int [**ip\_forward**](http://lxr.free-electrons.com/ident?i=ip_forward) (struct [sk\_buff](http://lxr.free-electrons.com/ident?i=sk_buff) \*[skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank))

{

[**U32**](http://lxr.free-electrons.com/ident?i=u32)[**mtu**](http://lxr.free-electrons.com/ident?i=mtu)**;**  //**static int** [**mtu**](http://lxr.free-electrons.com/ident?i=mtu)**[**[**MAX\_UNITS**](http://lxr.free-electrons.com/ident?i=MAX_UNITS)**]; */\* Jumbo MTU for interfaces. \*/***

A **maximum transmission unit** (**MTU**) is the largest size packet or frame, specified in octets (eight-bit bytes), that can be sent in a packet- or frame-based **network** such as the Internet. The Internet's Transmission Control Protocol (TCP) uses the **MTU** to determine the maximum size of each packet in any transmission.

[00191](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html) struct [sk\_buff](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html) {

00192 /\* These two members must be first. \*/

[00193](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m0) struct [sk\_buff](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html) \*next;

[00194](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m1) struct [sk\_buff](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html) \*prev;

00195

[00196](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m2) struct [sk\_buff\_head](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff__head.html) \*list;

[00197](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m3) struct [sock](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsock.html) \*sk;

[00198](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m4) struct timeval stamp;

[00199](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m5) struct net\_device \*dev;

[00200](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m6) struct net\_device \*real\_dev;

00201

00202 union {

[00203](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m7) struct [tcphdr](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structtcphdr.html) \*th;

[00204](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m8) struct [udphdr](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structudphdr.html) \*uh;

[00205](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m9) struct icmphdr \*icmph;

[00206](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m10) struct igmphdr \*igmph;

[00207](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m11) struct [iphdr](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structiphdr.html) \*ipiph;

00208 struct ipv6hdr \*ipv6h;

00209 unsigned char \*raw;

00210 } h;

00211

00212 union {

[00213](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m15) struct [iphdr](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structiphdr.html) \*iph;

[00214](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m12) struct ipv6hdr \*ipv6h;

[00215](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m16) struct arphdr \*arph;

00216 unsigned char \*raw;

00217 } nh;

00218

00219 union {

[00220](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m18) struct ethhdr \*ethernet;

[00221](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m13) unsigned char \*raw;

00222 } mac;

00223

[00224](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m20) struct dst\_entry \*dst;

[00225](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m21) struct sec\_path \*sp;

00226

00227 /\*

00228 \* This is the control buffer. It is free to use for every

00229 \* layer. Please put your private variables there. If you

00230 \* want to keep them across layers you have to do a skb\_clone()

00231 \* first. This is owned by whoever has the skb queued ATM.

00232 \*/

[00233](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m22) char cb[48];

00234

[00235](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m25) unsigned int len,

00236 data\_len,

00237 mac\_len,

00238 csum;

[00239](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m29) unsigned char local\_df,

00240 cloned,

00241 pkt\_type,

00242 ip\_summed;

[00243](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m31) \_\_u32 priority;

[00244](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m33) unsigned short protocol,

00245 security;

00246

00247 void (\*destructor)(struct [sk\_buff](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html) \*skb);

00248 #ifdef CONFIG\_NETFILTER

00249 unsigned long nfmark;

00250 \_\_u32 nfcache;

00251 struct nf\_ct\_info \*nfct;

00252 #ifdef CONFIG\_NETFILTER\_DEBUG

00253 unsigned int nf\_debug;

00254 #endif

00255 #ifdef CONFIG\_BRIDGE\_NETFILTER

00256 struct nf\_bridge\_info \*nf\_bridge;

00257 #endif

00258 #endif /\* CONFIG\_NETFILTER \*/

00259 #if defined(CONFIG\_HIPPI)

00260 union {

00261 \_\_u32 ifield;

00262 } private;

00263 #endif

00264 #ifdef CONFIG\_NET\_SCHED

00265 \_\_u32 tc\_index; /\* traffic control index \*/

00266 #endif

00267

00268 /\* These elements must be at the end, see alloc\_skb() for details. \*/

[00269](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m35) unsigned int truesize;

[00270](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m36) atomic\_t users;

[00271](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html" \l "m39) unsigned char \*head,

00272 \*data,

00273 \*tail,

00274 \*end;

00275 };

00276

struct [sk\_buff\_head](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff__head.html) {

00117 /\* These two members must be first. \*/

[00118](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff__head.html" \l "m0) struct [sk\_buff](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html) \*next;

[00119](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff__head.html" \l "m1) struct [sk\_buff](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff.html) \*prev;

00120

[00121](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff__head.html" \l "m2) \_\_u32 qlen;

[00122](http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structsk__buff__head.html" \l "m3) spinlock\_t lock;

00123 };

**struct** [**iphdr**](http://lxr.free-electrons.com/ident?i=iphdr) **\*iph; */\* Our header \*/***

**struct** [**iphdr**](http://lxr.free-electrons.com/ident?i=iphdr) **{**

**#if defined([\_\_LITTLE\_ENDIAN\_BITFIELD](http://lxr.free-electrons.com/ident?i=__LITTLE_ENDIAN_BITFIELD))**

[**\_\_u8**](http://lxr.free-electrons.com/ident?i=__u8) **ihl:4,**

[**version**](http://lxr.free-electrons.com/ident?i=version)**:4;**

**#elif defined (**[**\_\_BIG\_ENDIAN\_BITFIELD**](http://lxr.free-electrons.com/ident?i=__BIG_ENDIAN_BITFIELD)**)**

[**\_\_u8**](http://lxr.free-electrons.com/ident?i=__u8)[**version**](http://lxr.free-electrons.com/ident?i=version)**:4,**

