Developing CPE Routers based on NetBSD: Fifteen Years of SEIL

Masanobu SAITOH <msaitoh@netbsd.org>

Hiroki SUENAGA <hsuenaga@iij.ad.jp>

contents

1. Our router and the usage
2. The difference between NetBSD and SEIL
3. New product development
   - Usual workflow
   - Changing NetBSD's base version
4. Conclusion
   - Problems
   - Solutions
   - Current status

1. Our router and the usage

CPE router for business use

- **C**ustomer **P**rimises **E**quipment
- Locations:
  - Branch office, satellite office
  - Shops
    - Convenience stores
    - Fast food stores
    - Gas stations
  - Data center
    - To terminate a lot of IPsec tunnels

A lot of requirements…

- Complicated network
  - A lot of routers
  - Tunneling
  - Redirect, proxy
  - Redundancy
- A lot of functions
  - Ethernet, ISDN, Mobile, WiFi
  - IPv4, IPv6
  - PPPoE, DHCP
  - NAT, NAPT
  - Filter, firewall
  - IPsec, SSTP
  - PPTP, L2TP, L2TPv3
  - Dynamic routing
  - Policy routing
  - Transparent proxy
  - VRRP
  - QoS
- Manageability
- Reliability (e.g. MTBF)
- Stability

… make configuration very difficult.

SEIL Project

- "SEIL" is a brand name of router products.
- "**S**imple and **E**asy **I**nternet **L**ife"
  - To make management simple and easy
    - Configurations
    - Operations
    - Monitoring
- Started in 1997

By the way, 2014 -1997 = ?

Developing CPE Routers based on NetBSD: ~~Fifteen~~ Years of SEIL Seventeen

Masanobu SAITOH <msaitoh@netbsd.org>

Hiroki SUENAGA <hsuenaga@iij.ad.jp>

Why an ISP develops router?

- To manage CPEs from ISP polietry.
- To develop new service with our own router.

Router Products

MMU-less

Wan Interfaces	Lan Interfaces	CPU(Model)	Released
128Kbps BRI	10Mbps Ethernet	Hitachi SH2(SH7604)@20MHz	Aug 1998
1.5Mbps PRI	10Mbps Ethernet	Hitachi SH3(SH7709A)@133MHz	Dec 1999
128Kbps BRI	100Mbps Ethernet	Hitachi SH4(SH7750)@200MHz	Oct 2001
1.5Mbps PRI	100Mbps Ethernet	Hitachi SH4(SH7750)@200MHz	Oct 2001
100Mbps Ethernet	100Mbps Ethernet	Hitachi SH4(SH7750)@200MHz	Nov 2001
25Mbps ATM	100Mbps Ethernet	Hitachi SH4(SH7750)@200MHz	Oct 2002
1Gbps Ethernet	1Gbps Ethernet	Freescale PowerPC G4(MPC7745)@600MHz	Jun 2003
100Mbps Ethernet	100Mbps Ethernet	Intel XScale(IXP425)@400MHz	Dec 2003
1Gbps Ethernet USB 3G/LTE Modem	1Gbps Ethernet	Cavium Octeon(CN3010)@300MHz	Feb 2008
1Gbps Ethernet USB 3G/LTE Modem	1Gbps Ethernet	Cavium Octeon(CN3120)@500MHz	Feb 2008
1Gbps Ethernet USB 3G/LTE Modem 128Kbps BRI	100Mbps Ethernet	Intel XScale(IXP432)@400MHz	Oct 2008
1Gbps Ethernet USB 3G/LTE Modem	1Gbps Ethernet 802.11n Wireless LAN	Marvell Kirkwood(88F6281)@1.2GHz	Feb 2013

NetBSD

SMF

- **S**EIL **M**anagement **F**ramework
  - Zero Configuration.
  - Manage Network via server
- We started this service in 2003.

