

The FreeBSD Project

<http://www.freebsd.org/>



IPv6 in FreeBSD

Joseph Koshy

jkoshy@freebsd.org



Structure of this talk

- What is FreeBSD?
- The FreeBSD development model
- The KAME project
- IPv6 support in FreeBSD
- Configuring IPv6 in FreeBSD
- Available IPv6 capable application software.
- Tracking IPv6 development under FreeBSD
- Conformance and performance
- A look at IPv6 in operation; demos
- Conclusion



What is FreeBSD?

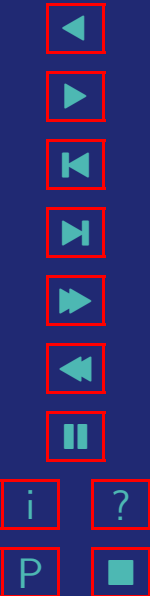
<http://www.freebsd.org/>

- A high-performance, volunteer developed, open source, protected mode operating system.
- University of California, Berkeley heritage (*BSD4.4-Lite*). BSD License.
- Released for Intel (x86) PCs, laptops and Alpha workstations.
- Most recent stable release: *FreeBSD v4.2*
- Linux, SVR4 (Solaris), SCO application compatibility.
- A large number of ported applications (over 4000).
- used in very high load scenarios (YAHOO, HOTMAIL, ...); excellent as a workstation OS.
- *Possesses an excellent IPv6 stack.*

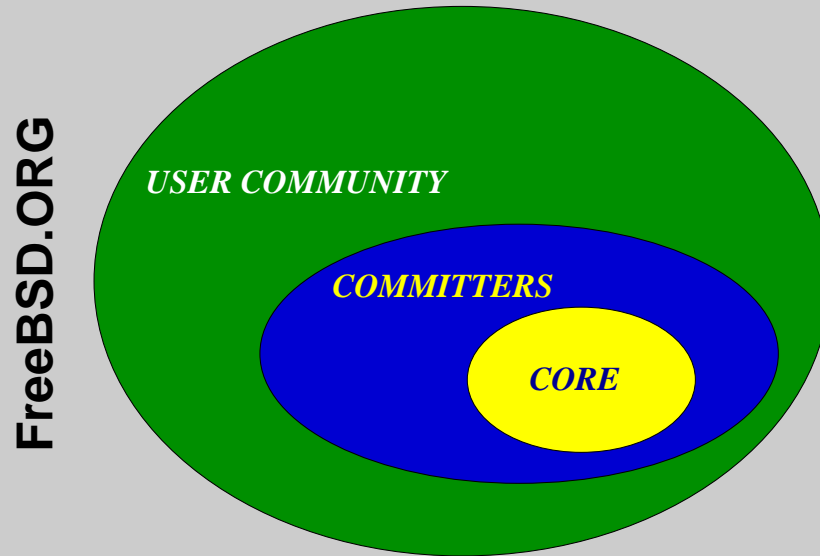


The BSD License

- very open to commercial use
- unlike the GPL, there is no requirement that you have to release your modified sources to the public
- only requires you to acknowledge authorship of the code
- no warranties on the code
- many companies are using the FreeBSD code base in their products
e.g. the Interjet from Whistle Inc. (an internet appliance), many router products ...



