Implementing an IPv6 and Mobile IPv6 testbed using FreeBSD 4.9 and KAME

Lawrence Stewart, Mai Banh, Grenville Armitage
Centre for Advanced Internet Architectures. Technical Report 040331A
Swinburne University of Technology
Melbourne, Australia
{lastewart,mbanh,garmitage}@swin.edu.au

***Abstract*-This technical report explains and documents the process of implementing a mobile IPv6 testbed using VIA EPIA-V Mini-ITX based machines running FreeBSD 4.9 and the development code from the KAME project. The steps taken to verify the functionality of the testbed have also been documented.**

Keywords- IPv6, Mobile IPv6, KAME, FreeBSD, Mini-ITX

I. Introduction

The exponential growth of the Internet in recent years has posed a number of problems for engineers to solve. Every node that expects to be publicly visible and reachable on the Internet must have an IP (Internet Protocol) address that is unique among all the other publicly visible and reachable nodes on the Internet [1]. When IP was conceived and developed in the late 70s, the designers allocated 32 bits to represent an IP address, which translates to just under 4.3 billion addresses. No one expected the Internet to gain as much popularity as it has, and 32 bits was therefore thought to be far more than necessary at the time of inception.

As a result of IPv4's hierarchical address allocation scheme (subdividing the IPv4 address space into networks and subnetworks) and phenomenal growth in popularity and scope of the Internet [2], the Internet Engineering Task Force (IETF) began work on a successor to IPv4 in the early 1990s - which became IP version 6 (IPv6) [3][4]. Although IPv6 retains the same hierarchical address assignment approach, its expanded 128bit address fields provide substantial breathing room in the face of further growth in the numbers of Internet-attached devices.

Accompanying the Internet boom has been a huge increase in the demand for mobile communications and services. Mobile phones, for example, are now not only used to make calls, but send messages, pictures, video content and even check email.

The inclusion of support for Mobile IP in the base IPv6 specification and development effort is another improvement [5]. Mobile IPv6 grew out of experiences with Mobile IPv4 [6], itself an attempt to enable IP-attached devices to migrate between physical networks without having to change the publicly visible IP address by which they were uniquely known to the rest of the Internet.

The potential for mobile IPv6 in applications such as 4th generation mobile phones and other mobile devices is huge, and as a result of this, a large number of companies are putting money into research and development. The KAME project [10], which is the source of the mobile IPv6 code used to build our testbed, is a joint venture between 6 different companies in Japan.

We found the setting up of an experimental mobile IPv6 testbed to be quite challenging. However, with our documented steps, some basic FreeBSD knowledge and some hardware, you too can have your own testbed running in a couple of hours.

II. Testbed Layout

Figure 1 shows the physical layout for our testbed and the interface/address information for each host.