**ihl:4;**

**#else**

**#error *"Please fix <asm/byteorder.h>"***

**#endif**

[**\_\_u8**](http://lxr.free-electrons.com/ident?i=__u8) **tos;**

[**\_\_be16**](http://lxr.free-electrons.com/ident?i=__be16) **tot\_len;**

[**\_\_be16**](http://lxr.free-electrons.com/ident?i=__be16)[**id**](http://lxr.free-electrons.com/ident?i=id)**;**

[**\_\_be16**](http://lxr.free-electrons.com/ident?i=__be16) **frag\_off;**

[**\_\_u8**](http://lxr.free-electrons.com/ident?i=__u8)[**ttl**](http://lxr.free-electrons.com/ident?i=ttl)**;**

[**\_\_u8**](http://lxr.free-electrons.com/ident?i=__u8)[**protocol**](http://lxr.free-electrons.com/ident?i=protocol)**;**

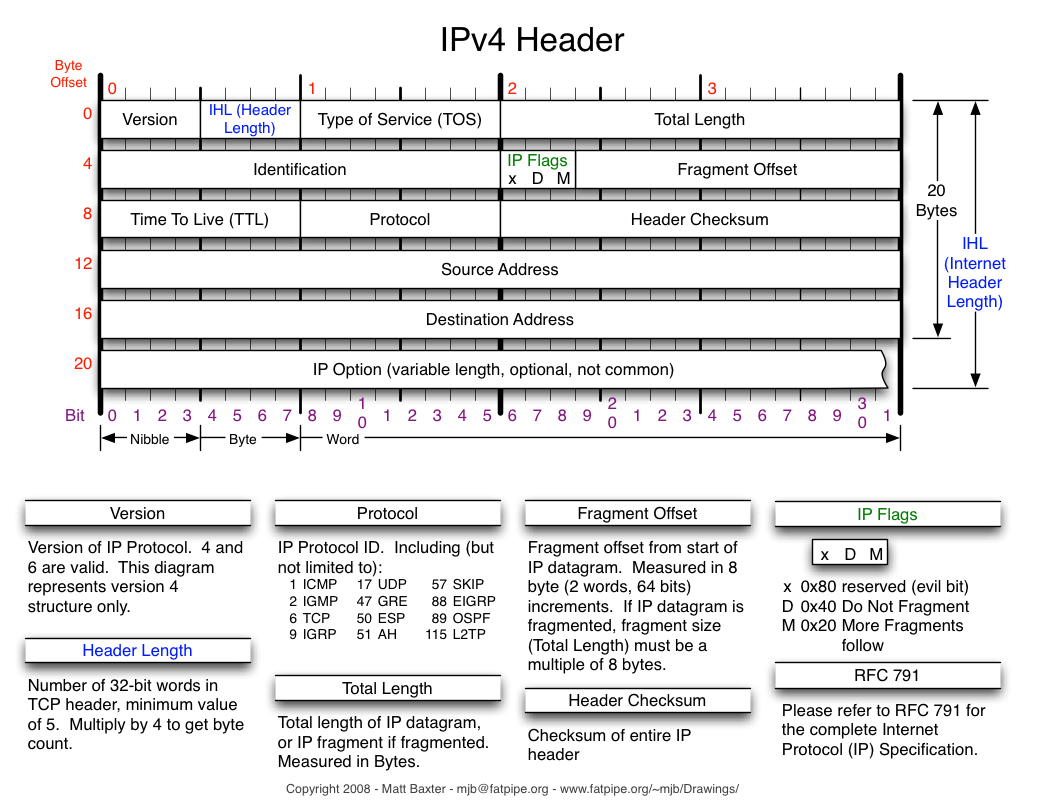
[**\_\_sum16**](http://lxr.free-electrons.com/ident?i=__sum16)[**check**](http://lxr.free-electrons.com/ident?i=check)**;**

[**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32)[**saddr**](http://lxr.free-electrons.com/ident?i=saddr)**;**

[**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32)[**daddr**](http://lxr.free-electrons.com/ident?i=daddr)**;**

***/\*The options start here. \*/***

**};**



**struct** [**rtable**](http://lxr.free-electrons.com/ident?i=rtable) **\*[rt](http://lxr.free-electrons.com/ident?i=rt" \t "_blank); */\* Route we use \*/***

Each rtable data structure holds information about the route to take in order to send packets to an IP host. rtable data structures are used within the IP route cache.

**struct** [**rtable**](http://lxr.free-electrons.com/ident?i=rtable) **{**

**struct** [**dst\_entry**](http://lxr.free-electrons.com/ident?i=dst_entry)[**dst**](http://lxr.free-electrons.com/ident?i=dst)**;**

**int rt\_genid;**

**unsigned int rt\_flags;**

[**\_\_u16**](http://lxr.free-electrons.com/ident?i=__u16) **rt\_type;**

[**\_\_u8**](http://lxr.free-electrons.com/ident?i=__u8) **rt\_is\_input;**

[**\_\_u8**](http://lxr.free-electrons.com/ident?i=__u8) **rt\_uses\_gateway;**

**int rt\_iif;**

***/\* Info on neighbour \*/***

[**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32) **rt\_gateway;**

***/\* Miscellaneous cached information \*/***

[**u32**](http://lxr.free-electrons.com/ident?i=u32) **rt\_pmtu;**

**struct** [**list\_head**](http://lxr.free-electrons.com/ident?i=list_head) **rt\_uncached;**

**#define** [**list\_head**](http://lxr.free-electrons.com/ident?i=list_head)**(**[**list**](http://lxr.free-electrons.com/ident?i=list)**,** [**type**](http://lxr.free-electrons.com/ident?i=type)**, member) \**