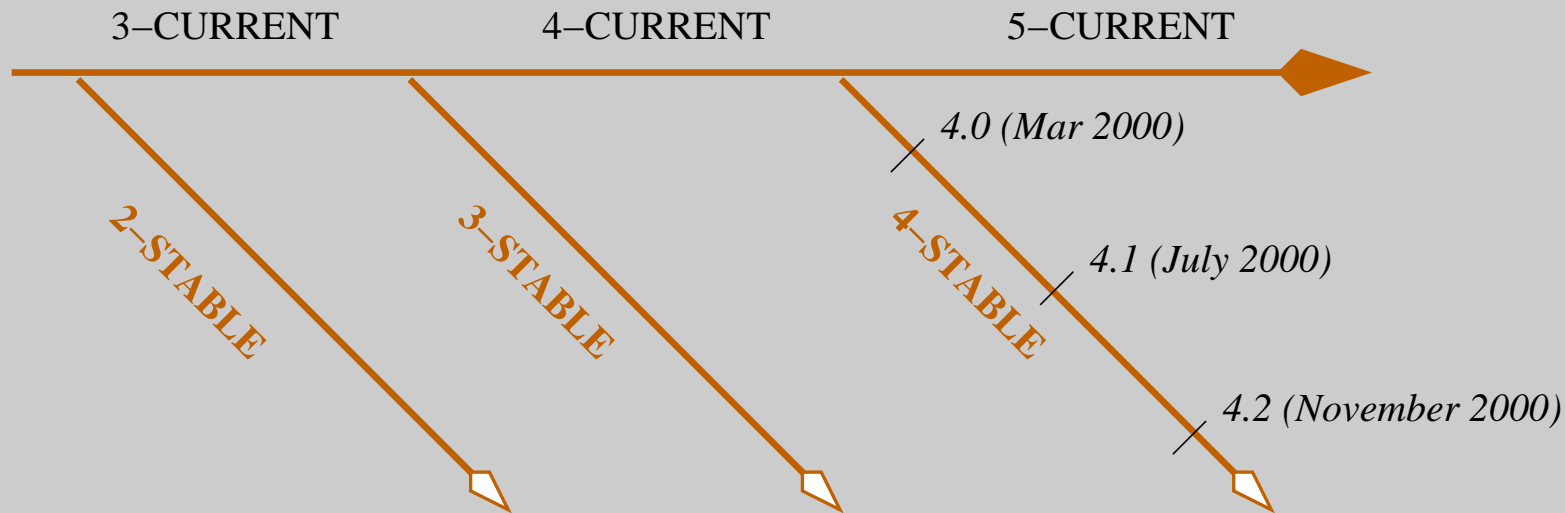
The FreeBSD Organization



- FreeBSD is developed and managed by its users.
- 3-level organization:
 - Over 75% of changes to the source tree originate from the FreeBSD user community.
 - COMMITTERS are individuals who have the rights to change the source tree.
 - CORE **core@freebsd.org** sets administrative policy; elected from COMMITTERS.



Development lines: -CURRENT vs -STABLE



- *One source base* for the entire working system: kernel, utilities, libraries
- two development lines **-CURRENT** and **-STABLE**
- new development happens on **-CURRENT** (*the bleeding edge!*)
- **-STABLE** lines are bug-fixed, release quality offerings
- the project follows excellent release engineering practices
- FreeBSD **-STABLE** has been IPv6 capable since 4.0 (March 2000)

End-users should use -STABLE!



The KAME Project



The KAME project (<http://www.kame.net/>).

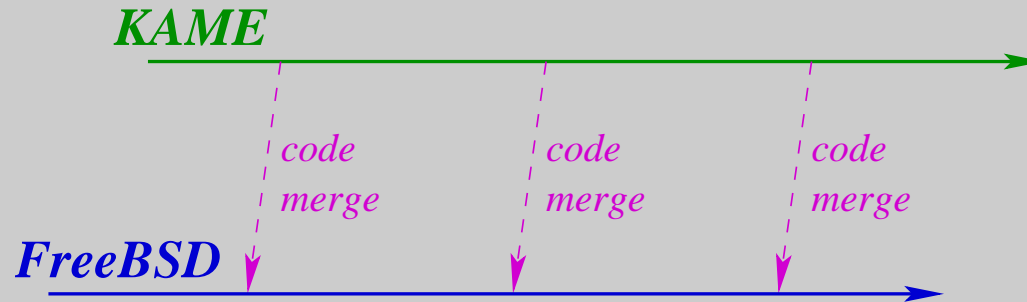
- The KAME project supports most of the BSD family of OSes: NetBSD, OpenBSD, FreeBSD, BSDI.
- The KAME project exports their code in terms of *patches* to the OSes released code base.
- The KAME project supports older releases of some OSes: e.g. FreeBSD v3.5, FreeBSD v2.2.8, and BSD v3.1.
- KAME code is released under a BSD style license.
- IPv6 in FreeBSD vs KAME?
 - The KAME code is upto-date and has experimental features.
 - KAME code in FreeBSD is better tested, more integrated.



