



# NetOps and SecOps in the Dataplane

(extreme flexibility at very interesting speeds and quantities)

**Michael Reed**

Vice President, Engineering

[reed@mantisnet.com](mailto:reed@mantisnet.com)

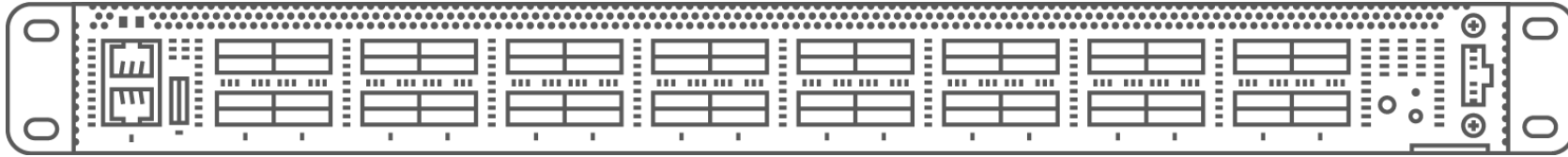
# The Problems

---

- 100G is the new 10G (and 400G is right around the corner)
- Network standards are evolving faster than ever (overlays, underlays, new protocols)
- Traditional switch/router vendors are designing gear for efficient transport, not intelligent analysis
- Custom silicon solutions take years to develop and fixed-function devices are obsolete almost as fast as they come on the market



# MantisNet RFP-NG



# P4 – “Hello Packet”

```
reed — root@rfrp-ng: ~/rfrp-ng/Main — ssh developer@192.168.100.205 — bash — 80x45
/* -*- P4_14 -*- */

#include <tofino/intrinsic_metadata.p4>

header_type ethernet_t
{
    fields
    {
        DMAC      : 48;
        SMAC      : 48;
        etherType  : 16;
    }
}
header ethernet_t ethernet;

parser start
{
    extract(ethernet);
    return ingress;
}

action echo_port()
{
    modify_field(ig_intr_md_for_tm.ucast_egress_port, ig_intr_md.ingress_port);
}

table echo
{
    actions
    {
        echo_port;
    }
    default_action : echo_port;
}

control ingress
{
    apply(echo);
}

control egress
{
}

example1.p4 (END)
```

