When sockets send and receive packets, they must be properly charged for the amount of system memory consumed by that packet.

So different attributes of it are:

'skb->truesize' : We keep track of how many bytes of system memory are consumed by a packet using this. This is the total of how large a data buffer we allocated for the packet, plus the size of 'struct sk\_buff' itself.

'sk->sk\_sndbuf' : The total number of bytes of send packet memory, a socket may use is limited by this.

'sk->sk\_rcvbuf' : The total number of bytes of for receive packets memory, a socket may use is limited by this.

**{ Actually above one are used in used for datagram protocols. Stream protocols, such as TCP, use a more elaborate scheme. }**

**Functions:**

1.

static inline void skb\_set\_owner\_r(struct sk\_buff \*skb, struct sock \*sk)

{

skb->sk = sk;

skb->destructor = sock\_rfree;

atomic\_add(skb->truesize, &sk->sk\_rmem\_alloc);

}

This function is used when a received packet is to be charged to a socket.

First,It sets 'skb->sk' to the socket.Secondly, it calls up a destructor function, and accounts the data bytes in 'sk->sk\_rmem\_alloc' for memory allocation.

2.

void sock\_rfree(struct sk\_buff \*skb)

{

struct sock \*sk = skb->sk;

atomic\_sub(skb->truesize, &sk->sk\_rmem\_alloc);

}

It is destructor function.

Later when the packet is freed up the destructor is invoked. So destructor 'sock\_rfree()' is invoked to do this operation. It releases the buffer allocate space from 'sk->sk\_rmem\_alloc'.

There are many other function which are not so important for forwarding.