P4 Developers Day Presentation

P4 in SDN-based Attack detection and AI-driven Security Mechanisms

Hello, I’m Reza

Welcome, Everyone! Thank you for being here today.

My name is Reza Fallahi Kapourchali, I have a Master’s degree in Computer Networks, and for the past five years, I’ve been working on P4, a journey filled with learning and growth.

This is a P4 developers day presentation and I hope it sparks some great discussions and ideas.

01

Aim.

Aim of this Introduction

I sincerely hope that this information proves valuable not only to the P4 community but also to the broader research community, contributing to the effective utilization and ongoing enhancement of P4.

Out Aim of this presentaion

- Brief overview of SDN
- Issues and limitaions of SDN
- Potential threats
- Network Attacks
- Why focusing on DDoS?
- Previous security mechanism in SDN
- AI solutions
- P4 as the best Solution
- Experience of P4+AI
- Limitations and discussions

Table of Contents.

01 > Our Aim

- Brief overview of SDN
- Importance of DDoS detection in SDN
- Role of P4

02 > Background, Motivation

- Traditional DDoS detection methods
- SDN advanced traffic analysis and mitigation
- The need for **P4**

03 > P4 Use Cases here

- Traffic filtering and anomaly detection
- Real-time packet inspection and classification
- Dynamic rule updates for mitigation

04 > AI & attack Detection in SDN

- Types of AI
- Role of ML in identifying DDoS attack patterns
- Feature extraction from SDN network flow data
- Example ML models
- ML-based vs. traditional rule-based detection

05 > Experience of P4 + AI

- Integration of ML with P4-based programmable switches
- Setup and dataset used for testing
- Performance evaluation

06 > Final Thougths

- How **P4** and **ML** can evolve in SDN security
- Questions

02

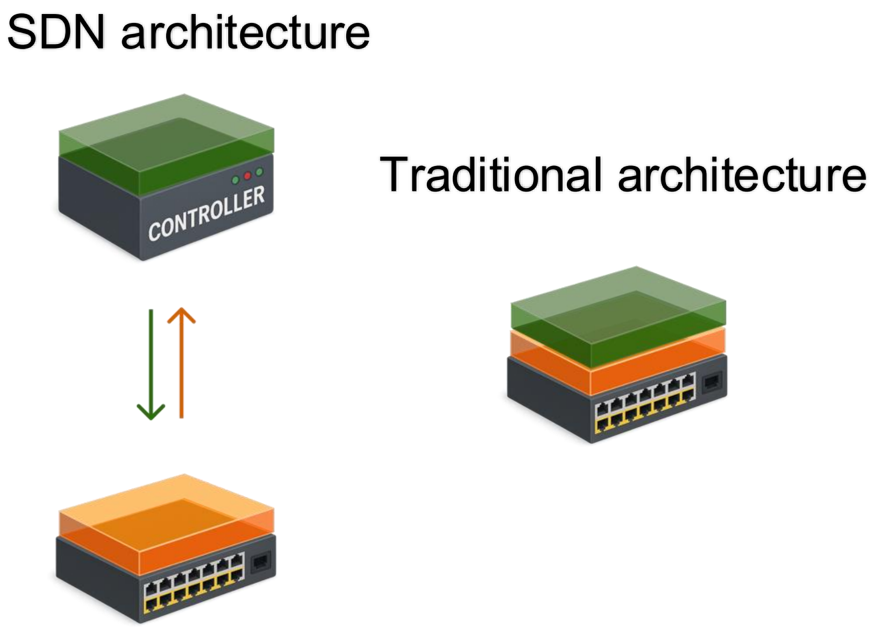
SDN and security concerns

Aim of this section

We will be discussing Software-Defined Networking (SDN) and security concerns to review its main challenges and limitations, and to explore how P4 is helping us to address these issues.