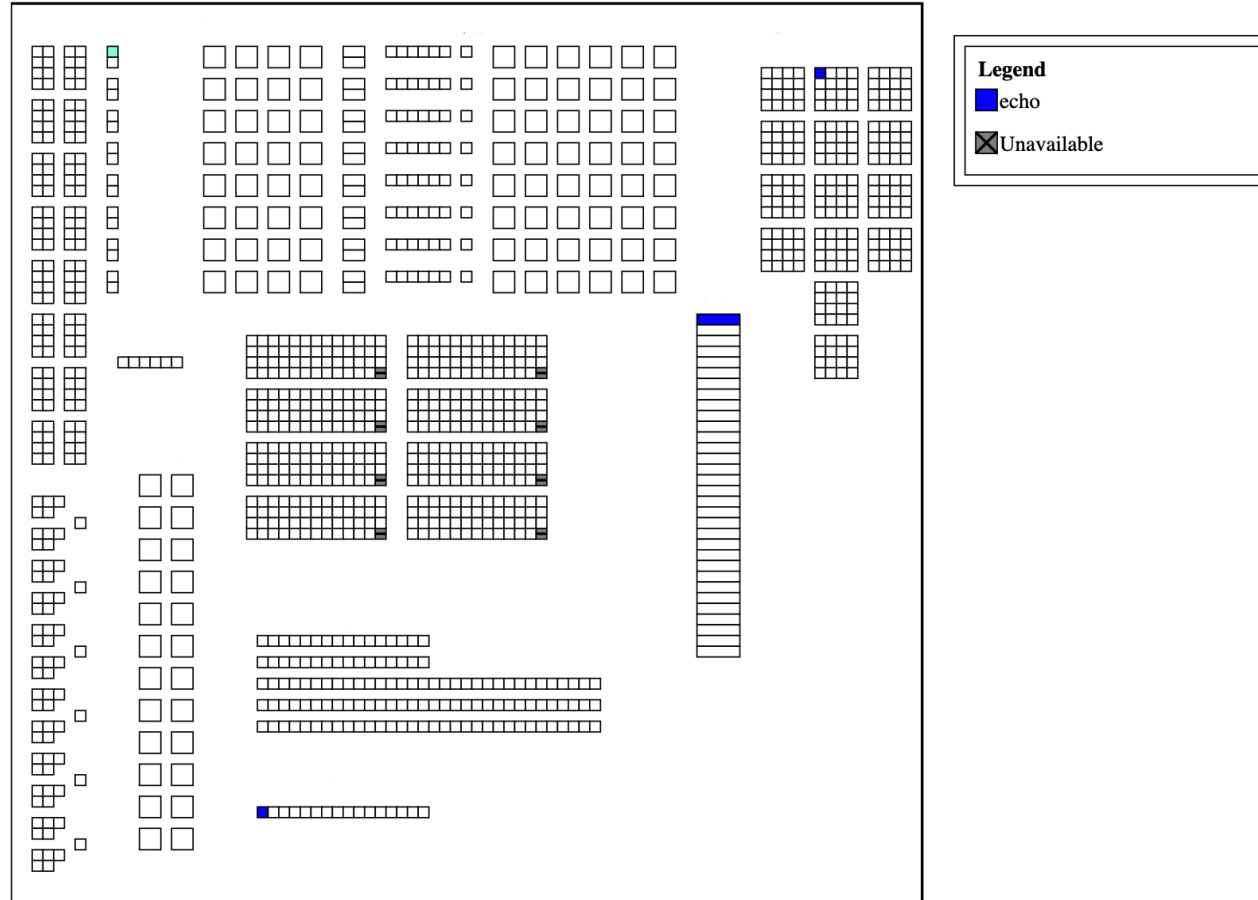


# P4 – Resource Utilization (“Hello Packet”)

0	0.00%	0.00%	0.00%	0.00%	6.25%	0.00%	0.00%	0.00%	3.12%	0.00%	0.00%	0.00%	6.25%	6.25%	0.00%	0.00%	0.00%	0.00%	0.00%	6.25%
1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
3	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
4	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
8	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
11	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Average	0.00%	0.00%	0.00%	0.00%	0.52%	0.00%	0.00%	0.00%	0.26%	0.00%	0.00%	0.00%	0.52%	0.52%	0.00%	0.00%	0.00%	0.00%	0.00%	0.52%



# P4 – MAU Stage 0 (“Hello Packet”)



# P4 – Add MAC-swap

```
reed — root@rfp-ng: ~/rfp-ng/Main — ssh developer@192.168.100.205 — bash — 80x45
/* -*- P4_14 -*- */

#include <tofino/intrinsic_metadata.p4>

header_type ethernet_t
{
    fields
    {
        DMAC      : 48;
        SMAC      : 48;
        etherType : 16;
    }
}
header ethernet_t ethernet;

parser start
{
    extract(ethernet);
    return ingress;
}

action echo_port()
{
    swap(ethernet.DMAC, ethernet.SMAC);
    modify_field(ig_intr_md_for_tm.ucast_egress_port, ig_intr_md.ingress_port);
}

table echo
{
    actions
    {
        echo_port;
    }
    default_action : echo_port;
}

control ingress
{
    apply(echo);
}

control egress
{
}

example2.p4 (END)
```



