

# Experiences from IPv6-Only Networks with Transition Technologies in the WIDE Camp Spring 2012

H. Hazeyama \*, R. Hiromi, T. Ishihara,  
O. Nakamura

[draft-hazeyama-widencamp-ipv6-only-  
experience-01](#)

# Aim and goal of IPv6 only experiment in the WIDE camp

- evaluate lots of devices, OSs, applications on the IPv6 enabled networks for IPv6 deployment
- operate IPv6 with transition (translation and encapsulation) technologies on experimental networks and get TIPS on operation
- determine clients' IPv6 readiness on experimental networks
- share the result and consider what we can / should do against remained IPv6 unreadiness

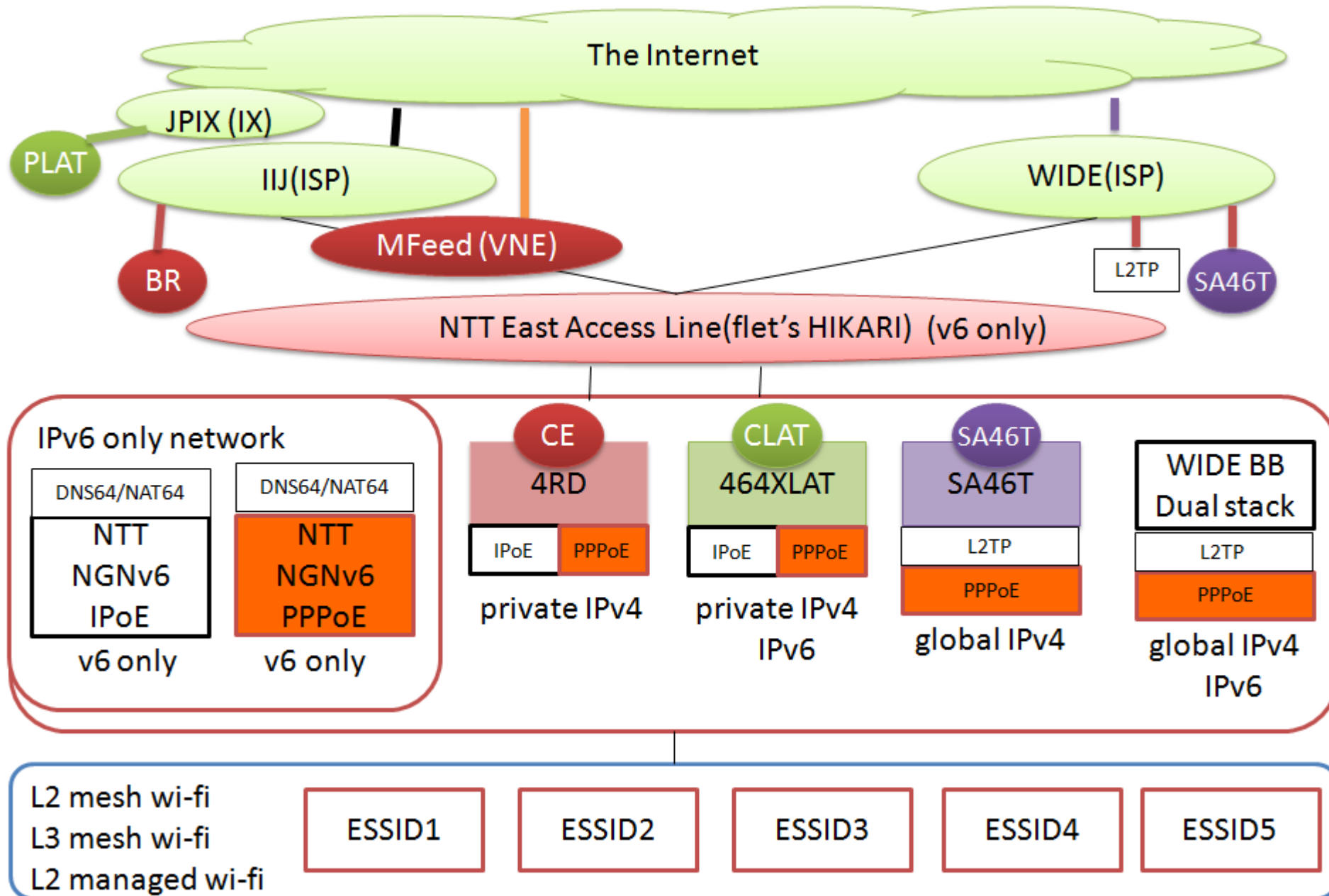
# 1<sup>st</sup> experiment

- The summary was reported in v6ops BoF of IETF 82 TAIPEI
  - [draft-hazeyama-widencamp-ipv6-only-experience-00](#)
  - Sep. 5<sup>th</sup> – Sep. 8<sup>th</sup>, 2011 at Matsushiro Royal Hotel, Nagano, Japan
  - 153 participants joined
  - Main test: **live in an IPv6 only network with DNS64/NAT64**
    - The IPv6 only network was constructed by WIDE BB address block through IPv6 L2TP on a commercial IPv6 service
  - Additional test: **availability of SA46T and 4RD**
    - Both SA46T and 4RD-BR were located in WIDE BB
  - We reported participants' impression and troubles we met

# 2<sup>nd</sup> experiment

- Today's presentation
  - [draft-hazeyama-widcamp-ipv6-only-experience-01](#)
    - This draft has several undocumented parts and typos
    - We will update this draft after IETF 83
- Mar. 5<sup>th</sup> – Mar. 8<sup>th</sup>, 2011 at Matsushiro Royal Hotel, Nagano, Japan
  - The same place of the 1<sup>st</sup> experiment
- 171 participants joined
- Main test: **live in two commercial IPv6 services with DNS64/NAT64, SA46T, 4RD and 464XLAT**
- Other tests: LISP+VXLAN, OSLR based L3 mesh using WIDE BB address blocks
  - Similar to the 1<sup>st</sup> experiment except for routing mechanisms
  - out of scope in this draft