Brief review of SDN

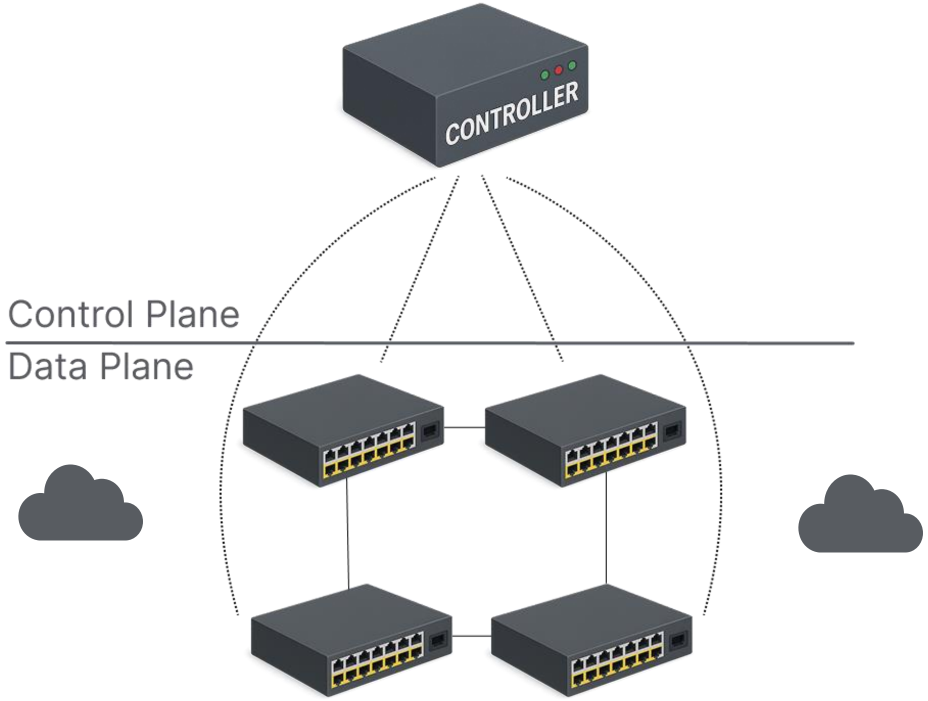
- SDN’s objective
  - Flexibility
  - Modularity
  - Control
- SDV vs. Traditional
  - Separated CP from DP
  - General-purpose devices
- Controller’s resources
  - Resource managements
  - Large-scale networks
  - Management decisions



Background and Motivation: SDN

What’s the problem?

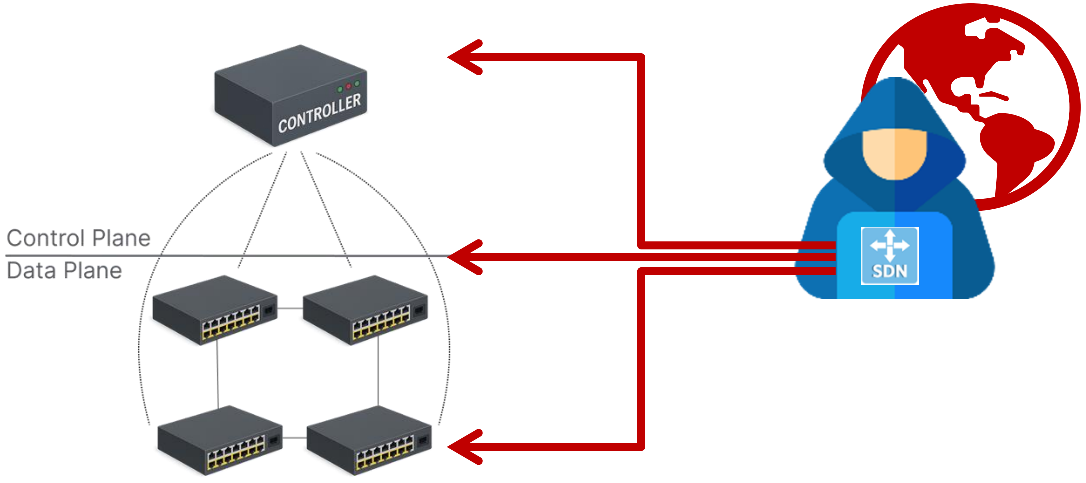
- SDN’s objective
  - Flexibility
  - Modularity
  - Control
- SDV vs. Traditional
  - Separated CP from DP
  - General-purpose devices
- Controller’s resources
  - Resource managements
  - Large-scale networks
  - Management decisions