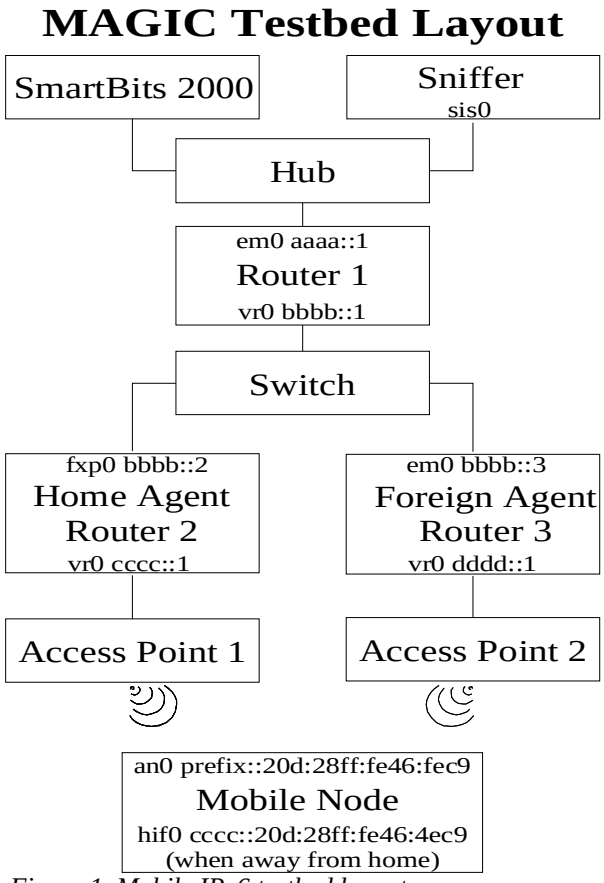

Figure 1. Mobile IPv6 testbed layout

III. Testbed Hardware and Software

VIA Mini-ITX based systems were used for the various devices in the testbed and consisted of:

- VIA EPIA-V series Mini-ITX motherboard [7]
- VIA 800MHz C3 E-Series processor
- On board 10/100 NIC (VT6102 chipset)
- 80 Gig Seagate Barracuda 7200 RPM HDD
- 256 Meg PC133 RAM

Router 1, 2 and 3 had an extra NIC added to their free PCI slot which was either an Intel 1000MT (82540em chipset) or an Intel 100/Pro (82559 chipset).

The mobile node had a Cisco Aironet 350 Series PCMCIA wireless network card in a PCI to PCMCIA cradle inserted into its free PCI port.

Two Cisco Aironet 1200 series wireless access points were used in the testbed.

The switch used was an Alloy NS05C 5 port Nway Mini Switch [8].

All NICs (excluding the wireless adapter) were explicitly configured to run at 10Mbps in full duplex mode.

All systems ran FreeBSD 4.9 [9] as the operating system. To provide the mobile IPv6 features, we used the KAME [10] code snap kit dated 22/03/2004 for FreeBSD 4.x series [11].

It should be noted that the KAME software is still under heavy development and often has features that are not completely implemented, buggy or being changed from snap kit to snap kit. The KAME newsletter [12] and mailing lists [13] can provide some help and information.

IV. Configuration

A. The KAME snap kit

Extracting the KAME snap kit from the archive results in the creation of the top level directory “kame” in which all snap kit related files are placed. The “INSTALL” file within this directory is of most interest, as it outlines the necessary steps that need to be taken to prepare the system.

As we were building the code for a new installation of FreeBSD 4.9, the only command we needed to issue was “make TARGET=freebsd4 prepare”.

This creates all the necessary symbolic links to build a kernel from the FreeBSD and KAME source trees.

The newly created directory named “freebsd4” contains an “INSTALL” file that walks you though the rest of the process, which includes rebuilding the FreeBSD kernel and building the KAME userland applications.

B. FreeBSD 4.9 kernel

We rebuilt the FreeBSD kernel using a modified version of the default GENERIC.KAME configuration file that came with the snap kit.

As the VIA processors used were i686 equivalents, we removed kernel support for i386, i486 and i586 processors, as it was suggested in the kernel configuration options file “LINT” that doing this would make some parts of the system run faster. We also added the kernel options “options MIP6”, “options MIP6_DEBUG”, “device acpica” and “options HZ=1000” for all machines.

The home agaent (HA) was compiled with the added option “options MIP6_HOME_AGENT”.

The mobile node (MN) was compiled with the added options “options MIP6_MOBILE_NODE” and “options hif 1”.

Interestingly, the KAME source code would automatically configure the IPv6 address of the hif0 interface to use the onboard NIC's MAC address instead of the Cisco Wireless Card's MAC address, even though the NIC had not been activated or configured. As a result of this finding, we had to remove kernel support for the on board VIA ethernet NIC on the MN. A loadable kernel module for the VIA NIC is provided in /modules/if_vr.ko and can be explicitly loaded using the “kldload” command after the boot sequence if support for the on board NIC is required. Following this procedure will eliminate the observed problem.