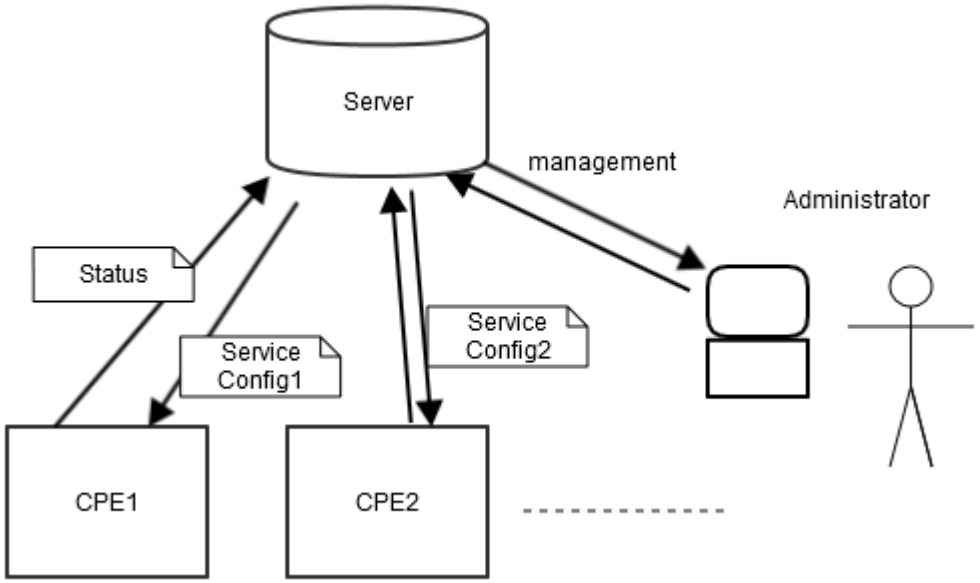
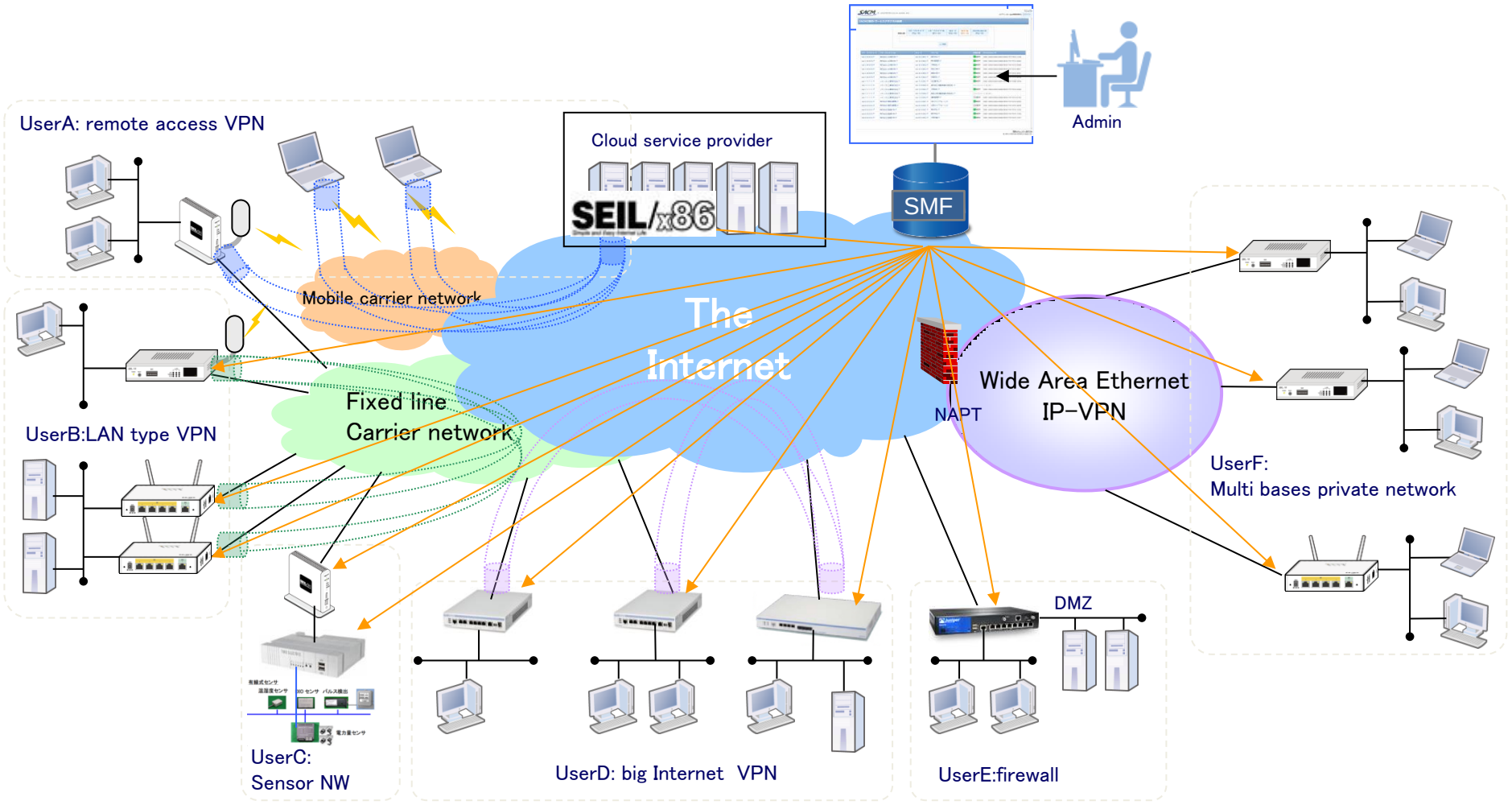