[**list\_entry**](http://lxr.free-electrons.com/ident?i=list_entry)**((**[**list**](http://lxr.free-electrons.com/ident?i=list)**)->**[**next**](http://lxr.free-electrons.com/ident?i=next)**,** [**type**](http://lxr.free-electrons.com/ident?i=type)**, member)**

**struct** [**uncached\_list**](http://lxr.free-electrons.com/ident?i=uncached_list) **\*rt\_uncached\_list;**

**struct** [**uncached\_list**](http://lxr.free-electrons.com/ident?i=uncached_list) **{**

[**spinlock\_t**](http://lxr.free-electrons.com/ident?i=spinlock_t)[**lock**](http://lxr.free-electrons.com/ident?i=lock)**;**

**struct** [**list\_head**](http://lxr.free-electrons.com/ident?i=list_head)[**head**](http://lxr.free-electrons.com/ident?i=head)**;**

**};**

**};**

**------------------------------------------------------------------------------------------------------------------------------- -**

***/\* that should never happen \*/***

struct [ip\_options](http://lxr.free-electrons.com/ident?i=ip_options) \*opt = &([IPCB](http://lxr.free-electrons.com/ident?i=IPCB)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank))->opt);

**struct** [**ip\_options**](http://lxr.free-electrons.com/ident?i=ip_options) **{**

[**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32) **faddr;**

[**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32) **nexthop;**

**unsigned char** [**optlen**](http://lxr.free-electrons.com/ident?i=optlen)**;**

**unsigned char** [**srr**](http://lxr.free-electrons.com/ident?i=srr)**;**

**unsigned char rr;**

**unsigned char** [**ts**](http://lxr.free-electrons.com/ident?i=ts)**;**

**unsigned char is\_strictroute:1,**

**srr\_is\_hit:1,**

**is\_changed:1,**

**rr\_needaddr:1,**

**ts\_needtime:1,**

**ts\_needaddr:1;**

**unsigned char router\_alert;**

**unsigned char cipso;**

**unsigned char \_\_pad2;**

**unsigned char \_\_data[0];**

**}**

**is\_strictroute:1 /\*Strict source route \*/**

**srr\_is\_hit:1, /\*Packet destination address was our one \*/**

**is\_changed:1, /\*IP checksum more not valid \*/**

**rr\_needaddr:1, /\*Need to record addr of outgoing dev \*/**

**ts\_needtime:1, /\*Need to record timestamp \*/**

**ts\_needaddr:1; /\*Need to record addr of outgoing dev \*/**

**IPCB**

**Defined as a preprocessor macro**

**#define** [**IPCB**](http://lxr.free-electrons.com/ident?i=IPCB)**([skb](http://lxr.free-electrons.com/ident?i=skb)) ((struct** [**inet\_skb\_parm**](http://lxr.free-electrons.com/ident?i=inet_skb_parm)**\*)(([skb](http://lxr.free-electrons.com/ident?i=skb))->[cb](http://lxr.free-electrons.com/ident?i=cb)))**

[86](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L86" \t "_blank) if ([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank)->pkt\_type != [PACKET\_HOST](http://lxr.free-electrons.com/ident?i=PACKET_HOST))

[87](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L87" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

**static void** [**drop**](http://lxr.free-electrons.com/ident?i=drop) **(void) {** [**flushline**](http://lxr.free-electrons.com/ident?i=flushline)**(**[**false**](http://lxr.free-electrons.com/ident?i=false)**); }**

[88](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L88" \t "_blank)

[89](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L89" \t "_blank) if ([unlikely](http://lxr.free-electrons.com/ident?i=unlikely)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank)->sk))

[90](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L90" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

**Defined as preprocessor macro:**

**#define** [**unlikely**](http://lxr.free-electrons.com/ident?i=unlikely)**([cond](http://lxr.free-electrons.com/ident?i=cond)) ([cond](http://lxr.free-electrons.com/ident?i=cond))**

[91](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L91" \t "_blank)

[92](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L92" \t "_blank) if ([skb\_warn\_if\_lro](http://lxr.free-electrons.com/ident?i=skb_warn_if_lro" \t "_blank)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank)))

**static inline** [**bool**](http://lxr.free-electrons.com/ident?i=bool)[**skb\_warn\_if\_lro**](http://lxr.free-electrons.com/ident?i=skb_warn_if_lro)**(const struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*[skb](http://lxr.free-electrons.com/ident?i=skb))**

**{**

***/\* LRO sets gso\_size but not gso\_type, whereas if GSO is really***

***\* wanted then gso\_type will be set. \*/***

**const struct** [**skb\_shared\_info**](http://lxr.free-electrons.com/ident?i=skb_shared_info) **\*shinfo =** [**skb\_shinfo**](http://lxr.free-electrons.com/ident?i=skb_shinfo)**([skb](http://lxr.free-electrons.com/ident?i=skb));**

**if ([skb\_is\_nonlinear](http://lxr.free-electrons.com/ident?i=skb_is_nonlinear)([skb](http://lxr.free-electrons.com/ident?i=skb)) && shinfo->gso\_size != 0 &&**

[**unlikely**](http://lxr.free-electrons.com/ident?i=unlikely)**(shinfo->gso\_type == 0)) {**

[**\_\_skb\_warn\_lro\_forwarding**](http://lxr.free-electrons.com/ident?i=__skb_warn_lro_forwarding)**([skb](http://lxr.free-electrons.com/ident?i=skb));**

**return** [**true**](http://lxr.free-electrons.com/ident?i=true)**; }**

[93](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L93" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

[94](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L94" \t "_blank)

[95](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L95" \t "_blank) if (![xfrm4\_policy\_check](http://lxr.free-electrons.com/ident?i=xfrm4_policy_check)([NULL](http://lxr.free-electrons.com/ident?i=NULL), XFRM\_POLICY\_FWD, [skb](http://lxr.free-electrons.com/ident?i=skb)))

[96](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L96" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

[97](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L97" \t "_blank)

[98](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L98" \t "_blank) if ([IPCB](http://lxr.free-electrons.com/ident?i=IPCB)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank))->opt.router\_alert && [ip\_call\_ra\_chain](http://lxr.free-electrons.com/ident?i=ip_call_ra_chain)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank)))

[99](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L99" \t "_blank) return [NET\_RX\_SUCCESS](http://lxr.free-electrons.com/ident?i=NET_RX_SUCCESS);

ip\_call\_ra\_chain:-

Is called early in the forwarding chain from ip\_forward and ip\_mr\_input which makes skb->dev the correct expression to get the input network device and dev\_net(skb->dev) a correct expression for the network namespace the packet is being processed in.