# Main IPv4/IPv6 Network of 2<sup>nd</sup> exp.



# User Subnets and DNS settings

Label	Subnet IP Prefix	DNS
Native IPoE	2409:150:8000:10::/64	2001:200:0:ff60::58 (DNS LB)
Native PPPoE	2001:240:2002:6d10::/64	2001:200:0:ff60::58 (DNS LB)
4RD/IPoE	192.168.12.0/24	210.130.1.1 (via proxy)
4RD/PPPoE	192.168.22.0/24	210.130.1.1 (via proxy)
464XLAT/IPoE	192.168.13.0/24 2409:150:8000:30::/64	2404:1a8:7f01:b::3 2001:4860:4860::8888
464XLAT/PPPoE	192.168.23.0/24 2001:240:2002:6d30::/64	2001:240::13 2001:4860:4860::8888
SA46T-FA, SA46T-FK	203.178.156.0/25	203.178.159.58 (DNS LB)

# participants' choice of networks

- **We changed provided networks through Wi-Fi in each lunch and dinner time**
  - 4 or 5 ESSIDs with mixing v4 networks and v6 only networks
- **Participants changed networks in order to get IPv4 or to avoid network congestion**
  - Network congestion mainly occurred by CPU overload on packet fragment handling
- **Around 20 - 60 participants constantly lived in IPv6 only networks**

# Evaluations

## **1. User Survey through face to face interview**

- Used devices, OSes, applications
- Troubles

## **2. MTU, fragmentation and NAT Traversal tests by KONAMI Digital Entertainment**

- Availability tests were conducted from the view of consumer (P2P) network games
- MTU and fragmentation were evaluated by STUN
- NAT Traversal was evaluated according to RFC4787

## **3. Analysis of DNS64 log about inappropriate AAAA replies for DNS64 fallback**

- We compared Unbound and BIND in failed AAAA queries



# User Survey through F2F interview

- **Hearing from almost of all participants**
  - 166 of 171 participants
- **297 unique devices were brought in the camp**
  - At least, 2 devices per person
  - Mac OS variants (Mac OS X / iOS) were much popular both for Smart phone and Personal computer
- **VPN is one of the most important applications**
  - IPv4 addresses were needed at anytime due to the lack of IPv6 support on the server side in companies
- **Many participants more or less faced troubles**
  - DNS configuration, Address configuration
  - Switching over suitable AP without any complaints

# Same troubles on the 1<sup>st</sup> experiment

- **Most troubles were same as those of the 1<sup>st</sup> experiment of us or draft-arkko**
  - Manual settings of resolver on older OSES due to lack of DHCP6 support
  - Lack of IPv6 support of OSES, applications, hardwares
  - Errors of name resolving on DNS64 environments
  - Unable to use VPN applications due to lack of IPv6 support on servers
  - IPv4 address literals
  - and so on
- **Most of them have not been fixed yet**
- **Pains of long fallback routine in connectivity check might be relaxed in many latest OSES**
  - In Windows 7, about 30 sec. in 2<sup>nd</sup> exp., 2min. In 1<sup>st</sup> exp.
  - We need further investigation

# New Troubles

- **We had new troubles as follows**
  - Failure of IPv6 neighbor discovery due to the on-link assumption of IPv4 network
  - Locked out IPv6 by vendor in android devices
  - Inappropriate selection of DNS resolvers
  - Previous configuration lived after moving to another Wi-Fi network
  - Crash of an application by IPv4 only plug-in
  - Difficultness on tuning of Happy Eyeball implementation with turn-on/off switch
  - Different behavior among OSes on MTU handling
- **Detail of each trouble is attached in appendix**

# MTU and fragment, and NAT Traversal tests by KONAMI

- **Max MTU size (with DF bit) were different on each network**
  - From 1260 (464XLAT) to 1500 (native IPv6 on IPoE)
  - In some case, MTU from client to server was different from MTU size from server to client in a same network
- **Fragmentation was not available in 4RD on PPPoE and SA46T**
  - In 4RD/PPPoE : an implementation problem of 4RD-CE on handling ICMP Too Big message with PPPoE session was found
  - In SA46T : two of three SA46T implementations did not support fragmentation handling because fragmentation support is out of scope SA46T spec. in the current draft
- **Address and Port dependent mapping was detected in 4RD/PPPoE**
  - A limitation of 4RD specification?
- **Lack of hair-pinning support were detected in 4RD and 464XLAT**
  - lack of implementation or operational mistakes on configuration of NAPT

# Analysis of inappropriate AAAA replies for DNS64 failback

- According to RFC4074, an inappropriate AAAA reply of a name is such reply that contain **NXDOMAIN** although **the A record of the name exists**
- Analysis result during the camp days
  - 45,633 unique names were queried in AAAA
  - Number of inappropriate replies to AAAA queries were different between BIND and Unbound
    - BIND : 201 names
    - Unbound: 69 names
    - Both : 15 names (11 domains)

# New Lessons from the 2<sup>nd</sup> experiment

- **Take care of MTU handling not only on network devices but also on OSES when an MTU black hole occurs**
- **Take care of hair-pinning support and address port mapping algorithms on NAPT implementations for consumer (P2P) network games**
- **Failures of DNS64 fallback are changed by DNS implementations on authoritative servers**
- **Latest OSES work well in IPv6 only networks, however, several OSES require users to turn off IPv4 property in IPv6 only networks or refresh configurations**
- **Trouble shoot is very hard**
  - Root causes of many troubles are still ambiguous

# Open Issues

- **We listed 3 items in -01.txt**
  - Dependency between IPv4 and IPv6 address configuration
  - PMTUD, MTU mismatch and NAT/NAPT behavior problems
  - Workaround for DNS64 problems
- **We will add another item**
  - Trouble shooting methods
- **Are they appropriate ?**

# Future work

- **Revise -01.txt**
  - Write down undocumented parts
    - Much of log data on the 2<sup>nd</sup> exp. have not been analyzed yet
  - Revise along with advices and comments
    - Should we change more appropriate file name ?
- **Next experiment ?**
  - Now under planning
  - The next camp will be held in Sep. 3<sup>rd</sup> to Sep. 6<sup>th</sup> ,2012
- **Give us your feedbacks or comments**