IPv6 on FreeBSD



FreeBSD supports a *dual stack* (IPv4 + IPv6).



- FreeBSD IPv6 code tracks that of the KAME project.
- IPv4 code is a BSD 4.4 Lite based stack with enhancements and bug fixes.
- the core FreeBSD networking utilities (**telnet**, **ftp**) support IPv6.
- the default FreeBSD 4-STABLE install is IPv6 enabled.
- many 3rd party applications (eg:- **mpg123**, **ssh**) have been IPv6 enabled by the FreeBSD Ports team and the KAME project working together.



Application Ports

FreeBSD supports a sophisticated way to build third party applications (from source) on your local machine.

- automates the process of fetching and building an application from source:

```
ob47191# pwd
/home/ports/net/mtr
ob47191# ls -CF
CVS/ distinfo pkg-comment pkg-plist
Makefile files/ pkg-descr
ob47191# make
>> mtr-0.42.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://ftp.bitwizard.nl/mtr/.
Receiving mtr-0.42.tar.gz (84767 bytes): 100%
84767 bytes transferred in 20.7 seconds (4.01 kBps)
>> mtr-042-v6-20000719.diff.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://ftp.kame.net/pub/kame/misc/.
Receiving mtr-042-v6-20000719.diff.gz (23526 bytes): 100%
23526 bytes transferred in 5.9 seconds (3.86 kBps)
====> Extracting for mtr-gtk-0.42
>> Checksum OK for mtr-0.42.tar.gz.
>> Checksum OK for mtr-042-v6-20000719.diff.gz.
====> mtr-gtk-0.42 depends on executable: gmake - found
====> mtr-gtk-0.42 depends on shared library: X11.6 - found
...
====> mtr-gtk-0.42 depends on shared library: gtk12.2 - found
====> Patching for mtr-gtk-0.42
====> Applying distribution patches for mtr-gtk-0.42
```



Ports (contd)

```
====> Applying FreeBSD patches for mtr-gtk-0.42
====> Configuring for mtr-gtk-0.42
...
the binary gets built
====> Installing for mtr-gtk-0.42
...
install -c -s -m 4755 -o root -g wheel /home/ports/net/mtr/work/mtr-0.42/mtr /usr/local/sbin
...
====> Registering installation for mtr-gtk-0.42
====> SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
195718  122 -rwsr-xr-x 1 root wheel 61680 Jan 3 10:59 /usr/local/sbin/mtr
...
```

- the ports mechanism can combine IPv6 patches from the KAME project with other patches needed to otherwise compile the code.
- Ports are classified according to functionality: shells, editors, devel, lang, WWW, ...
- currently there are 50+ applications that use IPv6. These include nearly every popular open source network capable application (e.g. **emacs**).
- many network analysis tools (**tcpdump**, **mtr**, ...) are already IPv6 enabled.



IPv6 Feature Support Status



As of Dec 2000:

<i>Feature</i>	<i>Status in -STABLE</i>	<i>Status in -CURRENT</i>
KAME Source base	Early Jun '00	Early Jul '00
KAME IPv4 IPsec	Yes	Yes
KAME IPv6 IPsec	Yes	Yes
Advanced API	RFC2292	RFC 2292
NDP Support	Yes	Yes
NUD on P2P links	Yes	Yes
V6 NFS	No	No
V6 RPC	No	No



IPv6 Feature Support (contd)



Tools and utilities:

<i>Feature</i>	<i>in -STABLE</i>	<i>in -CURRENT</i>	<i>Comments</i>
V6 Userland	Yes	Yes	telnet, ftp etc.
route6d	Yes	Yes	RIP6 routing
bgpd	No	No	Only in KAME sourcebase
pim6dd/pim6sd	Yes	Yes	Multicasting
rtsol	Yes	Yes	Router solicitation
rtadvd	Yes	Yes	Route advertisement
rrrenumd	Yes	Yes	Router renumbering
faithd	Yes	Yes	V6 to V4 TCP relay



Tracking IPv6 on FreeBSD

- KAME code keeps evolving, you may want to keep upto-date.
- *option*: track FreeBSD-STABLE (or even -CURRENT)
 - retrieve sources over the Internet using **AnonCVS** or **cvsup** (5mts daily).
 - build the “world” from source (500+ MB disk space, 4hrs).
 - will see stable, tested IPv6 code.
- *option*: mirror the CVS repository locally
 - **cvsup** does this very efficiently.
 - requires 1GB+ space.
 - fast access to full history of the project in the CVS logs.
- *option*: use a released version of FreeBSD with the KAME patch kit.



