Implementing IPv6: experiences at KAME project

Jun-ichiro Hagino, IIJ research laboratory
itojun@{iijlab.net,kame.net}

Outline

- What is IPv6, and why IPv6 (brief summary)

- What is KAME project
- Some technical insights observed at KAME

- Implmentation status
- What are the TODOs, issues from both specification/implementation

What is IPv6?

- Expansion of address space - 32bit -> 128bit
  - 32bit: 4.3 billion nodes maximum
    - not sufficient, blocking new IP-based applications from appearing
  - 128bit: 3.4 x 10^38 nodes maximum
- Make new technologies mandatory
  - IPv4: designed in 1970’s
  - IPv6: designed in 1990’s
  - autoconfiguration, multicast, security, ...

- Why? - IPv4 address space is filled up, NAT is killing us all
- When? - Already there, so you should
- How? - (next slide)

How to operate IPv6?

- IPv6 is "IP with bigger address space", almost no difference with IPv4
  - bigger address space makes a huge difference
- Base spec
  - NAT-free 128bit address, simpler base header, extensible header format
  - Enables many future application deployment and uses
- Routing - OSPFv3, RIPng, BGP4+
- QoS - diffserv, RSVP (separate effort from IPv6 itself)
  - more friendly than IPv4
- Mobility - mobile-ip6
  - No foreign agent necessary
- Security - IPsec (separate effort from IPv6 itself)
  - A "fully conformant IPv6 implementation" must have IPsec code
- Autoconfiguration
  - stateless autoconf, DHCPv6
- Multicast - PIM, MLD (= IGMP)
- Applications - HTTP, FTP, VoIP, whatever you do with IPv4
  - New applications would appear when IPv6 hits the critical mass

Problem we had in 1995...

- IPv6 specification is out, but there’s no codebase/testbed
- "No codebase" means "high deployment hurdle"
  - IPv4 was deployed because it was bundled into BSD and other operating systems
  - BSD served as the reference codebase for others
  - With IPv6, only spec is there, no code
- We don’t really know if the spec will work fine or not
  - Need to check if specification works right, by actual operational experiences
- Our answer:
  - WIDE IPv6 backbone: IPv6 network operation, with real daily traffic
  - KAME project: IPv6 reference implementation for BSDs

WIDE IPv6 backbone

- WIDE operates nationwide IPv6 backbone since 1996/06/09
  - We started with native, not tunnel!
- WIDE operates Gbps-class IPv6 backbone, will start operating 10Gbps nationwide backbone
- Days with tunnel/experiment/testbed is already over, IPv6 is for daily use for many people in WIDE and outside of WIDE
- Dr Hiroshi Esaki’s presentation (yesterday) covered more details

KAME Project goals

- Implement IPv6, IPsec, and whatever interests us onto *BSD
- Redistribute under BSD license
  - Research reference, education, and deployment
- Consortium of universities and companies
  - Keio-U, U-Tokyo, Fujitsu, Hitachi, IIJ, NEC, Toshiba, Yokogawa
- 10 core implementers, and number of supporting casts
- April 1998 - March 2004 (extension?)
- Supported platforms
  - OpenBSD, NetBSD, FreeBSD, BSD/OS
- Many commercial router/OS vendors are using our code
  - Juniper, Extreme Networks, IIJ, Fujitsu, Hitachi, ...
  - Apple MacOS X, BSD/OS, VxWorks...
- Active feedback to IETF specifications

KAME IPv6 code

- Rock stable, has been available since 1997
  - IPv6 operational experience in WIDE research community
  - (torture-test)
- High coverage of specs
  - more than IPv6/IPsec 40 RFCs
- Ultra spec-conformant
  - Conformance tested by TAHI verification team
- Ready for daily use (we actually are using it every day)
  - Everything is IPv6 ready
- Highly integrated
  - *BSD integration - IPv6 ready from boot floppy!
- Heavily documented and publically available source code
  - Good for learning, testing, and deployment

other advanced/experimental items

- ALTQ: alternate queueing framework
  - for diffserv/traffic management
- mobile-ip6
- multicast DNS lookup
- icmp6 name lookup
- IPv6 DNS lookups - EDNS0 considerations
- application supports
  - Mozilla, apache, ssh, perl, python, everything you want
- IPv6 PPP, NFS and NIS (NetBSD integration)
- DHCPv6 prefix delegation