Background and Motivation: SDN

Security concerns in SDN

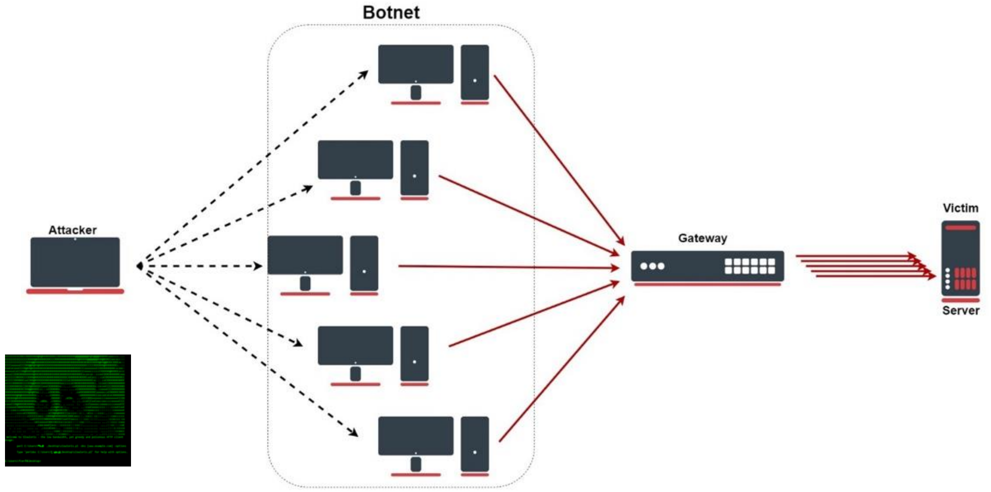
- SDN’s security
  - More critical
  - Controller for Whole network
- Attack targets on SDN
- Example: DDoS
  - Least effort
  - Destructive results
  - Smart!
- Previous proposed Solutions
  - General-purpose devices
  - Controller dependent
  - Cause limitations



Example: DDoS

- SDN's security
  - More critical
  - Controller for Whole network
- Attack targets on SDN
- Example: DDoS
  - Least effort
  - Destructive results
  - Smart!
- Previous proposed Solutions
  - General-purpose devices
  - Controller dependent
  - Cause limitations

