Ralink

Kernel-Mode NVRAM

Jun 2011

Agenda

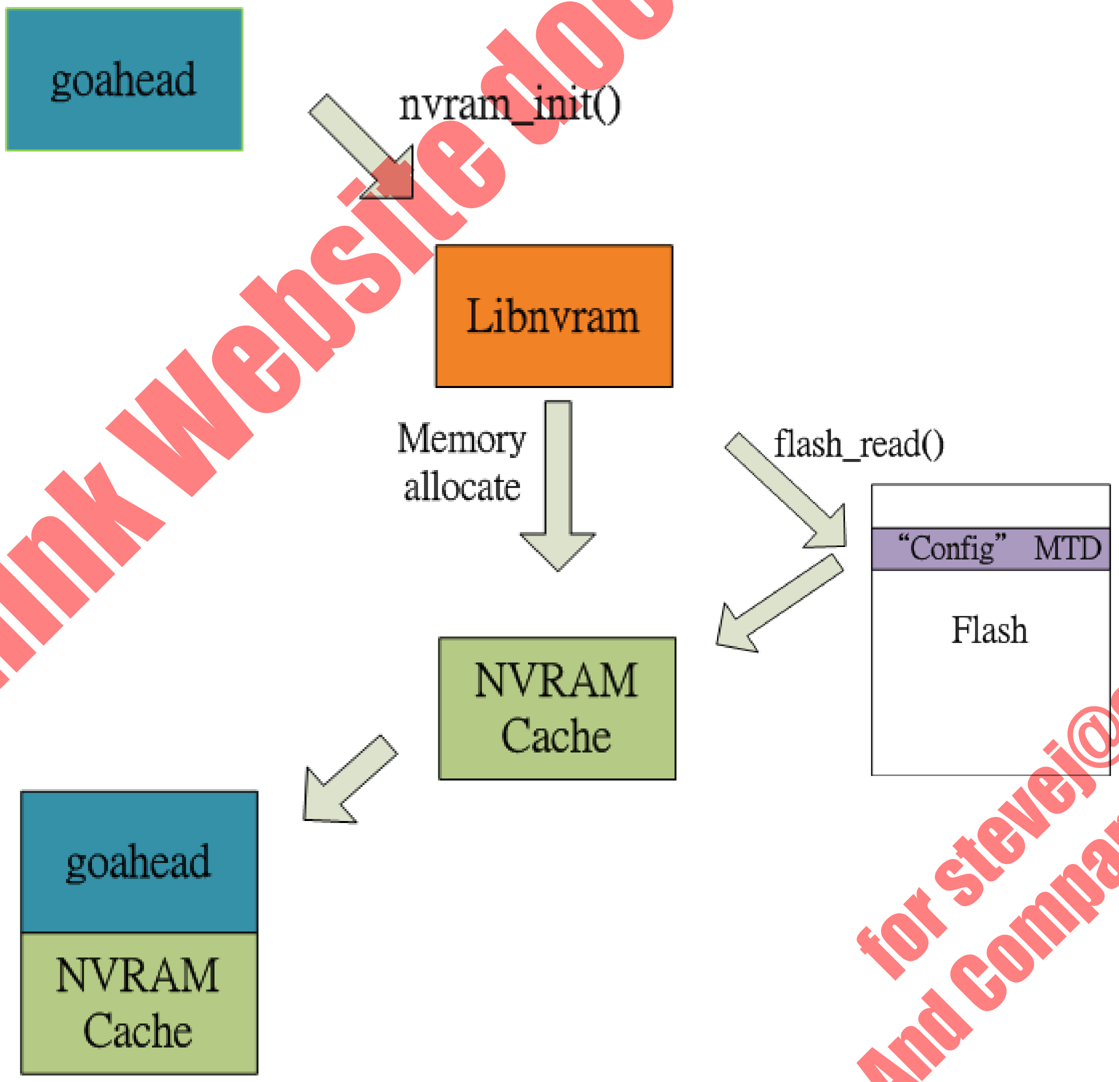
- Original Design
- Issues
- New Design
- Code Review
- Test Scenario

Original Design

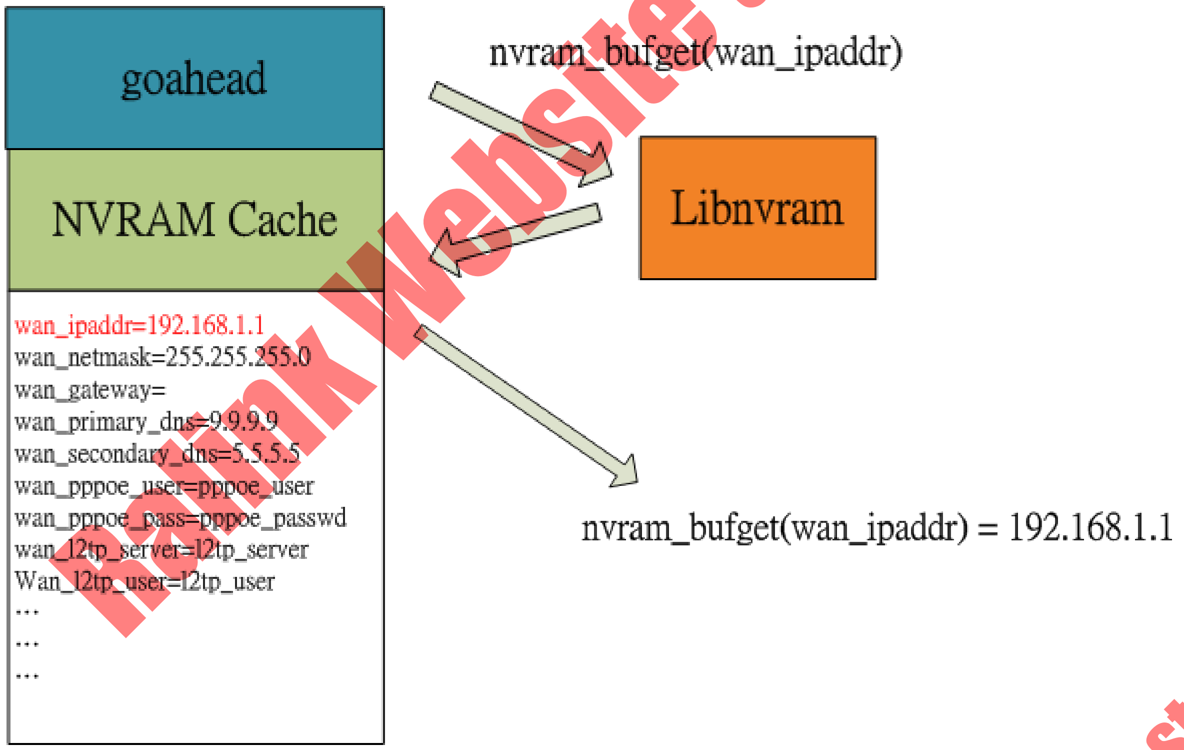
- User-space library
- Provide APIs for NVRAM operations
- Allocate a user-space memory region to store NVRAM data from Flash, and each user-space program has individual one
- Data is stored in ASCII format
- API:

  nvram_init, nvram_get, nvram_bufget, nvram_set, nvram_bufset,nvram_commit, nvram_close, nvram_clear