Returns a Boolean data type.

[**153**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L153)[**bool**](http://lxr.free-electrons.com/ident?i=bool)[**ip\_call\_ra\_chain**](http://lxr.free-electrons.com/ident?i=ip_call_ra_chain)**(struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*[skb](http://lxr.free-electrons.com/ident?i=skb))**

[**154**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L154) **{**

[**155**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L155) **struct** [**ip\_ra\_chain**](http://lxr.free-electrons.com/ident?i=ip_ra_chain) **\*[ra](http://lxr.free-electrons.com/ident?i=ra);**

[**156**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L156)[**u8**](http://lxr.free-electrons.com/ident?i=u8)[**protocol**](http://lxr.free-electrons.com/ident?i=protocol) **=** [**ip\_hdr**](http://lxr.free-electrons.com/ident?i=ip_hdr)**([skb](http://lxr.free-electrons.com/ident?i=skb))->**[**protocol**](http://lxr.free-electrons.com/ident?i=protocol)**;**

[**157**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L157) **struct** [**sock**](http://lxr.free-electrons.com/ident?i=sock) **\***[**last**](http://lxr.free-electrons.com/ident?i=last) **=** [**NULL**](http://lxr.free-electrons.com/ident?i=NULL)**;**

[**158**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L158) **struct** [**net\_device**](http://lxr.free-electrons.com/ident?i=net_device) **\***[**dev**](http://lxr.free-electrons.com/ident?i=dev) **=** [**skb**](http://lxr.free-electrons.com/ident?i=skb)**->**[**dev**](http://lxr.free-electrons.com/ident?i=dev)**;**

[**159**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L159)

[**160**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L160) **for ([ra](http://lxr.free-electrons.com/ident?i=ra) =** [**rcu\_dereference**](http://lxr.free-electrons.com/ident?i=rcu_dereference)**([ip\_ra\_chain](http://lxr.free-electrons.com/ident?i=ip_ra_chain));** [**ra**](http://lxr.free-electrons.com/ident?i=ra)**;** [**ra**](http://lxr.free-electrons.com/ident?i=ra) **=** [**rcu\_dereference**](http://lxr.free-electrons.com/ident?i=rcu_dereference)**([ra](http://lxr.free-electrons.com/ident?i=ra)->**[**next**](http://lxr.free-electrons.com/ident?i=next)**)) {**

[**161**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L161) **struct** [**sock**](http://lxr.free-electrons.com/ident?i=sock) **\*sk =** [**ra**](http://lxr.free-electrons.com/ident?i=ra)**->sk;**

[**162**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L162)

[**163**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L163) ***/\* If socket is bound to an interface, only report***

[**164**](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c#L164) ***\* the packet if it came from that interface.***

**[165](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L165)  *\*/***

**[166](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L166) if (sk &&** [**inet\_sk**](http://lxr.free-electrons.com/ident?i=inet_sk)**(sk)->[inet\_num](http://lxr.free-electrons.com/ident?i=inet_num) ==** [**protocol**](http://lxr.free-electrons.com/ident?i=protocol) **&&**

**[167](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L167) (!sk->[sk\_bound\_dev\_if](http://lxr.free-electrons.com/ident?i=sk_bound_dev_if) ||**

**[168](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L168) sk->[sk\_bound\_dev\_if](http://lxr.free-electrons.com/ident?i=sk_bound_dev_if) ==** [**dev**](http://lxr.free-electrons.com/ident?i=dev)**->ifindex) &&**

**[169](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L169)** [**net\_eq**](http://lxr.free-electrons.com/ident?i=net_eq)**([sock\_net](http://lxr.free-electrons.com/ident?i=sock_net)(sk),** [**dev\_net**](http://lxr.free-electrons.com/ident?i=dev_net)**(**[**dev**](http://lxr.free-electrons.com/ident?i=dev)**))) {**

**[170](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L170) if ([ip\_is\_fragment](http://lxr.free-electrons.com/ident?i=ip_is_fragment)([ip\_hdr](http://lxr.free-electrons.com/ident?i=ip_hdr)([skb](http://lxr.free-electrons.com/ident?i=skb)))) {**

**[171](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L171) if ([ip\_defrag](http://lxr.free-electrons.com/ident?i=ip_defrag)([skb](http://lxr.free-electrons.com/ident?i=skb), IP\_DEFRAG\_CALL\_RA\_CHAIN))**

**[172](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L172) return** [**true**](http://lxr.free-electrons.com/ident?i=true)**;**

**[173](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L173) }**

**[174](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L174) if (**[**last**](http://lxr.free-electrons.com/ident?i=last)**) {**

**[175](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L175) struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*skb2 =** [**skb\_clone**](http://lxr.free-electrons.com/ident?i=skb_clone)**([skb](http://lxr.free-electrons.com/ident?i=skb),** [**GFP\_ATOMIC**](http://lxr.free-electrons.com/ident?i=GFP_ATOMIC)**);**