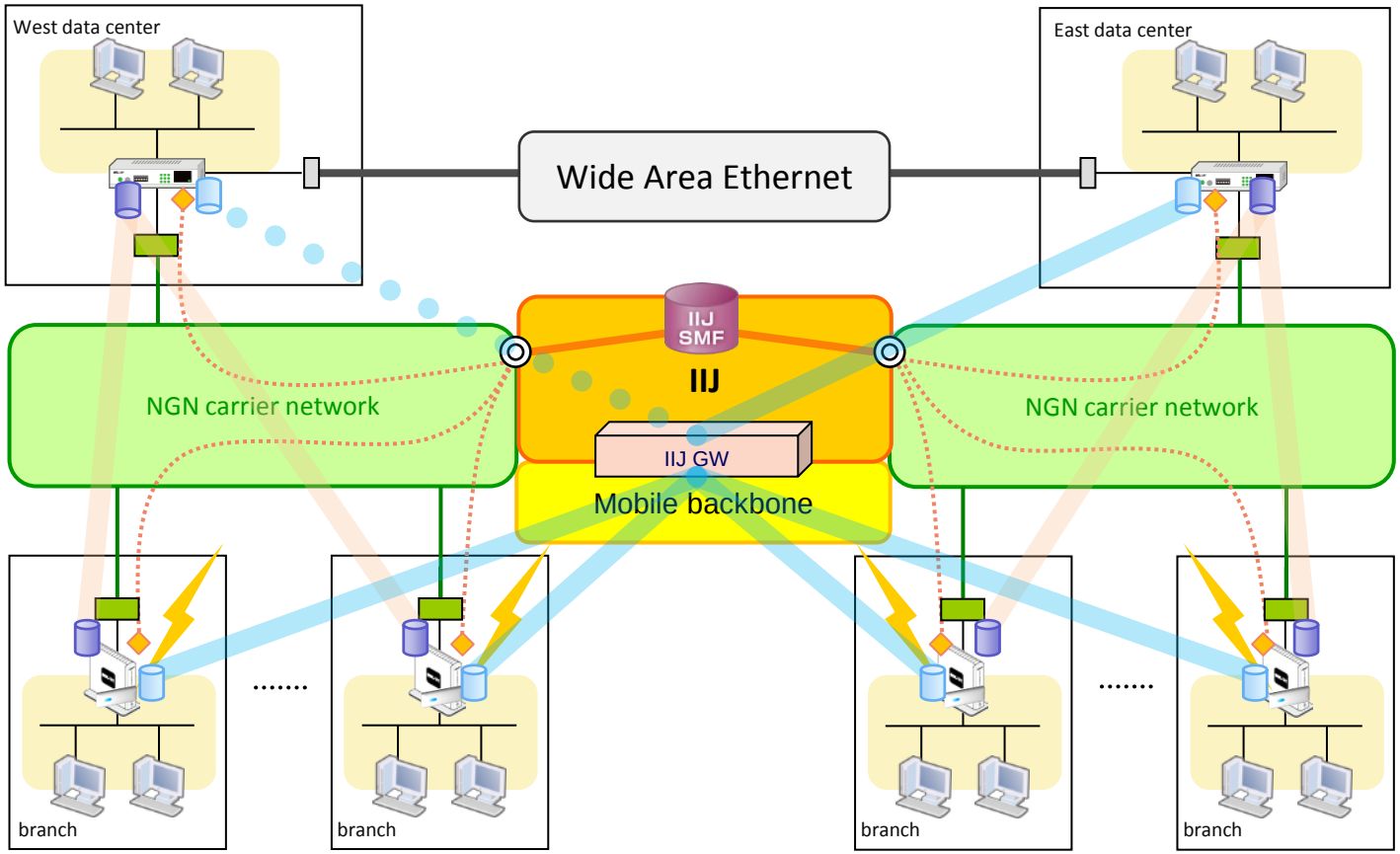
Image of management

- Manage different types of network in one system



An example network

- Disaster recovery
- Mobile backup



2. The difference between NetBSD and SEIL

Network devices

- Device drivers
  - ISDN(BRI)
  - 3G, LTE modem
    - Emulate serial port
    - Emulate Ethernet device
  - Ethernet switch
    - will be described in the next presentation

Daemon for P2P link

- connmgrd(8)
  - Connection manager daemon
  - Manipulate various type of P2P connections
    - ISDN
    - 3G, LTE modem
    - L2TP
  - Status, statistics

About link status

- Don’t enqueue outgoing packets if the link is down.
  - Because very old enqueued packet might cause a trouble after the link up
  - Not to waste mbuf resources

Pseudo devices

- IPsec tunnel interface
  - Routing-based IPsec
  - will be described in the next presentation
- Hyper-V driver
  - FreeBSD have another implementation.
  - At that time, FreeBSD's driver didn't exist.
  - Duplicated implementation
- L2TPv3
  - described later.

Extending IP networking stack

- iipf and iipfnat
  - IIJ original IP filter and NAT
  - Filter rule optimization
    - will be described in the next presentation
- IPsec
  - Added caching layer on Security Policy Database (SPD) and Security Association Database(SAD).
    - will be described in the next presentation

Cryptographic accelerator

