
# A Journey from OpenFlow to P4

## Improved Performance and Reduced Development Time

Jeff Elpern
Director of Product Management
NoviFlow

# NoviFlow: From OpenFlow to P4

Match-Action Programmable Pipeline
This is our DNA

Applications
- CyberMapper
- CGNMapper


Network Operating System
- OpenFlow
- P4 & P4 Speaker

# OpenFlow to P4: Order-of-Magnitude Results

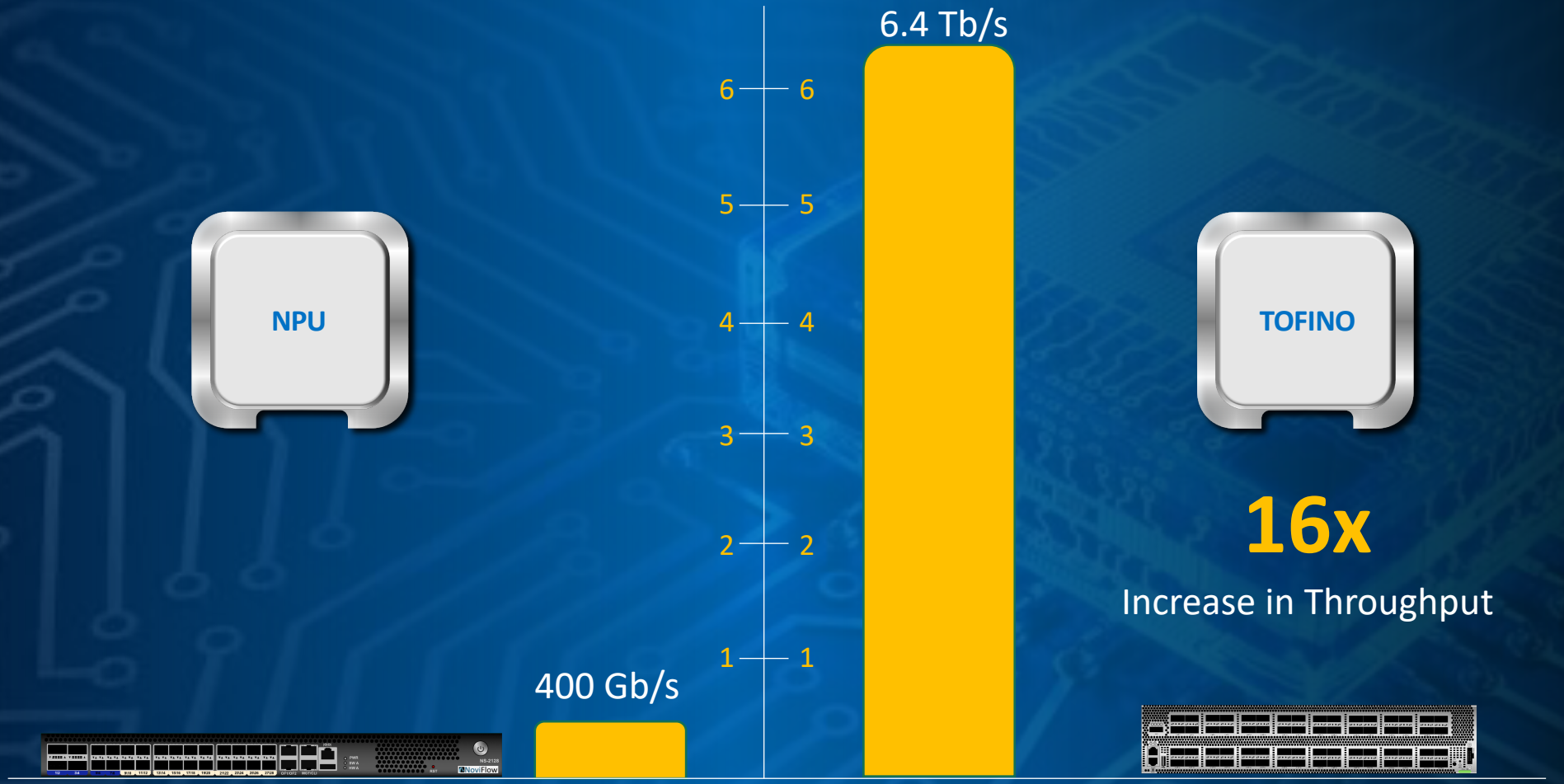
Match-Action Programmable Pipeline
This is our DNA

