

Austin's blog

I make things

Sending raw Ethernet packets from a specific interface in C on Linux

Posted on ~~September 14, 2011~~[June 3, 2012](#) by [austinmarton](#)

Lately I've been writing some code to send packets to a specific MAC address from a specific interface. I'm sure this will come in handy again so here is how it goes:

Includes:

(might not need all of these)

```
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <net/if.h>
#include <netinet/ip.h>
#include <netinet/udp.h>
#include <netinet/ether.h>
#include <linux/if_packet.h>
```

Open the raw socket:

```
int sockfd;
...
/* Open RAW socket to send on */
if ((sockfd = socket(AF_PACKET, SOCK_RAW, IPPROTO_RAW)) == -1) {
    perror("socket");
}
```

Get the index of the interface to send on:

```
struct ifreq if_idx;
...
memset(&if_idx, 0, sizeof(struct ifreq));
strncpy(if_idx.ifr_name, "eth0", IFNAMSIZ-1);
if (ioctl(sock, SIOCGIFINDEX, &if_idx) < 0)
    perror("SIOCGIFINDEX");
```

Get the MAC address of the interface to send on:

```
struct ifreq if_mac;
...
memset(&if_mac, 0, sizeof(struct ifreq));
strncpy(if_mac.ifr_name, "eth0", IFNAMSIZ-1);
if (ioctl(sock, SIOCGIFHWADDR, &if_mac) < 0)
    perror("SIOCGIFHWADDR");
```

Get the IP address of the interface to send on:

```

struct ifreq if_ip;
...
memset(&if_ip, 0, sizeof(struct ifreq));
strncpy(if_ip.ifr_name, "eth0", IFNAMSIZ-1);
if (ioctl(sock, SIOCGIFADDR, &if_ip) < 0)
    perror("SIOCGIFADDR");

```

Construct the Ethernet header:

```

int tx_len = 0;
char sendbuf[1024];
struct ether_header *eh = (struct ether_header *) sendbuf;
...
memset(sendbuf, 0, 1024);
/* Ethernet header */
eh->ether_shost[0] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[0];
eh->ether_shost[1] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[1];
eh->ether_shost[2] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[2];
eh->ether_shost[3] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[3];
eh->ether_shost[4] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[4];
eh->ether_shost[5] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[5];
eh->ether_dhost[0] = MY_DEST_MAC0;
eh->ether_dhost[1] = MY_DEST_MAC1;
eh->ether_dhost[2] = MY_DEST_MAC2;
eh->ether_dhost[3] = MY_DEST_MAC3;
eh->ether_dhost[4] = MY_DEST_MAC4;
eh->ether_dhost[5] = MY_DEST_MAC5;
eh->ether_type = htons(ETH_P_IP);
tx_len += sizeof(struct ether_header);

```

Construct the IP header:

```

struct iphdr *iph = (struct iphdr *) (sendbuf + sizeof(struct ether_header));
...
/* IP Header */
iph->ihl = 5;
iph->version = 4;
iph->tos = 16; // Low delay
iph->id = htons(54321);
iph->ttl = ttl; // hops
iph->protocol = 17; // UDP
/* Source IP address, can be spoofed */
iph->saddr = inet_addr(inet_ntoa(((struct sockaddr_in *)&if_ip.ifr_addr)->sin_addr));
// iph->saddr = inet_addr("192.168.0.112");
/* Destination IP address */
iph->daddr = inet_addr("192.168.0.111");
tx_len += sizeof(struct iphdr);

```

Construct the UDP header:

```

struct udphdr *udph = (struct udphdr *) (sendbuf + sizeof(struct iphdr) + sizeof(struct ether_header));
...
/* UDP Header */
udph->source = htons(3423);
udph->dest = htons(5342);
udph->check = 0; // skip
tx_len += sizeof(struct udphdr);

```

Fill in UDP payload:

```
/* Packet data */
sendbuf[tx_len++] = 0xde;
sendbuf[tx_len++] = 0xad;
sendbuf[tx_len++] = 0xbe;
sendbuf[tx_len++] = 0xef;
```