- We have our own implementation
- The strategy is the same as opencrypt and FAST_IPSEC (originally implemented in OpenBSD)
  - Duplicated implementation
- Abandoned our implementation and switched to use NetBSD's.

Implementing new network protocols

- Tunneling protocols
  - PPTP
  - L2TP
- PIPEX
  - An in-kernel cut-through forwarding mechanism
  - Already merged to OpenBSD

L2TPv3

- A kind of Ethernet encapsulation and tunneling protocol described in RFC3931.
- The pseudo device acts as a kind of Ethernet device, and can be added to an Ethernet bridging group.
- Virtual Ethernet HUB
  - Multiple L2TPv3 interfaces can be added into one bridging group

MAP

- draft-ietf-softwire-map-xx
- One of experimental implementations of new Internet drafts.
- We are so interested in new protocols.

getifaddrs()

- Many pseudo network interface to provide various tunnel connections

- Add cache layer on getifaddrs()
- Add getiffaddrs_up() to get list of interfaces which link-state is up

sendfromto, recvfromto

- Problem
  - A router has many addresses (including alias addresses).
  - When a request received at an address, the reply packet's source address should be the dest address of the request packet.
  - Usually a deamon make as many number of sockets as the number of addresses.
    - This way will be complex if the number of addresses is big or an address changed.

sendfromto, recvfromto (2)