Original Design: nvram_init()



Original Design: nvram_bufget()

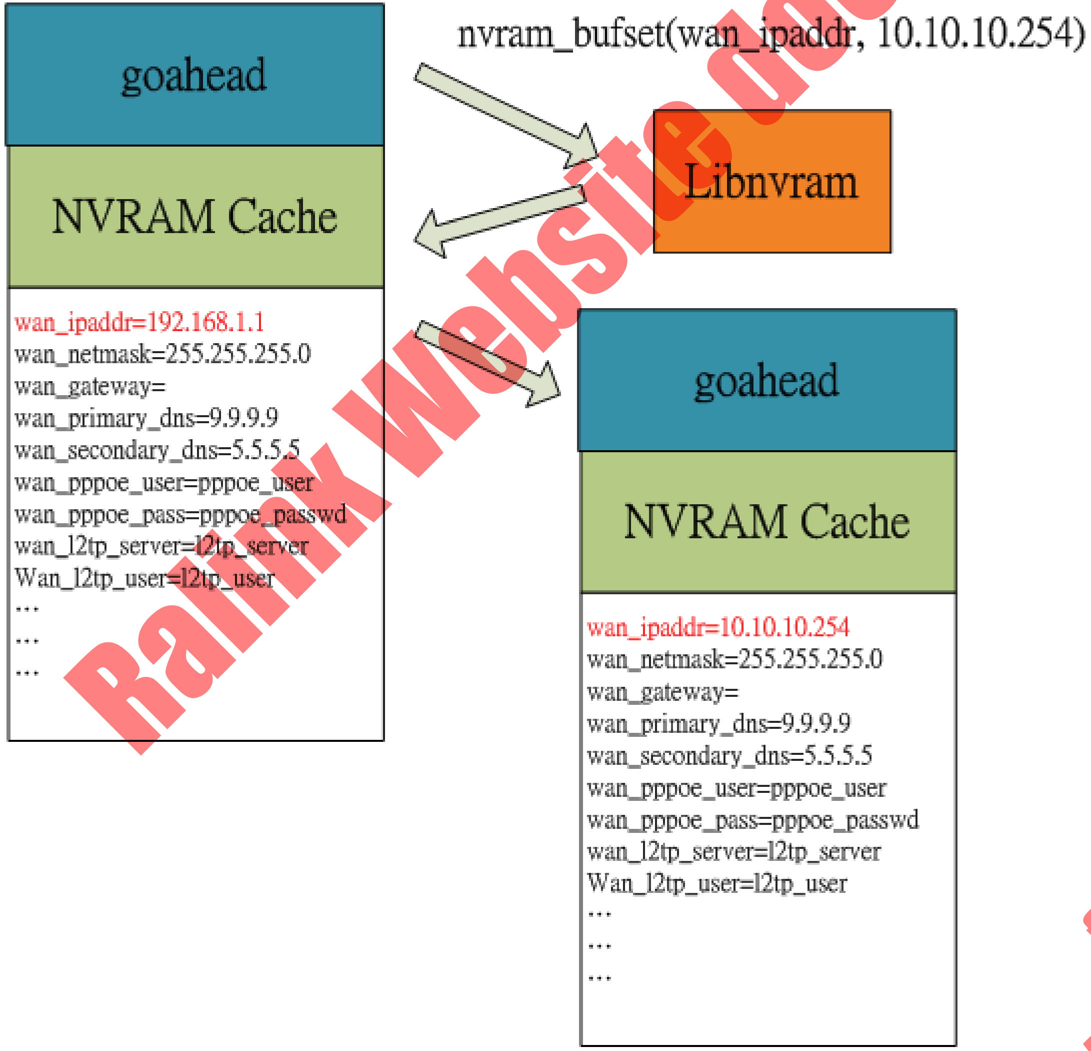


nvram_bufget(wan_ipaddr)

wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…

nvram_bufget(wan_ipaddr) = 192.168.1.1

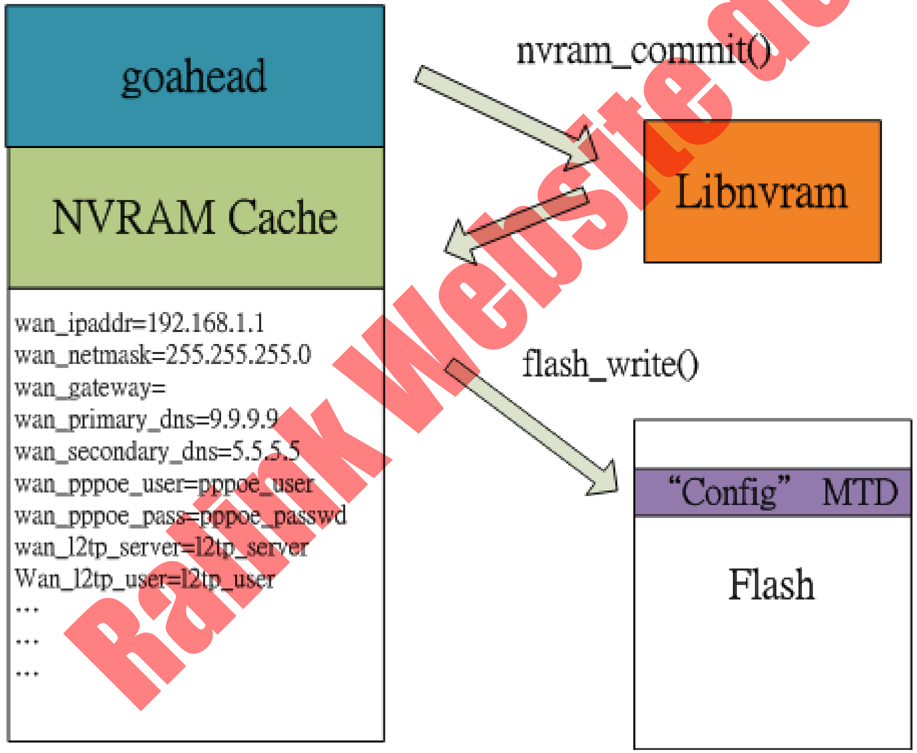
Original Design: nvram_bufset()



```
wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

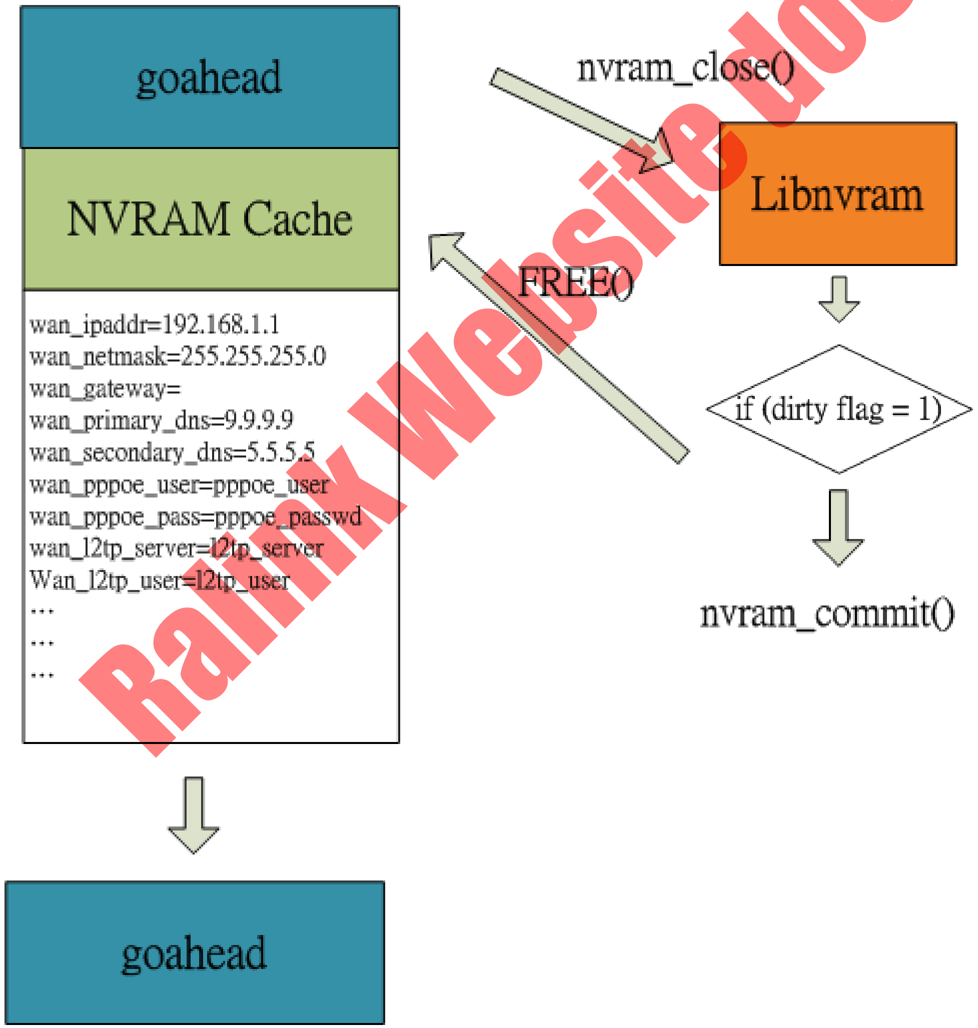
```
wan_ipaddr=10.10.10.254
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

Original Design: nvram_commit()



```
wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

Original Design: nvram_close()



goahead

NVRAM Cache

wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…

goahead

nvram_close()

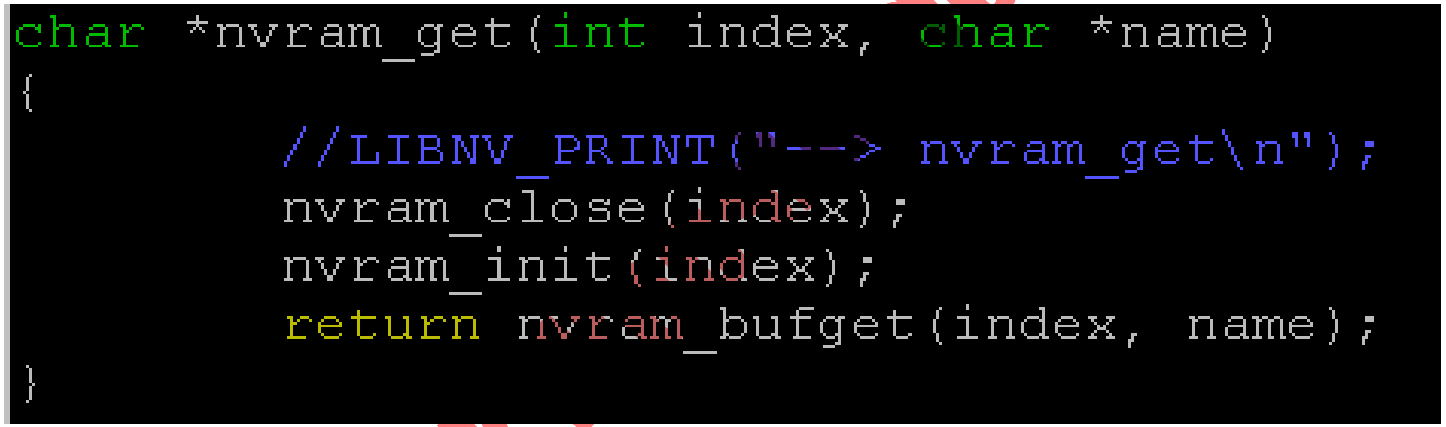
Libnvram

FREE()

if (dirty flag = 1)

nvram_commit()

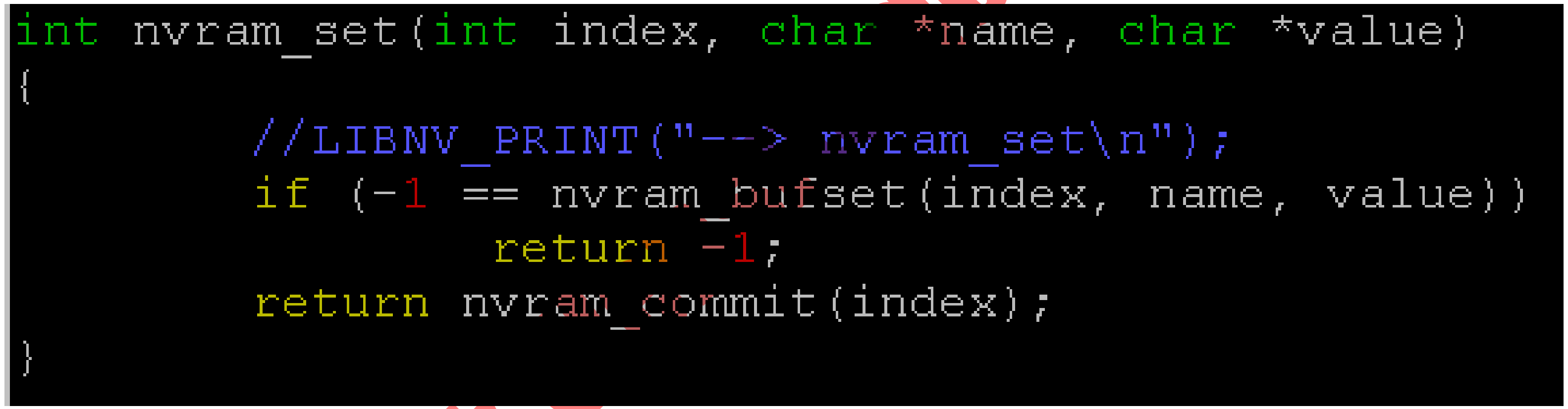
Original Design: nvram_get()