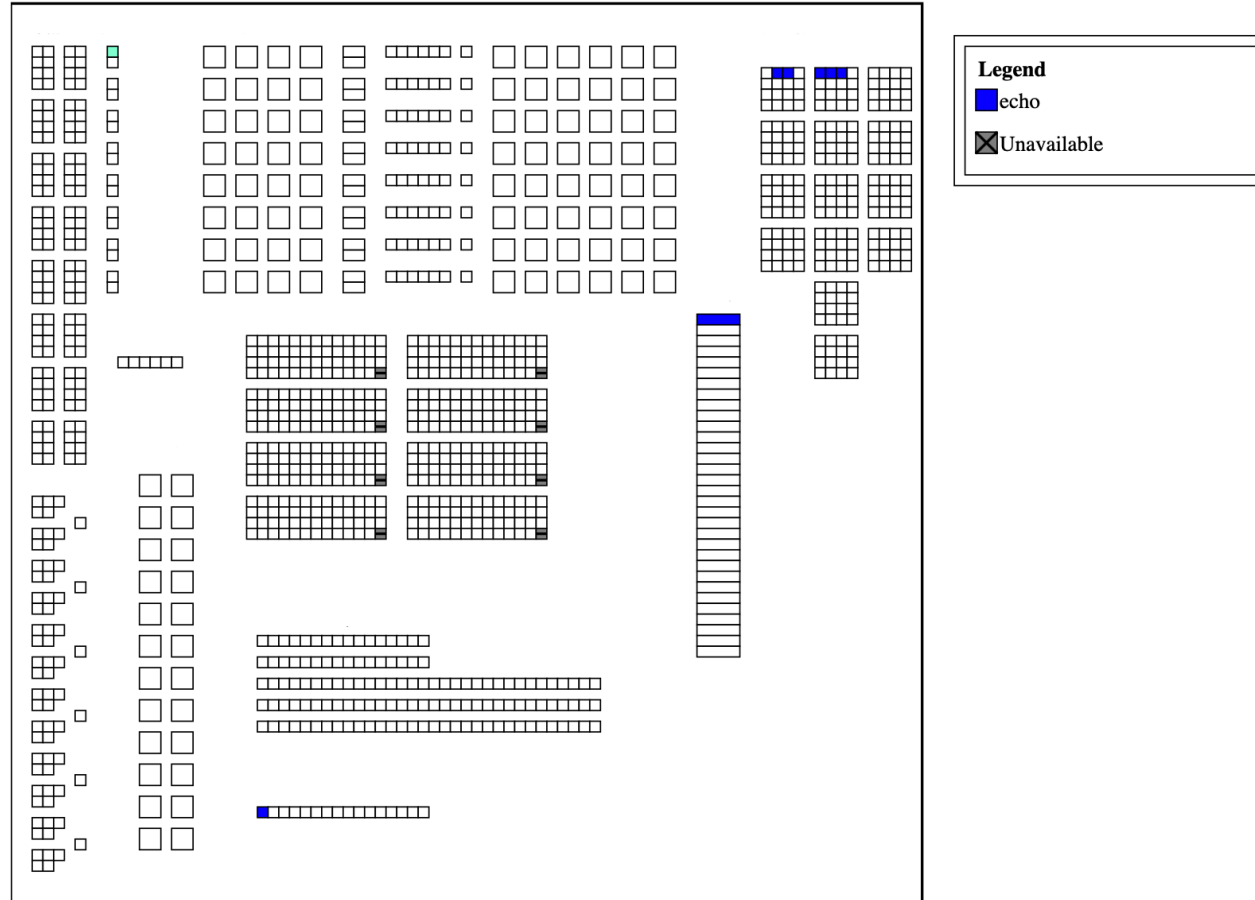
# P4 – Resource Utilization (MAC-swap)

0	0.00%	0.00%	0.00%	0.00%	6.25%	0.00%	0.00%	0.00%	3.12%	0.00%	0.00%	0.00%	6.25%	6.25%	0.00%	0.00%	0.00%	0.00%	0.00%	6.25%
1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
3	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
4	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
8	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
11	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Average	0.00%	0.00%	0.00%	0.00%	0.52%	0.00%	0.00%	0.00%	0.26%	0.00%	0.00%	0.00%	0.52%	0.52%	0.00%	0.00%	0.00%	0.00%	0.00%	0.52%





# P4 – MAU Stage 0 (MAC-swap)



# MantisNet RFP-NG

---

- Rules match:
  - L2/2.5 Match (SMAC, DMAC, EtherType, 802.1Q, MPLS)
  - L3 IPv4, IPv6
  - L4 TCP, UDP, SCTP, ICMP, IGMP
  - L4+ GTP, DPI\*
- Actions:
  - Permit, Deny, Push/Pop/Replace 802.1Q, NetFlow
- Flow-aware load balancing, round-robin distribution, replication
- Counters galore!

