IPv6 Stack Overview
-Linux Is IPv6 Ready-

Linux Networking [IPv4/IPv6] Co-maintainer
USAGI/WIDE Project Co-Chair
Keio University

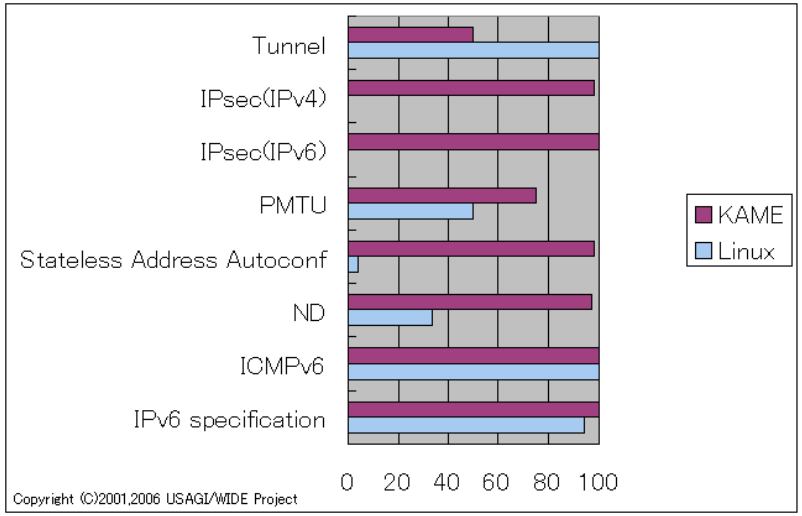
Hideaki YOSHIFUJI
<yoshfuji@linux-ipv6.org>

IPv6 -Internet Protocol version 6-

- Standardized by IETF (1994)
  - Scalability, performance, extentability, privacy, security
- Supported by various network devices and softwares
  - Commercial services by xSP available
  - Practical stage as the standard for connecting "everything"
- Linux
  - Pedro Roque implementation in 2.1 (1996)
  - EXPERIMENTAL
  - "useless" because of quality
    - Unstable
    - Interoperability issues
      - API, Neight Discovery, Stateless Address Autoconfiguration
    - Missing pieces
      - IPsec, Mobile IPv6, packet filter

Quality Test for Linux 2.2.15

- By the TAHI IPv6 Conformance Test Suite
- Far less fair results
  - Stateless Address Configuration
  - Neighbor Discovery



IPv6 Core

- Issues
  - Incorrect state transition of neighbor cache
  - Inaccurate or lazy lifetime management
  - Insufficient test of incoming packet
- Solutions
  - Rearrangement of dependency, mutual exclusion etc.
  - Introduction of common procedures

  ->Improvement of maintenance

IPv6 Core (cont'ed)
New Functions (selected)

- Privacy Extensions
  - Periodical formation of pseudo-random interface identifier
  - Important in terms of security
- Router Selection
  - Preference (Hi/Med/Low)
  - Route Information
  - Router Reachability Probing
- Node Information Query
  - Node information Query/Reply
  - Daemon in user space
    - Implementation in kernel at the early stage

IPsec

- XFRM appeared after FreeS/WAN / IABG and USAGI
  - By David S. Miller, Alexey N. Kuznetsov
  - IPv6 support by Kunihiro Ishiguro and USAGI (Kazunori Miyazawa, Mitsuru Kanda and Hideaki Yoshifuji)
- Features
  - IPv4/IPv6 unified framework
  - Cryptoapi
    - Pluggable algorithms
  - XFRM (Transform) processing engine
    - Abstraction by flowi and agressive use of cache
    - Stackable Destination
  - Dual Interface
    - Netlink and PF_KEYv2 (with KAME extensions)
    - Racoon / Racoon2 / Pluto
  - High Quality

IPsec (cont'ed)

- Challenges
  - Cross-protocol Tunneling
    - IPv4/IPv6 / IPv6/IPv4
  - AES-128-XCBC-96
    - coming soon
  - Extended Sequence Number
    - 64bit
  - BEET mode

Packet Filter

- Development and Maintenance mainly by Netfilter Project
  - Only fundermental IPv6 packet filter was available
- USAGI Commitment
  - 2.6.15: nf_conntrack
    - Network protocol independent framework
  - 2.6.16-: state module
    - Filter by the state of “connection” (w/ iptables 1.3.4-)
  - Participation in Netfilter Project core team
    - Kozakai (2005-)
- Challenges
  - IPv4 NAT
  - More efficient packet filter system

Mobile IPv6

- Technology to maintain constant address after movement, using IPv6 features
- Linux Implementation
  - MIPL (Mobile IPv6 for Linux)
    - Go-Core Project in HUT(Helsinki University of Technology)
  - USAGI implementation
    - MIPL-based
    - Supporting Latest specifications
    - IPsec friendly
    - More stable
  - Issues
    - Everything in kernel...too many/big changes

Mobile IPv6 (cont'ed)

- MIPL2
  - Cooperation between Go-Core Project (Helsinki University of Technology) and USAGI Project
  - Allocation of the roles in kernel and space daemon
    - Kernel: packet treatment
      - XFRM framework
    - Daemon: signaling, movement detection
  - Aimed at high specification conformity
    - Using TAHI test suite, IPv6 Ready Logo Self Test
  - Cooperative to IPsec and key exchange (IKE)
    - Standardization (migrate)
  - 2.0.1 release
  - Main-line merge in preparation/progress
    - Rearrangement and functional separation