Fill in remaining header info:

```
unsigned short csum(unsigned short *buf, int nwords)
{
    unsigned long sum;
    for(sum=0; nwords>0; nwords--)
        sum += *buf++;
    sum = (sum >> 16) + (sum & 0xffff);
    sum += (sum >> 16);
    return (unsigned short)(~sum);
}
...
/* Length of UDP payload and header */
udph->len = htons(tx_len - sizeof(struct ether_header) - sizeof(struct iphdr));
/* Length of IP payload and header */
iph->tot_len = htons(tx_len - sizeof(struct ether_header));
/* Calculate IP checksum on completed header */
iph->check = csum((unsigned short *) (sendbuf+sizeof(struct ether_header)), sizeof(struct iphdr)/2);
```

Send the raw Ethernet packet:

```
/* Destination address */
struct sockaddr_ll socket_address;
...
/* Index of the network device */
socket_address.sll_ifindex = if_idx.ifr_ifindex;
/* Address length */
socket_address.sll_halen = ETH_ALEN;
/* Destination MAC */
socket_address.sll_addr[0] = MY_DEST_MAC0;
socket_address.sll_addr[1] = MY_DEST_MAC1;
socket_address.sll_addr[2] = MY_DEST_MAC2;
socket_address.sll_addr[3] = MY_DEST_MAC3;
socket_address.sll_addr[4] = MY_DEST_MAC4;
socket_address.sll_addr[5] = MY_DEST_MAC5;
/* Send packet */
if (sendto(sock, sendbuf, tx_len, 0, (struct sockaddr*)&socket_address, sizeof(struct sockaddr_ll)) < 0)
    printf("Send failed\n");
```

Update:

As in the comments, I've written a working example that can be found here: <https://gist.github.com/1922600> (<https://gist.github.com/1922600>)

Change the destination MAC address (e.g. 00:11:22:33:44:55) and compile:

```
gcc sendRawEth.c -o sendRawEth
```

In one terminal run tcpdump to observe the packets:

```
sudo tcpdump -nettti eth0 '(ether dst host 00:11:22:33:44:55)'
```

And in another run the program as root:

```
sudo ./sendRawEth eth0
```

59 thoughts on “Sending raw Ethernet packets from a specific interface in C on Linux”

1. Jonathan Andrews says:
[February 10, 2012 at 11:30 am](#)
Thank you 😊 Exactly the example I was looking for, is it possible to download the source ?

[Reply](#)
o austinmarton says:
[February 13, 2012 at 11:23 pm](#)
Hi Jon,

I'm stoked someone else has found it useful! Those snippets should include everything you need but I'll try put a full compiling example up on github for you soon.

Cheers,
Austin.

[Reply](#)
2. Jonathan Andrews says:
[February 16, 2012 at 6:28 pm](#)
I have it working 😊

I'm sending 8 UDP broadcast packets each with a different IP address from a single ethernet card. Its being used to test an audio streaming application (it emulates 8 stream sources at once). Linux will let you have multiple IP addresses per card, but it seems will not let you elect to UDP broadcast from a specific one. Many thanks, great work, Jon

[Reply](#)
3. Awais says:
[February 24, 2012 at 5:17 am](#)
Hi All
i just want to send ethernet packet, a frame not layer 3,4 packet. is it possible to do this in java or C ?
Thanks

[Reply](#)
o austinmarton says:
[February 27, 2012 at 8:45 am](#)
Hi Awais,

Of course it is possible to send a raw Ethernet packet in C! Just ignore the section of the above code that does the UDP and IP headers.

I've written a little example here:

1	/*
2	* This program is free software: you can redistribute it and/or modify
3	* it under the terms of the GNU General Public License as published by
4	* the Free Software Foundation, either version 3 of the License, or
5	* (at your option) any later version.
6	*/
7	
8	#include <arpa/inet.h>
9	#include <linux/if_packet.h>