**[176](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L176) if (skb2)**

**[177](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L177)** [**raw\_rcv**](http://lxr.free-electrons.com/ident?i=raw_rcv)**(**[**last**](http://lxr.free-electrons.com/ident?i=last)**, skb2);**

**[178](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L178) }**

**[179](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L179)** [**last**](http://lxr.free-electrons.com/ident?i=last) **= sk;**

**[180](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L180) }**

**[181](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L181) }**

**[182](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L182)**

**[183](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L183) if (**[**last**](http://lxr.free-electrons.com/ident?i=last)**) {**

**[184](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L184)** [**raw\_rcv**](http://lxr.free-electrons.com/ident?i=raw_rcv)**(**[**last**](http://lxr.free-electrons.com/ident?i=last)**,** [**skb**](http://lxr.free-electrons.com/ident?i=skb)**);**

**[185](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L185) return** [**true**](http://lxr.free-electrons.com/ident?i=true)**;**

**[186](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L186) }**

**[187](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L187) return** [**false**](http://lxr.free-electrons.com/ident?i=false)**;**

**[188](http://lxr.free-electrons.com/source/net/ipv4/ip_input.c" \l "L188) }**

First you declare a pointer ra

Store the type of protocol.

Declare pointer of type sock,name it as last and store NULL value.

**NET\_RX\_SUCCESS**

Defined as a preprocessor macro in:

**#define NET\_RX\_SUCCESS 0 */\* keep 'em coming, baby \*/***

[100](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c#L100)

[101](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L101" \t "_blank) [skb\_forward\_csum](http://lxr.free-electrons.com/ident?i=skb_forward_csum)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank));

**static inline void** [**skb\_forward\_csum**](http://lxr.free-electrons.com/ident?i=skb_forward_csum)**(struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*[skb](http://lxr.free-electrons.com/ident?i=skb))**

**[3409](http://lxr.free-electrons.com/source/include/linux/skbuff.h" \l "L3409) {**

**[3410](http://lxr.free-electrons.com/source/include/linux/skbuff.h" \l "L3410) */\* Unfortunately we don't support this one. Any brave souls? \*/***

**[3411](http://lxr.free-electrons.com/source/include/linux/skbuff.h" \l "L3411) if ([skb](http://lxr.free-electrons.com/ident?i=skb)->ip\_summed ==** [**CHECKSUM\_COMPLETE**](http://lxr.free-electrons.com/ident?i=CHECKSUM_COMPLETE)**)**

**[3412](http://lxr.free-electrons.com/source/include/linux/skbuff.h" \l "L3412)** [**skb**](http://lxr.free-electrons.com/ident?i=skb)**->ip\_summed =** [**CHECKSUM\_NONE**](http://lxr.free-electrons.com/ident?i=CHECKSUM_NONE)**;**

**[3413](http://lxr.free-electrons.com/source/include/linux/skbuff.h" \l "L3413) }**

[102](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L102" \t "_blank)

[103](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L103" \t "_blank) ***/\****

[104](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L104" \t "_blank)  ***\* According to the RFC, we must first decrease the TTL field. If***

[105](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L105" \t "_blank)  ***\* that reaches zero, we must reply an ICMP control message telling***

[106](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L106" \t "_blank)  ***\* that the packet's lifetime expired.***

[107](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L107" \t "_blank)  ***\*/***

[108](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L108" \t "_blank) if ([ip\_hdr](http://lxr.free-electrons.com/ident?i=ip_hdr" \t "_blank)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank))->[ttl](http://lxr.free-electrons.com/ident?i=ttl" \t "_blank) <= 1)

[109](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L109" \t "_blank) goto too\_many\_hops;

[110](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L110" \t "_blank)

[111](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c#L111) if (![xfrm4\_route\_forward](http://lxr.free-electrons.com/ident?i=xfrm4_route_forward)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank)))

**static inline int** [**xfrm4\_route\_forward**](http://lxr.free-electrons.com/ident?i=xfrm4_route_forward)**(struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*[skb](http://lxr.free-electrons.com/ident?i=skb))**

**[1134](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1134) {**

**[1135](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1135) return** [**xfrm\_route\_forward**](http://lxr.free-electrons.com/ident?i=xfrm_route_forward)**([skb](http://lxr.free-electrons.com/ident?i=skb),** [**AF\_INET**](http://lxr.free-electrons.com/ident?i=AF_INET)**);**

**[1136](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1136) }**

**static inline int** [**xfrm\_route\_forward**](http://lxr.free-electrons.com/ident?i=xfrm_route_forward)**(struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*[skb](http://lxr.free-electrons.com/ident?i=skb), unsigned short** [**family**](http://lxr.free-electrons.com/ident?i=family)**)**

**[1125](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1125) {**

**[1126](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1126) struct** [**net**](http://lxr.free-electrons.com/ident?i=net) **\***[**net**](http://lxr.free-electrons.com/ident?i=net) **=** [**dev\_net**](http://lxr.free-electrons.com/ident?i=dev_net)**([skb](http://lxr.free-electrons.com/ident?i=skb)->**[**dev**](http://lxr.free-electrons.com/ident?i=dev)**);**

**[1127](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1127)**

**[1128](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1128) return ![net](http://lxr.free-electrons.com/ident?i=net)->xfrm.policy\_count[XFRM\_POLICY\_OUT] ||**

**[1129](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1129) ([skb\_dst](http://lxr.free-electrons.com/ident?i=skb_dst)([skb](http://lxr.free-electrons.com/ident?i=skb))->**[**flags**](http://lxr.free-electrons.com/ident?i=flags) **&** [**DST\_NOXFRM**](http://lxr.free-electrons.com/ident?i=DST_NOXFRM)**) ||**

