

A Full Specification

```
1  from state import flow_emap, int_devices
2  EXP_TIME = 10 * 1000
3  EXT_DEVICE = 1
4
5  if a_packet_received:
6      flow_emap.expire_all(now - EXP_TIME)
7
8  h3 = pop_header(tcpudp, on_mismatch=([], []))
9  h2 = pop_header(ipv4, on_mismatch=([], []))
10 h1 = pop_header(ether, on_mismatch=([], []))
11
12 if received_on_port == EXT_DEVICE:
13     internal_flow = FlowIdc(h3.dst_port, h3.src_port, h2.daddr, h2.saddr, h2.npid)
14     if flow_emap.has(internal_flow):
15         fl_id = flow_emap.get(internal_flow)
16         flow_emap.refresh_idx(fl_id, now)
17         out_port = vector_get(int_devices, fl_id)
18         return ([out_port], [ether(h1, saddr=..., daddr=...), ipv4(h2, cksun=...), tcpudp(h3)])
19     else:
20         return ([], [])
21 else:
22     internal_flow = FlowIdc(h3.src_port, h3.dst_port, h2.saddr, h2.daddr, h2.npid)
23     if flow_emap.has(internal_flow):
24         fl_id = flow_emap.get(internal_flow)
25         flow_emap.refresh_idx(fl_id, now)
26     else:
27         if not flow_emap.full():
28             fl_id = the_index_allocated
29             flow_emap.add(internal_flow, fl_id, now)
30             vector_set(int_devices, fl_id, received_on_port)
31     return ([EXT_DEVICE], [ether(h1, saddr=..., daddr=...), ipv4(h2, cksun=...), tcpudp(h3)])
```

A Partial Specification

```
1  if received_on_port == EXT_DEVICE and flow_emap.has(internal_flow):  
2      return ([...], [ether(h1, saddr=..., daddr=...), ipv4(h2, cksun=...), tcpudp(h3)])  
3  else:  
4      pass
```