10	#include <stdio.h>
11	#include <string.h>
12	#include <stdlib.h>
13	#include <sys/ioctl.h>
14	#include <sys/socket.h>
15	#include <net/if.h>
16	#include <netinet/ether.h>
17	
18	#define MY_DEST_MAC0 0x00
19	#define MY_DEST_MAC1 0x00
20	#define MY_DEST_MAC2 0x00
21	#define MY_DEST_MAC3 0x00
22	#define MY_DEST_MAC4 0x00
23	#define MY_DEST_MAC5 0x00
24	
25	#define DEFAULT_IF "eth0"
26	#define BUF_SIZ 1024
27	
28	int main(int argc, char *argv[])
29	{
30	int sockfd;
31	struct ifreq if_idx;
32	struct ifreq if_mac;
33	int tx_len = 0;
34	char sendbuf[BUF_SIZ];
35	struct ether_header *eh = (struct ether_header *) sendbuf;
36	struct iphdr *iph = (struct iphdr *) (sendbuf + sizeof(struct ether_header));
37	struct sockaddr_ll socket_address;
38	char ifName[IFNAMSIZ];
39	
40	/* Get interface name */
41	if (argc > 1)
42	strcpy(ifName, argv[1]);
43	else
44	strcpy(ifName, DEFAULT_IF);
45	
46	/* Open RAW socket to send on */
47	if ((sockfd = socket(AF_PACKET, SOCK_RAW, IPPROTO_RAW)) == -1) {
48	perror("socket");
49	}
50	
51	/* Get the index of the interface to send on */
52	memset(&if_idx, 0, sizeof(struct ifreq));
53	strncpy(if_idx.ifr_name, ifName, IFNAMSIZ-1);
54	if (ioctl(sockfd, SIOCGIFINDEX, &if_idx) < 0)
55	perror("SIOCGIFINDEX");
56	/* Get the MAC address of the interface to send on */
57	memset(&if_mac, 0, sizeof(struct ifreq));
58	strncpy(if_mac.ifr_name, ifName, IFNAMSIZ-1);
59	if (ioctl(sockfd, SIOCGIFHWADDR, &if_mac) < 0)
60	perror("SIOCGIFHWADDR");
61	
62	/* Construct the Ethernet header */
63	memset(sendbuf, 0, BUF_SIZ);
64	/* Ethernet header */
65	eh->ether_shost[0] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[0];
66	eh->ether_shost[1] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[1];
67	eh->ether_shost[2] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[2];
68	eh->ether_shost[3] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[3];
69	eh->ether_shost[4] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[4];
70	eh->ether_shost[5] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[5];
71	eh->ether_dhost[0] = MY_DEST_MAC0;
72	eh->ether_dhost[1] = MY_DEST_MAC1;
73	eh->ether_dhost[2] = MY_DEST_MAC2;

74	eh->ether_dhost[3] = MY_DEST_MAC3;
75	eh->ether_dhost[4] = MY_DEST_MAC4;
76	eh->ether_dhost[5] = MY_DEST_MAC5;
77	/* Ethertype field */
78	eh->ether_type = htons(ETH_P_IP);
79	tx_len += sizeof(struct ether_header);
80	
81	/* Packet data */
82	sendbuf[tx_len++] = 0xde;
83	sendbuf[tx_len++] = 0xad;
84	sendbuf[tx_len++] = 0xbe;
85	sendbuf[tx_len++] = 0xef;
86	
87	/* Index of the network device */
88	socket_address.sll_ifindex = if_idx.ifr_ifindex;
89	/* Address length */
90	socket_address.sll_halen = ETH_ALEN;
91	/* Destination MAC */
92	socket_address.sll_addr[0] = MY_DEST_MAC0;
93	socket_address.sll_addr[1] = MY_DEST_MAC1;
94	socket_address.sll_addr[2] = MY_DEST_MAC2;
95	socket_address.sll_addr[3] = MY_DEST_MAC3;
96	socket_address.sll_addr[4] = MY_DEST_MAC4;
97	socket_address.sll_addr[5] = MY_DEST_MAC5;
98	
99	/* Send packet */
100	if (sendto(sockfd, sendbuf, tx_len, 0, (struct sockaddr*)&socket_address, sizeof(struct sockaddr_ll)) < 0)
101	printf("Send failed\n");
102	
103	return 0;
104	}

view raw

sendRawEth.c

hosted with ❤ by [GitHub](#)

If you compile and run that app as root you can view the packets going out on the specified interface using wireshark.

