

Push-Button

Pay-As-You-Go

Full-Stack

- Softwarization —for rapid innovation
- Traditional verification cost: 5-10x development effort[†]
- Ultimate goal: verification at no cost

[†]seL4 SOSP 2009
Ironclad OSDI 2014
HACL*. CCS 2017

Push-Button

Pay-As-You-Go

Full-Stack

EXT_PORT = 2

if received_on_port != EXT_PORT: # Heartbeat
return ([], [])

And not:

```
1 from state import flow_emap, flow_id_to_backend_id, backends, backend_ip_emap, cht
2 EXP_TIME = 10 * 1000
3 BACKEND_EXP_TIME = 3600000000 * 1000
4 EXT_PORT = 2
5
6 if a_packet_received:
7     flow_emap.expire_all(now - EXP_TIME)
8     backend_ip_emap.expire_all(now - BACKEND_EXP_TIME)
9
10 h3 = pop_header(tcpudp, on_mismatch=([], []))
11 h2 = pop_header(ipv4, on_mismatch=([], []))
12 h1 = pop_header(ether, on_mismatch=([], []))
13
14 assert a_packet_received
15 assert h1.type == 8 # 0x0800 == IPv4 in big endian
16 assert h2.npid == 6 or h2.npid == 17 # 6/17 -> TCP/UDP
17
18 if received_on_port == EXT_PORT: # Packet from the external network - client
19     packet_flow = LoadBalancedFlowc(h2.saddr, h2.daddr, h3.src_port, h3.dst_port, h2.npid)
20     alloc_flow_and_process_packet = False;
21     if flow_emap.has(packet_flow):
22         flow_id = flow_emap.get(packet_flow)
23         backend_id = flow_id_to_backend_id.get(flow_id)
24         if backend_ip_emap.has_idx(backend_id):
25             flow_emap.refresh_idx(flow_emap.get(packet_flow), now)
26             backend = backends.get(backend_id)
27             return ([backend.nic],
28                     [ether(h1, saddr=..., daddr=backend.mac),
29                      ipv4(h2, cksum=..., daddr=backend.ip),
30                      tcpudp(h3)])
31     else:
32         flow_emap.erase(packet_flow)
33         alloc_flow_and_process_packet = True
```

```
34 else:
35     alloc_flow_and_process_packet = True
36     if alloc_flow_and_process_packet:
37         if backend_ip_emap.exists_with_cht(cht, _LoadBalancedFlow_hash(packet_flow)):
38             bknd = backend_ip_emap.choose_with_cht(cht, _LoadBalancedFlow_hash(packet_flow))
39             if not flow_emap.full():
40                 idx = the_index_allocated
41                 flow_emap.add(packet_flow, idx, now)
42                 flow_id_to_backend_id.set(idx, bknd)
43                 backend = backends.get(bknd)
44                 return ([backend.nic],
45                         [ether(h1, saddr=..., daddr=backend.mac),
46                          ipv4(h2, cksum=..., daddr=backend.ip),
47                          tcpudp(h3)])
48         else:
49             return ([], [])
50     else: # A heartbeat from a backend
51         bknd_addr = ip_addr(h2.saddr)
52         if backend_ip_emap.has(bknd_addr):
53             backend_ip_emap.refresh_idx(backend_ip_emap.get(bknd_addr), now)
54         else:
55             if not backend_ip_emap.full():
56                 idx = the_index_allocated
57                 backend_ip_emap.add(bknd_addr, idx, now)
58                 backends.set(idx, LoadBalancedBackendc(received_on_port, h1.saddr, h2.saddr))
59     return ([], [])
```