

Run your next CGN on a \$20 OpenWRT

Andrew Yourtchenko

@ayourtch

What is this talk about ?

- There're plenty of interesting technologies emerging
Let's pick MAP: a sustainable life-support for IPv4
- Not all of them are on the shelves yet
There are some CPE vendors working on it, but I want one *now*
- Practical steps to make your own CPE for experimental purposes

Post-IPv4 SP technologies

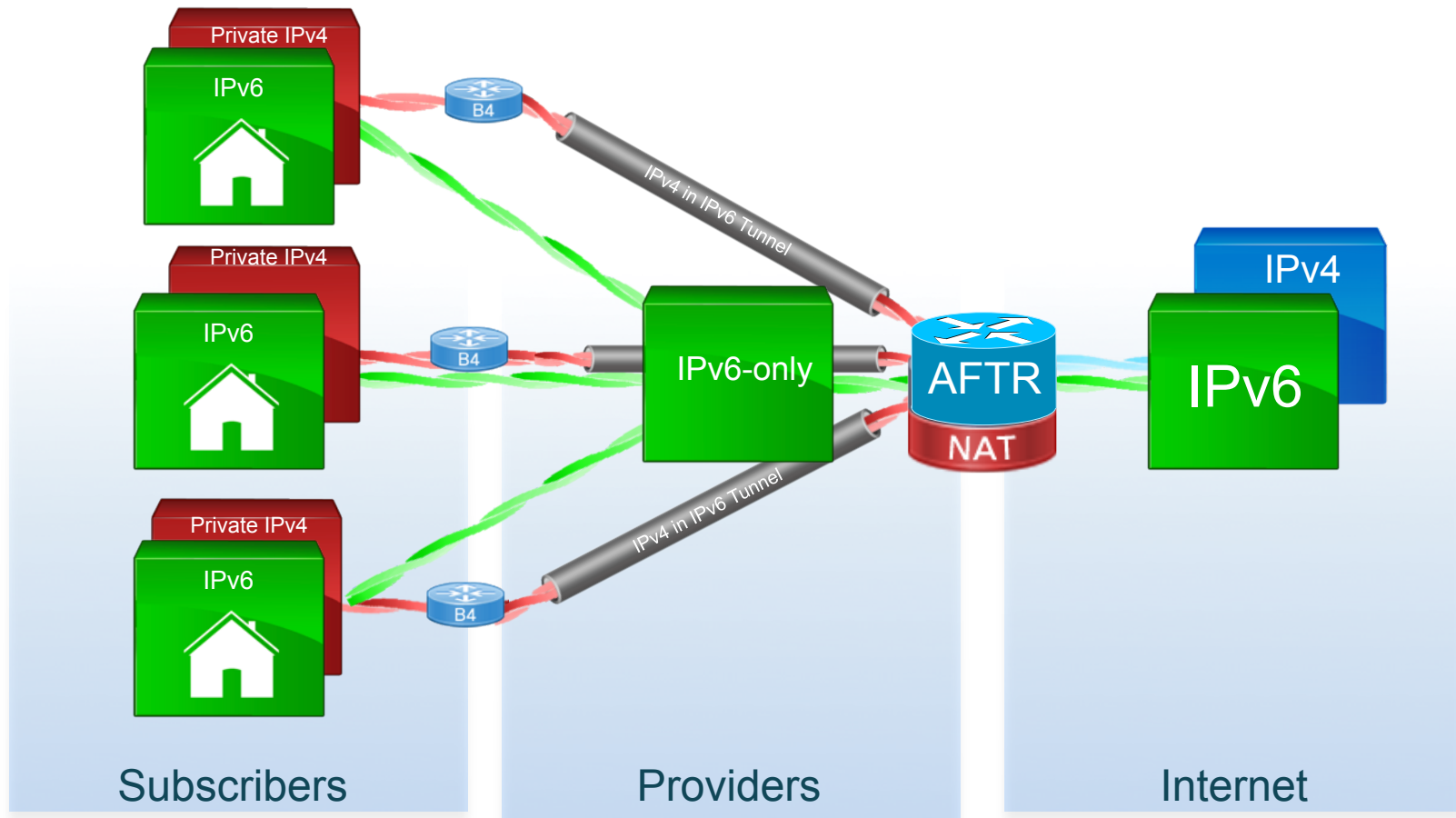


WARNING: IPv6-only ahead !

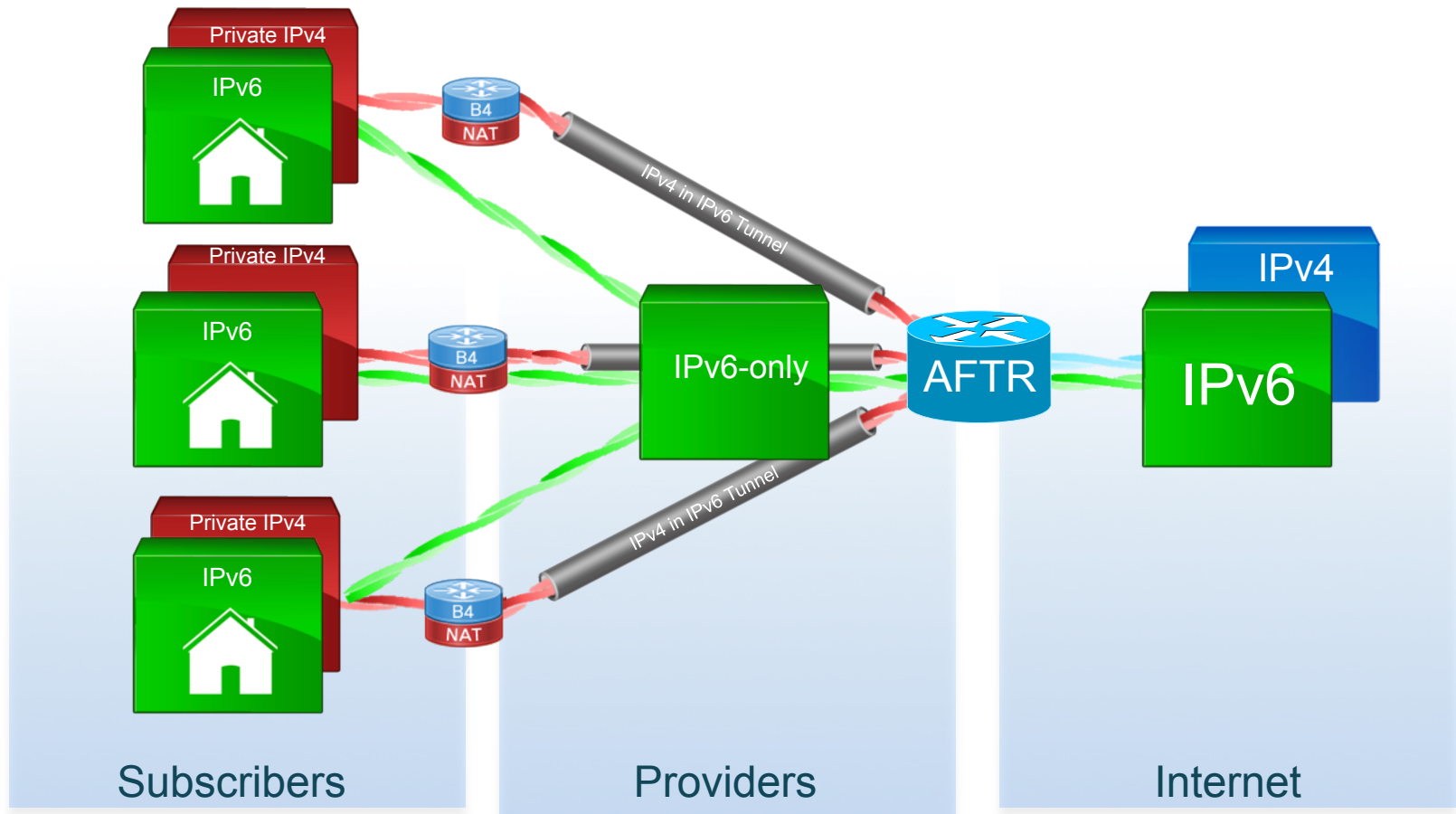
- Requires IPv6 in production
- “post-IPv4”: IPv4 as a service