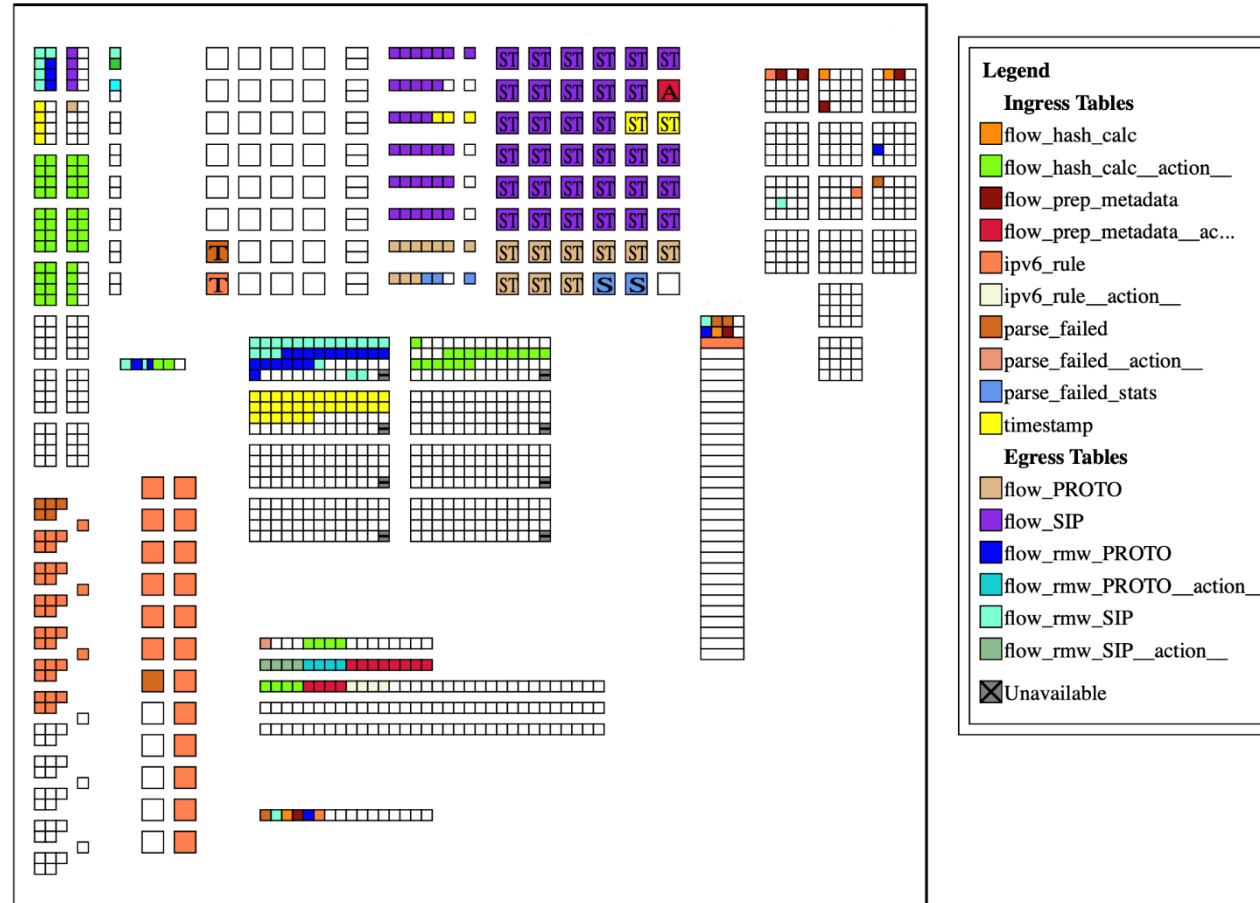


# MantisNet RFP-NG Resource Utilization

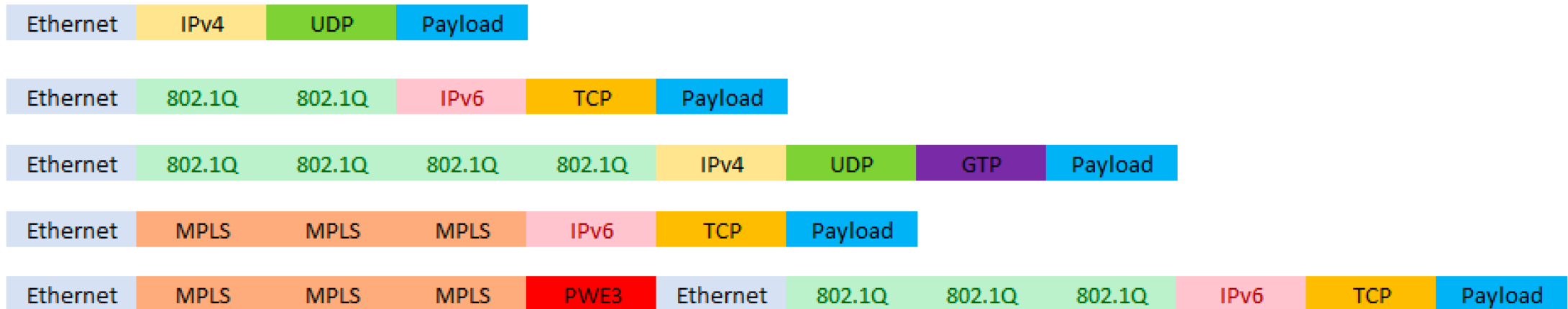
0	47.66%	57.58%	20.43%	83.33%	18.75%	61.25%	95.83%	79.17%	9.38%	75.00%	25.00%	0.00%	12.50%	18.75%	18.75%	25.78%	65.62%	18.75%	28.12%	37.50%
1	8.59%	98.48%	8.17%	50.00%	12.50%	58.75%	91.67%	100.00%	12.50%	50.00%	25.00%	0.00%	6.25%	12.50%	6.25%	34.38%	37.50%	25.00%	34.38%	18.75%
2	5.47%	98.48%	4.09%	33.33%	12.50%	48.75%	75.00%	100.00%	12.50%	50.00%	25.00%	0.00%	6.25%	12.50%	6.25%	34.38%	12.50%	37.50%	34.38%	18.75%
3	7.81%	98.48%	8.17%	50.00%	18.75%	58.75%	91.67%	100.00%	12.50%	50.00%	25.00%	0.00%	12.50%	18.75%	6.25%	34.38%	37.50%	25.00%	34.38%	25.00%
4	11.72%	98.48%	7.21%	0.00%	0.00%	10.00%	4.17%	100.00%	12.50%	0.00%	25.00%	6.25%	6.25%	6.25%	6.25%	28.12%	12.50%	25.00%	28.12%	12.50%
5	4.69%	98.48%	4.09%	33.33%	6.25%	27.50%	39.58%	100.00%	12.50%	25.00%	25.00%	0.00%	6.25%	6.25%	6.25%	31.25%	12.50%	31.25%	31.25%	12.50%