C. FreeBSD rc files

We backed up `/etc/rc.network6` and copied the KAME supplied version of this script and rc.mobileip6 from `/<path_to_unziped_kame_snapshot>/kame/freebsd4/etc` to `/etc`. The contents of the rc.conf files for the different machines are shown in Appendix 1. Note that adding the “`PATH=...`” statement is important to ensure that KAME binaries located in `/usr/local/v6/{bin, sbin}` are used in preference to any other system binaries with the same name.

We discovered two errors in the rc.mobileip6 file supplied as part of the snap kit. Line 101 reads: `prefix='expr “$home_prefix” : “\(.*\)::/.*”'`, where `home_prefix` contains the list of home prefixes given in the `rc.conf` option “`ipv6_mobile_home_prefixes=`”. This does not function as it was supposed to and needed to be changed to: `prefix='expr “$home_prefix” : “\(.*::\)”'`.

Line 103 reads: `ifconfig ${ipv6_mobile_home_link} inet6 ${prefix}:fdff:ffff:ffff:fffe prefixlen 64 anycast alias`

As `prefix` already contains the ::, the line should be changed to: `ifconfig ${ipv6_mobile_home_link} inet6 ${prefix}fdff:ffff:ffff:fffe prefixlen 64 anycast alias`.

D. The Router Advertisement Daemon: rtadvd

Rtadvd is responsible for sending out router advertisements to aid in the automatic address configuration processes implemented as part of IPv6. Rtadvd sends router advertisements at 200-600 second intervals by default. For a wired, fairly static LA,N this

is completely acceptable. However, mobile nodes use router advertisements to realise that they have moved from one network to another. As such, the speed with which they get advertisements will affect the amount of time a mobile node is unreachable because of expired routing information.

As a result of this, we decreased the router advertisement time period to 4-6 seconds as recommended in the KAME newsletter dated 20031007 [14]. We created the file `/usr/local/etc/rtadvd.conf` and added the lines:

```
vr0:\
        :maxinterval#6:mininterval#4:
```

The HA rc.mobileip6 needed line 107 which reads: `rtadvd -m ${ipv6_mobile_home_link}` changed to: `rtadvd -c /usr/local/etc/rtadvd.conf -m ${ipv6_mobile_home_link}`.

The FA required an executable script placed in `/usr/local/etc/rc.d/<scriptname>.sh` with the line: `rtadvd -c /usr/local/etc/rtadvd.conf -m vr0` added to it.

E. IPSec

Initially, we attempted to use the IP Security protocol for HA to MN tunneling and vice versa. It functioned correctly where conversation between the HA and MN was concerned. However, when an outside source attempted to ping the MN when it was not at home, the echo reply would arrive from the MN to the HA in encrypted form and then be sent to the corresponding node with the encryption still in place. This resulted in the corresponding node being unable to "see" the echo reply and therefore resulted in unsuccessful pinging.

Owing to time considerations, we did not pursue the cause of this behaviour. We suspect that IPSec needed to be configured on the corresponding node, as IPSec is supposed to be used end to end which means traffic from the MN to the CN should be encrypted as well as traffic between the HA and MN. However, we will briefly document the steps involved in getting the IPSec tunneling working, which came from a KAME newsletter article [14].

After creating the directories `/usr/local/v6/etc/mobileip6` and `/usr/local/v6/mobileip6/<node_name>`, create a file named config in the `/usr/local/v6/mobileip6/<node_name>/` directory. Appendix 2 includes a copy of our config file. Note that the hmac-sha1 authentication algorithm uses a 160 bit keylength i.e. 20 characters x 8 bits per character.

Issuing the command makeconfig.sh <node_name> should result in a number of files being created in the working directory.

Modifying the /etc/rc.conf file by replacing the line `ipv6_mobile_security_enable="NO"` with `ipv6_mobile_config_dir="/usr/local/v6/etc/mobileip6"` will result in the use of the generated IPSec files from the previous step.