Dual Stack Lite (DS-Lite)

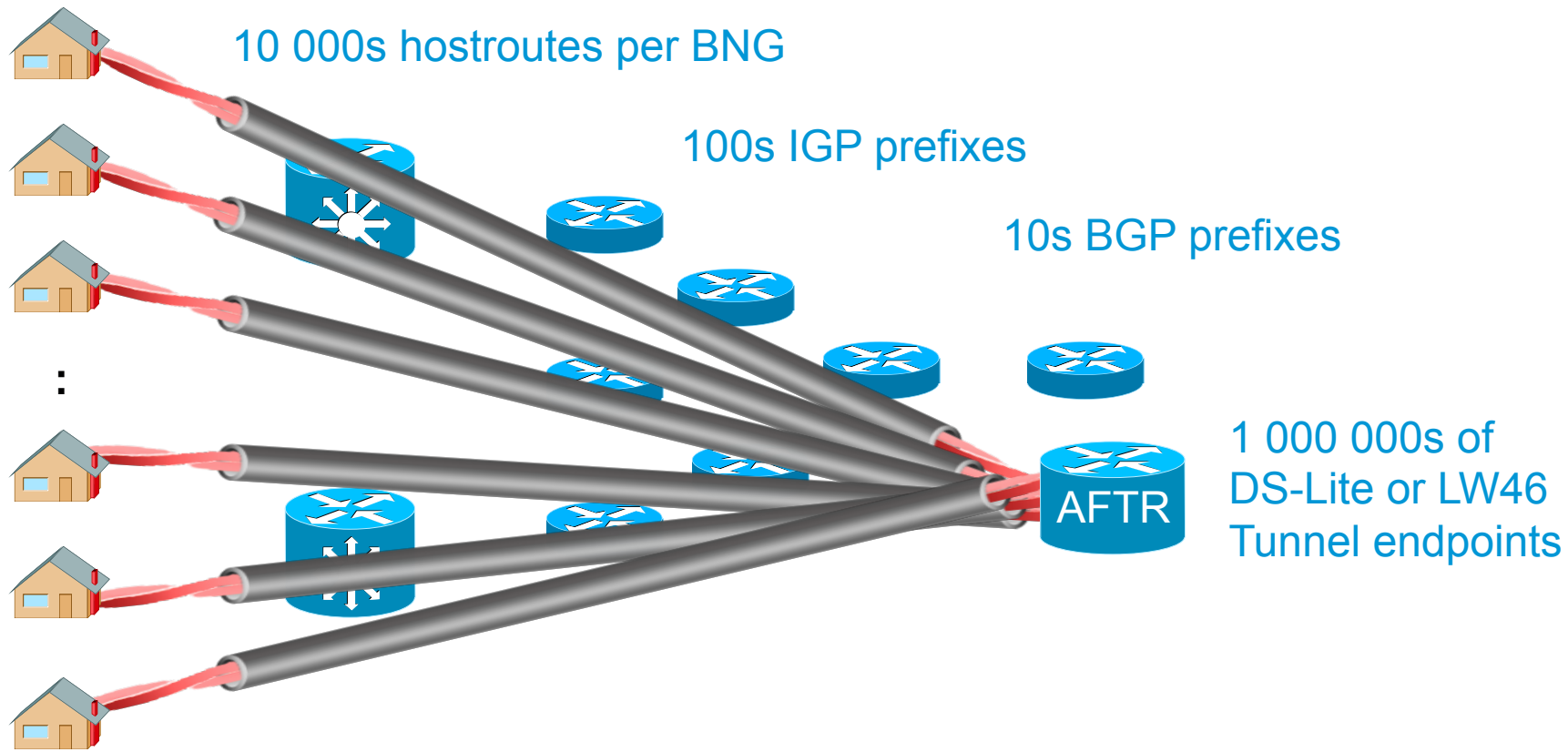


“Lightweight 4 over 6” (also “Public 4over6”)



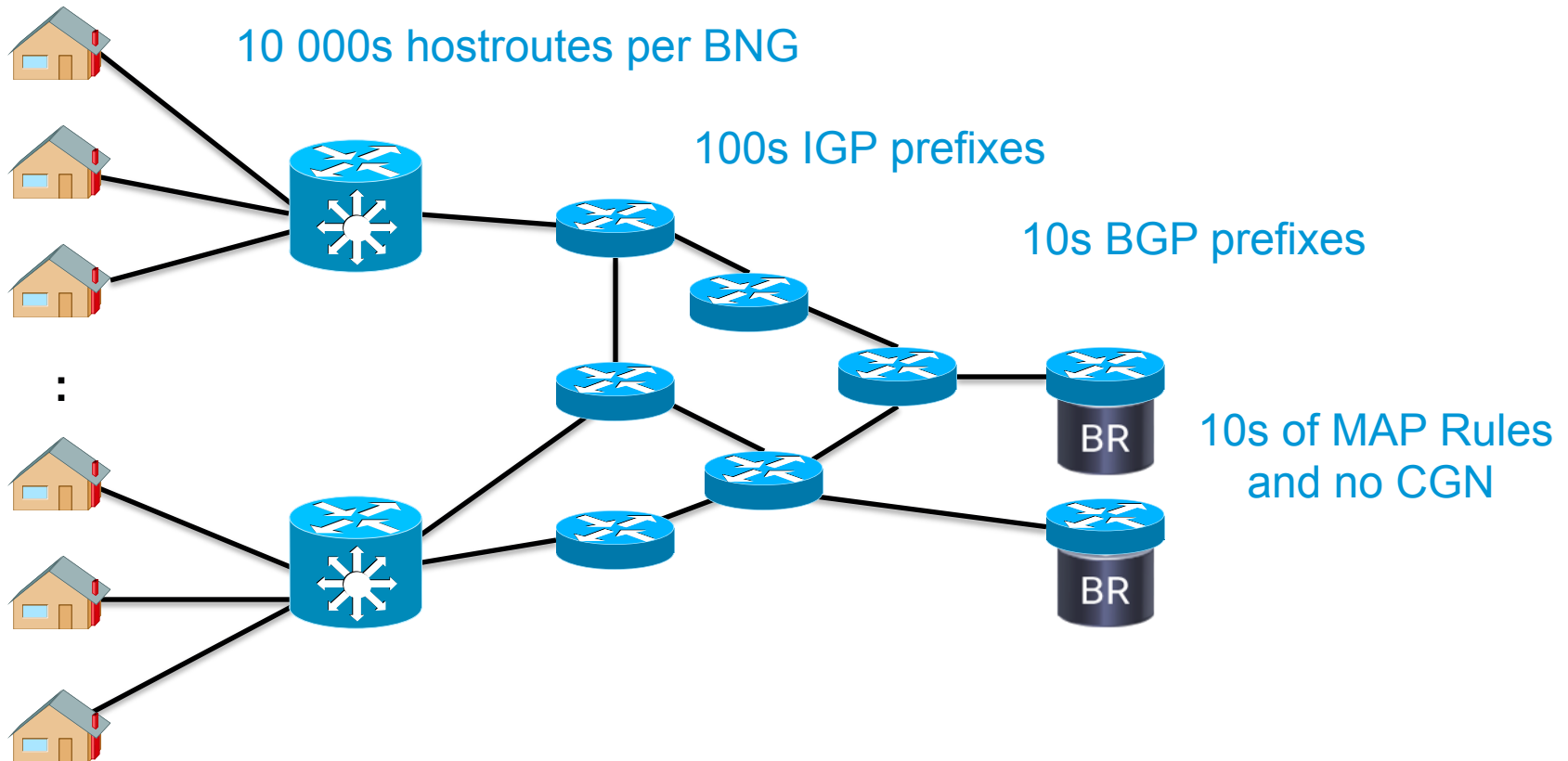
DS-Lite/LW46/Public 4over6 – Per-subscriber tunnels