1. Huge Increase in Throughput

2. Huge improvement in application development time

# First Order-of-Magnitude: Throughput

## When dedicated to Match-Action Programmable Pipeline




NoviSwitch 2128
4x 40G + 24x 10G

5164 Whitebox
64x 100G





NoviFlow

# Second Order-of-Magnitude Result: Development

Match-Action Programmable Pipeline
This is our DNA

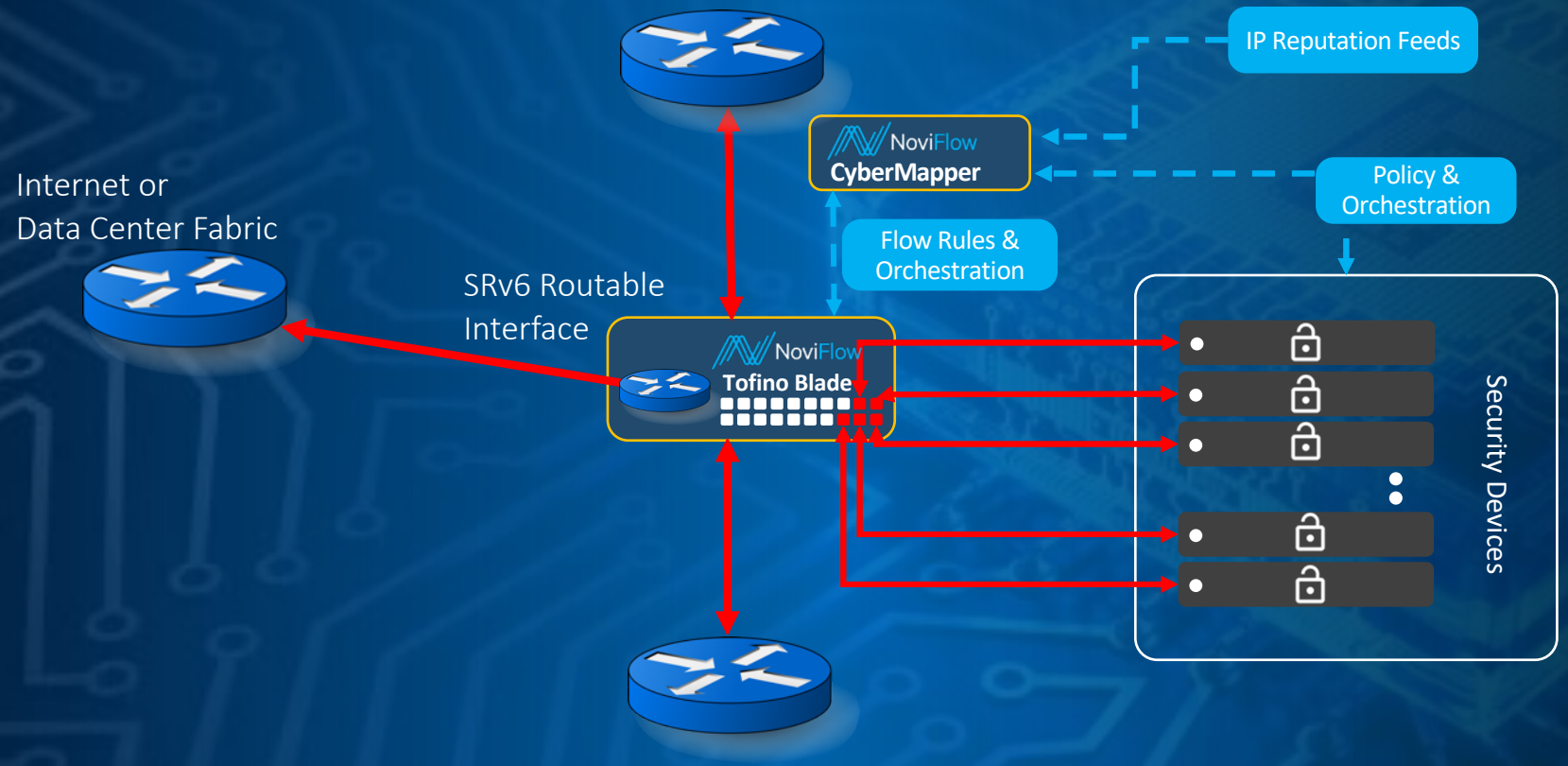
1. Big surprise

2. Headers moved out of specification and into programming language

3. Real world example – Implement SRv6 for CyberMapper