```
char *nvram_get(int index, char *name)
{
        //LIBNV_PRINT("--> nvram_get\n");
        nvram_close(index);
        nvram_init(index);
        return nvram_bufget(index, name);
}
```

- **Sync data with Flash and return the freshest value**

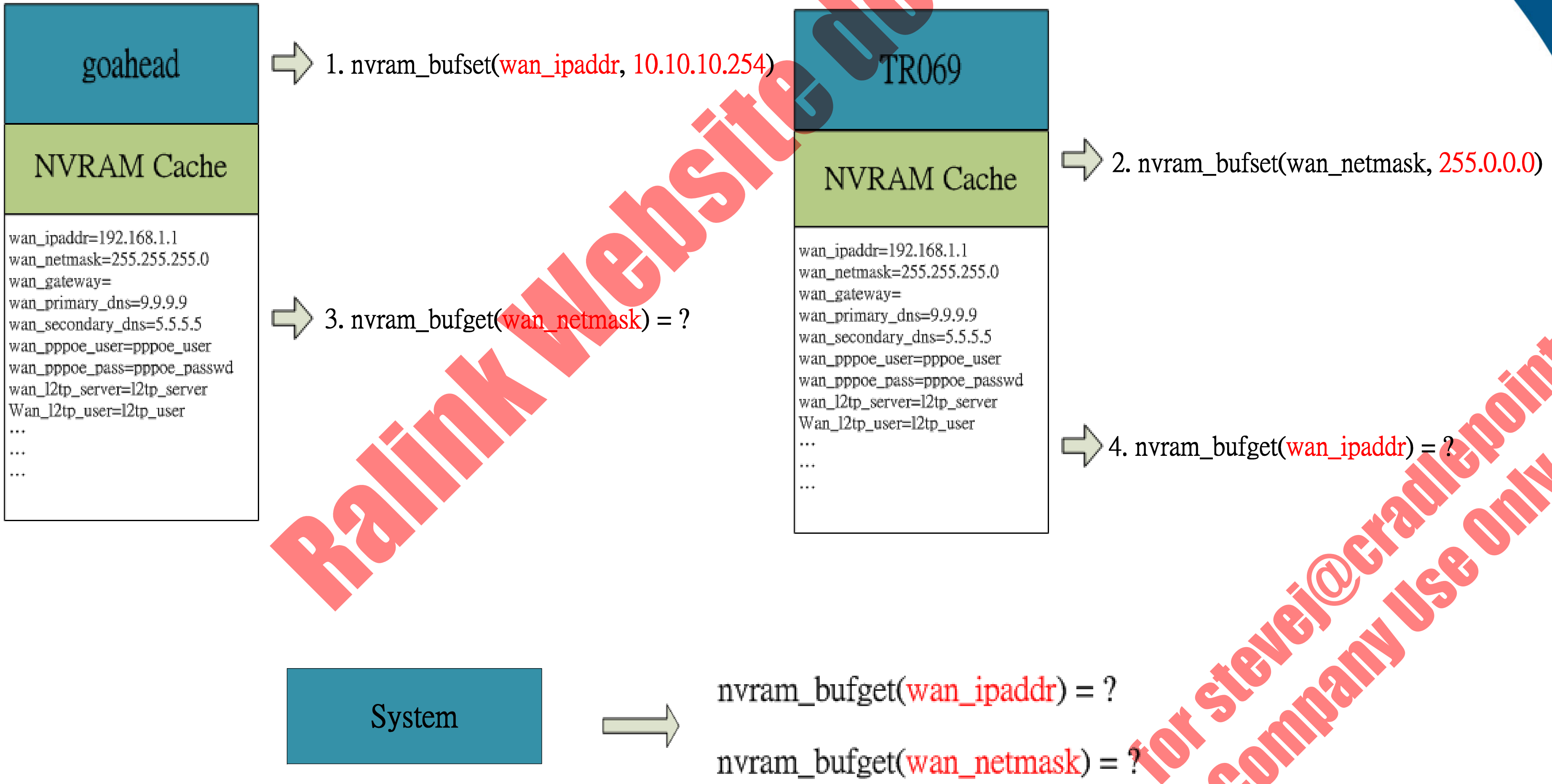
Original Design: nvram_set()



```
int nvram_set(int index, char *name, char *value)
{
        //LIBNV_PRINT("--> nvram_set\n");
        if (-1 == nvram_bufset(index, name, value))
                return -1;
        return nvram_commit(index);
}
```

- **Set the value and commit current data to Flash**

Issues?