Tracking FreeBSD Development (contd)



- Sources for the base OS are kept in a publically available (CVS) repository. This repository contains the *entire* history of the project since its inception.
- The repository is efficiently mirrored using John Polstra's **cvsup** tool.
- Another tool, **CTM**, allows tracking of the sources using email.
- Once you have the sources, the *whole system* can be built with a single command:

```
# cd /usr/src  
# make world
```

This takes approximately 1.5–12 hours depending on your machine speed.

- Nearly every aspect of the project is accessible to the public; discussion lists, bug reports, security advisories ...



Conformance

- RFC compliance:

1886/2671 (DNSv6), 1981 (Path MTU), 2080 (RIPng), 2292 (Advanced API), 2373 (Addressing Arch.), 2374/2375/2710 (Multicasting), 2428(FTP Extensions), 2460 (IPv6), 2461 (Neighbour Discovery), 2462 (Stateless Autoconfig), 2463 (ICMPv6), 2464 (v6/Enet), 2467 (v6/FDDI), 2472 (v6/PPP), 2675 (Jumbograms), 2732 (literal address format), 2766 (NAT), 2894 (Router renumbering), and others ...

- the code also implements a few draft standards related to tunnelling, potential abuse issues during transition etc.
- the IMPLEMENTATION document distributed by the KAME project has the complete details of conformance.
- conformance testing is being done by the Tahi project:

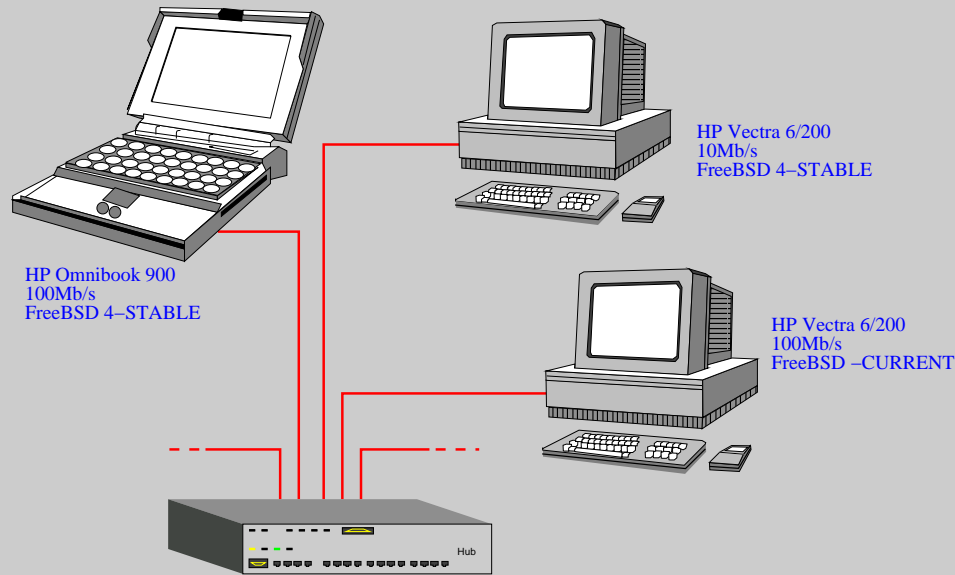
<http://www.tahi.org/>



Performance



Some very ad-hoc performance testing, just for this presentation:



- timed the transfer of a large file using **ftp -6** and **ftp -4**
- Conclusion: IPv6 appears to be as fast as IPv4 (1.1MB/s).
- The Tahi Project (<http://www.tahi.org/>) has more authentic performance test reports.