F. Cisco Aironet 1200 Series Wireless Access Point

A Cisco serial cable and RJ-45 to DB9 connector were used to connect each AP to the serial port of a host machine on the network. By using the "`tip`" command e.g. `tip com1`, we were able to access the console interface of the AP and configure it from our FreeBSD host. The output of the `show run` command from each of the APs is given in Appendix 3.

V. Verifying the Testbed

There are a number of KAME userland utilities that are provided to configure and examine the state of IPv6 and mobile IPv6. These tools are located in `/usr/local/v6/bin` and `/usr/local/v6/sbin`.

A. Mip6control

This tool resides in `/usr/local/v6/sbin` and is probably the most useful for examining what the mobile IP code is actually doing. "`man mip6control`" will list the available command line options (although we found one option listed in the man page to have been removed from the binary, which goes to show that the documentation is often behind the code). For verifying mobile node functions, the -a and -b options show the home/foreign agents list and binding update list respectively.

For verifying the functions of all other nodes, the -c option shows the binding cache list.

B. Mip6stat

This tool resides in `/usr/local/v6/sbin` and is useful to simply verify that mobility related packets are being sent and received by the node in question.

C. Ping6

This tool resides in `/usr/local/v6/sbin` and can be used to test the end to end network connectivity between two IPv6 hosts.

D. Ifconfig

This tool resides in `/usr/local/v6/sbin` and is used to view/modify the local network interface configuration settings.

E. Useful tests

The most obvious question is how to trigger a mobility event to occur, such as the MN moving from the home network to the foreign network. We found the simplest way to do this was by issuing the command: `ifconfig an0 ssid "magicap2"` on the mobile node. This resulted in a binding update event occurring once a router solicitation from the foreign router had been received.

Once the transition to the new network has been made, issuing the command `ifconfig` on the mobile node shows 2 changes from when the node was on the home network. `Hif0` has the mobile node's home IPv6 address (that `an0` had when the mobile node was at home) and `an0` now has the same host address as before but with the prefix of the foreign network. `Mip6control -a` will now show an entry for the home agent marked

"Home" and an entry for the foreign network marked "Foreign". `Mip6control -b` shows a binding update entry for the address of the home agent.

`Mip6control -c` on the home agent lists a binding cache entry for the mobile node's home address with CoA address being the new address of `an0` on the mobile node.

Running `mip6stat` on either node will display a summary list of the mobility headers being exchanged between the nodes.

Pinging from a non mobile IPv6 aware node to the mobile node while it is at home should be successful. Whilst still pinging, change the mobile node over to the foreign network and a break in ping activity for a number of seconds will be observed while the mobile node reconfigures itself, followed by the resumption of ping activity.

`Tcpdump` [15] is also extremely useful for observing the traffic flowing through an interface. `Tcpdump -i <interface> ip6` will display only IPv6 traffic, which is obviously desired for our purposes.

VI. Conclusion

By installing FreeBSD 4.9 and the KAME project's code snap kit dated 22/03/04 on 4 VIA EPIA-V Mini-ITX based machines, we were able to construct a functional, IPv6 and mobile IPv6 testbed. We were able to trigger mobile IPv6 hand over events by switching the ssid that the MN was associated with, which resulted in the MN being placed on a foreign network. By tuning the router advertisement behaviour of both the HA and FA, we were able to improve the speed with which hand over events occurred, thus reducing the amount of time the mobile node was unreachable. We were able to document the steps taken to create the working testbed and verify it worked correctly. The heavy development of the KAME project code tree means some of the steps outlined in this document may not be completely relevant if you are using a much newer or older snap kit. Always read the documentation (namely the INSTALL files in the snap kit) to get the most up to date information.