1 000 000s of subscribers

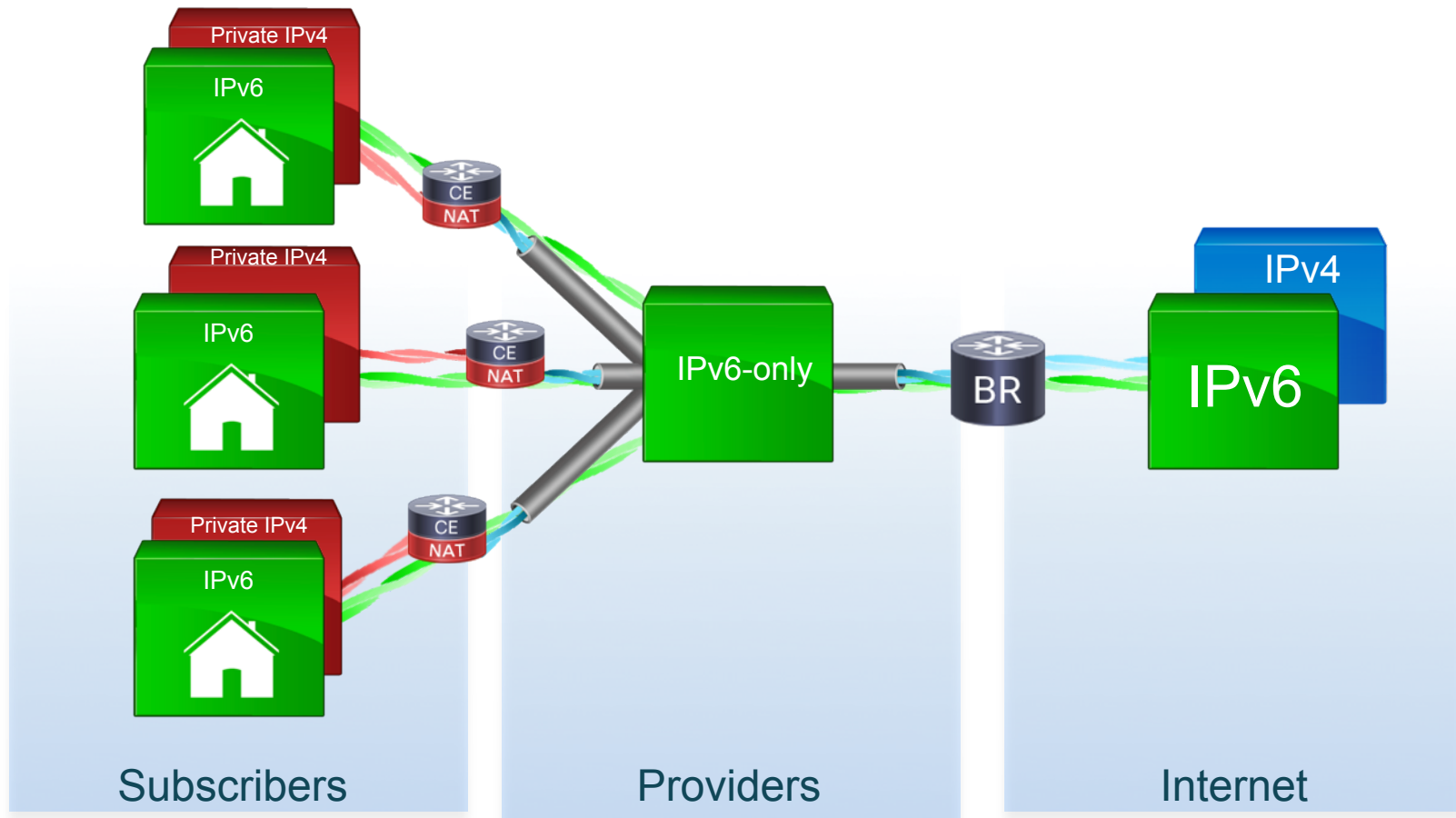


MAP Exploits Aggregation in IPv6 Routing

1 000 000s of subscribers



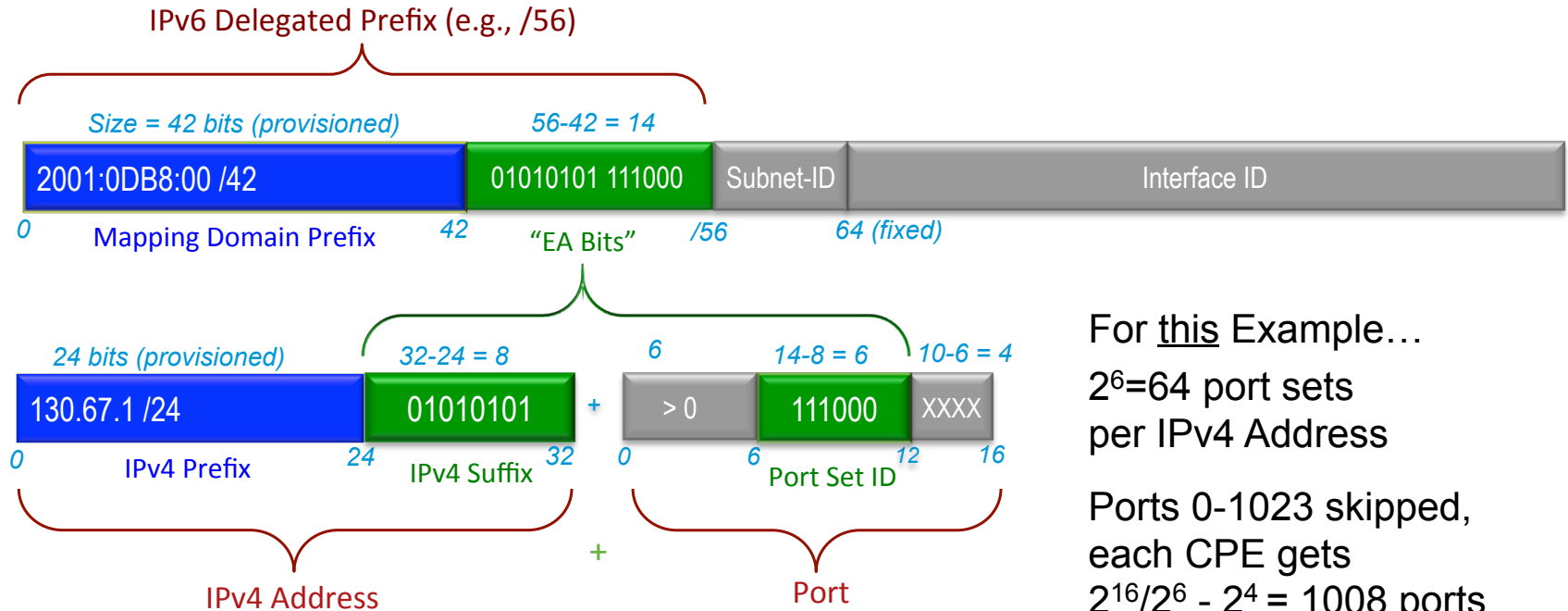
Mapping Address + Port (MAP)



Stateless Address Sharing With MAP

- A public IPv4 address: $(32 - \text{MAP IPv4 prefix len}) = p$ bits
- PSID: Port Set ID: q bits
- $p + q = \text{DHCPv6-PD (user) pref.len.} - \text{MAP Rule IPv6 pref. len}$

Stateless Address Sharing: Example



For this Example...