```
wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

1. nvram_bufset(wan_ipaddr, 10.10.10.254)

3. nvram_bufget(wan_netmask) = ?

```
wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

2. nvram_bufset(wan_netmask, 255.0.0.0)

4. nvram_bufget(wan_ipaddr) = ?

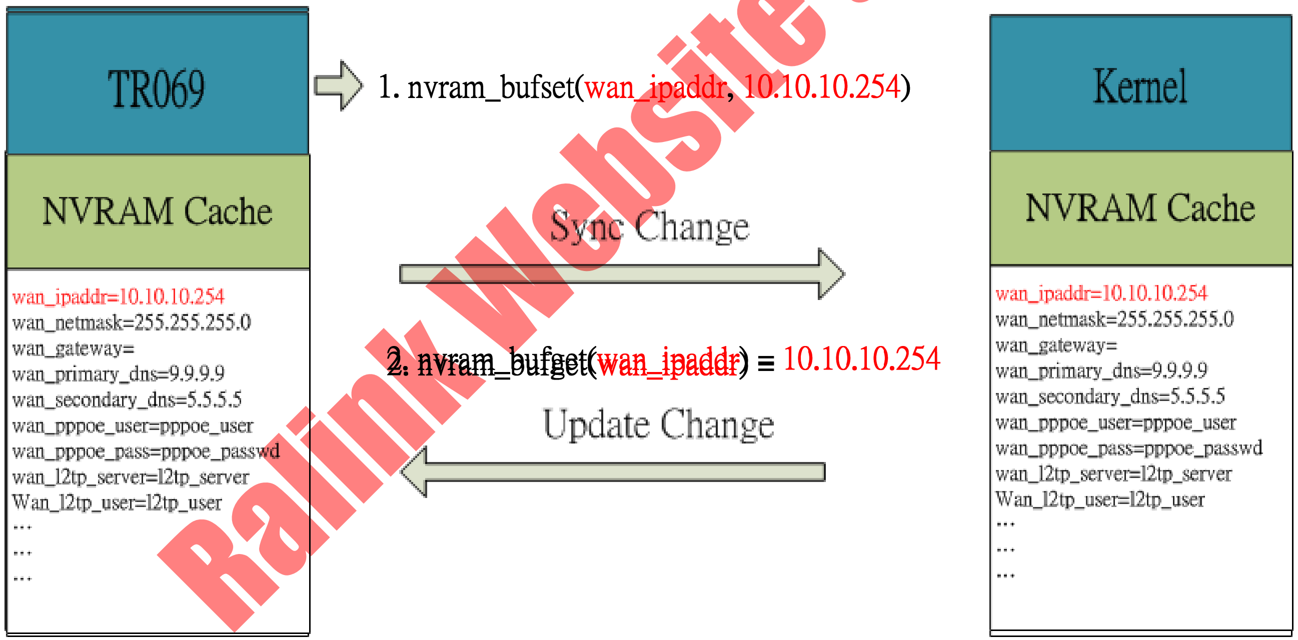
nvram_bufget(wan_ipaddr) = ?

nvram_bufget(wan_netmask) = ?

Solutions?

- **Use nvram_get() and nvram_set() instead**
  - nvram_commit() takes long time (flash erase and write)
  - Not efficient
- **Notification**
  - Need to know which program needs to be notified(is using NVRAM)
  - Other program may not be interested in it
- **Centralized Database**
  - Every NVRAM user synchronize changes with it
  - Keep the freshest data

New Design



TR069

NVRAM Cache

```
wan_ipaddr=10.10.10.254
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

1. nvram_bufset(wan_ipaddr, 10.10.10.254)