References

[1] Postel, J., Editor, "Internet Protocol DARPA Internet Program Protocol Specification." RFC 791, Internet Engineering Task Force, September 1981

[2] Fuller, V., Li, T., Yu, J., Varadhan, K., "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," RFC 1519, Internet Engineering Task Force, September 1993

[3] S. Bradner and A. Mankin, "The Recommendation for the IP Next Generation Protocol," RFC 2460, Internet Engineering Task Force, January 1995

[4] S. Deering and R. Hinden, "Internet Protocol, Version 6 (Ipv6) Specification," RFC 2460, Internet Engineering Task Force, December 1998

[5] D. Johnson,C. Perkins, J. Arkko "Mobility Support in IPv6." draft-ietf-mobileip-ipv6-14.txt, Internet Engineering Task Force, July 2003

[6] Perkins, C., "IP Mobility Support," RFC 3344, Internet Engineering Task Force, August 2002

[7] VIA EPIA V Mainboard, http://www.viavpsd.com/product/epia_v_spec.jsp?motherboardId=141 (as of March 2004)

[8] 10/100Mbps Nway Mini Switches by Alloy, http://www.alloy.com.au/products/ns05c.htm (as of March 2004)

[9] The FreeBSD Project, http://www.freebsd.org/ (as of March 2004)

[10] The KAME Project, http://kame.net/ (as of March 2004)

[11] KAME 22/03/04 code snap kit , ftp://ftp.kame.net/pub/kame/snap/kame-20040322-freebsd49-snap.tgz (as of March 2004)

[12] KAME Newsletter, http://kame.net/newsletter/ (as of March 2004)

[13] KAME-snap kit, http://www.kame.net/snap-users/ (as of March 2004)

[14] KAME Mobile Ipv6 How To, http://www.kame.net/newsletter/20031007/ (as of March 2004)

[15] Tcpdump, http://www.tcpdump.org/ (as of March 2004)

Appendix 1

rc.conf for Router 1:

```
hostname="magic1.caia.swin.edu.au"
ifconfig_vr0="media 10baseT/UTP mediaopt full-duplex"
ifconfig_em0="media 10baseT/UTP mediaopt full-duplex"
gateway_enable="YES"
kern_securelevel_enable="NO"
saver="logo"
linux_enable="YES"
moused_enable="YES"
nfs_reserved_port_only="YES"
sendmail_enable="NONE"
sshd_enable="YES"
usbd_enable="YES"
PATH=/usr/local/v6/sbin:/usr/local/v6/bin:${PATH}
ipv6_enable="YES"
ipv6_gateway_enable="YES"
ipv6_network_interfaces="vr0 em0"
ipv6_ifconfig_em0_alias0="aaaa::1 prefixlen 64"
ipv6_ifconfig_vr0_alias0="bbbb::1 prefixlen 64"
ipv6_static_routes="1 2"
ipv6_route_1="cccc:: -prefixlen 64 bbbb::2"
ipv6_route_2="dddd:: -prefixlen 64 bbbb::3"
ipv6_mobile_enable="YES"
ipv6_mobile_debug_enable="YES"
ipv6_mobile_security_enable="NO"
```

rc.conf for HA:

```
gateway_enable="YES"
hostname="magic2.caia.swin.edu.au"
ifconfig_vr0="media 10baseT/UTP mediaopt full-duplex"
```

```
   ifconfig_fxp0="media 10baseT/UTP mediaopt full-
duplex"
   saver="logo"
   kern_securelevel_enable="NO"
   linux_enable="YES"
   moused_enable="YES"
   nfs_reserved_port_only="YES"
   sendmail_enable="NONE"
   sshd_enable="YES"
   usbd_enable="YES"
   PATH=/usr/local/v6/sbin:/usr/local/v6/bin:${PATH}
   ipv6_enable="YES"
   ipv6_gateway_enable="YES"
   ipv6_network_interfaces="fxp0 vr0"
   ipv6_ifconfig_fxp0_alias0="bbbb::2 prefixlen 64"
   ipv6_ifconfig_vr0_alias0="cccc::1 prefixlen 64"
   ipv6_static_routes="1 2"
   ipv6_route_1="dddd:: -prefixlen 64 bbbb::3"
   ipv6_route_2="aaaa:: -prefixlen 64 bbbb::1"
   ipv6_mobile_enable="YES"
   ipv6_mobile_debug_enable="YES"
   ipv6_mobile_security_enable="NO"
   ipv6_mobile_nodetype="HOME_AGENT"
   ipv6_mobile_home_link="vr0"
   ipv6_mobile_home_prefixes="cccc::/64"
```