# Appendix

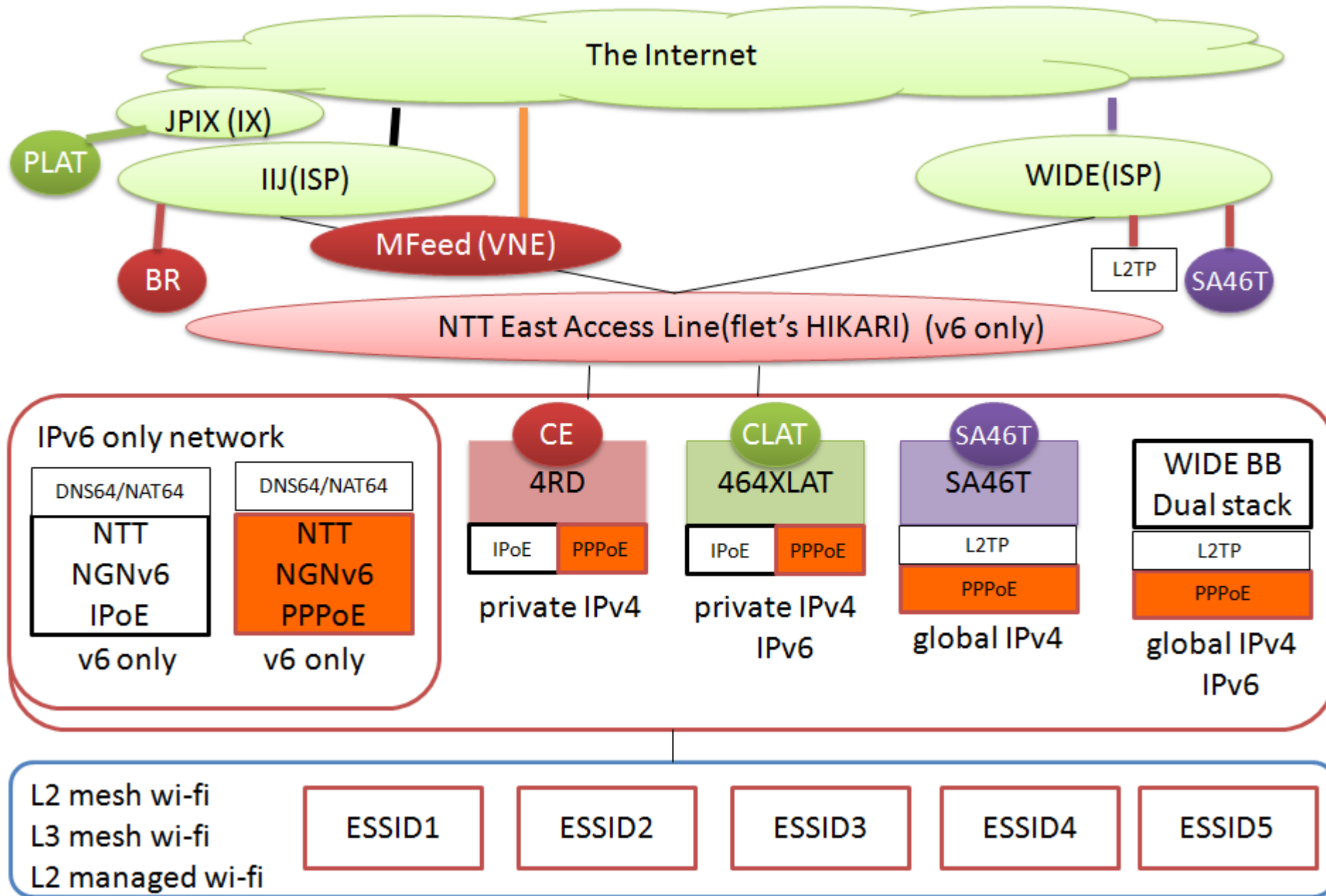
# User access networks evaluated by IPv6 exp. on the 2<sup>nd</sup> experiment

- NTT NGNv6 IPoE / PPPoE with 64 / 464 translators
  - Native IPv6 (global IPv6 only)
    - Operated by IIJ, Internet MultiFeed, NTT-East
  - DNS64/NAT64
    - Operated by WIDE Project and NTT Advanced Technology
  - 4RD (private IPv4 only)
    - Operated by IIJ
  - 464XLAT (private IPv4 and global IPv6)
    - Operated by JPIX and NEC AccessTechnica
  - SA46T (global IPv4 only)
    - Operated by Keio Univ. and Fujitsu group
  - 4RD + ULA IPv6 (private IPv4 and closed IPv6)
    - Operated by IIJ and WIDE project

# User access networks evaluated by IPv6 exp. on the 2<sup>nd</sup> experiment

- Other experiments
  - IPv6 with DNS64/NAT64 over layer 3 mesh wi-fi
    - Operated by Univ. of Tokyo
  - Layer 2 mesh wi-fi
    - Operated by IJ Innovation Institute and CISCO systems
  - LISP+VXLAN
    - Operated by Keio Univ.
  - Dual stack with DNS64/NAT64
    - Operated by WIDE Project

# Main IPv4/IPv6 Network of 2<sup>nd</sup> exp.



# server settings

- **DNS**
  - 4RD
    - IJ's DNSv4 via DNS proxy
  - 464XLAT
    - DNSv6 on each network as primary
    - Google's DNSv6 as secondary
  - Others
    - DNS load balancing by F5 BIG IP
    - BIND for DNS64 (just forwarder)
    - BIND and Unbound for authoritative servers on WIDE BB
- **NAT64**
  - Linuxnat64 (Mar. 5<sup>th</sup> – Mar. 8<sup>th</sup>)
  - F5 BIG IP's NAT64 function (Mar. 7<sup>th</sup> – Mar. 8<sup>th</sup>)
- **DHCP (DHCP4, DHCP6)**
  - ISC DHCP