Sync Change

2. nvram_bufget(wan_ipaddr) = 10.10.10.254

Update Change

Kernel

NVRAM Cache

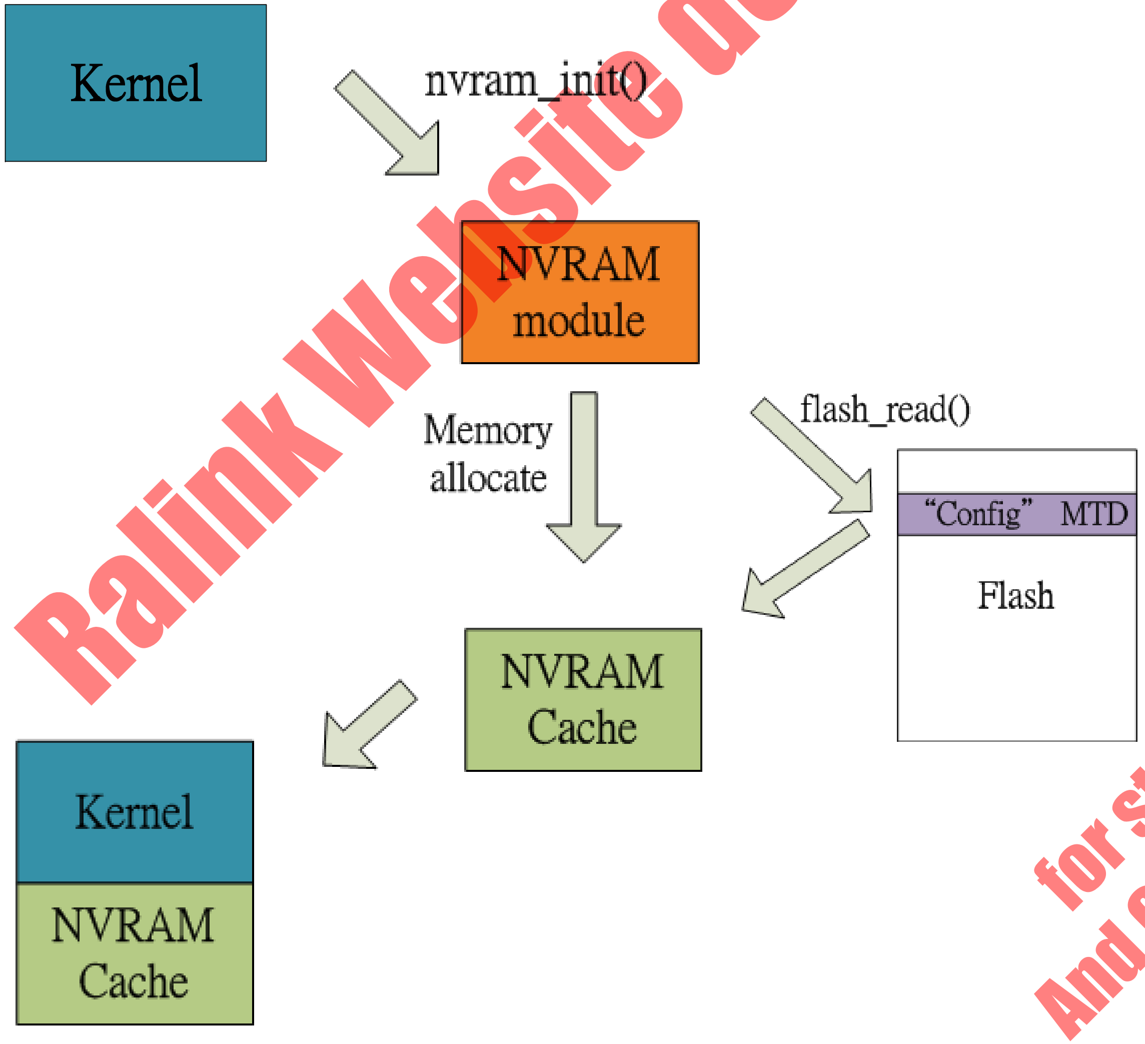
```
wan_ipaddr=10.10.10.254
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

New Design: nvram_init()

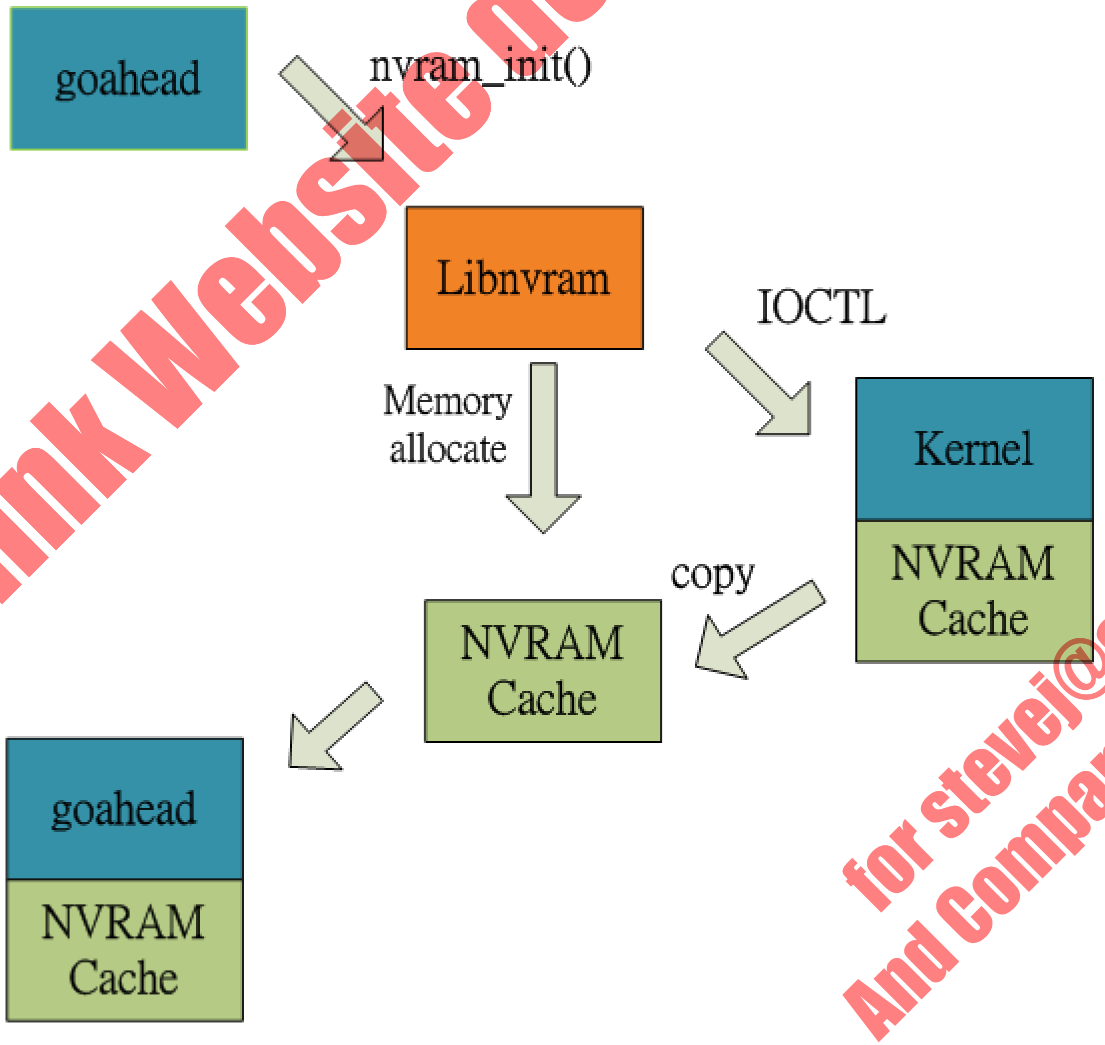
- Kernel NVRAM module will invoke nvram_init() at the boot time which does the same thing with previous user-mode nvram_init()

- User-space program also needs to call libnvram nvram_init(), but it gets the NVRAM data from Kernel NVRAM cache instead, not directly from Flash.