rc.conf for FA:

```
   gateway_enable="YES"
   hostname="magic3.caia.swin.edu.au"
   ifconfig_vr0="media 10baseT/UTP mediaopt full-
duplex"
   ifconfig_em0="media 10baseT/UTP mediaopt full-
duplex"
   saver="logo"
   kern_securelevel_enable="NO"
   linux_enable="YES"
   moused_enable="YES"
   nfs_reserved_port_only="YES"
   sendmail_enable="NONE"
   sshd_enable="YES"
   usbd_enable="YES"
   PATH=/usr/local/v6/sbin:/usr/local/v6/bin:${PATH}
   ipv6_enable="YES"
   ipv6_gateway_enable="YES"
   ipv6_network_interfaces="em0 vr0"
   ipv6_static_routes="1 2"
   ipv6_route_1="aaaa:: -prefixlen 64 bbbb::1"
   ipv6_route_2="cccc:: -prefixlen 64 bbbb::2"
   ipv6_ifconfig_em0_alias0="bbbb::3 prefixlen 64"
   ipv6_ifconfig_vr0_alias0="dddd::1 prefixlen 64"
   ipv6_mobile_enable="YES"
   ipv6_mobile_debug_enable="YES"
   ipv6_mobile_security_enable="NO"
```

rc.conf for MN:

```
   hostname="magic4.caia.swin.edu.au"
   kern_securelevel_enable="NO"
   saver="logo"
   linux_enable="YES"
   moused_enable="YES"
   nfs_reserved_port_only="YES"
   sendmail_enable="NONE"
   sshd_enable="YES"
   usbd_enable="YES"
   PATH=/usr/local/v6/sbin:/usr/local/v6/bin:${PATH}
   pccard_enable="YES"
   ipv6_enable="YES"
   ipv6_network_interfaces="an0 hif0"
   pccard_ifconfig="ssid magicap1"
   ifconfig_hif0="up"
   ipv6_mobile_enable="YES"
   ipv6_mobile_debug_enable="YES"
   ipv6_mobile_security_enable="NO"
   ipv6_mobile_nodetype="MOBILE_NODE"
   ipv6_mobile_home_prefixes="cccc::/64"
```

Appendix 2

Contents of /usr/local/v6/etc/mobileip6/HA1/config:

```
   mobile_node="cccc::28ff:fe46:4ec9"
   home_agent="cccc::1"
   transport_spi_mn_to_ha=2000
   transport_spi_ha_to_mn=2001
   transport_protocol=esp
   transport_esp_algorithm=blowfish-cbc
   transport_esp_secret='"magic"'
   transport_auth_algorithm=hmac-sha1
```

```
transport_auth_secret='"abcdefghijklmnopqrst"'
tunnel_spi_mn_to_ha=2002
tunnel_spi_ha_to_mn=2003
tunnel_uid_mn_to_ha=2002
tunnel_uid_ha_to_mn=2003
tunnel_esp_algorithm=blowfish-cbc
tunnel_esp_secret='"magic"'
tunnel_auth_algorithm=hmac-sha1
tunnel_auth_secret='"abcdefghijklmnopqrst"'
```