# User Subnets and DNS settings

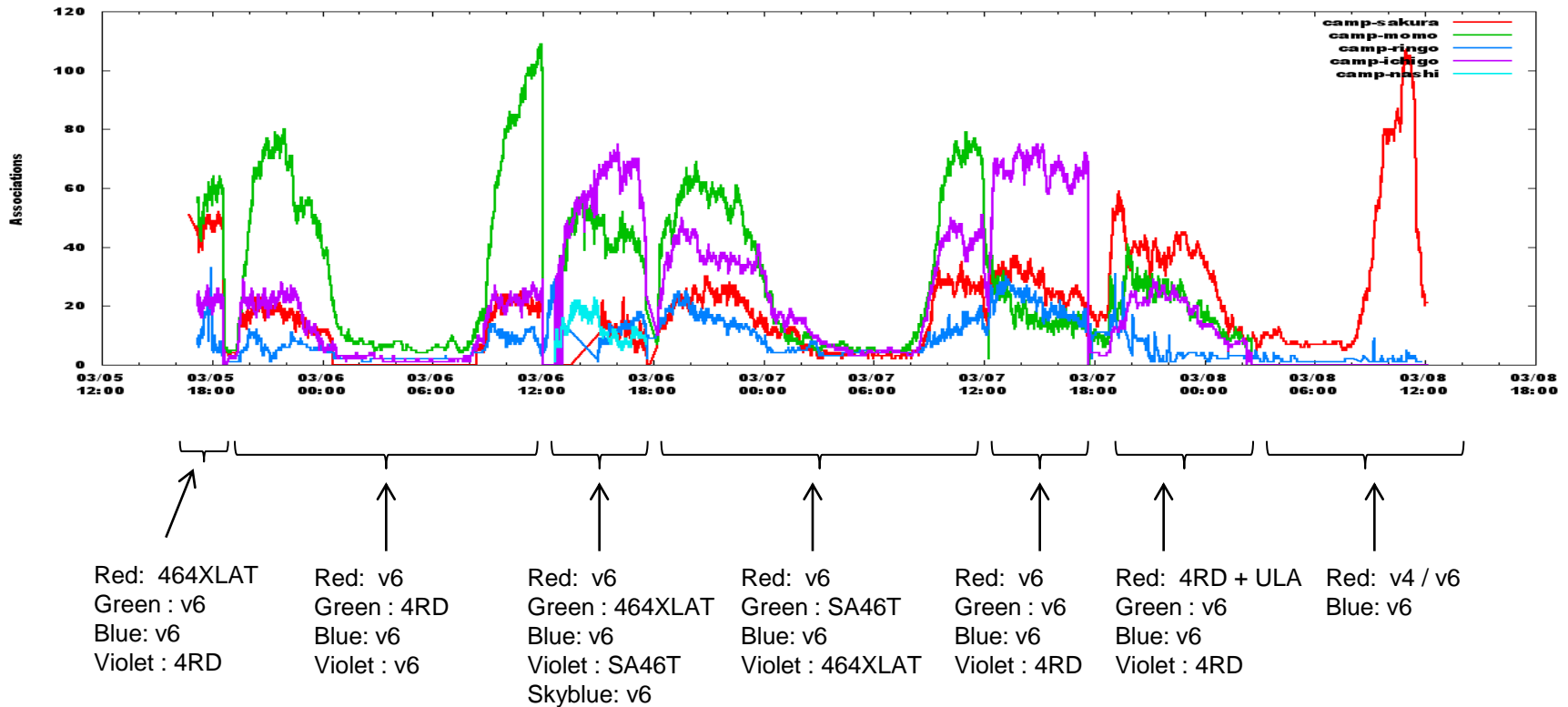
Label	Subnet IP Prefix	DNS
Native IPoE	2409:150:8000:10::/64	2001:200:0:ff60::58 (DNS LB)
Native PPPoE	2001:240:2002:6d10::/64	2001:200:0:ff60::58 (DNS LB)
4RD/IPoE	192.168.12.0/24	210.130.1.1 (via proxy)
4RD/PPPoE	192.168.22.0/24	210.130.1.1 (via proxy)
464XLAT/IPoE	192.168.13.0/24 2409:150:8000:30::/64	2404:1a8:7f01:b::3 2001:4860:4860::8888
464XLAT/PPPoE	192.168.23.0/24 2001:240:2002:6d30::/64	2001:240::13 2001:4860:4860::8888
SA46T-FA, SA46T-FK	203.178.156.0/25	203.178.159.58 (DNS LB)

# Time line of provided networks

time SSID	3/5		3/6			3/7			3/8
	afternoon	night	morning	afternoon	night	morning	afternoon	night	morning
ESSID 1 (sakura)	464XLAT/ IPoE	Native IPoE		L3mesh (IPv6)	Native PPPoE		Native PPPoE	4RD/PPP oE + ULA v6 addr.	Dual stack
ESSID 2 (ichigo)	4RD/IPoE	Native PPPoE		SA46T- FA	464XLAT/IPoE		4RD /IPoE	4RD IPoE through IEEE 8021.11 b	N/A
ESSID 3 (momo)	Native IPoE	4RD/PPPoE		464XLAT/ PPPoE	SA46T-FK		Native IPoE	Native IPoE	N/A
ESSID 4 (nashi)	N/A	N/A		Native IPoE	N/A		N/A	rogue	N/A
ESSID 5 (ringo)	LISP / VXLAN (IPv6)								

lunch/dinner time : configuration changed

# participants' choice of networks



- Participants changed networks in order to get IPv4 or to avoid network congestion
- Around 20 - 60 participants constantly lived in IPv6 only networks



# Same troubles on the 1<sup>st</sup> experiment

- **Most troubles were same as troubles mentioned in the 1<sup>st</sup> experiment of us or in draft-arkko**
  - Manual settings of resolver on older OSes due to lack of DHCP6 support
  - Lack of IPv6 support of OSes, applications, hardwares
  - Errors of name resolving on DNS64 environments
  - Unable to use VPN applications due to lack of IPv6 support on servers
  - IPv4 address literals
  - and so on

# Troubles on VPN applications

- **Troubles reported with specific network name**
  - In DNS64/NAT64 and 464XLAT
    - SSH / IPsec VPN, PPTP-GRE to IPv4 server did not work
      - Out of scope on the spec.
  - In 4RD
    - CISCO IPsec client (Apple has it) did not work in some mode
      - Although the 4RD implementation was improved according to the results of 1<sup>st</sup> experiment
  - In SA46T
    - PPTP client of Mac OS X Lion did not work well
      - However, PPTP client of Windows 7 worked well
- **Troubles reported without specific network name**
  - Dropbox, Oauth, Mobile Me and My Mac, Samba

# New Troubles

- **We had new troubles as follows**
  - Failure of IPv6 neighbor discovery due to the on-link assumption of IPv4 network
  - Locked out IPv6 by vendor in android devices
  - Inappropriate selection of DNS resolvers
  - Previous configuration lived after moving to another WiFi network
  - Crash of an application by IPv4 only plug-in
  - Difficultness on tuning of Happy Eyeball implementation with turn-on/off switch
  - Different behavior among OSes on MTU handling

# Failure of IPv6 neighbor discovery due to the on-link assumption of IPv4 network

- **Following OSes or Wireless access controller needed turning off IPv4 property in IPv6 only networks (reported from participants)**
  - Lenovo Access Connections 5.72 in Windows XP
  - Lenovo Access Connections 5.85 83C771WW in Windows 7
    - But, another participant answered that he did not meet such trouble with Lenovo Access Connection 5.85 83C7711WW in Windows 7 with turning on IPv4 property
  - Ubuntu 11.10
  - iPhone iOS (version unknown)
- We need further investigations