Automatic Test System Background

- Linux is growing day by day
  - Basically... :-p
- It occasionally include bugs
  - Revert or fix?
    - Fix!
- Important are early finding and early treatment
- Test tool utilization
  - Tests are performed on problem report and occasionally
  - A lot of work and time
    - Setup, compile, test, analysis, release
    - Frequent tests needed

Automatic Test System Components and Tools

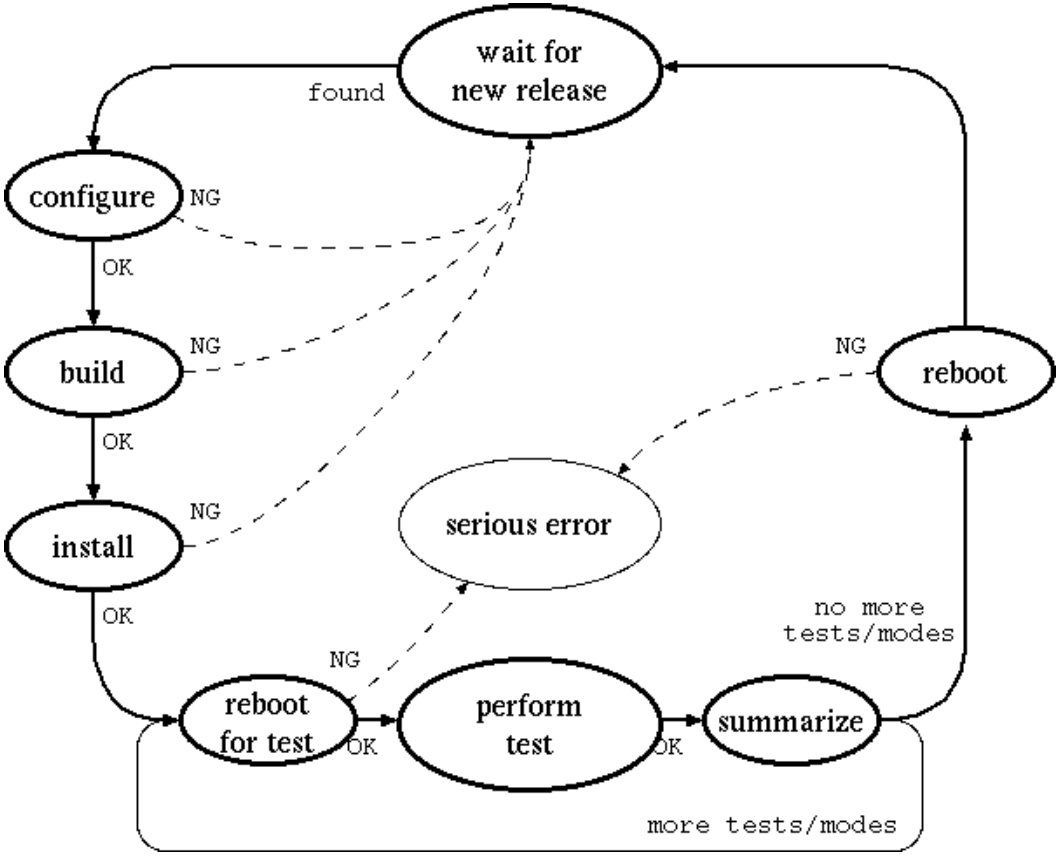
- Components
  - TN (Test Node)
  - NUT (Node Under Test)
  - 2 (two) back-to-back links
  - Parallel test can be perfomed by 1 (one) Test Node using several NUTs

- Tools (selected)
  - Grub
  - Setup scripts
    - Switch setups (init scripts and configuration files) by kernel parameter
    - Allow several setups on one kernel

Automatic Test System Workflow

- Flow Chart
  - Waiting
    - Not only released version but rc and git versions targeted
  - Building
    - Oldconfig
  - Test
    - Specify an appropriate mode and setup, boot, and perform test
      - Host / Router
    - Analysis
      - Abstract of the different information between the key version
- Data Collection
  - Source, binary, test results, executed logs, etc.
  - Public via HTTP at http://testlab.linux-ipv6.org

Automatic Test System Flow Chart



Automatic Test System Perform Tests

- IPv6 Ready Logo Phase-1 Self-Test
  - Host
  - Router
- IPv6 Ready Logo Phase-2 Self-Test
  - Host, IPsec End-node
  - Router, IPsec Security-gateway
  - Mobile IPv6 MN/CN/HA (planned)
- TAHI Conformance Test
  - IPsec end-node (selected)
  - IPsec security-gateway (selected)

- Total period of time over 24 hours...

Automatic Test System Effects and Future Challenges

- Free from daily tests
- Efficient finding and identification of problems
  - Tremendous improvement has beed accomplished corresponding to the test frequencies, even though manual operations

- Further challenges
  - Development of more easily accessible interface
  - Automatic finding and identification
    - Automatic alert
    - git-bisect
  - False-positive / sporadic phenomena
  - Visualization

Linux Is IPv6 Ready

- Certification of IPv6 Ready Logo
  - 2.6.11-rc2: Phase-1 (Host, Router)
  - 2.6.15: Phase-2 (Host + IPsec End-node)
- No longer EXPERIMENTAL
  - 2.6.12-rc1
- Mobile IPv6 will be merged soon
  - 2.6.18-19: multiple tables / policy routing
  - 2.6.19-20
- Various possibility in the IPv6 world
  - HIP, SHIM6, XCAST, ...
- Further Challenges
  - To establish user-friendly system for IPv6
  - To promote documents and know-how