/\*

\* remove\_from\_pending\_list: removes the pkt from the list when

\* it is no longer needed. pending\_list is a global variable.

\*/

**void** **remove\_from\_pending\_list**(packet \*p)

{

packet \*\*i = &pending\_list;

**for** (;(\*i) != NULL && ((\*i) != p); \*i = ((\*i)->next)) { [1]

/\* do nothing \*/;

}

**if** (\*i != NULL) { [2]

(\*i) = (\*i)->next;

}

**if** (p != NULL) { [3]

packetfree(p);

}

}

Remarks:

This is wrong implementation. We lose the pointer to the rest elements of the list after we release element p.

We “break” the list. We don’t save the pervious element (the element which comes before element p in the list). By not doing so we lose the connection between the previous and the next element of element p.

For example: remove element 3.

After step [1] p and (\*i) are equal

After step [2] (\*i) points to element 4

After step [3] element 2 doesn’t point to element 4, but to the released p element. And then we break the list chain (an also causing memory leak)

p

\*i