# Locked out IPv6 by vendor in android devices

- **Two different android devices by different vendors**
  - **Same OS version**
    - of course, IPv6 is officially supported
  - **Same mobile company**
  - **Same behavior in IPv4**
  - **Different behavior in IPv6**
    - One could run IPv6 function, the other could not

# Inappropriate selection of DNS resolvers

- **Mainly this trouble was derived from wrong configuration of DHCP6 or DNS proxy**

# Previous configuration lived after moving to another Wi-Fi network

- **Previous resolver settings were remained**
  - Reported from several windows 7 users
  - Some troubles were derived due to wrong configuration of DHCP6
    - Mismatch between announce resolver address on DHCP6 and local address of the resolver on the access network
    - Too long lease time
  - Root causes of other patterns were not clarified
    - No resolver entry of /etc/resolv.conf in Mac OS X Lion
- **It need turning off/on wireless interface or refresh dhcp client**

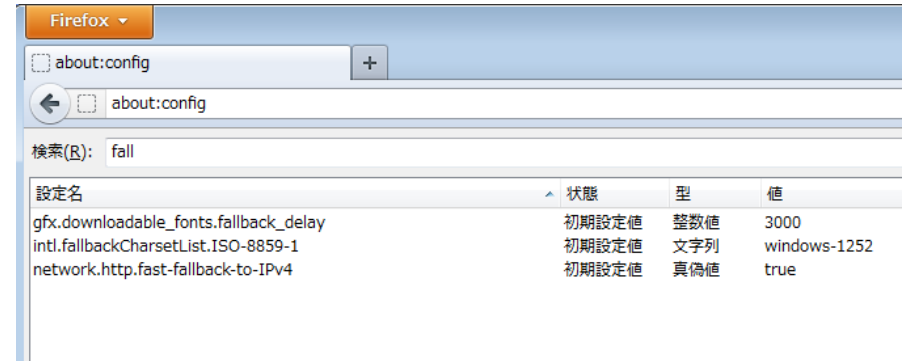
# Crash of an application by plug-in

- **Crash of web browsers due to lack of IPv6 support plug-in apps**
  - Reported in Firefox and safari of Mac OS X users



# Difficultness on tuning of Happy Eyeball

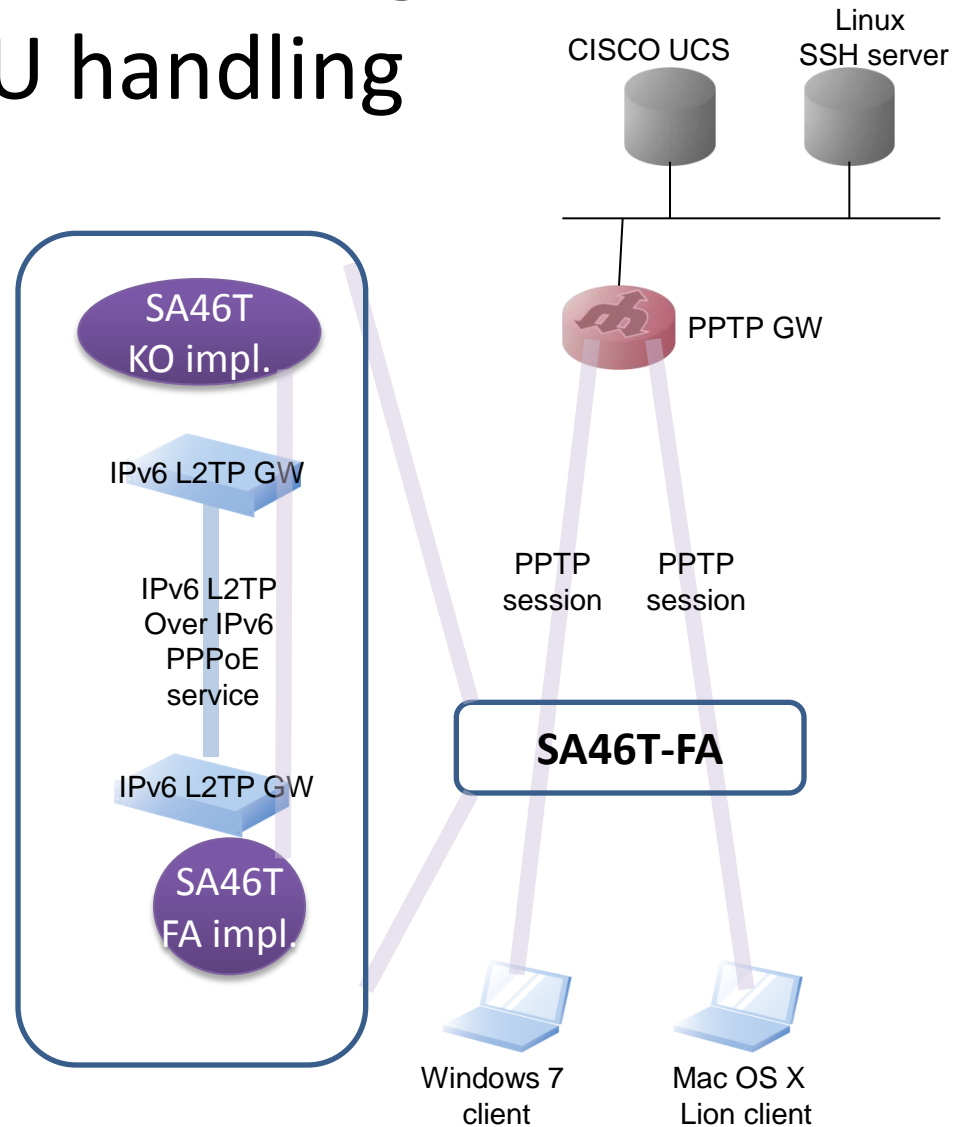
- **Happy eyeball**
  - Increase performance on web browsing where an IPv4 network has better performance than an IPv6 network
  - may decrease performance on web browsing where an IPv6 network has better performance than an IPv4 network
- **Turning on / off of some happy eyeball implementations were difficult or could not be easily found**



Example of turn of / off switch  
(Firefox 11.0)