Non x86 architectures?



What if you are looking at bringing IPv6 to a machine that is not running an x86 family processor?

- FreeBSD runs on Alpha processor based machines
- *NetBSD* <http://www.netbsd.org/>
- *OpenBSD* <http://www.openbsd.org/>
- NetBSD is perhaps the most portable open source OS today
- OpenBSD has their own implementation of IPsec
- KAME IPv6 code is used in {Open,Net}BSD projects
- The developer community for these projects is active, but is not as large as that of FreeBSD.



Configuring IPv6



A simple 3-step procedure:

- build and install a kernel with IPv6 capabilities (default **GENERIC** kernel is already IPv6 capable).
- turn on the knobs in `/etc/rc.conf`, setup `/etc/hosts` ...
- reboot

Building and Installing the kernel

- modify the existing kernel configuration file
(copy `/usr/src/sys/i386/conf/GENERIC` and edit to suit)
- add IPv6 related options
- build and install



IPv6 Kernel Configuration Options



Edit your kernel configuration file:

/usr/src/sys/i386/conf/OB47191

```
...
options      INET          #InterNETworking
options      INET6        #IPv6 communications protocols
options      IPSEC        #IP security
options      FFS          #Berkeley Fast Filesystem
...
pseudo-device gif 4      # IPv6 and IPv4 tunneling
pseudo-device faith 1    # IPv6-to-IPv4 relaying (translation)
pseudo-device stf 1      # 6to4 IPv6 over IPv4 encapsulation
...
pseudo-device bpf        #Berkeley packet filter
```



Building and Installing the new kernel

```
# cd /usr/src
# make buildkernel KERNEL=OB47191
... build output deleted
# make installkernel KERNEL=OB47191
...
```

- a simple 2 step procedure:
- the new kernel will be installed as **/kernel** on 4-STABLE machines.
- **/usr/src/sys/i386/conf/LINT** lists all the supported kernel options.



Edit /etc/rc.conf

```
ipv6_enable="YES" # Set to YES to set up for IPv6.
...
ipv6_defaultrouter="NO" # Set to IPv6 default gateway
ipv6_gateway_enable="NO" # host will be a gateway?
ipv6_router_enable="NO" # run the IPv6 routing daemon?
rtadvd_enable="YES" # enable IPv6 router adv.
mroute6d_enable="NO" # Do IPv6 multicast routing?
gif_interfaces="gif0 gif1" # Examples
stf_interface_ipv4addr="" # 6to4 IPv6 over IPv4
ipv6_firewall_enable="NO" # enable IPv6 firewall
...
```

Setup /etc/hosts

```
# IPV6 addresses
fe80::260:b0ff:fe57:fe42%xe0 pc45026-v6
fe80::280:c8ff:fe3d:f5e5%xe0 csypcnt3-v6
fe80::210:a4ff:fe07:2d2c%xe0 ob47191-v6
```



Running IPv6



Check your **ifconfig** output:

```
$ ifconfig xe0
xe0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
  inet 15.10.47.191 netmask 0xfffff800 broadcast 15.10.47.255
  inet6 fe80::210:a4ff:fe07:2d2c%xe0 prefixlen 64 scopeid 0x7
  ether 00:10:a4:07:2d:2c
  media: autoselect (100baseTX)
  supported media: autoselect 10baseT/UTP 100baseTX

$ ifconfig lo0
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
  inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
  inet6 ::1 prefixlen 128
  inet 127.0.0.1 netmask 0xff000000

$ ifconfig gif0
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
```



What applications are available?