**[1130](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1130)** [**\_\_xfrm\_route\_forward**](http://lxr.free-electrons.com/ident?i=__xfrm_route_forward)**([skb](http://lxr.free-electrons.com/ident?i=skb),** [**family**](http://lxr.free-electrons.com/ident?i=family)**);**

**[1131](http://lxr.free-electrons.com/source/include/net/xfrm.h" \l "L1131) }**

[112](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L112" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

[113](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L113" \t "_blank)

[114](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L114" \t "_blank) [rt](http://lxr.free-electrons.com/ident?i=rt) = [skb\_rtable](http://lxr.free-electrons.com/ident?i=skb_rtable)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank));

[115](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L115" \t "_blank)

[116](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L116" \t "_blank) if (opt->is\_strictroute && [rt](http://lxr.free-electrons.com/ident?i=rt)->rt\_uses\_gateway)

[117](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L117" \t "_blank) goto sr\_failed;

[118](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L118" \t "_blank)

[119](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L119" \t "_blank) [IPCB](http://lxr.free-electrons.com/ident?i=IPCB)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank))->[flags](http://lxr.free-electrons.com/ident?i=flags) |= [IPSKB\_FORWARDED](http://lxr.free-electrons.com/ident?i=IPSKB_FORWARDED);

[120](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L120" \t "_blank) [mtu](http://lxr.free-electrons.com/ident?i=mtu) = [ip\_dst\_mtu\_maybe\_forward](http://lxr.free-electrons.com/ident?i=ip_dst_mtu_maybe_forward)(&[rt](http://lxr.free-electrons.com/ident?i=rt" \t "_blank)->[dst](http://lxr.free-electrons.com/ident?i=dst" \t "_blank), [true](http://lxr.free-electrons.com/ident?i=true));

[121](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L121" \t "_blank) if ([ip\_exceeds\_mtu](http://lxr.free-electrons.com/ident?i=ip_exceeds_mtu" \t "_blank)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank), [mtu](http://lxr.free-electrons.com/ident?i=mtu))) {

[122](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L122" \t "_blank) [IP\_INC\_STATS](http://lxr.free-electrons.com/ident?i=IP_INC_STATS)([dev\_net](http://lxr.free-electrons.com/ident?i=dev_net" \t "_blank)([rt](http://lxr.free-electrons.com/ident?i=rt" \t "_blank)->[dst](http://lxr.free-electrons.com/ident?i=dst).[dev](http://lxr.free-electrons.com/ident?i=dev)), IPSTATS\_MIB\_FRAGFAILS);

[123](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L123" \t "_blank) [icmp\_send](http://lxr.free-electrons.com/ident?i=icmp_send)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank), [ICMP\_DEST\_UNREACH](http://lxr.free-electrons.com/ident?i=ICMP_DEST_UNREACH), [ICMP\_FRAG\_NEEDED](http://lxr.free-electrons.com/ident?i=ICMP_FRAG_NEEDED),

[124](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L124" \t "_blank) [htonl](http://lxr.free-electrons.com/ident?i=htonl)([mtu](http://lxr.free-electrons.com/ident?i=mtu" \t "_blank)));

[125](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L125" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

[126](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L126" \t "_blank) }

**static inline unsigned int** [**ip\_dst\_mtu\_maybe\_forward**](http://lxr.free-electrons.com/ident?i=ip_dst_mtu_maybe_forward)**(const struct** [**dst\_entry**](http://lxr.free-electrons.com/ident?i=dst_entry) **\*[dst](http://lxr.free-electrons.com/ident?i=dst),**

[**bool**](http://lxr.free-electrons.com/ident?i=bool)[**forwarding**](http://lxr.free-electrons.com/ident?i=forwarding)**)**

**{**

**struct** [**net**](http://lxr.free-electrons.com/ident?i=net) **\***[**net**](http://lxr.free-electrons.com/ident?i=net) **=** [**dev\_net**](http://lxr.free-electrons.com/ident?i=dev_net)**([dst](http://lxr.free-electrons.com/ident?i=dst)->**[**dev**](http://lxr.free-electrons.com/ident?i=dev)**);**

**if (**[**net**](http://lxr.free-electrons.com/ident?i=net)**->ipv4.sysctl\_ip\_fwd\_use\_pmtu ||**

[**dst\_metric\_locked**](http://lxr.free-electrons.com/ident?i=dst_metric_locked)**([dst](http://lxr.free-electrons.com/ident?i=dst),** [**RTAX\_MTU**](http://lxr.free-electrons.com/ident?i=RTAX_MTU)**) ||**

**!**[**forwarding**](http://lxr.free-electrons.com/ident?i=forwarding)**)**

**return** [**dst\_mtu**](http://lxr.free-electrons.com/ident?i=dst_mtu)**([dst](http://lxr.free-electrons.com/ident?i=dst));**

**return** [**min**](http://lxr.free-electrons.com/ident?i=min)**([dst](http://lxr.free-electrons.com/ident?i=dst)->**[**dev**](http://lxr.free-electrons.com/ident?i=dev)**->[mtu](http://lxr.free-electrons.com/ident?i=mtu),** [**IP\_MAX\_MTU**](http://lxr.free-electrons.com/ident?i=IP_MAX_MTU)**);**

**}**

**IP\_INC\_STATS:**

**Defined as preprocessor macro:**

**#define** [**IP\_INC\_STATS**](http://lxr.free-electrons.com/ident?i=IP_INC_STATS)**(**[**net**](http://lxr.free-electrons.com/ident?i=net)**,** [**field**](http://lxr.free-electrons.com/ident?i=field)**)** [**SNMP\_INC\_STATS64**](http://lxr.free-electrons.com/ident?i=SNMP_INC_STATS64)**((**[**net**](http://lxr.free-electrons.com/ident?i=net)**)->mib.ip\_statistics,** [**field**](http://lxr.free-electrons.com/ident?i=field)**)**

icmp\_send:

**void** [**icmp\_send**](http://lxr.free-electrons.com/ident?i=icmp_send)**(struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*skb\_in, int** [**type**](http://lxr.free-electrons.com/ident?i=type)**, int** [**code**](http://lxr.free-electrons.com/ident?i=code)**,** [**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32)[**info**](http://lxr.free-electrons.com/ident?i=info)**)**

**be32:**

**typedef** [**u32**](http://lxr.free-electrons.com/ident?i=u32)[**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32)**;**

[127](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c#L127)

[128](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L128" \t "_blank) ***/\* We are about to mangle packet. Copy it! \*/***

[129](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L129" \t "_blank) if ([skb\_cow](http://lxr.free-electrons.com/ident?i=skb_cow" \t "_blank)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank), [LL\_RESERVED\_SPACE](http://lxr.free-electrons.com/ident?i=LL_RESERVED_SPACE)([rt](http://lxr.free-electrons.com/ident?i=rt" \t "_blank)->[dst](http://lxr.free-electrons.com/ident?i=dst).[dev](http://lxr.free-electrons.com/ident?i=dev))+[rt](http://lxr.free-electrons.com/ident?i=rt" \t "_blank)->[dst](http://lxr.free-electrons.com/ident?i=dst" \t "_blank).[header\_len](http://lxr.free-electrons.com/ident?i=header_len)))

[130](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L130" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

[131](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L131" \t "_blank) iph = [ip\_hdr](http://lxr.free-electrons.com/ident?i=ip_hdr)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank));

[132](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L132" \t "_blank)

[133](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L133" \t "_blank) ***/\* Decrease ttl after skb cow done \*/***

[134](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L134" \t "_blank) [ip\_decrease\_ttl](http://lxr.free-electrons.com/ident?i=ip_decrease_ttl)(iph);

[135](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L135" \t "_blank)

[136](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L136" \t "_blank) ***/\****

[137](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L137" \t "_blank)  ***\* We now generate an ICMP HOST REDIRECT giving the route***

[138](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L138" \t "_blank)  ***\* we calculated.***

[139](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L139" \t "_blank)  ***\*/***

[140](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L140" \t "_blank) if ([IPCB](http://lxr.free-electrons.com/ident?i=IPCB)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank))->[flags](http://lxr.free-electrons.com/ident?i=flags) & [IPSKB\_DOREDIRECT](http://lxr.free-electrons.com/ident?i=IPSKB_DOREDIRECT) && !opt->[srr](http://lxr.free-electrons.com/ident?i=srr" \t "_blank) &&

[141](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L141" \t "_blank) ![skb\_sec\_path](http://lxr.free-electrons.com/ident?i=skb_sec_path" \t "_blank)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank)))

[142](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L142" \t "_blank) [ip\_rt\_send\_redirect](http://lxr.free-electrons.com/ident?i=ip_rt_send_redirect)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank));

[143](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L143" \t "_blank)

[144](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L144" \t "_blank) [skb](http://lxr.free-electrons.com/ident?i=skb)->[priority](http://lxr.free-electrons.com/ident?i=priority) = [rt\_tos2priority](http://lxr.free-electrons.com/ident?i=rt_tos2priority)(iph->tos);

[145](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L145" \t "_blank)

[146](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L146" \t "_blank) return [NF\_HOOK](http://lxr.free-electrons.com/ident?i=NF_HOOK)(NFPROTO\_IPV4, NF\_INET\_FORWARD, [NULL](http://lxr.free-electrons.com/ident?i=NULL), [skb](http://lxr.free-electrons.com/ident?i=skb),

[147](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L147" \t "_blank) [skb](http://lxr.free-electrons.com/ident?i=skb)->[dev](http://lxr.free-electrons.com/ident?i=dev), [rt](http://lxr.free-electrons.com/ident?i=rt)->[dst](http://lxr.free-electrons.com/ident?i=dst).[dev](http://lxr.free-electrons.com/ident?i=dev), [ip\_forward\_finish](http://lxr.free-electrons.com/ident?i=ip_forward_finish));

[148](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L148" \t "_blank)

[149](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L149" \t "_blank) sr\_failed:

[150](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L150" \t "_blank) ***/\****

[151](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L151" \t "_blank)  ***\* Strict routing permits no gatewaying***

[152](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L152" \t "_blank)  ***\*/***

[153](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L153" \t "_blank) [icmp\_send](http://lxr.free-electrons.com/ident?i=icmp_send)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank), [ICMP\_DEST\_UNREACH](http://lxr.free-electrons.com/ident?i=ICMP_DEST_UNREACH), [ICMP\_SR\_FAILED](http://lxr.free-electrons.com/ident?i=ICMP_SR_FAILED), 0);

ICMP\_DEST\_UNREACH

Defined as a preprocessor macro

**#define** [**ICMP\_DEST\_UNREACH**](http://lxr.free-electrons.com/ident?i=ICMP_DEST_UNREACH) **3 */\* Destination Unreachable \*/***

ICMP\_SR\_FAILED

**#define** [**ICMP\_SR\_FAILED**](http://lxr.free-electrons.com/ident?i=ICMP_SR_FAILED) **5 */\* Source Route failed \*/***

[154](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L154" \t "_blank) goto [drop](http://lxr.free-electrons.com/ident?i=drop);

[155](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L155" \t "_blank)

[156](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L156" \t "_blank) too\_many\_hops:

[157](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L157" \t "_blank) ***/\* Tell the sender its packet died... \*/***