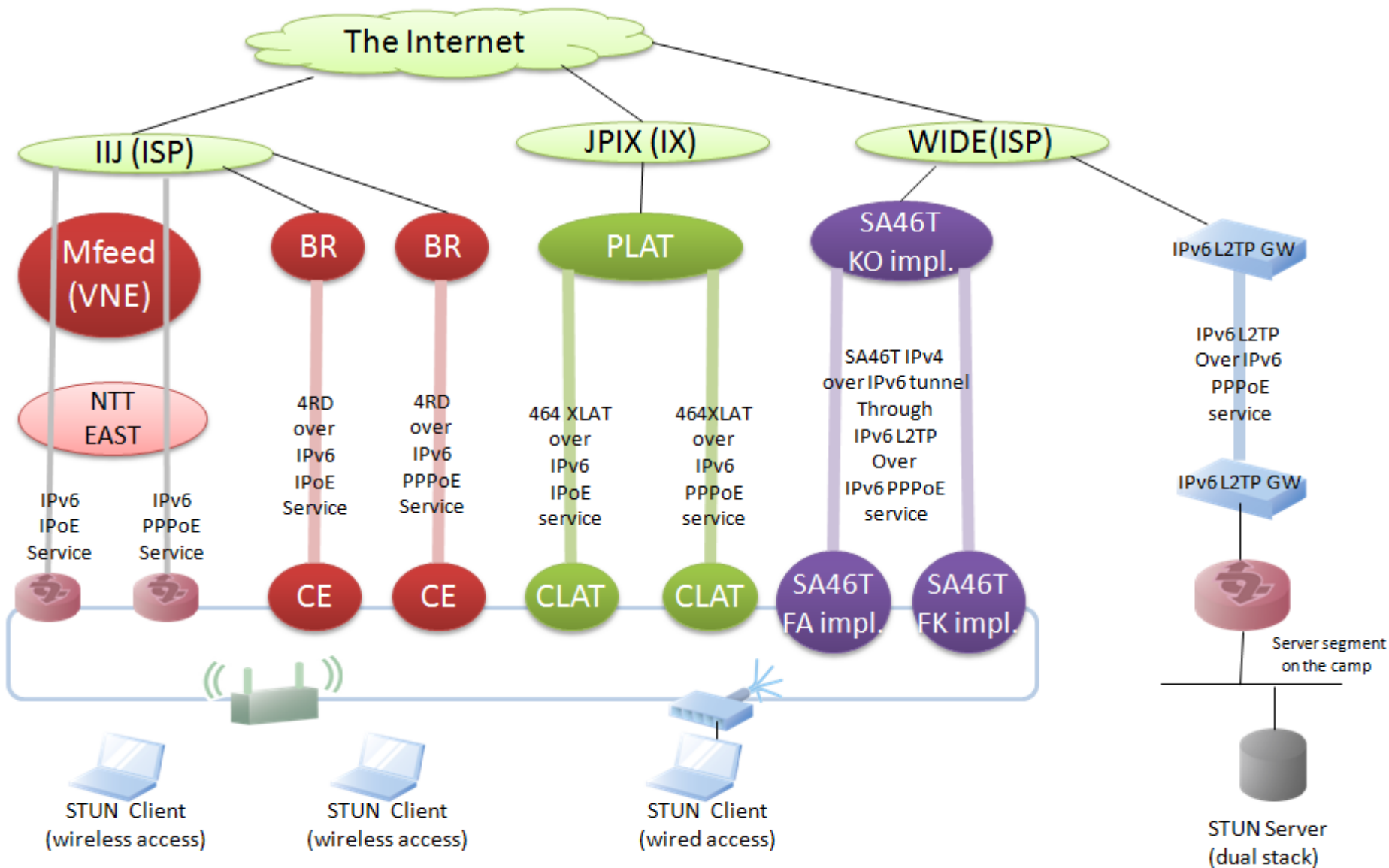
# Different behavior among OSes on MTU handling

- **Test PPTP over SA46T IPv4 over IPv6 tunnel again**
  - Unavailability of PPTP through SA46T was one of reported trouble in the 1<sup>st</sup> experiment
    - Especially Mac OS X Lion users claimed
  - Compare windows 7 PPTP client and Mac OS X Lion PPTP client
    - Remote KVM access of CISCO UCS server
    - SSH session to a linux server
- **PPTP on Windows 7 worked without any problems**
  - Ping from the Windows client to the linux server worked up to 1460 MTU with DF bit, also much more MTU size packets were fragmented without DF bit
- **Packets from Servers stopped after a few minute in PPTP on Mac OS X Lion**
  - Ping from Mac OS X Lion client to the linux server worked up to around 1411 to 1420 MTU size without DF bit
- **According to this result, something was wrong on MTU handling of Mac OS X**
  - We could not clarify actual root causes of this trouble
- **We need further tests**



# MTU, Fragment, NAT Traversal tests by KONAMI Digital Entertainment

# Test Topology



# Test methods

- **MTU size and Fragmentation were evaluated by STUN between server and client**
- **NAT Traversal behaviors were evaluated according to RFC4787**
  - UDP Between Server and Client
  - UDP Between Clients
    - Clients got each address through the STUN server

# MTU and fragment tests by KONAMI

		IPoE	PPPoE	4rd-IPoE	4rd-PPPoE	464XLAT-IPoE	464XLAT-PPPoE	SA46T-fa	SA46T-fk
	protocol	v6	v6	v4	v4	v4	v4	v4	v4
Client→Server	PMTU	1500	1452	1452	1452	1476	1260	1460	1460
	Fragment (DF=0)	○	○	○	×	○	○	○	×
Server→Client	PMTU	1500	1500	1452	1280	1480	1260	1460	1460
	Fragment (DF=0)	○	○	○	○	○	○	×	×

- Disability of Fragmentation C => S on 4RD-PPPoE was derived from an implementation issue around MTU handling function of the 4RD-CE router
- Difference among Fragmentation of SA46T was derived from difference of fragment support function of three SA46T implementations

# NAT Traversal tests by KONAMI

Network	Protocol	Connectivity
Native PPPoE	IPv6	Good (no NAT)
Native IPoE	IPv6	Good (no NAT)
4RD/PPPoE	IPv4	Bad (Address and Port dependent mapping)
4RD/IPoE	IPv4	Bad (no hairpinning)
464XLAT/PPPoE	IPv4	Bad (no hairpinning)
464XLAT/IPoE	IPv4	Bad (no hairpinning)
SA46T-FA	IPv4	Good (no NAT)
SA46T-FK	IPv4	Good (no NAT)