# CM Application Environment

## Bump in the Wire vs. SRv6







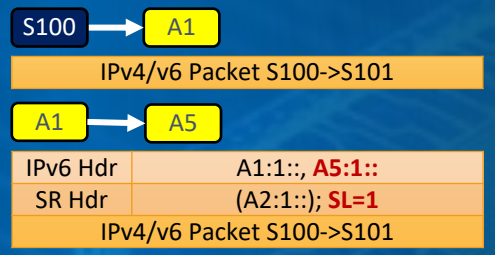

NoviFlow

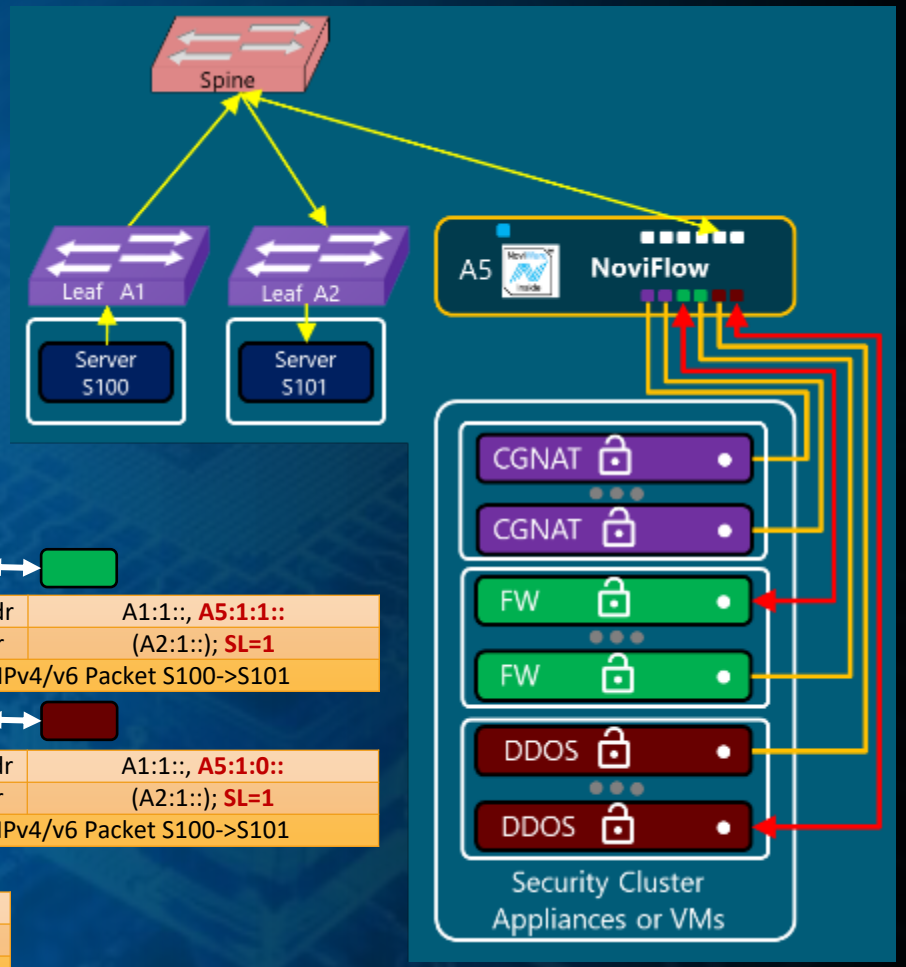
# SRv6 Project – Example of P4 Power

Task: Developed a new algorithm for CM that make the switch an addressable Security Proxy that load balances to a pool of Security tool --- SRv6 app

## Tasks

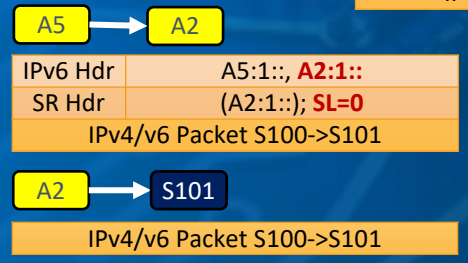
- Accept SRv6 packets
- Match on SRv6 SID for target security service
- Load Balance to Tool Farms
- Update SRv6 header and route