APPENDIX 3

AP1 show run output:

```
Building configuration...

Current configuration : 1662 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname magicap1
!
enable secret 5 $1$K6fT$9ZlUUnozsLVA5yVh4r3Aw0
!
username Cisco password 7 047802150C2E
ip subnet-zero
!
!
bridge irb
!
!
interface Dot11Radio0
 no ip address
 no ip route-cache
 !
 ssid magicap1
    authentication open
    guest-mode
 !
 speed basic-1.0 basic-2.0 basic-5.5 basic-11.0
 rts threshold 2312
 channel 2412
 station-role root
 bridge-group 1
 bridge-group 1 subscriber-loop-control
 bridge-group 1 block-unknown-source
 no bridge-group 1 source-learning
 no bridge-group 1 unicast-flooding
 bridge-group 1 spanning-disabled
!
interface Dot11Radio1
 no ip address
 no ip route-cache
 shutdown
 !
 ssid magicap1
    authentication open
    guest-mode
 !
 speed basic-6.0 9.0 basic-12.0 18.0 basic-24.0
36.0 48.0 54.0
 rts threshold 2312
 station-role root
 bridge-group 1
 bridge-group 1 subscriber-loop-control
 bridge-group 1 block-unknown-source
 no bridge-group 1 source-learning
 no bridge-group 1 unicast-flooding
 bridge-group 1 spanning-disabled
!
interface FastEthernet0
 no ip address
 no ip route-cache
 speed 10
 full-duplex
 bridge-group 1
 no bridge-group 1 source-learning
 bridge-group 1 spanning-disabled
!
interface BVI1
 ip address 192.168.100.1 255.255.255.0
 no ip route-cache
!
ip http server
```

```
   ip http help-path
http://www.cisco.com/warp/public/779/smbiz/prodconfi
g/help/eag/ivory/1100
   cdp timer 5
   no cdp run
   bridge 1 route ip
   !
   !
   line con 0
    password 7 13261E010803
    login
   line vty 0 4
    password 7 13261E010803
    login
   line vty 5 15
    password 7 062506324F41
    login
   !
   end
```

AP2 show run output:

```
   Building configuration...

   Current configuration : 1657 bytes
   !
   version 12.2
   no service pad
   service timestamps debug datetime msec
   service timestamps log datetime msec
   service password-encryption
   !
   hostname magicap2
   !
   enable secret 5 $1$zLEF$mZs5.d.n1teyILMkzmR420
   !
   username Cisco password 7 00271A150754
   ip subnet-zero
   !
   !
   bridge irb
   !
   !
   interface Dot11Radio0
    no ip address
    no ip route-cache
    !
    ssid magicap2
       authentication open
       guest-mode
    !
    speed basic-1.0 basic-2.0 basic-5.5 basic-11.0
    rts threshold 2312
    channel 2412
    station-role root
    bridge-group 1
    bridge-group 1 subscriber-loop-control
    bridge-group 1 block-unknown-source
    no bridge-group 1 source-learning
    no bridge-group 1 unicast-flooding
    bridge-group 1 spanning-disabled
   !
   interface Dot11Radio1
    no ip address
    no ip route-cache
    shutdown
    !
    ssid magicap2
       authentication open
       guest-mode
    !
    speed basic-6.0 9.0 basic-12.0 18.0 basic-24.0
36.0 48.0 54.0
    rts threshold 2312
    station-role root
    bridge-group 1
    bridge-group 1 subscriber-loop-control
    bridge-group 1 block-unknown-source
    no bridge-group 1 source-learning
    no bridge-group 1 unicast-flooding
    bridge-group 1 spanning-disabled
   !
   interface FastEthernet0
    no ip address
    no ip route-cache
    speed 10
    full-duplex
    bridge-group 1
```

```
 no bridge-group 1 source-learning
 bridge-group 1 spanning-disabled
!
interface BVI1
 ip address 10.0.0.1 255.255.255.0
 no ip route-cache
!
ip http server
ip http help-path
http://www.cisco.com/warp/public/779/smbiz/prodconfi
g/help/eag/ivory/1100
cdp timer 5
no cdp run
bridge 1 route ip
!
!
line con 0
 password 7 13261E010803
 login
line vty 0 4
 password 7 047802150C2E
 login
line vty 5 15
 password 7 096F471A1A0A
 login
!
end
```