Cheers,

Austin.

[Reply](#).

4. hassan says:

[March 6, 2012 at 6:58 am](#)

Hi Austin,

I have tried your example code <https://gist.github.com/1922600> . Its compilation is fine with no errors but I'm not getting packets going out of the interface using wireshark.....Just for confirmation.... Have you run that code or you have just provided the code???.....And guide me if there are more steps involved which I am not completing....And one thing more At run time what arguments is it needed...??

Thanks

[Reply](#).

o austinmarton says:

[March 7, 2012 at 1:03 am](#)

I have compiled and run the code on Linux:

```
gcc sendRawEth.c -o sendRawEth
```

You will need to run the executable as root, or using "sudo":

```
sudo ./sendRawEth eth0
```

Check that you are watching on the same interface as the packet is being sent.

Also, because I have used the IP EtherType (0x0800) and then just put random data in, it will come up as "Malformed Packet" in Wireshark. Unlikely but possible your NIC would reject it on that basis but you should see it in Wireshark at least.

Cheers,

Austin.

[Reply](#).

o Victor Ramos says:

[December 28, 2012 at 8:50 am](#)

How did you put random data in? Could you add discrete data instead?

o austinmarton says:

December 29, 2012 at 11:49 pm

Sorry I shouldn't have said "random" data, I was referring to the bytes that I had hard coded as the packet data (0xde, 0xad, 0xbe, 0xef – see line 81 of the gist).

You can easily add your own payload by copying the bytes into 'sendbuf' and incrementing 'tx_len'.

I could add some code to parse packet data from a command line argument if needed.

- o **Victor Ramos** says:

December 30, 2012 at 6:47 am

Where exactly would you expect to see the data if viewing packets in Wireshark?? What I'm trying to do is send raw data to a wireless transceiver that tells a PIC chip to turn on a SSR. Any ideas? YThanks for the help!

- o **Chris Miller** says:

November 15, 2013 at 1:50 pm

I too witnessed "packet not going out" which had me going for some time. I was using an Asus laptop with its built-in NIC set up with six VLANs. All the VLANs were basically working in that I could ping devices on all different VLANs, and then look at the arp table and see the expected entries against each sub-interface, but packets generated using this code were not going out – not according to tcpdump, at least.

The same software was working on another machine in the same version of Fedora, so I was pretty sure the software was correct.

In the end, I added a second NIC (elderly 100Mbit/s Belkin, USB-attached), reconfigured my interfaces to run over that NIC and everything started working correctly.

My tentative conclusion is that the drivers or the hardware of the built-in NIC were not handling RAW sockets correctly over sub-interfaces. The suspect NIC reports as "Tigon3 [partno(BCM95764m) rev 5784100]" in a dmesg report, and its MAC is interpreted as Wistron by Wireshark. Kernel version is "2.6.35.14-106"

Reply

- o **Chris Miller** says:

November 15, 2013 at 11:15 pm

When putting up the previous post, I completely and shamefully forgot to thank Austin for putting up this code – it helped me a very great deal.

- o **austinmarton** says:

November 21, 2013 at 9:57 am

Good info, cheers for your comment Chris

5. **Mehdi** says:

March 6, 2012 at 10:08 am

Hi Austin,

I am a student, currently trying to develop an Ethernet communication link using a research protocol. But I need first to send raw ethernet packets from a machine to an other using MAC addresses. Since I'm a newbie in the field, I would need some help if you don't mind.

I downloaded your file but I think I need the "receive" file to run on the other machine. Can you provide me with this?

Cheers,

Mehdi

Reply

- o **Mehdi** says:

March 10, 2012 at 1:42 pm

Anyone here? I would be very grateful if could give me some help. Thanks

Reply

- o **austinmarton** says:

March 10, 2012 at 9:28 pm

Hi Mehdi,

Don't worry I haven't forgotten about this. I will try find some time to put up the receive code soon.

Cheers,

Austin.