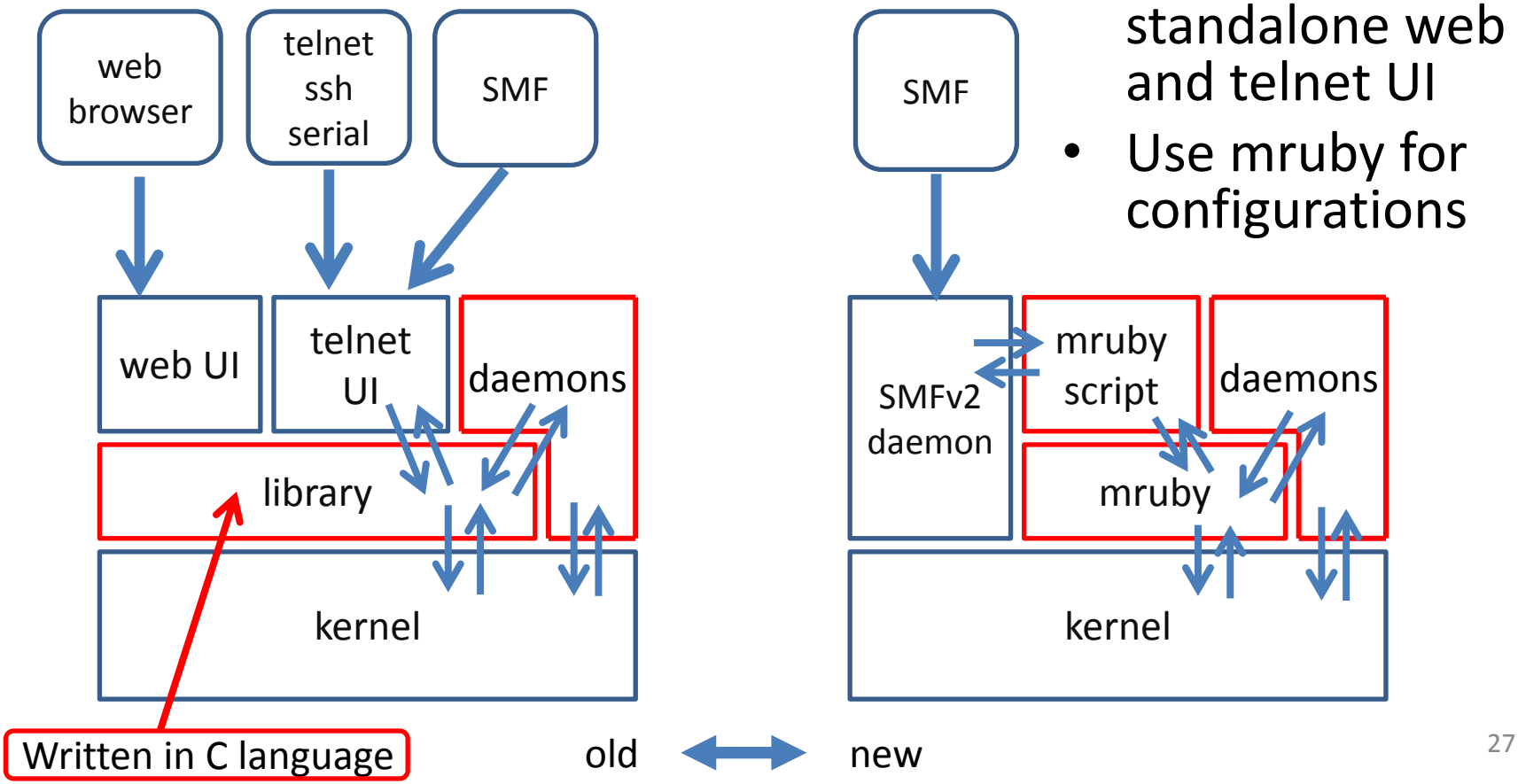
- Functions:
  - recvfromto(3)
    - like recvfrom(2). It can receive packet with destination address and destination port simultaneously
  - sendfromto(3)
    - like sendto(2). It can send packet with specified source address and source port simultaneously
- Implemented with setsockopt(IP_PKTINFO)
  - Add IP_PKTINFO
    - Linux and FreeBSD have it.
    - NetBSD and OpenBSD doesn't have it.

Manageability of UNIX like OS based CPE

- This is not Free UNIX like system but CPE, so the following things are important:
  - Automatic
    - Changes should be automatically propagated to other functions.
  - Easy to understand
    - without based OS's knowledge
    - what happened
    - whether an event is problem or not
  - Stable
    - Not panic. Strong against attack from others (e.g. DoS).