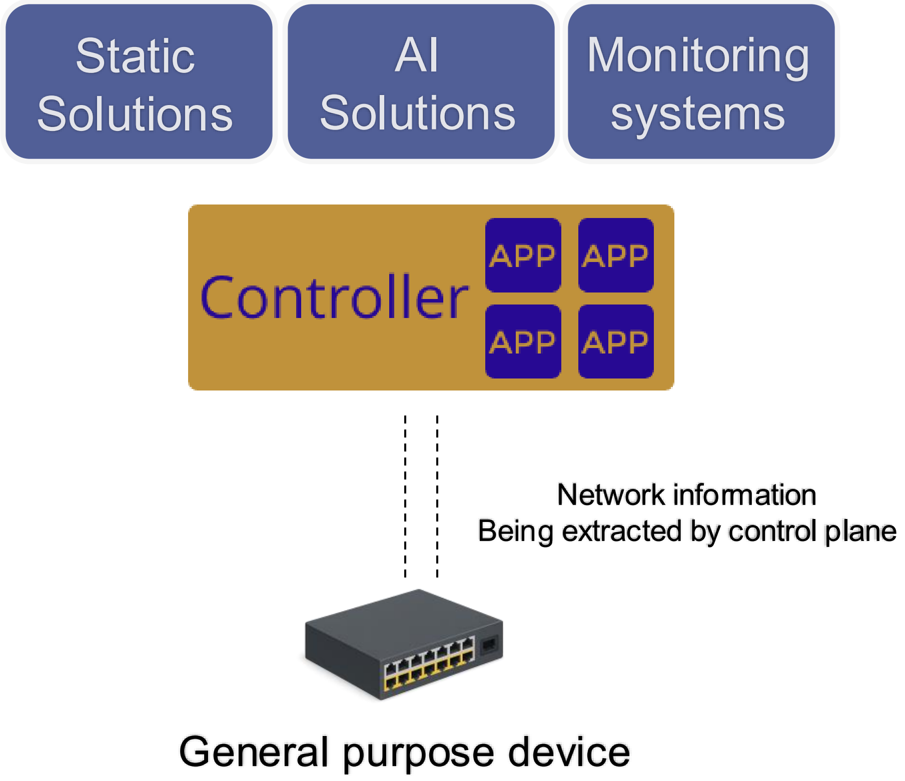
Distributed Denial of Service



AI + P4: Experience of P4 along with AI solution

Controller-dependent Proposed solution

- SDN’s security
  - More critical
  - Controller for Whole network
- Attack targets on SDN
- Example: DDoS
  - Least effort
  - Destructive results
  - Smart!
- Previous proposed Solutions
  - General-purpose devices
  - Controller dependent or 3rd party tools
  - What can we learn here?

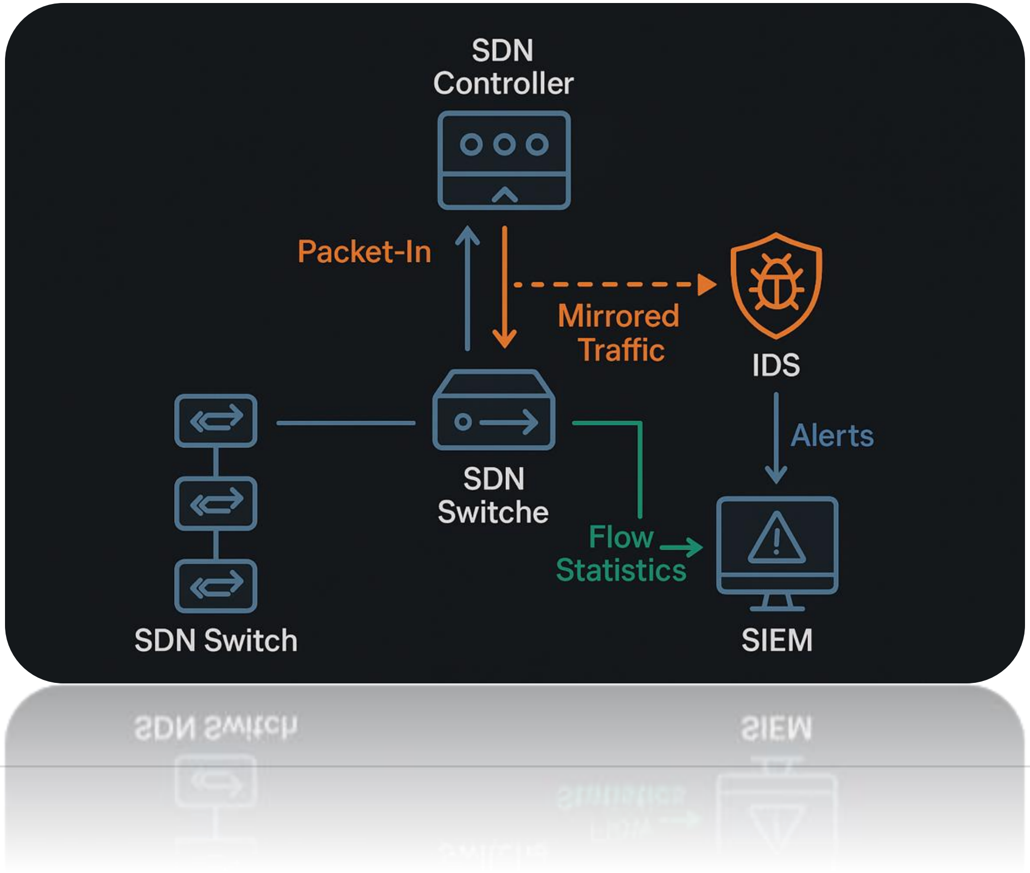


Background and Motivation: SDN

3rd party devices solution

- Previous proposed Solutions
  - General-purpose devices
  - Controller dependent or 3rd party tools
  - What can we learn here?