6. **Mehdi** says:

March 15, 2012 at 8:55 pm

Actually, I'm trying to learn by myself how to code with RAW sockets. Do you have any kind of tutorial for newbies or some step-by-step guide? Thank you for your help 😊

Reply

- o **austinmarton** says:

June 3, 2012 at 8:21 am

Not sure if you are still looking but I finally got around to putting up the code for receiving on raw sockets:

<https://austinmarton.wordpress.com/2012/06/03/receiving-raw-packets-in-linux-without-pcap/>

Reply

7. **swehack** says:

March 29, 2012 at 11:49 am

Hi Austin

I tried your code from the github gist posting and it compiles fine but I can't see any packet going out on eth0.

I changed MY_DEST_MAC0 to my destination MAC in the same LAN using the format 0x001122334455.

Then I setup tcpdump -nettti eth0 |grep 44:55 to find a packet going to that mac address but nothing comes out. I also tried without the grep but with a bpf to filter out the noise on heavily used ports but still no packet.

So what am I doing wrong here?

Reply

- o **austinmarton** says:

March 29, 2012 at 9:12 pm

I observed the same problem when using grep on tcpdump, but when I ran tcpdump by itself I can see the packets (on a separate interface that isn't flooded with other traffic).

In short – you need to use tcpdump filters instead of grep, the following command will show packets with destination MAC address 00:11:22:33:44:55 on interface eth0:

```
sudo tcpdump -nettti eth0 '(ether dst host 00:11:22:33:44:55)'
```

Cheers,

Austin.

Reply

- o **swehack** says:

March 30, 2012 at 4:57 am

You're right I do faintly remember being told never to grep tcpdump, thanks for showing how to filter out mac addresses.

However, this time I tried it with a filter setting of ether dst host 11:22:33... on a LAN and still I see no packets. I tried pinging the same LAN host that I stole the destination mac address from and sure enough tcpdump shows an icmp echo request on that mac.

- o **austinmarton** says:

March 30, 2012 at 9:53 am

Not sure what's different with your set up. I modified the code from github like so:

```
#define MY_DEST_MAC0 0x00
#define MY_DEST_MAC1 0x11
#define MY_DEST_MAC2 0x22
#define MY_DEST_MAC3 0x33
#define MY_DEST_MAC4 0x44
#define MY_DEST_MAC5 0x55
```

recompiled:

```
gcc sendRawEth.c -o sendRawEth
```

then ran tcpdump:

```
sudo tcpdump -nettti eth0 '(ether dst host 00:11:22:33:44:55)'
```

then ran the program:

```
sudo sendRawEth
```

and in tcpdump I can see the packets.

Tested on Ubuntu 10.04 and 11.10.

- o **swehack** says:

March 30, 2012 at 10:00 am

Oh I see Austin! This is exactly why I specified the format I used on the mac address, because I wasn't sure it was correct. You've shown me now that the correct format is to use all 6 macros.

And now it works! You're the best Austin. ❤️

- o **Frank** says:

October 30, 2012 at 1:27 pm

Jenius! Austin! You save me. haha. For me, I use this example for wireless (e.g. ad-hoc), the both wireless cards have a same essid & channel. And run : sudo tcpdump -nettti wlan0 '(ether dst host 00:11:22:33:44:55)' On the other card: sudo ./sendRawEth wlan0 everything goes well.....

- o **austinmarton** says:

October 30, 2012 at 9:22 pm

That's very cool! I hadn't thought about using it for ad-hoc wifi networks. Thanks for the info. Austin.

- o **mrfrankliu** says:

October 30, 2012 at 10:21 pm

cheers

- o **Frank LIU** says:

November 2, 2012 at 9:01 pm

cool

8. **manrico** says:

April 7, 2012 at 9:32 am

is there a similar code in C for Windows?

thanks

Reply

9. **Receiving raw packets in Linux without pcap « Austin's Weblog** says:

June 3, 2012 at 8:18 am

[...] there was a lot of interest in my post on sending raw Ethernet packets, this is an example of receiving packets on a raw socket. A compiling version can be found on [...]

[Reply](#)

10. **Chethan Pandarinath** says:

[September 20, 2012 at 3:48 pm](#)

This code is fantastic, thanks so much for posting it. Tried out the linked github source and it worked flawlessly the first time. After beating my head against networking code for several hours, this made my day.