Automatic

- Changes are automatically propagated by …



- Abandoned standalone web and telnet UI
- Use mruby for configurations

Easy to understand(1)

- Rename interface names
  - For usability
    - e.g. wm0, bge0 -> lan0, lan1…
  - Some different implementations
    A) Use fixed table
    B) Automatically rename around if_init()
    C) Use ifconfig(8)
      - The latest CPE uses this way
      - “ifconfig mvgbe0 rename ge0”

Easy to understand(2)

- Log
  - Add new logs
    - For example, if a link goes down, send a log with LOG_CRIT.
  - Change log level if it's inappropriate
  - Modify log text if it's inappropriate

Easy to understand(3)

- For debugging
  - Our CPE has no storage device, so it doesn't dump a core. Instead, the kernel makes the userland program's information (include stack trace!) and records it into dmesg buffer.
  - The panic() function is extended and it records some inportant information into dmesg buffer.

stable(1)

- Heavy Ethenet rx interrupt and MCLGETI()
  - We have made a few implementation to avoid live lock so far.
    - Stop rx interrupt when live lock is detected.
      - It's little difficult when interrupt will be enabled again.
    - How to detect live lock
      - watermark in protocol queues
      - CPU utilization
  - It was very difficult, so I switched to use OpenBSD's MCLGETI().

3. New product development

Usual workflow(1)

1. Create plain new port of NetBSD
   – e.g. copy arch/evbarm to arch/seil6 and modify it.
2. Add new device driver if it's needed.
3. Run as plain NetBSD machine and make it stable.
4. Create customized ramdisk of the product.
5. Launch an NTP daemon and check the clock jitter and drift.

Usual workflow(2)

6. Check if dmesg buffer isn’t cleared after reboot.
7. Send/receive various size of Ethernet frames to find bugs. Frame with vlan tag often reveals MTU handling problem of the Ethernet driver.
8. Check counters. If an value isn’t visible, add it. If an counter is not incremented on some cases, fix it.
9. Throttling log. Some logs might be frequently generated.

Usual workflow(3)

10. Measure primitive performance (e.g. CPU(INT, FLOAT), memory, cryptographic, systemcall). Tune up if it can.
11. Measure total performance. Tune up if it can.
    - Bridge
    - IP forwarding
    - Filter, nat
    - IPsec
    - etc.

Usual workflow(4)

12. Do tests.
13. Release!

Changing NetBSD's base version(1)

- The base version of NetBSD is always release branch, not –current.
- Sometimes we upgrade NetBSD's base version when we make a new product.
- The frequency is very low. For example, we currently use the following two branches:
  - netbsd-3
  - netbsd-6

Changing NetBSD's base version(2)

- Upgrading is very heavy work
  - Check the difference between old branch and new branch.
  - Make patches and merge into new branch.
  - Sometimes the same function that we wrote was added into NetBSD. We have to choose one of them.

conclusion

Problems(1)

- We can’t feedback to NetBSD well
  - We are almost always busy.
  - Language problem
    - Japanese!
    - Sometime discussion is required.
  - Some people are shy :-)

Problems(2)

- On some cases, it’s difficult to feedback codes because we are developing software not based on –current but based on release branch (e.g. netbsd-3, netbsd-6)
- Sometimes other people develop the same function. (The same function in different implementation)

Solutions(1)

- Ideas
  1. Increase number of NetBSD developers in IIJ.
  2. Make a collaboration space outside of IIJ.
  3. Develop new function based on –current first if we can.

Solutions(2)

- Yet another cvs2git
  - We tried some tools but all of them don’t satisfy our requirement.
  - Our requirement is:
    - The following branches can be converted and synchronized correctly.
      - maintrunk
      - netbsd-3 (our products use it)
      - netbsd-6 (our product uses it)
      - rmind-smpnet

Current status

- ryo@n.o is working to make a tool to convert/sync from NetBSD's cvs repository to git.
  - The jobs will be finished in a few weeks.
  - When you won't see our tree in a few weeks, blame him ☺

Thank you.