$2^6=64$ port sets
per IPv4 Address

Ports 0-1023 skipped,
each CPE gets
 $2^{16}/2^6 - 2^4 = 1008$ ports
One IPv4 /24 serves
 $2^{(6+8)} \approx 16,384$ (vs. ≈ 256)
subscribers



MAP Simulation Tool (beta)

Video tutorial

Highly editable elements

Add a new MAP rule

Remove all MAP rules

Load rules from text

Save rules to text

Create a link to these rules

Paste previously saved set of rules here.

Rule 0

Delete

Advanced

Example

/56

IPv6

2001:db8:9500:0

/40

EA Bits
(16 = 8 + 8)

Subnet
(8)

Interface ID (64)

IPv4 : Port

198.51.100.0 /24

Suffix
(8)

:

(4)

PSID
(8)

(4)

256 IPv4 addresses, 65536 users, 240 ports each (1:256)

In order to help us understand how this tool is being used and to improve it in the future, it will periodically save anonymous usage information for analysis. This does NOT include your IP address or any other information not needed by the tool itself. If you wish, you may override this by unchecking the box below.

☒ Data collection is currently on.

MAP Simulation tool created by [Arthur Lacoste](#) of Cisco Systems based on [this IETF draft](#).

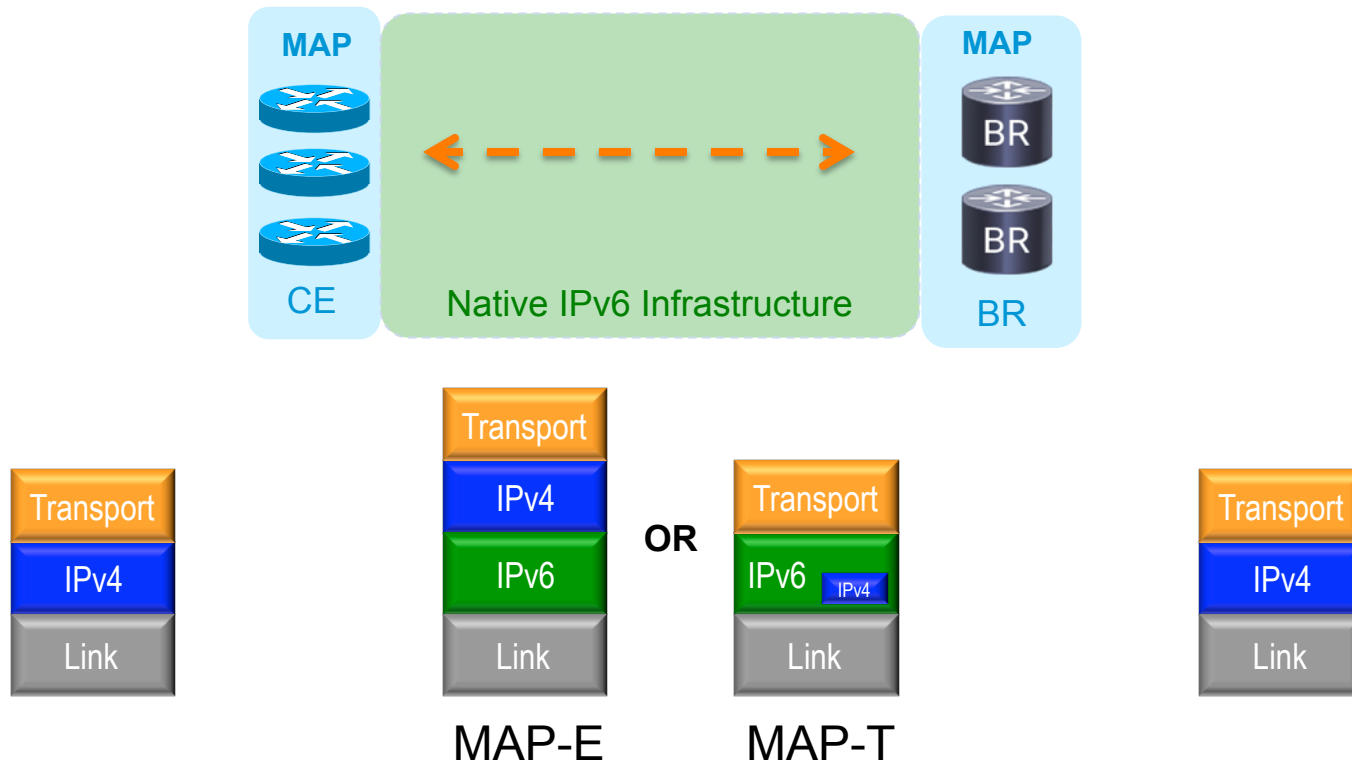
A [quick video tutorial](#) for this tool is available on youtube.

Please send comments, bug reports, and other feedback to : [map46-tool-feedback\[at\]external.cisco.com](mailto:map46-tool-feedback[at]external.cisco.com)

Last updated: 6/19/2012

http://6lab.cisco.com/map

Encapsulation or Translation – Boils down to 20 bytes



Standardizing MAP in the IETF

- MAP-E will be a Standards Track RFC
<http://tools.ietf.org/html/draft-ietf-softwire-map-07>
- MAP-T, 4rd, etc. will be Experimental or Informational
<http://tools.ietf.org/html/draft-ietf-softwire-map-t-01>
- LW46/Pubilc4over6 can be viewed as “special cases” of MAP
- Goal: One unified standard for CPE vendors
- Stretch Goal: One unified standard for BR/AFTR vendors

Running code

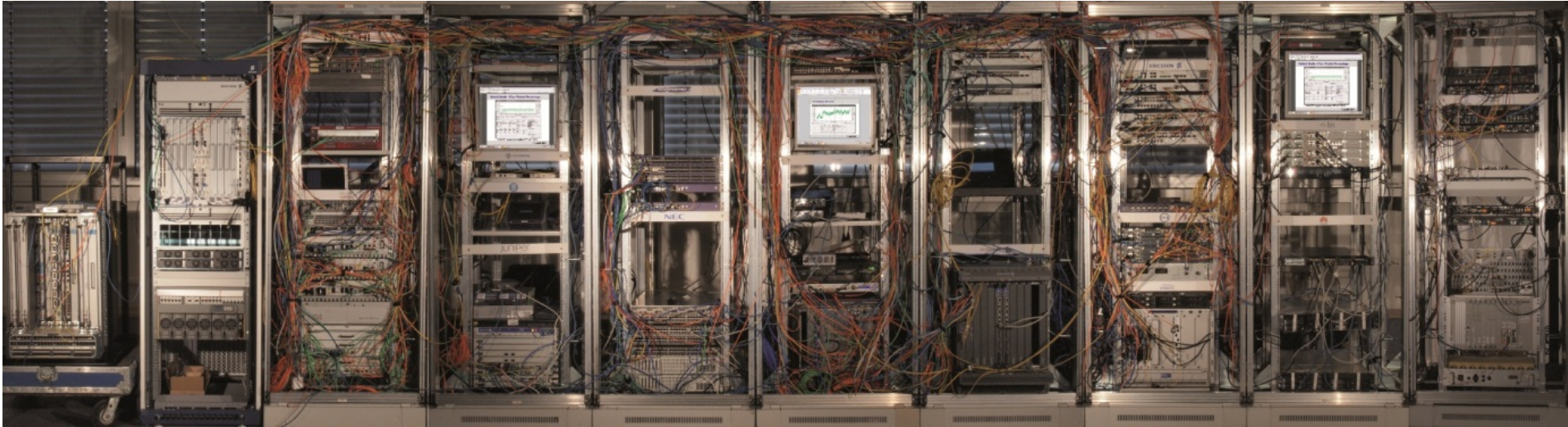


MAP testing by NIC.br

- “The working applications had no need of a special configuration to work.”
- Most of the applications work OK
- FTP active mode does not work.
(But, it's 2013...)
- More info:

<http://tools.ietf.org/html/draft-cordeiro-experience-mapt-testing-00>

European Advanced Networking Test Center



IPv6 MAP Testing at Multi-Vendor Interoperability Test Event 2013

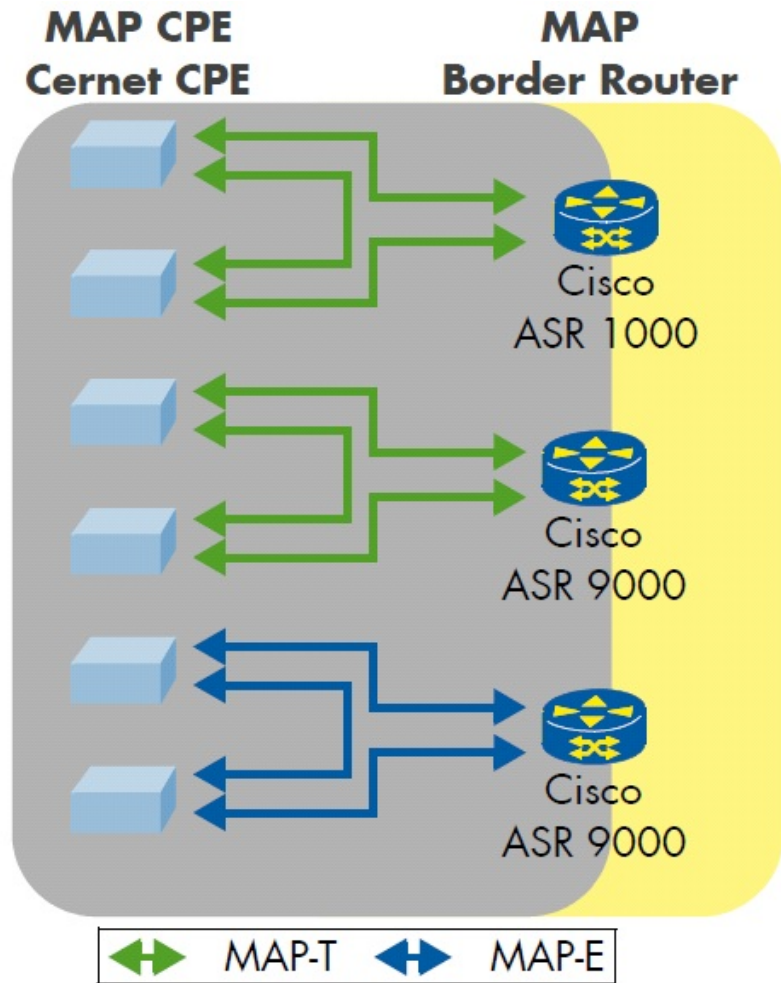
Mapping of Address and Port (MAP)

IPv6 Tests

- Stateless counterpart to DS-Lite
- Designed to be used without Carrier-Grade NAT
- Cisco ASR1000, ASR9000 and Cernet (CPE) participated

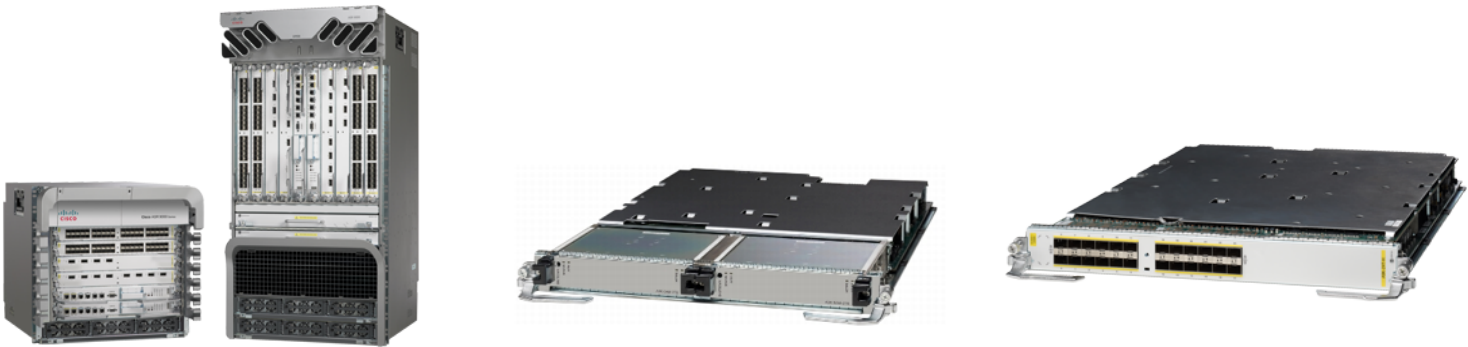
Successfully tested:

- Mapping of Address and Port with Encapsulation (MAP-E)
- Mapping of Address and Port using Translation (MAP-T)



MAP on ASR 9K

- MAP does not route traffic through the ISM Blade, yielding line rate performance.
 - Using A9K-24x10G line cards = 240 Gbps per slot!
 - $7 \times 240 = 1.68 \text{ Tbps}$ on a 9010 chassis.
- DS-Lite routes traffic through the ISM Blade
 - 14Gbps per slot



CPE code: <http://github.com/cernet/MAP>

The screenshot shows the GitHub web interface for the repository **cernet / MAP**. The browser address bar displays <https://github.com/cernet/MAP>. The repository is public and has 0 pull requests, 0 issues, and 1 branch (master). The description states: "An open source CPE implementation of MAP-E/MAP-T which can be run on Linux and Openwrt. — [Read more](#)".

Clone options include:

- Clone in Mac
- ZIP
- HTTP
- SSH
- Git Read-Only
- Repository URL: <https://github.com/cernet/MAP.git>

The commit history table shows the following entries:

Commit Message	Author	Time	Details
fix a small comment mistake	cernet	an hour ago	fix a small comment mistake [cernet]
MAP-1.0 version	cernet	4 hours ago	MAP-1.0 version [cernet]

DIY CPE: How To



“E” or “T” ?

<http://tools.ietf.org/html/draft-ietf-software-map>

Standards Track

Running code on ASR9k

<http://tools.ietf.org/html/draft-ietf-software-map-t>

Experimental Track

Running code on ASR9k, ASR1k

My deciding factor: the size of the box. Also, I like NATs. “T”.

Your own CPE: OpenWRT

- Great platform support
- Well documented
- Open Source



My own CPE: the hardware (TP-Link)

- X86 VM
The cheapest
- TL-WR703N
The smallest
- TL-MR3020
Feels more polished
- TL-WR1043ND
PoC platform of choice
- TL-WDR4300
The luxury CPE.