50+ network applications are IPv6 ready today. Some examples:

<i>Application</i>	<i>Description</i>
apache+ipv6-1.3.14	The extremely popular Apache http server.
bind-9.0.1	The Berkeley Internet Name Daemon
emacs-20.7	GNU editing macros
ethereal-0.8.14	An X11/GTK network analyzer/capture tool
fetchmail-5.6.1	Mail retrieval utility for pop2, pop3, apop, imap
irc-2.10.3	The 'Internet Relay Chat' Server
mozilla+ipv6-M18	The mozilla ver 0.0 communicator web-surfboard
sendmail-8.11.1	Reliable, highly configurable mail transfer agent
ssh-1.2.27	Secure shell client and server (remote login program)
zebra-0.89a	RIPv1, RIPv2, OSPFv2, BGP4 route software



IPv6 in operation: Network traces



```
# tcpdump -n -e -f -i de0 ip6
```

```
11:08:57.048493 0:80:c8:3d:f5:e5 0:10:a4:7:2d:2c 86dd 94:  
fe80::280:c8ff:fe3d:f5e5.1005 > fe80::210:a4ff:fe07:2d2c.22:  
S 1201176387:1201176387(0) win 16384 <mss 1440,nop,wscale 0,  
nop,nop,timestamp 59810721 0>
```

```
11:08:57.788120 0:10:a4:7:2d:2c 33:33:ff:3d:f5:e5 86dd 86:  
fe80::210:a4ff:fe07:2d2c > ff02::1:ff3d:f5e5:  
icmp6: neighbor sol: who has fe80::280:c8ff:fe3d:f5e5
```

```
11:08:57.788288 0:80:c8:3d:f5:e5 0:10:a4:7:2d:2c 86dd 86:  
fe80::280:c8ff:fe3d:f5e5 > fe80::210:a4ff:fe07:2d2c:  
icmp6: neighbor adv: tgt is fe80::280:c8ff:fe3d:f5e5
```

```
11:08:57.788703 0:10:a4:7:2d:2c 0:80:c8:3d:f5:e5 86dd 78:  
fe80::210:a4ff:fe07:2d2c.22 > fe80::280:c8ff:fe3d:f5e5.1005:  
S 1234536047:1234536047(0) ack 1201176388 win 17280 <mss 1440>  
[flowlabel 0x60a5c]
```



Demos: FTP and Ping



Ping:

```
(orthanc) 16:08 ~ % ping6 ob47191-v6
PING6(56=40+8+8 bytes) fe80::280:c8ff:fe3d:f5e5%de0 -->
    fe80::210:a4ff:fe07:2d2c%de0
16 bytes from fe80::210:a4ff:fe07:2d2c%de0, icmp_seq=0 hlim=64 time=0.62 ms
16 bytes from fe80::210:a4ff:fe07:2d2c%de0, icmp_seq=1 hlim=64 time=0.54 ms
^C
--- ob47191-v6 ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.54/0.58/0.62 ms
```

FTP:

```
(orthanc) 12:36 ~ % ftp -6 fe80::210:a4ff:fe07:2d2c%de0
Connected to fe80::210:a4ff:fe07:2d2c%de0.
220 ob47191 FTP server (Version 6.00LS) ready.
Name (fe80::210:a4ff:fe07:2d2c%de0:jkoshy): jkoshy
331 Password required for jkoshy.
Password: *****
... rest of the session
```



Demo: A Web Server / Web Client running IPv6

- Install port `apache+ipv6` or `thttpd` for the server.
- IPv6 capable WWW clients: `w3m`, `lynx`, `mozilla`, ...
- Capture and analyse traffic using `ethereal`.



Getting Started with FreeBSD

- Recommended: at least an i486 with 8MB RAM, 0.5GB Disk Space.
- Download and install over the Internet (requires good connectivity)
- 4 CD packs from Walnut Creek Inc. Cheaper 1 CD version from CheapBytes Inc. Installation via CD takes about 15 minutes.
- Other installation options: floppies, NFS, DOS partition ...
- Standard Unix administration knowledge will see you through most of your needs.
- The FreeBSD Handbook (<http://www.freebsd.org/handbook/>) is a good resource.
- Subscribe to the mailing lists (**freebsd-newbies**, **freebsd-questions**, **freebsd-hackers**, **freebsd-current**, **freebsd-stable**, ...)



Conclusion

FreeBSD is an excellent way to get started with IPv6.

- runs on commodity PCs
- supports a dual-stack (IPv4/IPv6)
- has stable code, good performance
- has an upto date IPv6 stack and utilities
- has a commercialization friendly BSD license
- has an active developer community
- provides tools to ease to tracking ongoing development

<http://www.freebsd.org/>