KAME experiences and enhancements

- Extension for scoped address format
- Denial-of-service by extension header chain
- (In)security of some of IPv6 specs
- Therer were many more issues we have faced
- refer to other papers/internet-drafts from us
- We provided feedbacks to IETF specifications already
  - As updates to existing i-d/rfc, or independent i-d/rfc

Extension for scoped address format

- IPv6 introduced "scoped address"
  - unicast: Link-local, site-local
  - multicast: 15 scopes
- Scoped address must come with scope identifier
  - They are ambiguous if we only have 128bit specified
- We need a common notation!
- fe80::1234:5432:abcd:ef00%ether0
- getaddrinfo/getnameinfo can support it cleanly
  - getaddrinfo fills in sin6_addr + sin6_scope_id
  - getnameinfo converts sin6_addr + sin6_scope_id into string

```
fe80::1 --- my machine --- fe80::1
```

Denial-of-service by extension header chain

- IPv6 employs the idea of "extension header chain"
  - Many extension headers can be attached to a packet
- There’s no upper limit to the number of extension headers!
  - Specwise, there’s no limitation at all
- KAME strategy:
  - Design function call tree so that there’s no kernel stack overflow
  - Limit number of extension headers acceptable (tunable)

(In)security of some of IPv6 specs

- Some of IPv6 specs talks conflicting thing
  - So many tunnelling specifications, with different definitions - inbound processing gets hairy
  - Use of special addresses (IPv4 mapped) - impose more work to third-party userland programs, insecure behavior by defualt
  - Issues with translators - tend to be configured as an open relay, help bad guys mistakenly
- Solution: careful implementation, feedback to specs
  - Diagnose each specs and implement those make sense only
  - Put enough warnings to users
  - Careful restrictions/tweaks into API, feedback to specs
- KAME case:
  - 6to4 is not enabled by default
  - IPv4 mapped address (inbound) is not enabled by default, or not supported at all
  - IPv4 mapped address (outbound) is supported only in some cases
  - Filter out some of misconfigured DNS database entries

Towards the real deployment

- Host OS, Router OS are all ready by now
  - *BSD, Solaris, Linux, whatever
  - Cisco, Juniper, Extreme, NEC, Fujitsu whatever
- It is just a matter of configuration to enable it

- Education is the key
  - Educate ISP operators
  - Educate university operators
  - Educate end users (may not be necessary, as IPv6 is hidden in the very bottom)

- Applications readiness
  - Upgrade application
  - Upgrade library (if library API is address-family neutral)

- Convince your boss, configure it and use it!

Deployment of WIDE IPv6 network

- 1996 - started with 64k leased lines
  - We are trying to avoid tunnels from day one
- 1997 - steal bandwidth from IPv4 backbone (ATM pipes)
- 1998 - JB: nationwide ATM research network, funded by government
  - With IPv6 we can get circuits cheap :-P
- 2000 - 200 to 300Mbps class IPv6 native backbone, nationwide
- 2002 - Gbps-class native backbone, nationwide

- Researchers have trouble with tight IPv4 address allocation policies
- IPv6 is the solution for them!
  - 30Mbps digital video multicast traffic, for remote classrooms (Wisconsin - Yokohama)
  - MP3 audio over multicast + IPsec
  - xcast (small group multicast)
  - IPv4/v6 over satellite medium
  - diffserv experiments/operations
  - Of course, routing experiments/operations - OSPFv3

Summary

- What is IPv6, and why IPv6 (brief summary)

- What is KAME project
- Some technical insights observed at KAME

- Implementation status
- What are the TODOs, issues from both specification/implementation
  - IPv6 is ready for everyone in every domain, so just configure it and use it!

- http://www.kame.net/