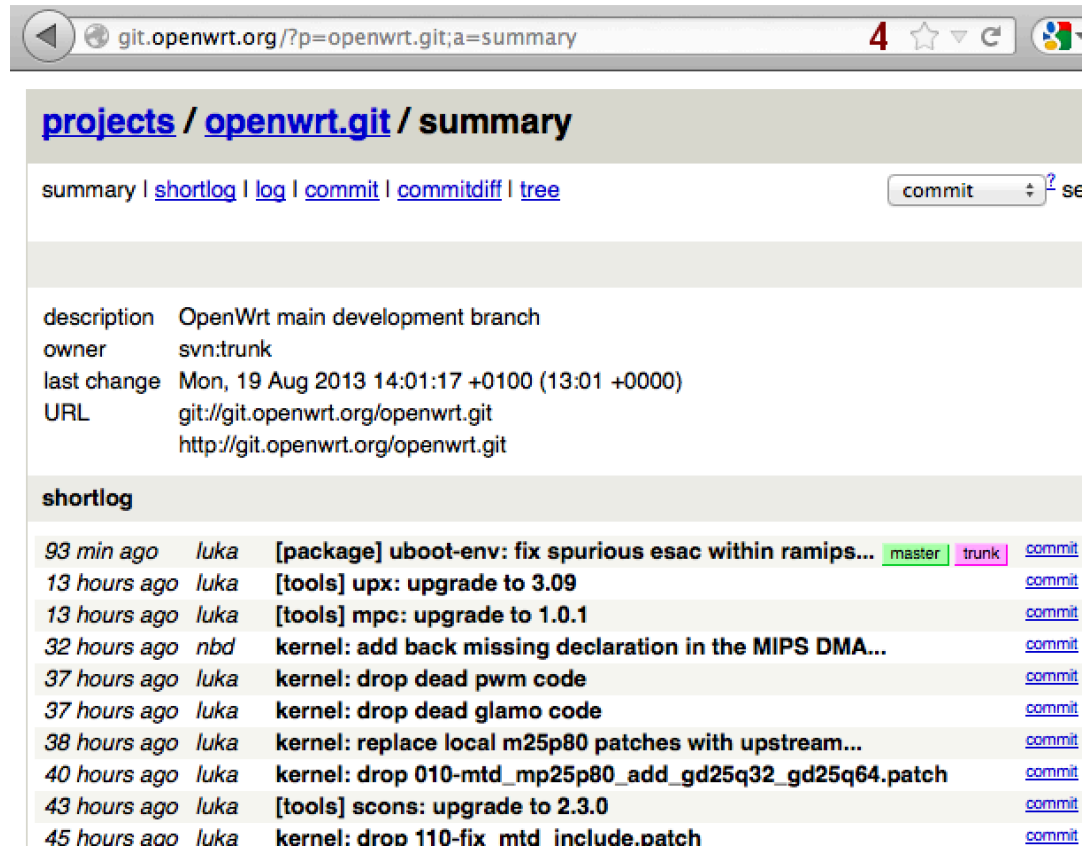
Getting your build environment

- Ubuntu 12.04 Server install with all-defaults
- In a VM => easy to rollback

```
sudo apt-get update  
sudo apt-get upgrade
```

```
sudo apt-get install build-essential subversion git-core libncurses5-dev  
sudo apt-get install zlib1g-dev gawk flex quilt libssl-dev unzip  
sudo apt-get install xsltproc libxml-parser-perl
```

Check out the trunk: “bleeding edge”



The screenshot shows a web browser window displaying the OpenWrt git repository summary page. The browser's address bar shows the URL `git.openwrt.org/?p=openwrt.git;a=summary`. The page title is **projects / openwrt.git / summary**. Below the title, there are links for [summary](#), [shortlog](#), [log](#), [commit](#), [commitdiff](#), and [tree](#). A search bar with the text "commit" and a magnifying glass icon is also present. The main content area is divided into two sections: "description" and "shortlog".

description

- OpenWrt main development branch
- owner: svn:trunk
- last change: Mon, 19 Aug 2013 14:01:17 +0100 (13:01 +0000)
- URL: [git://git.openwrt.org/openwrt.git](http://git.openwrt.org/openwrt.git)
<http://git.openwrt.org/openwrt.git>

shortlog

Time	Author	Commit Message	Branch	Action
93 min ago	luka	[package] uboot-env: fix spurious esac within ramips...	master trunk	commit
13 hours ago	luka	[tools] upx: upgrade to 3.09		commit
13 hours ago	luka	[tools] mpc: upgrade to 1.0.1		commit
32 hours ago	nbd	kernel: add back missing declaration in the MIPS DMA...		commit
37 hours ago	luka	kernel: drop dead pwm code		commit
37 hours ago	luka	kernel: drop dead glamo code		commit
38 hours ago	luka	kernel: replace local m25p80 patches with upstream...		commit
40 hours ago	luka	kernel: drop 010-mtd_mp25p80_add_gd25q32_gd25q64.patch		commit
43 hours ago	luka	[tools] scones: upgrade to 2.3.0		commit
45 hours ago	luka	kernel: drop 110-fix mtd include.patch		commit

`git clone git://git.openwrt.org/openwrt.git`

Update and add all the packages

```
cd openwrt  
./scripts/feeds update -a  
./scripts/feeds install -a
```

Configure your CPE a la carte!

```
.config - OpenWrt Configuration

OpenWrt Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc>
to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module
< > module capable

- Target System (x86) --->
  Subtarget (KVM Guest) --->
  Target Profile (Default) --->
  Target Images --->
  Global build settings --->
  [ ] Advanced configuration options (for developers) --->
  [ ] Build the OpenWrt Image Builder
  [ ] Build the OpenWrt SDK
  [ ] Build the OpenWrt based Toolchain
  [ ] Image configuration --->
    Package features --->
    Base system --->
    LuCI --->
    Kernel modules --->
    Boot Loaders --->
    Administration --->
    Video Streaming --->
    Xorg --->
    Mail --->
    Libraries --->
    Network --->
    LuCI2 --->
    Multimedia --->
  +(+)
```

<Select> < Exit > < Help > < Save > < Load >

make menuconfig

Run “make”: Take a break!

```
make[3] -C package/boot/grub2 host-compile
make[3] -C package/boot/grub2 compile
make[3] -C feeds/luci/contrib/package/freifunk-common compile
make[3] -C feeds/routing/olsrd compile
make[3] -C package/network/utils/iptables compile
make[3] -C package/network/config/firewall compile
make[3] -C package/network/utils/iproute2 compile
make[3] -C feeds/luci/contrib/package/freifunk-gwcheck compile
make[3] -C feeds/luci/contrib/package/freifunk-mapupdate compile
make[3] -C package/libs/cyassl compile
make[3] -C package/libs/ocf-crypto-headers compile
make[3] -C package/libs/zlib compile
make[3] -C package/libs/openssl compile
make[3] -C package/libs/ustream-ssl compile
make[3] -C package/network/services/uhttpd compile
make[3] -C package/network/utils/iwinfo compile
make[3] -C package/utils/lua host-compile
make[3] -C package/utils/px5g compile
make[3] -C feeds/luci/contrib/package/luci compile
make[3] -C feeds/packages/ipv6/tayga compile
make[3] -C package/kernel/linux compile
make[3] -C package/libs/libpcap compile
make[3] -C package/network/ipv6/6relayd compile
make[3] -C package/network/ipv6/odhcp6c compile
make[3] -C package/network/services/dnsmasq compile
make[3] -C package/network/services/dropbear compile
make[3] -C package/network/utils/linux-atm compile
make[3] -C package/network/utils/resolveip compile
make[3] -C package/network/services/ppp compile
make[3] -C package/network/utils/tcpdump compile
make[3] -C package/system/mtd compile
make[3] -C package/system/opkg compile
make[3] -C package/system/udev compile
make[3] -C package/utils/busybox compile
make[3] -C package/utils/mkelfimage compile
make[2] package/install
```

make

But, what about MAP ?



Several packages exist

- ASAMAP (kernel patches)
<http://enog.jp/~masakazu/vyatta/map/>
- CERNET MAP (kernel module)
<https://github.com/cernet/MAP>

CERNET MAP manual provisioning

- `ivictl -s -i br-lan -l wan0 -H -a 192.168.1.1/24 -A 1.1.1.1/32 -P 2001:6f8:147e:1000::/52 -R 16 -z 4 -o 14 -c 1234 -T`
- `ivictl -r -d -P 2610:d0:1208:cafe::/64 -T`

(does it look complicated to you too ?)

