
    - invalid query
  - hunt the vendor down and make them fix it
  - there are tons of products that does not work properly (i.e. seemingly work for IPv4 but does not implement other cases right)
    - web load balancers are the primary suspect
- Lame delegation in IPv6 DNS reverse lookup tree
  - Servers use reverse tree for logging
  - Lame delegation will make reverse lookup very slow
  - Solution: fix it!
- onlink assumption (RFC2461)
  - "if there's no default router, consider all destinations on-link"
  - the rule prevents fallback to IPv4 (or other IPv6 addrs)
  - the rule has been removed in RFC2461bis

Summary

- WHAT is IPv6?
- WHY do we need to transition to IPv6?

- Protocol technical details

- HOW should we transition to IPv6?
- WHERE ARE WE now in the transition process?
- What is blocking deployment today? How to boost deployment?