[Reply](#)

o **austinmarton** says:

[September 20, 2012 at 9:04 pm](#)

Great! I'm really pleased it was useful to you 😊

[Reply](#)

11. **Thomas Waldecker** says:

[September 22, 2012 at 8:19 am](#)

Hi Austin,

thanks for your code. Especially for the gist. I thought it is bad to compile the MAC address in the program, so I forked your gist and wrote the bits to specify the MAC address with a command line argument.

Kind regards,

Thomas

[Reply](#)

o **Thomas Waldecker** says:

[September 22, 2012 at 8:21 am](#)

Forgot the link:

1	/*
2	* Added the command line arguments for interface and MAC Address
3	*
4	* Based on raw Ethernet from austinmarton: https://gist.github.com/1922600
5	*
6	* This program is free software: you can redistribute it and/or modify
7	* it under the terms of the GNU General Public License as published by
8	* the Free Software Foundation, either version 3 of the License, or
9	* (at your option) any later version.
10	*/
11	
12	#include <arpa/inet.h>
13	#include <linux/if_packet.h>
14	#include <stdio.h>
15	#include <string.h>
16	#include <stdlib.h>
17	#include <sys/ioctl.h>
18	#include <sys/socket.h>
19	#include <net/if.h>
20	#include <netinet/ether.h>
21	
22	#define DEFAULT_IF "eth0"
23	#define BUF_SIZ 1024
24	
25	int main(int argc, char *argv[])
26	{
27	int sockfd;
28	struct ifreq if_idx;
29	struct ifreq if_mac;
30	int tx_len = 0;
31	char sendbuf[BUF_SIZ];
32	struct ether_header *ch = (struct ether_header *) sendbuf;
33	struct iphdr *iph = (struct iphdr *) (sendbuf + sizeof(struct ether_header));
34	struct sockaddr_ll socket_address;
35	char ifName[IFNAMSIZ];
36	unsigned int mac[6]; //using as uint8_t
37	

```

38  /* Get interface name */
39  if (argc > 1){
40      strcpy(ifName, argv[1]);
41      if (argc > 2) { /* 2 arguments, second argument is mac */
42          sscanf(argv[2], "%02x:%02x:%02x:%02x:%02x:%02x", &mac[0], &mac[1], &mac[2], &mac[3], &mac[4], &mac[5]);
43          //printf("mac: \n");
44          //printf("%02x:%02x:%02x:%02x:%02x:%02x\n", mac[0], mac[1], mac[2], mac[3], mac[4], mac[5]);
45      }
46  }
47  else
48      strcpy(ifName, DEFAULT_IF);
49
50  /* Open RAW socket to send on */
51  if ((sockfd = socket(AF_PACKET, SOCK_RAW, IPPROTO_RAW)) == -1) {
52      perror("socket");
53  }
54
55  /* Get the index of the interface to send on */
56  memset(&if_idx, 0, sizeof(struct ifreq));
57  strncpy(if_idx.ifr_name, ifName, IFNAMSIZ-1);
58  if (ioctl(sockfd, SIOCGIFINDEX, &if_idx) < 0)
59      perror("SIOCGIFINDEX");
60  /* Get the MAC address of the interface to send on */
61  memset(&if_mac, 0, sizeof(struct ifreq));
62  strncpy(if_mac.ifr_name, ifName, IFNAMSIZ-1);
63  if (ioctl(sockfd, SIOCGIFHWADDR, &if_mac) < 0)
64      perror("SIOCGIFHWADDR");
65
66  /* Construct the Ethernet header */
67  memset(sendbuf, 0, BUF_SIZ);
68  /* Ethernet header */
69  eh->ether_shost[0] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[0];
70  eh->ether_shost[1] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[1];
71  eh->ether_shost[2] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[2];
72  eh->ether_shost[3] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[3];
73  eh->ether_shost[4] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[4];
74  eh->ether_shost[5] = ((uint8_t *)&if_mac.ifr_hwaddr.sa_data)[5];
75  eh->ether_dhost[0] = mac[0];
76  eh->ether_dhost[1] = mac[1];
77  eh->ether_dhost[2] = mac[2];
78  eh->ether_dhost[3] = mac[3];
79  eh->ether_dhost[4] = mac[4];
80  eh->ether_dhost[5] = mac[5];
81  /* Ethertype field */
82  eh->ether_type = htons(ETH_P_IP);
83  tx_len += sizeof(struct ether_header);
84
85  /* Packet data */
86  sendbuf[tx_len++] = 0xde;
87  sendbuf[tx_len++] = 0xad;
88  sendbuf[tx_len++] = 0xbe;
89  sendbuf[tx_len++] = 0xef;
90
91  /* Index of the network device */
92  socket_address.sll_ifindex = if_idx.ifr_ifindex;
93  /* Address length */
94  socket_address.sll_halen = ETH_ALEN;
95  /* Destination MAC */
96  socket_address.sll_addr[0] = mac[0];
97  socket_address.sll_addr[1] = mac[1];
98  socket_address.sll_addr[2] = mac[2];
99  socket_address.sll_addr[3] = mac[3];
100 socket_address.sll_addr[4] = mac[4];
101 socket_address.sll_addr[5] = mac[5];