There's got to be a better way!

- IETF draft - **draft-ietf-softwire-map-dhcp-03**
- A new “MAP” DHCPv6 option
 - Rule option
 - DMR option
 - MAP Port Parameters
- *static* value, the same across the entire MAP domain
- Let's do some coding!



Odhcp6c custom scripting

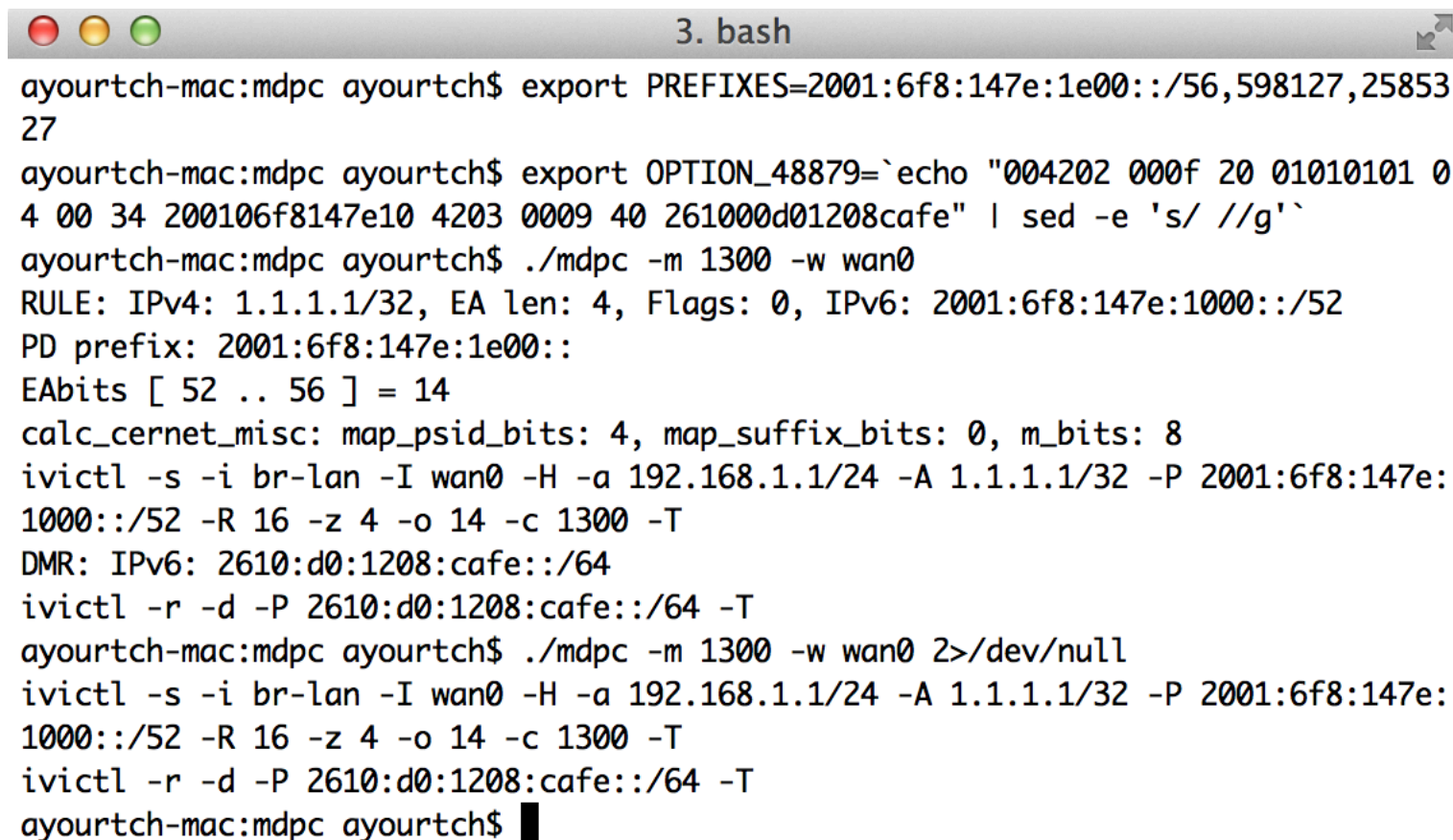
- Starts /etc/odhcp6c.user on addressing changes
- Preset environment variables
 - Allocated prefixes
 - DHCPv6 options requested

First implementation in shell

- ~1 day to write
- Works
- Problem: way too slow
- Need a rewrite!