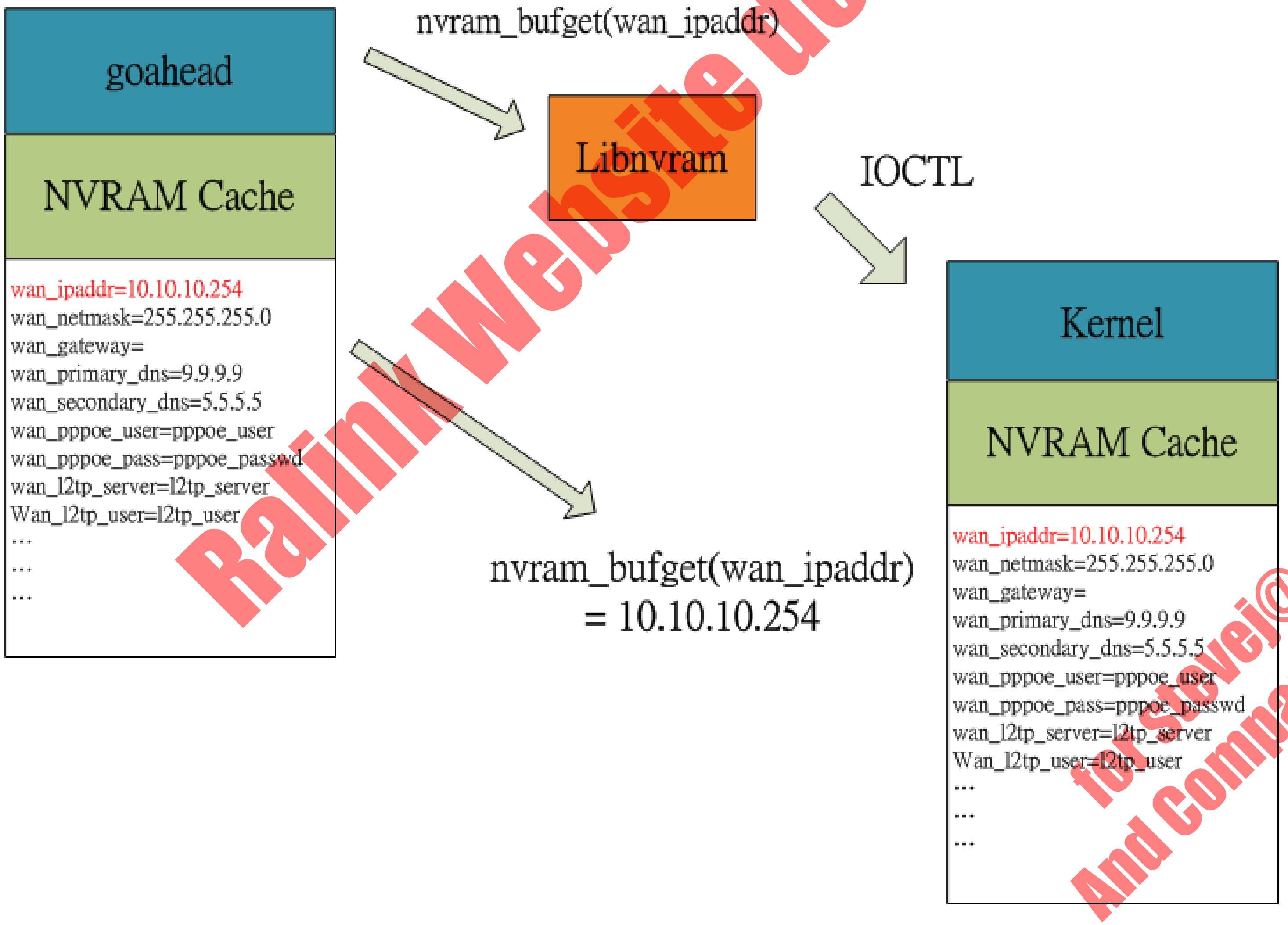
New Design: Kernel nvram_init()



New Design:
Libnvram nvram_init()



New Design: Libnvram nvram_get()



goahead

NVRAM Cache

wan_ipaddr=10.10.10.254
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…

nvram_bufget(wan_ipaddr)

Libnvram

IOCTL

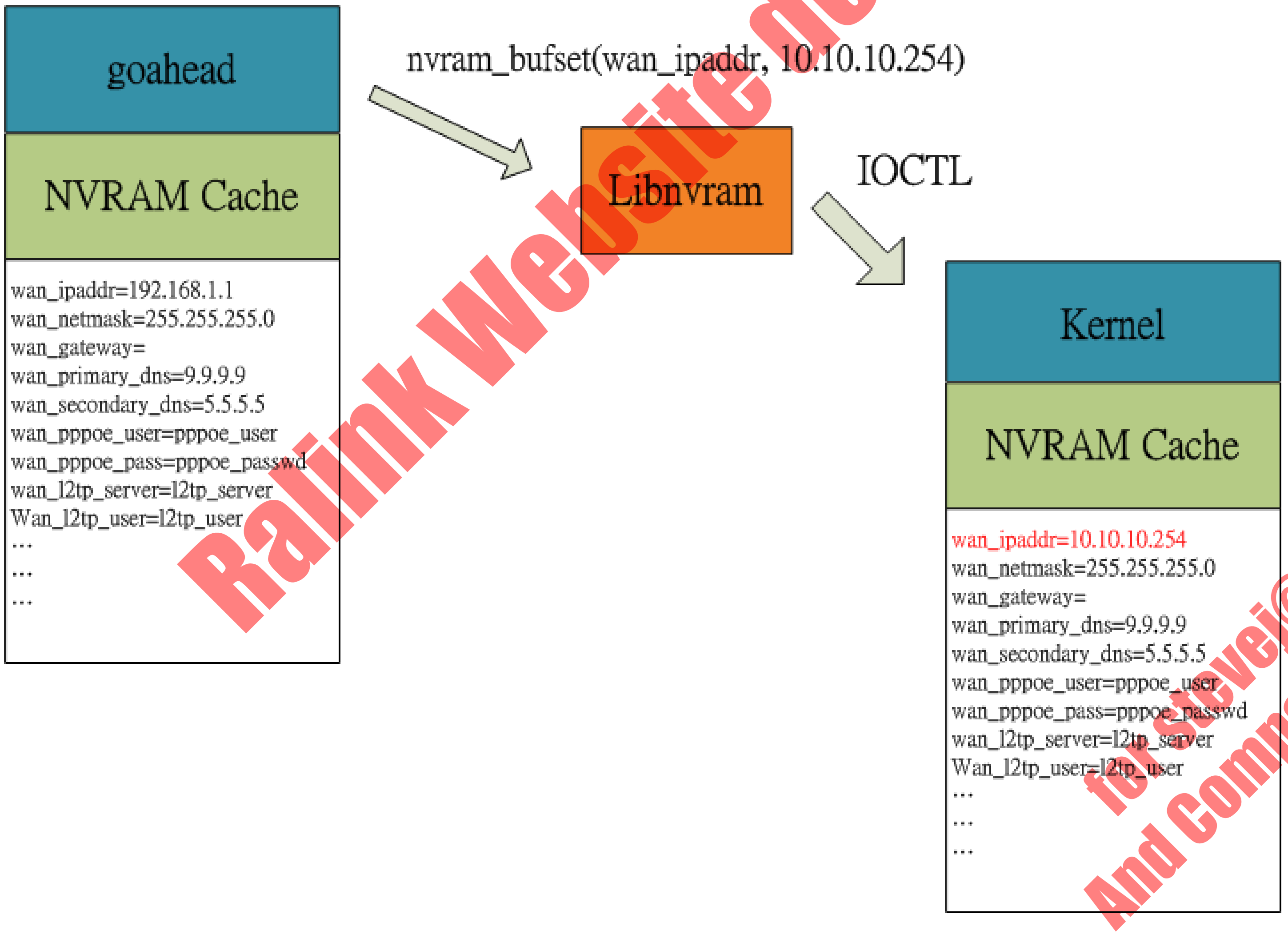
Kernel

NVRAM Cache

wan_ipaddr=10.10.10.254
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…

nvram_bufget(wan_ipaddr)
= 10.10.10.254

New Design:
Libnvram nvram_set()



goahead

NVRAM Cache