```

102	
103	/* Send packet */
104	if (sendto(sockfd, sendbuf, tx_len, 0, (struct sockaddr*)&socket_address, sizeof(struct sockaddr_ll)) < 0)
105	printf("Send failed\n");
106	
107	return 0;
108	}

view raw

sendRawEth.c

hosted with ❤ by GitHub

[Reply](#)

12. **Frank LIU** says:

[November 2, 2012 at 9:11 pm](#)

good

[Reply](#)

13. **Eric Mackay** says:

[November 5, 2012 at 7:31 pm](#)

Wow. You are a LIFESAVER! I have been searching for a clear example of using raw sockets online for weeks, but everyone always says "Don't use them!" and provides no information. This really saves my bacon. Thanks!

[Reply](#)

o **austinmarton** says:

[November 6, 2012 at 5:00 am](#)

No problem! Thanks for your feedback 😊

[Reply](#)

o **hemant** says:

[May 2, 2013 at 1:20 pm](#)

hey can u upload entire code ,that link u gave is expired

o **austinmarton** says:

[May 2, 2013 at 7:59 pm](#)

I think there was a problem with github yesterday, the links to gists weren't working but they seem to be fine now – try again today 😊

14. **Frank LIU** says:

[January 8, 2013 at 1:46 pm](#)

Reblogged this on [Here, we go...](#)!

[Reply](#)

15. **Yahya** says:

[February 25, 2013 at 7:15 am](#)

Hi great post and very useful, using this ive been able to send Ethernet packets over the wireless interface (802.11) just needed to know what kind of sorcery is this 😊 and how its even possible as ethernet (wired) and Wlan (wireless) follow different L2 protocol structures

However Thanks for this great post!

[Reply](#)

o **Vic** says:

[December 5, 2014 at 1:58 am](#)

That's what the data link layer is for

[Reply](#)

16. **hemant** says:

[May 2, 2013 at 2:13 pm](#)

hey ,

i need this code urgent " Sending raw Ethernet packets from a specific interface in C on Linux" can u please send it to my email thanks

[Reply](#)

17. **rrlagic** says:

[July 14, 2013 at 6:12 pm](#)

Simple, clear, useful.

[Reply](#)

18. **NS** says:

[October 21, 2013 at 8:06 am](#)

Hi all,

i have raw IPSEC protected packet which i have to send to my Ethernet interface,

Dst Addr : my IP , MAC address

Src Addr : Tunnel's other end point host ip and MAC,

- 1) Whether it will receive in the application running on my host ?
- 2) Will it go through ip(ipsec) layer of my host for decrypting the packet and pass to the application ?

While Trying this in wireshark, IPSEC packet can be captured but application is not received the packet...

Please help me.

[Reply](#)

19. **Aminos** says:

[May 27, 2014 at 3:22 pm](#)

it's very useful, thank you

[Reply](#)

20. **deepak** says:

[July 24, 2014 at 4:55 pm](#)

Hi ,

I want to send the LLDP packet from the linux kernel .