NoviFlow

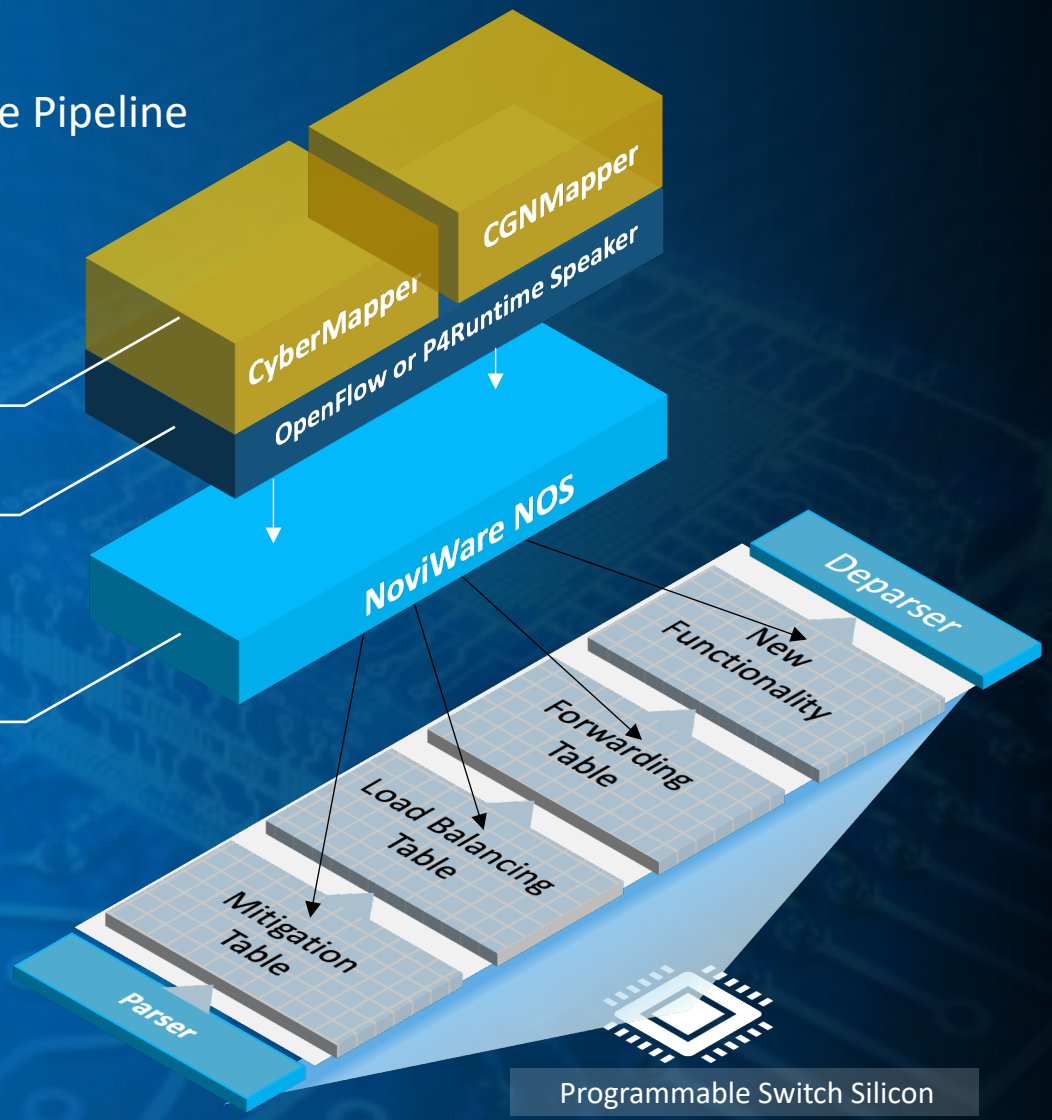
# NoviFlow: P4 Technology Stack

Match-Action Programmable Pipeline
This is our DNA

**Applications**
Load Balancing | Mitigation | NPB

**Controller**
P4Runtime Speaker

**Network Operating System**
NoviWare P4








NoviFlow

# Headers in NOS vs. Headers in P4 Code

## NOS Extensions for OpenFlow

- UDP Payload Matching & Handling
- IP Payload Matching & Handling
- L2MPLS Encapsulation / Decapsulation
- VxLAN Encapsulation / Decapsulation
- L2GRE Encapsulation / Decapsulation
- GTP Encapsulation / Decapsulation

Development moved from months to days

## SRv6 Header Defined in P4

```
// Main segment route header
header ipv6_srh_h {
    bit<8> next_header;   // Uses same values as IPv4 Protocol field
    bit<8> hdr_ext_len;   // Length of the header divided by 8 bytes, minus 1
    bit<8> routing_type;  // Identifies router header variant
    bit<8> segments_left; // Number of segments remaining
    bit<8> last_entry;    // Index in segment list of last element
    bit<8> flags;         // All unused and must be 0
    bit<16> tag;          // Tag for class or group of packets
}

// Individual segment items
header ipv6_srh_segment_list_t {
    bit<128> sid;       // IPv6 address representing a segment
}

parser ingress_parser( /* ... */ )
{