[158](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L158" \t "_blank) [IP\_INC\_STATS\_BH](http://lxr.free-electrons.com/ident?i=IP_INC_STATS_BH)([dev\_net](http://lxr.free-electrons.com/ident?i=dev_net" \t "_blank)([skb\_dst](http://lxr.free-electrons.com/ident?i=skb_dst" \t "_blank)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank))->[dev](http://lxr.free-electrons.com/ident?i=dev)), IPSTATS\_MIB\_INHDRERRORS);

IP\_INC\_STATS\_BH

Defined as IP preprossesor macro:

**#define** [**IP\_INC\_STATS\_BH**](http://lxr.free-electrons.com/ident?i=IP_INC_STATS_BH)**(**[**net**](http://lxr.free-electrons.com/ident?i=net)**,** [**field**](http://lxr.free-electrons.com/ident?i=field)**)** [**SNMP\_INC\_STATS64\_BH**](http://lxr.free-electrons.com/ident?i=SNMP_INC_STATS64_BH)**((**[**net**](http://lxr.free-electrons.com/ident?i=net)**)->mib.ip\_statistics,** [**field**](http://lxr.free-electrons.com/ident?i=field)**)**

dev\_net

**struct** [**net**](http://lxr.free-electrons.com/ident?i=net) **\*[dev\_net](http://lxr.free-electrons.com/ident?i=dev_net)(const struct** [**net\_device**](http://lxr.free-electrons.com/ident?i=net_device) **\***[**dev**](http://lxr.free-electrons.com/ident?i=dev)**)**

**{**

**return** [**read\_pnet**](http://lxr.free-electrons.com/ident?i=read_pnet)**(&**[**dev**](http://lxr.free-electrons.com/ident?i=dev)**->nd\_net);**

**}**

skb\_dst

**static inline struct** [**dst\_entry**](http://lxr.free-electrons.com/ident?i=dst_entry) **\*[skb\_dst](http://lxr.free-electrons.com/ident?i=skb_dst)(const struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*[skb](http://lxr.free-electrons.com/ident?i=skb))**

**{**

***/\* If refdst was not refcounted, check we still are in a***

***\* rcu\_read\_lock section***

***\*/***

[**WARN\_ON**](http://lxr.free-electrons.com/ident?i=WARN_ON)**(([skb](http://lxr.free-electrons.com/ident?i=skb)->\_skb\_refdst &** [**SKB\_DST\_NOREF**](http://lxr.free-electrons.com/ident?i=SKB_DST_NOREF)**) &&**

**![rcu\_read\_lock\_held](http://lxr.free-electrons.com/ident?i=rcu_read_lock_held)() &&**

**![rcu\_read\_lock\_bh\_held](http://lxr.free-electrons.com/ident?i=rcu_read_lock_bh_held)());**

**return (struct** [**dst\_entry**](http://lxr.free-electrons.com/ident?i=dst_entry) **\*)([skb](http://lxr.free-electrons.com/ident?i=skb)->\_skb\_refdst &** [**SKB\_DST\_PTRMASK**](http://lxr.free-electrons.com/ident?i=SKB_DST_PTRMASK)**);**

**}**

dev

Defined as preprocessor macro

[159](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c#L159) [icmp\_send](http://lxr.free-electrons.com/ident?i=icmp_send)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank), [ICMP\_TIME\_EXCEEDED](http://lxr.free-electrons.com/ident?i=ICMP_TIME_EXCEEDED), ICMP\_EXC\_TTL, 0);

icmp\_send

**void** [**icmp\_send**](http://lxr.free-electrons.com/ident?i=icmp_send)**(struct** [**sk\_buff**](http://lxr.free-electrons.com/ident?i=sk_buff) **\*skb\_in, int** [**type**](http://lxr.free-electrons.com/ident?i=type)**, int** [**code**](http://lxr.free-electrons.com/ident?i=code)**,** [**\_\_be32**](http://lxr.free-electrons.com/ident?i=__be32)[**info**](http://lxr.free-electrons.com/ident?i=info)**);**

ICMP\_TIME\_EXCEEDED

Defined as ICMP preprocessor macro

**#define** [**ICMP\_TIME\_EXCEEDED**](http://lxr.free-electrons.com/ident?i=ICMP_TIME_EXCEEDED) **11 */\* Time Exceeded \*/***

ICMP\_EXC\_TTL

Defined as preprocessor macro

**#define** [**ICMP\_EXC\_TTL**](http://lxr.free-electrons.com/ident?i=ICMP_EXC_TTL) **0 */\* TTL count exceeded \*/***

[160](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L160" \t "_blank) [drop](http://lxr.free-electrons.com/ident?i=drop):

[161](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L161" \t "_blank) [kfree\_skb](http://lxr.free-electrons.com/ident?i=kfree_skb)([skb](http://lxr.free-electrons.com/ident?i=skb" \t "_blank));

void kfree\_skb(struct sk\_buff \*skb);

void dev\_kfree\_skb(struct sk\_buff \*skb);

void dev\_kfree\_skb\_irq(struct sk\_buff \*skb);

void dev\_kfree\_skb\_any(struct sk\_buff \*skb);

Free a buffer. The *kfree\_skb* call is used internally by the kernel. A driver should use one of the forms of *dev\_kfree\_skb* instead: *dev\_kfree\_skb* for noninterrupt context,*dev\_kfree\_skb\_irq* for interrupt context, or *dev\_kfree\_skb\_any* for code that can run in either context.

[162](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L162" \t "_blank) return [NET\_RX\_DROP](http://lxr.free-electrons.com/ident?i=NET_RX_DROP);

**#define** [**NET\_RX\_DROP**](http://lxr.free-electrons.com/ident?i=NET_RX_DROP) **1 */\* packet dropped \*/***

[163](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L163" \t "_blank) }

[164](http://lxr.free-electrons.com/source/net/ipv4/ip_forward.c" \l "L164" \t "_blank)