```
wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

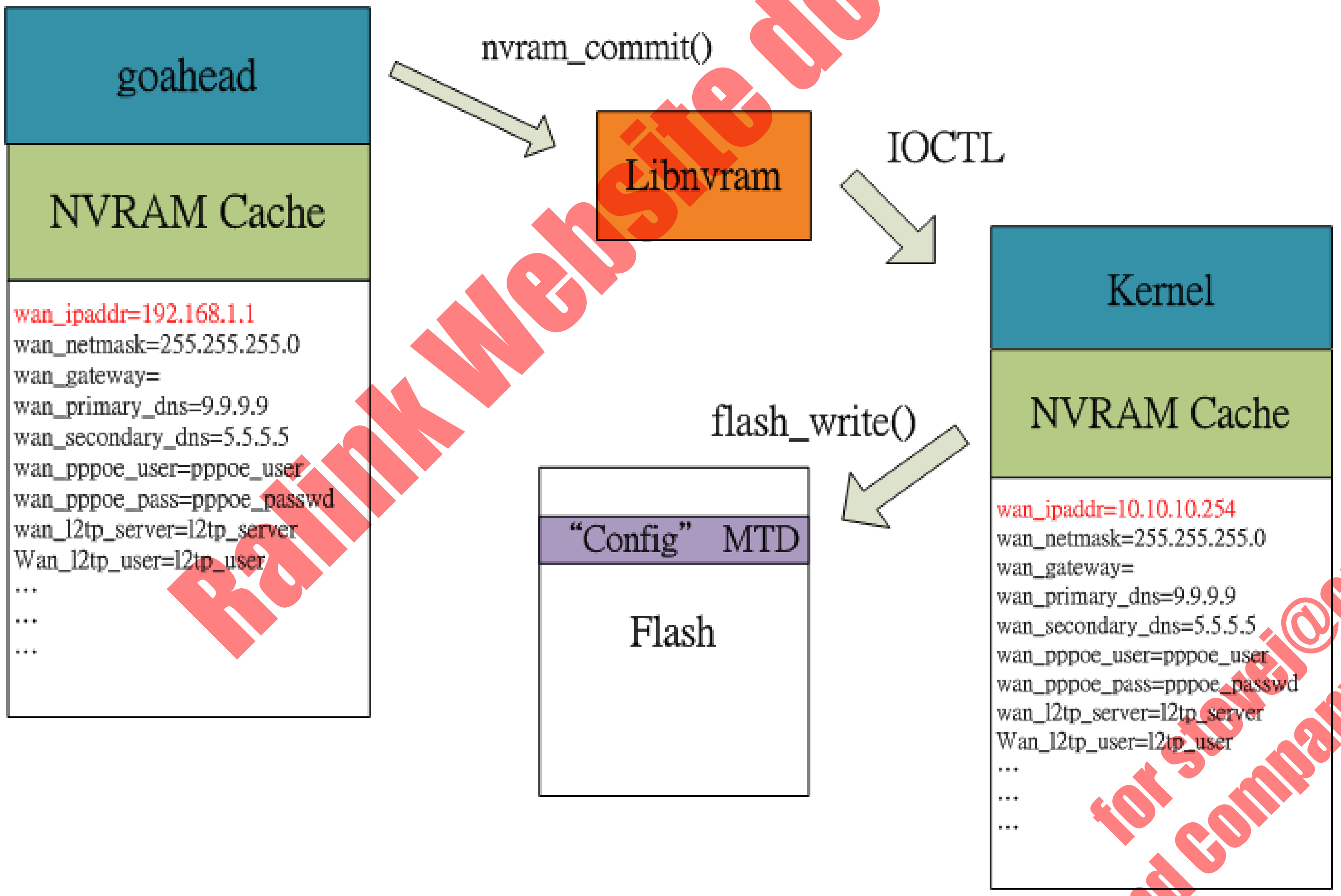
nvram_bufset(wan_ipaddr, 10.10.10.254)

Kernel

NVRAM Cache

```
wan_ipaddr=10.10.10.254
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…
```

New Design:
Libnvram nvram_commit()



goahead

NVRAM Cache

wan_ipaddr=192.168.1.1
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…

nvram_commit()

Libnvram

IOCTL

Kernel

NVRAM Cache

wan_ipaddr=10.10.10.254
wan_netmask=255.255.255.0
wan_gateway=
wan_primary_dns=9.9.9.9
wan_secondary_dns=5.5.5.5
wan_pppoe_user=pppoe_user
wan_pppoe_pass=pppoe_passwd
wan_l2tp_server=l2tp_server
Wan_l2tp_user=l2tp_user
…
…
…

flash_write()

"Config" MTD

Flash

New Design:
Libnvram nvram_commit()

- **Which copy of NVRAM cache will be committed to Flash?**
  - ➔ Kernel NVRAM cache

- **Why?**
  - ➔ Kernel NVRAM cache keeps the freshest data

Code Review

- **New added files:**
  - source/linux-2.6.21.x/arch/mips/rt2880/nvram.c
  - source/linux-2.6.21.x/arch/mips/rt2880/nvram.h
  - source/linux-2.6.21.x/arch/mips/rt2880/crc32.c
- **Modified files:**
  - source/lib/libnvram/nvram_env.c
  - source/lib/libnvram/nvram.h
  - source/vendors/Ralink/RT3052/makedevlinks

Test Scenario

- **Memory leak**

1. Program A keeps calling nvram_bufget()
2. Program B keeps calling nvram_bufset()
3. Use “free” command to check if system free memory is decreasing

- **Kernel NVRAM semaphore**

1. Program A calls X_1 times nvram_bufget(), Y_1 times nvram_bufset(), and Z_1 times nvram_commit()
2. Program A calls X_2 times nvram_bufget(), Y_2 times nvram_bufset(), and Z_2 times nvram_commit()
3. Increase global counter in Kernel NVRAM critical setction
4. Check if the value of global counter = “$X_1 + X_2 + Y_1 + Y_2 + Z_1 + Z_2$ “

Test Scenario

- **Synchronization**
  - Program A and Program B increase a nvram value in turn
  - Program A increase X times, Program B increase Y times
  - Check if that nvram value equals to (X + Y)