- No hairpinning of 464XLAT was derived from lack of function on the PLAT implementation
- No hairpinning of 4RD/IPoE was derived from mis-configuration
- Address and Port dependent mapping of 4RD/PPPoE was derived from the limitation of address mapping algorithm of 4RD

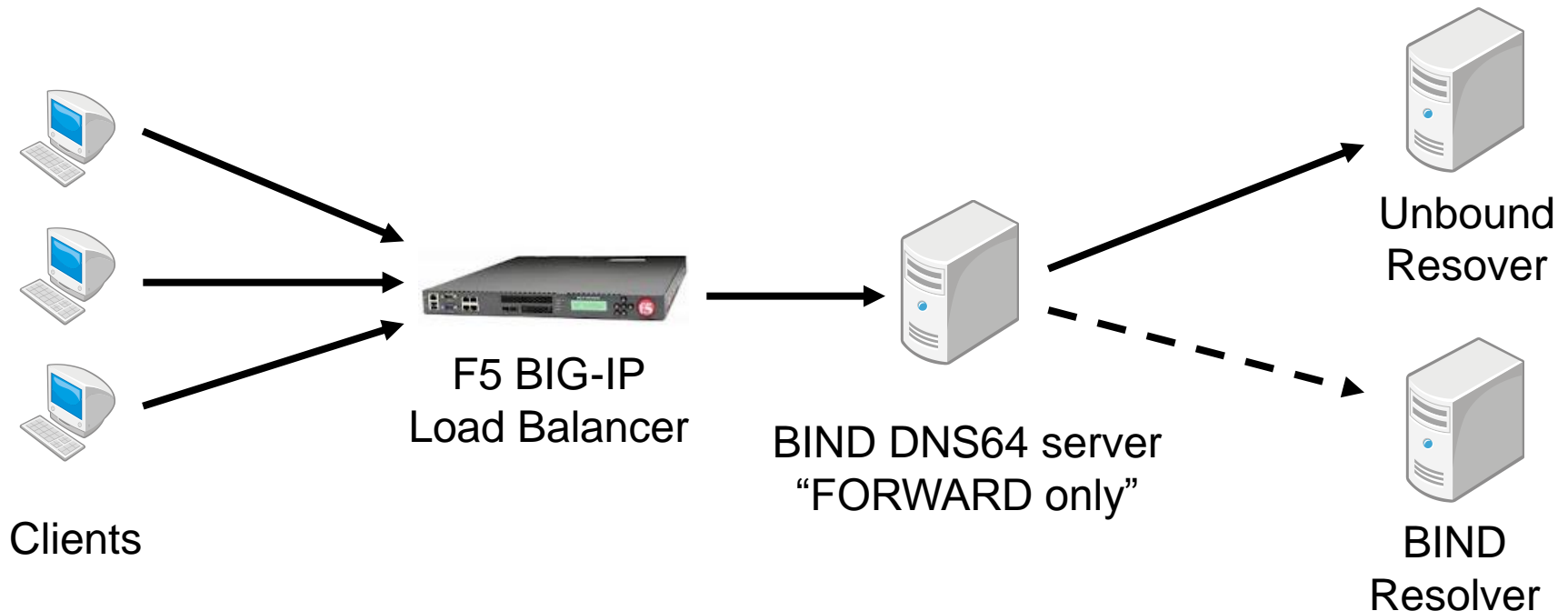
# Bogus authoritative server interfere DNS64

