



Actions

[Client] basicfwd uses rte_eth_rx_burst on 1 and captures SYN packet;

[Client] (TODO) Dysco agent inserts hardcoded payload on SYN packet;

[Client] basicfwd uses rte_eth_tx_burst on 0 and forwards SYN packet;

[Middlebox] basicfwd uses rte_eth_rx_burst on 0;

[Middlebox] Dysco agent intercepts rte_eth_rx_burst and receives SYN packet with payload;

[Middlebox] Dysco agent maps 4 tcp session:

(a) Client->MB => Client->Server

(b) Client->Server => MB->Server

(c) Server->MB => Server->Client

(d) Server->Client => MB->Client

[Middlebox] Removing first IP address from service chain (yourself) and uses it to source address to forward (on map values, as MB)

[Middlebox] Restoring original values of tcp session and return original function to basicfwd application.

[Middlebox] Dysco agent intercepts rte_eth_tx_burst and receives SYN packet without payload;

[Middlebox] Restoring payload received previosly (without first IP address on service chain) and with map value, it forwards (creating new values of TCP ports on map);

[Server] basicfwd uses rte_eth_rx_burst on 0

[Server] Dysco agent intercepts rte_eth_rx_burst and receives SYN packet with payload;

... same of Middlebox (with diferent values)

When non-syn packets are received, just search on map and 4-tuple values are modified.

Notes

All devices are DPDK compatible, even if are tap devices.

rte_eth_rx_burst and rte_eth_tx_burst functions are using on both interfaces.

basicfwd application algorithm: rte_eth_rx_burst(port) -> do nothing -> rte_eth_tx_burst(port ^1)