6	7.81%	98.48%	4.57%	33.33%	12.50%	50.00%	77.08%	100.00%	12.50%	50.00%	25.00%	0.00%	6.25%	12.50%	6.25%	34.38%	25.00%	31.25%	34.38%	18.75%
7	60.16%	0.00%	12.74%	33.33%	18.75%	52.50%	81.25%	0.00%	6.25%	50.00%	50.00%	12.50%	18.75%	18.75%	0.00%	6.25%	12.50%	6.25%	6.25%	25.00%
8	41.41%	4.55%	9.86%	33.33%	6.25%	52.50%	81.25%	4.17%	18.75%	50.00%	50.00%	6.25%	6.25%	12.50%	6.25%	9.38%	12.50%	12.50%	9.38%	18.75%
9	13.28%	0.00%	7.93%	33.33%	25.00%	56.25%	93.75%	0.00%	3.12%	100.00%	0.00%	0.00%	12.50%	25.00%	0.00%	0.00%	0.00%	0.00%	0.00%	25.00%
10	13.28%	0.00%	11.54%	50.00%	25.00%	60.00%	100.00%	0.00%	3.12%	100.00%	0.00%	0.00%	12.50%	25.00%	0.00%	0.00%	0.00%	0.00%	0.00%	25.00%
11	14.06%	0.00%	11.78%	50.00%	31.25%	60.00%	100.00%	0.00%	3.12%	100.00%	0.00%	0.00%	18.75%	31.25%	0.00%	0.00%	0.00%	0.00%	0.00%	31.25%
Average	19.66%	54.42%	9.21%	40.28%	15.62%	49.69%	77.60%	56.94%	9.90%	58.33%	22.92%	2.08%	10.42%	16.67%	5.21%	19.86%	19.01%	17.71%	20.05%	22.40%