Tomohiro Ishihara  
University of Tokyo  
Graduate School of Arts and Sciences

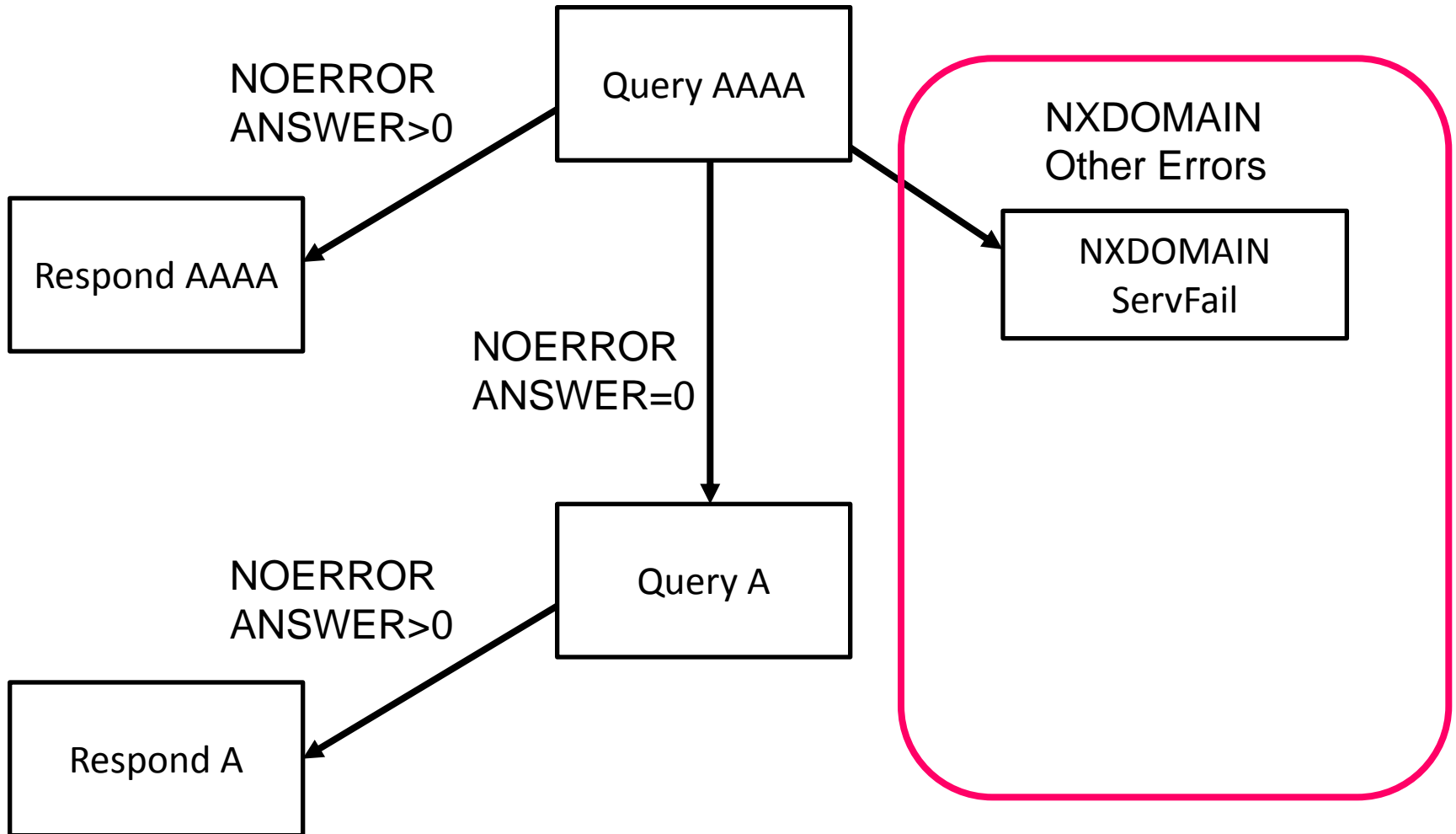


# DNS64 configuration in this CAMP

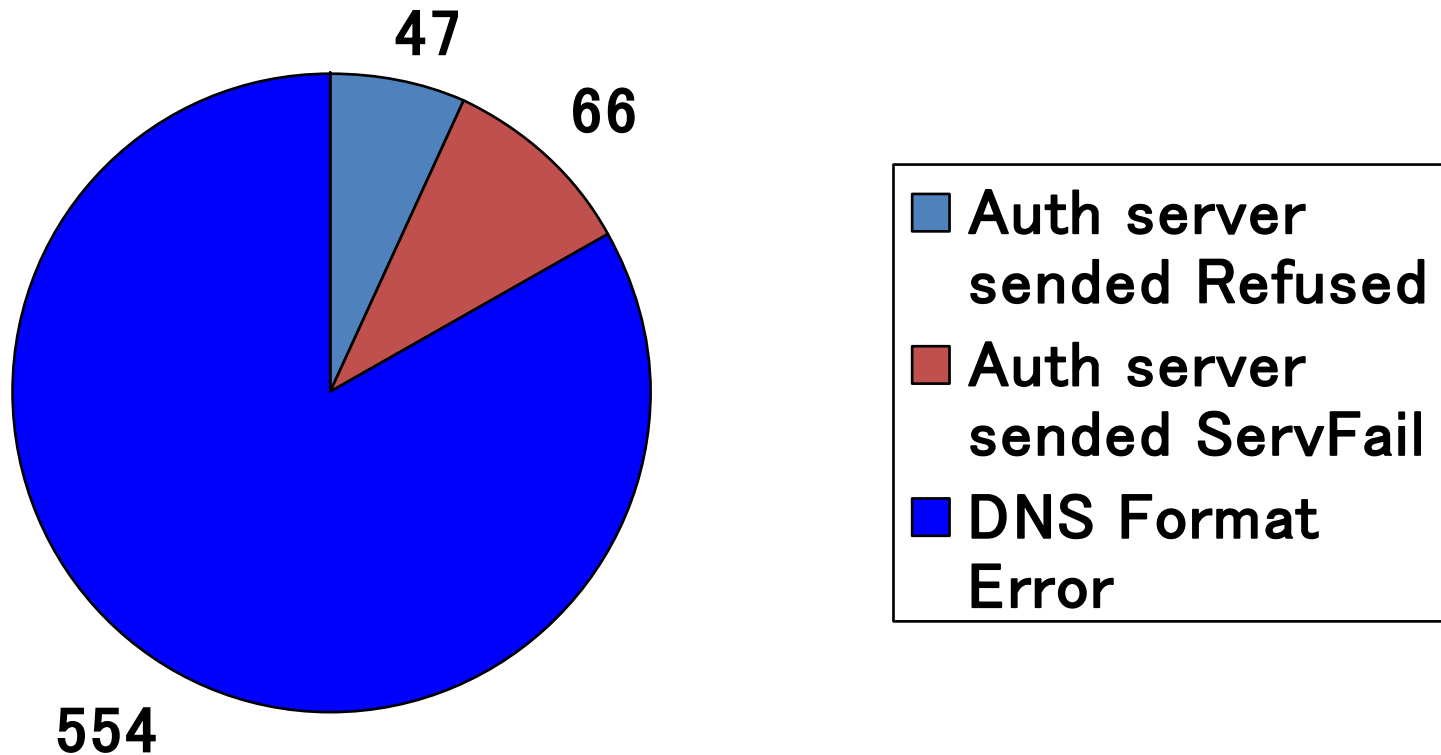
- BIG-IP forwarded queries to DNS64 server except 'www.camp.wide.ad.jp'
- DNS64 server only forwarded queries to Unbound/BIND resolver and did not do any iterative queries itself
- We had changed target resolver from bind server to unbound server at midnight on March 5th



# DNS64 problem



# Error Result in March 5<sup>th</sup> counting by unique name



# Re-check “FORMERR” in Mar. 5<sup>th</sup>

- Resolved 554 AAAA Records which occurs FORMERR both by BIND resolver and by Unbound resolver
- NXDOMAIN is inappropriate reply to failures of DNS64 fallback

	NXDOMAIN	NOERROR Record = 0	NOERROR Record > 0	others (ex. no reply)
Unbound	21	506	7	20
BIND	449	66	9	11

# Analysis of inappropriate AAAA replies for DNS64 fallback

Item	number
# of unique names (A, AAAA, PTR, SRV)	70,886
# of names queried by AAAA	45,633
# of names which returned NXDOMAIN in AAAA through Unbound	4,011
# of names which returned NXDOMAIN in AAAA through Unbound although their A records exist	69
# of names which returned NXDOMAIN in AAAA through BIND	4,234
# of names which returned NXDOMAIN in AAAA through BIND, although their A records exist	201
# of names which returned NXDOMAIN in AAAA through both BIND and Unbound, although their A records exist	15

- BIND was more restrict than Unbound on FormError check
- 15 names were surely inappropriate for DNS64 fallback

# RFC6147 says...

- Any other RCODE is treated as though the RCODE were 0 and the answer section were empty. This is because of the large number of different responses from deployed name servers when they receive AAAA queries without a AAAA record being available. Note that this means, for practical purposes, that several different classes of error in the DNS are all treated as though a AAAA record is not available for that owner name.
- It is important to note that, as of this writing, some servers respond with RCODE=3 to a AAAA query even if there is an A record available for that owner name. Those servers are in clear violation of the meaning of RCODE 3, and it is expected that they will decline in use as IPv6 deployment increases.