Can somebody please help me out .

Thanks

[Reply](#)

21. **bearvarinePat** says:

[December 4, 2014 at 11:42 pm](#)

Awesome example. Helped me understand why code I am debugging was not working. I should do a blog too! Thanks, -Pat

[Reply](#)

- o **Vic** says:

[December 5, 2014 at 1:56 am](#)

No problem man. This is a really neat program. I use it to send UDP packets to a wireless microcontroller. Works Great!

[Reply](#)

22. **vanatora** says:

[May 29, 2015 at 2:23 pm](#)

Hi Austin,

thanks for the blog.

I was wondering if its possible to create such raw socket without using any headers, no IP headers for example. I am writing an application which I want to inject my payload without any alteration (i.e. without addition of any headers) into the MAC/PHY.

[Reply](#)

- o **austinmarton** says:

[July 12, 2015 at 10:30 pm](#)

Hello, sorry for the slow reply. Yes that should be possible just skip the section for IP headers. You will of course still need the Ethernet header and to set up the destination socket info. Have you tried that?

[Reply](#)

23. **Govind** says:

[October 29, 2015 at 2:36 am](#)

Hello Austin, I am in a problem. I am writing packet encapsulator code. how to encapsulate ipv4 packet into DVB-S2 frame? upto layer 3 same as TCP/ip client /server but layer 2 frame is totally different than ethernet, iam confused, how to start. can anybody help me?

[Reply](#)

- o **austinmarton** says:

[November 5, 2015 at 10:37 am](#)

Not sure sorry, I have no experience with DVB-S2 but if it's different at layer 2 I would have thought it was handled in the kernel drivers. Couple of things that might be useful:

– <https://wiki.wireshark.org/DVB-S2>

– <http://www.linuxtv.org/wiki/index.php/Dvbstream>

– <http://dvbsnoop.sourceforge.net/dvbsnoop.html>

Cheers.

[Reply](#)

- o **Govind** says:

[November 21, 2015 at 3:28 pm](#)

Austin, thanks for reply.

Iam writing code for DVB-S2 Network program.

Can anybody help me- how to process layer 2 packet? or how to change the layer 2 header of IPv4 packet?

Ethernet layer 2 packet is: Source MAC+Dest MAC+ IPV4 packet+FCS

In my DVB-S2 Network case layer 2packet is like: Flags (4bit) + Length(12 bit)+ Total length (16 bit)+ Dest MAC+ IPv4 packet

Govind Singh Dhama

Kathmandu Nepal

24. **Govind** says:

[January 3, 2016 at 11:17 pm](#)

Dear Austin Thanks for previous support.

I am engaged in a work. can you help me on:

How to get layer-3 IPV4 packet from an audio/video source in c language? this is first part of my work.

Govind Singh Dhimi
Kathmandu, Nepal

[Reply](#)

25. **Eric** says:

[January 14, 2016 at 3:43 pm](#)

I ran your code on Ubuntu at my Mac by virtual machine Parallels 10.
But the following error code always appear!

```
parallels@ubuntu:~/Desktop$ sudo ./test2
SIOCGIFINDEX: No such device
SIOCGIFHWADDR: No such device
Send failed
```

[Reply](#)

o **austinmarton** says:

[January 23, 2016 at 8:18 am](#)

Probably eth0 doesn't exist on your virtual machine. Open a terminal and do `ifconfig` and see what interfaces you have and then modify the code.
You might need to do some additional setup to get networking in Parallels, I haven't used it myself.

[Reply](#)

26. **Govind** says:

[July 24, 2016 at 4:44 pm](#)

Dear Austin, i am working in a college thesis. I have to write "c program for udp streaming with VLC player". Can anybody help me

[Reply](#)

27. **Govind** says:

[August 20, 2016 at 2:48 pm](#)

can anybody help me, how to capture and save "udp packet" from vlc player?

Govind

[Reply](#)

28. **Rafael** says:

[April 1, 2017 at 5:52 am](#)

Thank you for posting this! I thought I'd be spending days trying to make this work

[Reply](#)

[Create a free website or blog at WordPress.com.](#)