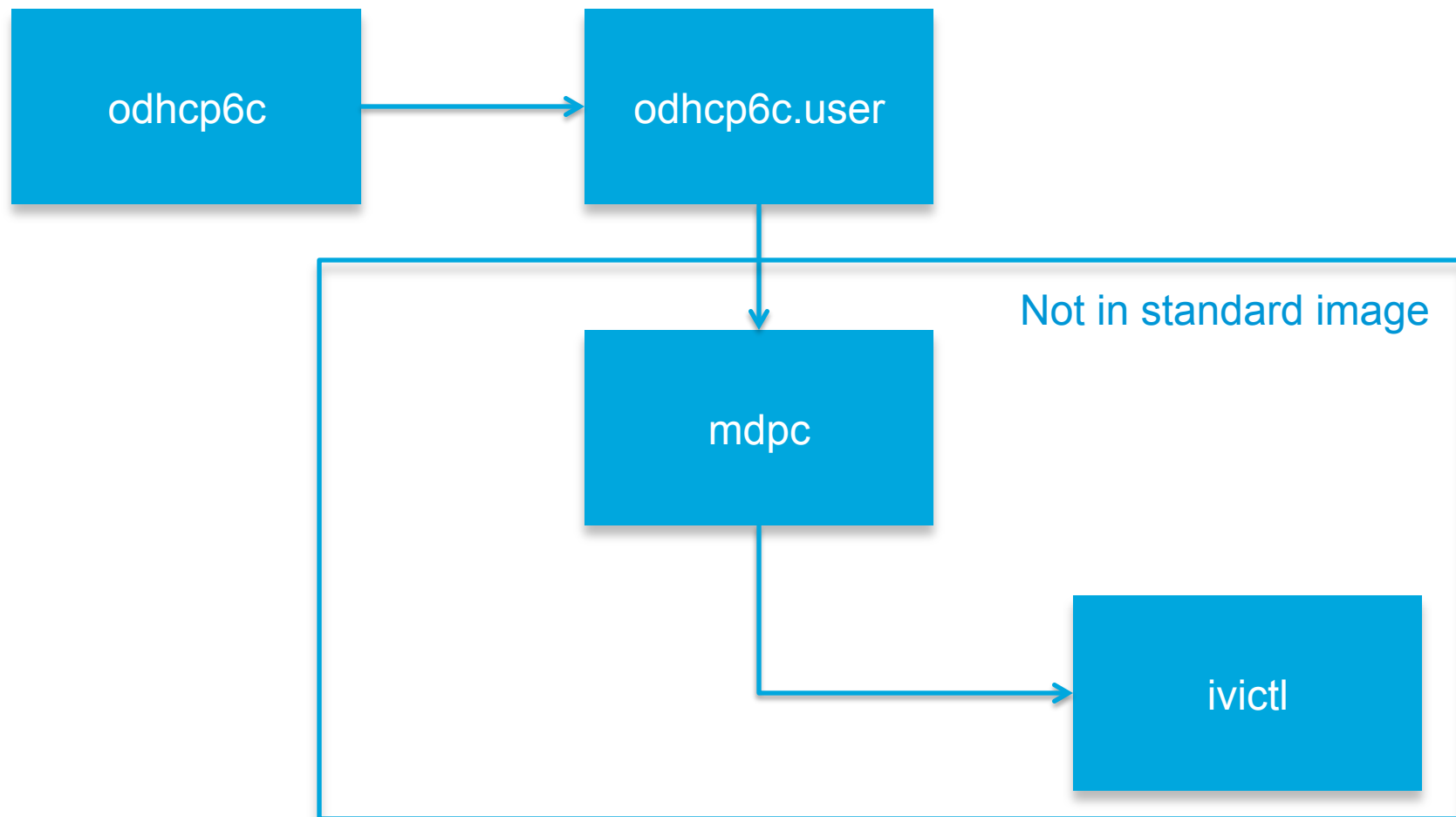
Let's do it in C

- <https://github.com/ayourtch/mdpc>



```
ayourtch-mac:mdpc ayourtch$ export PREFIXES=2001:6f8:147e:1e00::/56,598127,25853
27
ayourtch-mac:mdpc ayourtch$ export OPTION_48879=`echo "004202 000f 20 01010101 0
4 00 34 200106f8147e10 4203 0009 40 261000d01208cafe" | sed -e 's/ //g'`
ayourtch-mac:mdpc ayourtch$ ./mdpc -m 1300 -w wan0
RULE: IPv4: 1.1.1.1/32, EA len: 4, Flags: 0, IPv6: 2001:6f8:147e:1000::/52
PD prefix: 2001:6f8:147e:1e00::
EAbits [ 52 .. 56 ] = 14
calc_cernet_misc: map_psid_bits: 4, map_suffix_bits: 0, m_bits: 8
ivictl -s -i br-lan -I wan0 -H -a 192.168.1.1/24 -A 1.1.1.1/32 -P 2001:6f8:147e:
1000::/52 -R 16 -z 4 -o 14 -c 1300 -T
DMR: IPv6: 2610:d0:1208:cafe::/64
ivictl -r -d -P 2610:d0:1208:cafe::/64 -T
ayourtch-mac:mdpc ayourtch$ ./mdpc -m 1300 -w wan0 2>/dev/null
ivictl -s -i br-lan -I wan0 -H -a 192.168.1.1/24 -A 1.1.1.1/32 -P 2001:6f8:147e:
1000::/52 -R 16 -z 4 -o 14 -c 1300 -T
ivictl -r -d -P 2610:d0:1208:cafe::/64 -T
ayourtch-mac:mdpc ayourtch$
```

DHCPv6 interaction



Adding your stuff to default image



Packages and feeds

- Package

An OpenWRT-specific abstraction

Describes

- building process
- name and place in the “menuconfig” menu
- dependencies to enable

Very flexible retrieval mechanism (git, tarball, http, etc.)

- Feed

A collection of packages

Simple way to add functionality

Only one-line edit needed for the source!

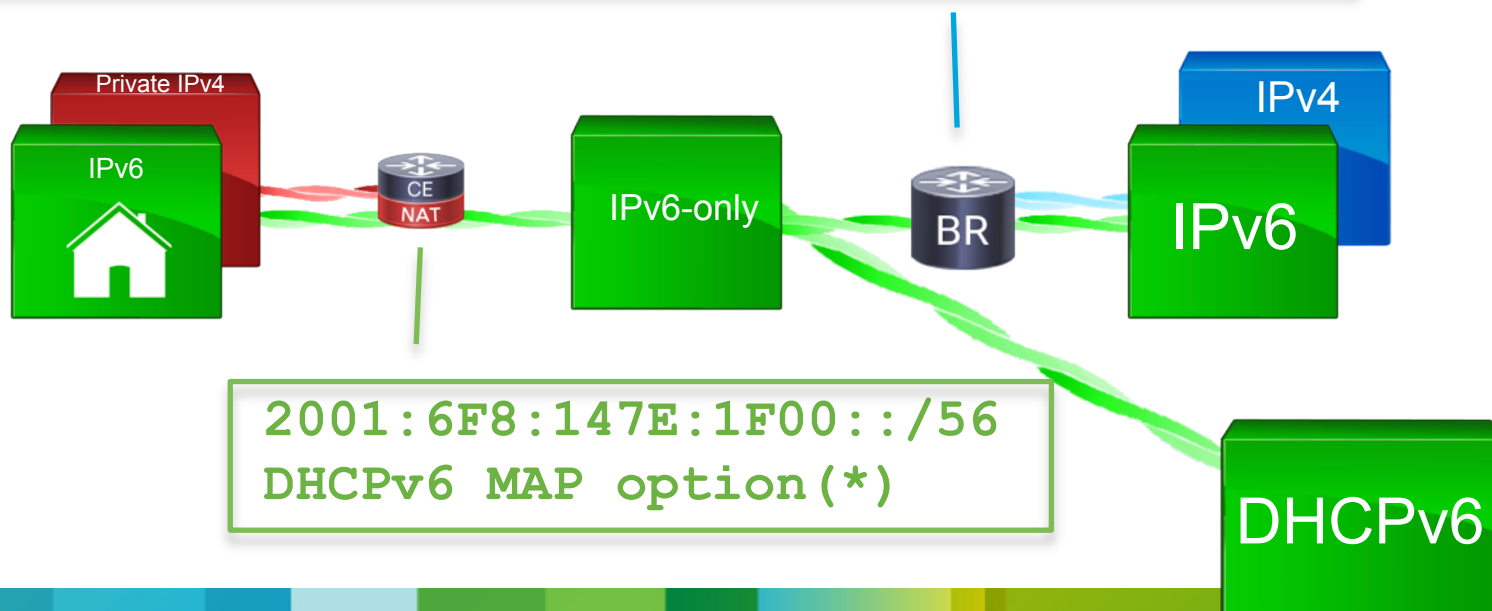


Openwrt-map: experimental feed

- <https://github.com/ayourtch/openwrt-map>
- Adds “CERNET MAP” package
- Adds “MDPC” package
- Tested on “Barrier Breaker” (trunk in October 2013)

MAP-T example demo configuration

```
nat64 map-t domain 1
  default-mapping-rule 2610:D0:1208:CAFE::/64
  basic-mapping-rule
    ipv6-prefix 2001:6F8:147E:1000::/52
    ipv4-prefix 153.16.17.83/32
    port-parameters share-ratio 16
```



How to construct the DHCPv6 option ?

DHCPv6 MAP provisioning

According to <https://github.com/ayourtch/mdpc/blob/master/draft/draft-ietf-softwire-map-dhcp-03.txt>, with arbitrary values for the with those arbitrarily chosen for mdpc).

Input

```
! paste the text config from http://6lab.cisco.com/map/MAP.php here
{Rule 0,2001:6f8:147e:1000/52,1.1.1.1/32,4,6,0}
```

BR address (/128) or DMR prefix (/64): Flavour:

Output

```
# For unit tests / shell scripts:
export OPTION_48879="004202000f2001010101040034200106f8147e104203000940261000d01208cafe"

# For ISC DHCPD dhcpd6.conf (ensure the lines are copied in full!)
option dhcp6.map-option code 48879 = string;
option dhcp6.map-option
00:42:02:00:0f:20:01:01:01:01:04:00:34:20:01:06:f8:14:7e:10:42:03:00:09:40:26:10:00:d0:12:08:ca:fe;
```

<https://github.com/ayourtch/mdpc/blob/master/html/provision-03.html>

End result: DHCPv6-provisioned MAP CPE

DIY demo: **<http://tinyurl.com/map-cpe>**

(links to <http://www.youtube.com/watch?v=UQUK5nnqilA>)

Summary

- ~~NATs are good!~~ **MAPs are good!**
- There's a MAP CPE ready for your experiments today
 - My home office connects through a MAP-T CPE and CSR1000V BR
 - Ask your CPE supplier for the production-grade code
- This model is replicable for other technologies
- Allows to evaluate the new tech w/o waiting for the vendors
 - The code they ship can contain lessons from early iterations

Thank you.