    // Only showing SRH states ...
    // Parse SRH before segments
    state parse_ipv6_srh {
        packet.extract(hdr.ipv6_srh);
        transition parse_ipv6_srh_segment_0;
    }

    // Parse SRH up to 6 segments, rejecting packet if more
    state parse_ipv6_srh_segment_0 {
        packet.extract(hdr.ipv6_srh_segment_list[0]);
        transition select(hdr.ipv6_srh.last_entry) {
            0: parse_ipv6_srh_next_header;
            default: parse_ipv6_srh_segment_1;
        }
    }
```





NoviFlow

# Second Order-of-Magnitude Result - Development

Separation of headers for specification and other adavntages

**Productivity Gains:**

- Accept SRv6 packets – Big Win (Parser)
- Match on SRv6 SID for target security service – Big Win (Metadata bus)
- Load Balance to Tool Farms – Equal between OpenFlow and P4
- Update SRv6 header and route – Big Win (Parse and Metadata bus)

Note:

- All together we believe dev effort reduced by an order of magnitude
- First release quality increased by programming headers in a high-level parser language





NoviFlow

# Summary

- Great experience
- P4 NoviWare NOS and P4Speaker delivered the potential of P4

- NoviFlow had productized the same tools used to create NoviFlow P4-based applications for other companies looking to develop in P4

# Thank You!

Learn more at
www.noviflow.com