# MantisNet RFP-NG MAU Stage 0

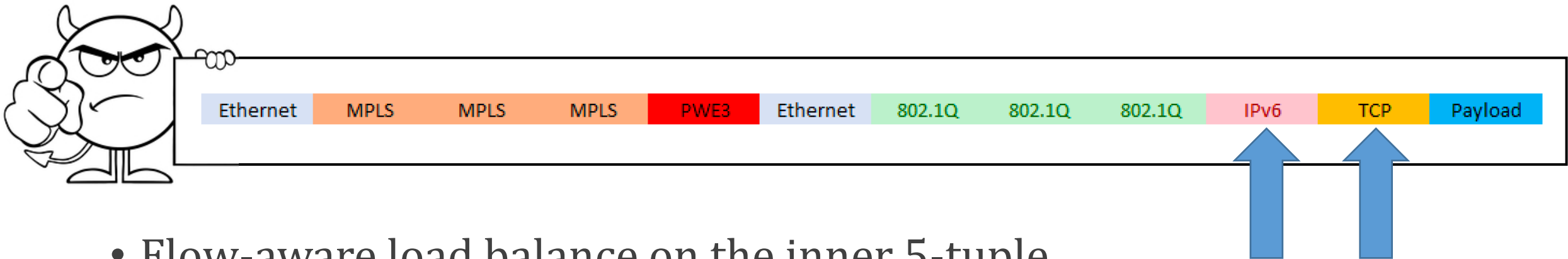


# So What Can Your IDS/Firewall Handle?



# CASE STUDY: Load Balancing From Hell

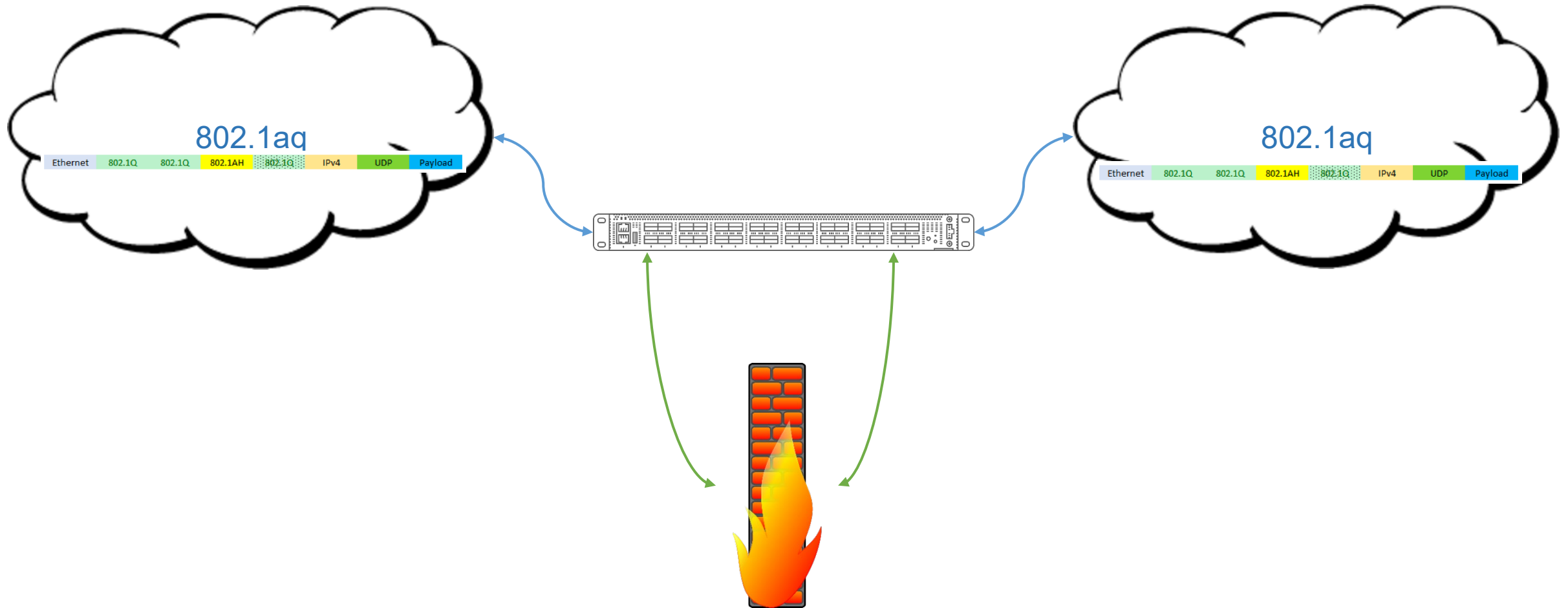
- 2x100G inputs
- 32x10G outputs
- Frame From Hell:



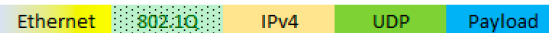
- Flow-aware load balance on the inner 5-tuple
- P4 Development time: 1 day



# CASE STUDY: MAC-in-MAC



P4 Development time: 3 days



# **CASE STUDY:** Intelligent Patch Panel/SDN

---

- Mix and match: 32x100G, 32x40G, or 128x10G
- Point-to-Point traffic direction
- “Monitor” (replicate to read-only port(s))
- P4 Development time: 1 day





# CASE STUDY: NetFlow

---

- 128k flowlet hash-based cache
- Selection criteria based on Packet Broker rule match
- Recorded Data:
  - Ingress port
  - IPv4 SIP/DIP/protocol
  - TCP/UDP/SCTP SPORT/DPORT
  - TCP flags
  - Start time/Update time
  - Packet count/Octet count
- 3 Different ways to egress flowlets
  - DMA rings, P4 generate\_digest() primitive, Packet cloning/repurposing
- P4 Development time: 1 month



# CASE STUDY: HTTP Attack

---

- 32k IPv4 CIDR SIP suppression table
- ~ Dozen throttle-attack detection signatures
- 8k URI signatures
- Clone key packets out of band to customer's follow on processor
  - Follow on processor populates IPv4 CIDR SIP rules to suppress/honeypot hosts based on key frame analysis
- P4 Development time: 2 days



# CASE STUDY: In-Band Telemetry

---

- Injected a customer header after Ethernet (custom etherType) containing:
  - Offset to start of inner-most IPv4/IPv6 header
  - Offset to start of payload
  - Original next etherType
- P4 Development Time: 2 days



# CASE STUDY: Out-of-Band Telemetry

---

- Used existing Packet Broker functionality to build an L4 port-based categorizer
  - 8k ports matched
  - 15 second update time
  - Raw and delta stats on a packet and octet basis
  - Output in JSON to a kafka queue
- P4 Development Time: 0 days! 😊



# Thank you!

## Questions?



**MantisNet**

### Contact:

Michael Reed

Vice President, Engineering

[reed@mantisnet.com](mailto:reed@mantisnet.com)