# A Generic C/C++ API for Wireless Interfaces in GNU/Linux Systems

Volkan Yazıcı
Özyeğin University
Dept. of Electrical & Electronics Engineering
volkan.yazici@ozyegin.edu.tr

WISERLAB Group Meeting, İstanbul, 2010

- Intro.
  - Motivation
  - Literature
- 2 Contrib.
- Status
  - Where Are We?
  - hostapd
  - Roadmap
- Demo
  - Getters
  - Scanning
- Doc.
- Questions

#### Motivation

- A generic C/C++ API for network interfaces.
  - Almost full coverage of wireless network configurations.
    - iwconfig
    - iwlist
    - wlanconfig
  - Generic network interface operations.
    - ifconfig
    - route
- A library, not a command line tool.
- A core component of the Connectivity Brokerage.

- A majority of the previous works were basically parsing command (e.g. iwlist) outputs.
- Accessing command line tools within a program execution flow is dangerous and not convenient.
  - Command outputs are unreliable.
    - High possibility of unknown outputs (bugs).
    - Hard to be in sync. with every release.
    - Requires a significant amount of work to parse and extract certain tokens.

# Literature (Cont'd)

- Other libraries with similar goals:
  - libiw of Wireless Tools for Linux,
  - libnm of NetworkManager.
- Cons:
  - not designed for embedded environments (library dependencies),
  - function calls are blocking,
  - lacks ifup/ifdown and routing table functionalities,
  - no fine-grained access to the configurations,
  - doesn't provide an easy to use unified API.

#### Contribution

- a unified C/C++ library with on par features compared to iwconfig, iwlist, wlanconfig, ifconfig, route tools.
- not yet another command wrapper, but a written from scratch library accessing network drivers through ioctl() calls to the kernel and nl80211 netlink.
- fine-grained access, that is, just get/set a single property at a time.
- non-blocking function calls.
- code base is not bloated with 10 year old backward compatibility issues.
- fully documented, clean code; suitable for research & education purposes.

Where Are We? hostand Roadman

#### Where Are We?

#### Finished!

- Website & Docs.: http://vy.github.com/wapi/
- Source Code: http://www.github.com/vy/wapi/
- Tested on various hardware. (Atheros, Broadcom, etc.)
- Fully documented code base. (Thanks Doxygen.)
- Now available as an official OpenWrt package!

Contrib. Status Demo Doc. Questions Where Are We? hostapd Roadmap

### hostapd

- We can alter client side configurations?
- What about AP side?
- hostapd is the de-facto user space daemon,
- but provides restrictions in terms of access:
  - You are not allowed to externally alter the configurations of wireless devices that are allocated by hostapd.
  - Access to hostapd is allowed in two ways:
    - hostapd control interface (undocumented)
    - hostapd.conf (in machine-readable, mature format)

tro. Contrib. Status Demo Doc. Questions Where Are We? hostapd Roadmap

#### Roadmap

- Get rid off libiw dependency.
  - Because of a race-condition bug in nl80211, delayed for a while.

#### Getters: Source Code

```
int ret:
double freq;
wapi_freq_flag_t freq_flag;
char essid[WAPI_ESSID_MAX_SIZE + 1];
wapi_essid_flag_t essid_flag;
/* freq */
ret = wapi_get_freq(sock, ifname, &freq, &freq_flag);
printf("freq: %g\n", freq);
printf("freq_flags: %s\n", wapi_freq_flags[freq_flag]);
/* essid */
ret = wapi_get_essid(sock, ifname, essid, &essid_flag);
printf("essid: %s\n", essid);
printf("essid_flag: %s\n", wapi_essid_flags[essid_flag]);
```

#### Getters: Output

```
ip: 192.168.1.111
netmask: 255.255.255.0
freq: 2.412e+09
freq_flag: WAPLFREQ.AUTO
chan: 1
freq: 2.412e+09
essid : ozu
essid_flag: WAPLESSID_ON
mode: WAPLMODE_MANAGED
ap: 00:14:C1:34:CE:83
wireless.c:451: wapi_get_bitrate(): Bitrate is disabled.
wireless.c:546: wapi_get_txpower():ioctl(SIOCGWTXPOW): Invalid argument
```

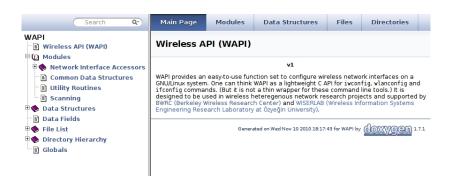
```
int sleepdur = 1;
int sleeptries = 5;
wapi_list_t list;
wapi_scan_info_t *info:
int ret;
/* start scan */
ret = wapi_scan_init(sock, ifname);
printf("wapi_scan_init(): ret: %d\n", ret);
/* wait for completion */
do {
    sleep (sleepdur):
    ret = wapi_scan_stat(sock, ifname);
    printf("wapi_scan_stat(): ret: %d, sleeptries: %d\n", ret, sleeptries);
} while (--sleeptries > 0 \&\& ret > 0);
if (ret < 0) return:
/* collect results */
bzero(&list , sizeof(wapi_list_t));
ret = wapi_scan_coll(sock, ifname, &list);
printf("wapi_scan_coll(): ret: %d\n", ret);
/* print found aps */
for (info = list.head.scan; info; info = info->next)
    printf(">> @%x| %s\n", (size_t) info, (info->has_essid ? info->essid : ""));
```

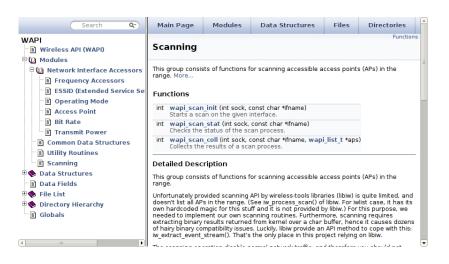
## Scanning: Output

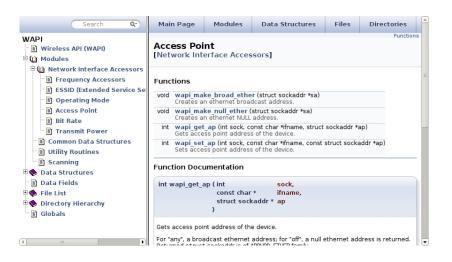
```
wapi_scan_init(): ret: 0
wapi_scan_stat(): ret: 1, sleeptries: 5
wapi_scan_stat(): ret: 0, sleeptries: 4
wapi_scan_coll(): ret: 0
>> @8214ce8l OzuNet. Visitors
>> @8214c901 Wozunacad
>> @8214c38L OzuNet. Visitors
>> @8214be0| canbazoglu
>> @8214b88I NWii
>> @8214b30I wozunguest
>> @8214ad8| Wozunacad
>> @8214a80| ZQNET29
>> @8214a281 wozunvisitor
>> ...
```

### Doxygen

- All available public API (functions, variables, constants, structs, enums, etc.) is fully documented.
- Driver specific turnarounds are documented.
- All documentation is integrated within the source code itself as comment lines.
- Using Doxygen to produce HTML, LaTeX, RTF, etc. output from these comment lines.









```
00483
          return ret;
00484 }
00485
00486
00487 int
00488 wapi set txpower(
00489
          int sock.
00490
          const char *ifname.
00491
          int power.
00492
          wapi txpower flag t flag)
00493 {
00494
          struct iwreq wrq;
00495
00496
          /* Construct the request. */
00497
          wrq.u.txpower.value = power;
00498
          switch (flag)
00499
          case WAPI TXPOWER DBM:
00500
              wrq.u.txpower.flags = IW TXPOW DBM;
00501
00502
              break;
00503
          case WAPI TXPOWER MWATT:
00504
              wrq.u.txpower.flags = IW TXPOW MWATT;
00505
              break:
00506
          case WAPI TXPOWER RELATIVE:
00507
              wrg.u.txpower.flags = IW TXPOW RELATIVE;
00508
              break:
00509
          }
00510
00511
          /* Issue the set command. */
          return wapi ioctl(sock, ifname, SIOCSIWTXPOW, &wrg);
00512
00513 }
00514
00515
```

#### Questions?



answers of questions

I'm Feeling Lucky