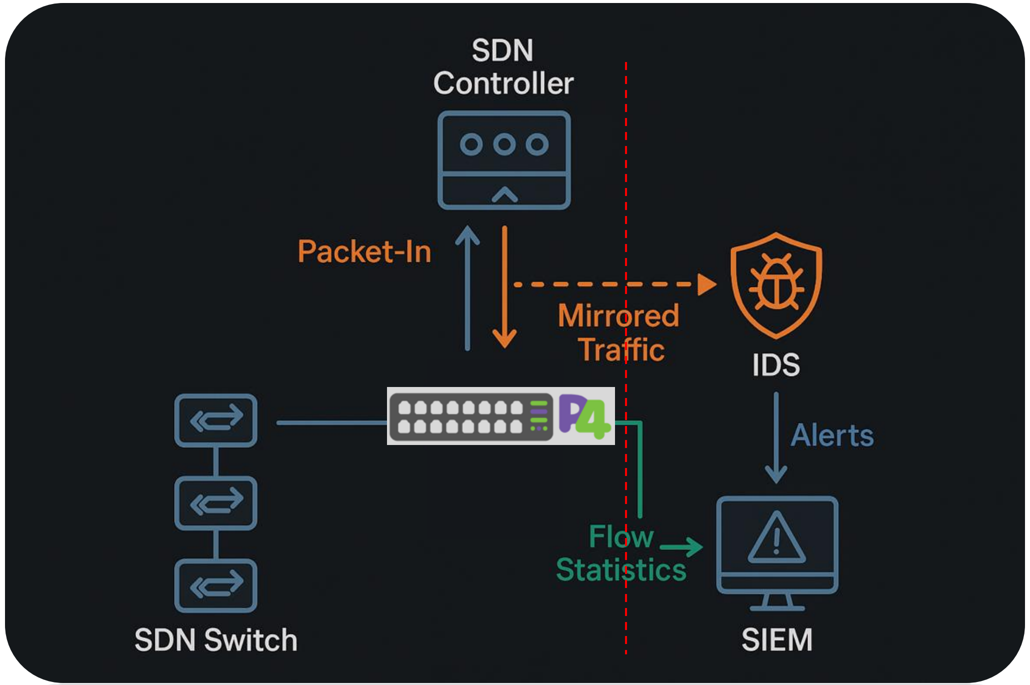
- Traffic Collection (packet/flow level)
  - sFlow, NetFlow, Wireshark
- Feature Extraction
  - CICFlowMeter, Scapy
- Anomaly Detection
  - Zeek, ML
- Signature Matching
  - Snort
- Alerting & Response
  - Splunk



Background and Motivation: SDN

Our Solution : P4

- How P4 can help?
  - No processing overhead for controller
  - Task offloading
  - Controller resource optimizations in large-scale networks
  - More personalized control over packet processing
  - Custom Monitoring
  - On-demand Implementation of ideas



03

P4 Use Cases here.

Aim of this section

In this section, we will explore how P4's programmability can help us advance to the next level in network innovation and flexibility. We will highlight the functionalities that P4 offers to enhance security mechanisms and improve the management of SDN, helping us overcome existing limitations.

P4 Use Cases: Traffic filtering and anomaly detection

Network monitoring with P4

- Traffic Collection
  - Packet inspections
- Feature Extraction
  - Flow-level inspections
- Anomaly Detection
  - Programmability itself
- Signature Matching
  - Conditions, Flow management and modularity of the code
- Alerting & Response
  - Cloning, Mirroring, etc.

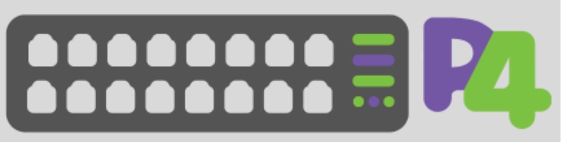
```
 ingress () {
ble routing {
   key = { hdr.ipv4.dstAddr : lpm; }
   actions = { drop; route; }
   size = 2048;

  {
   routing.apply();
}
```

We are talking about how P4 is improving the whole system

P4 Use Cases: Traffic filtering and anomaly detection

Traffic Collection with P4

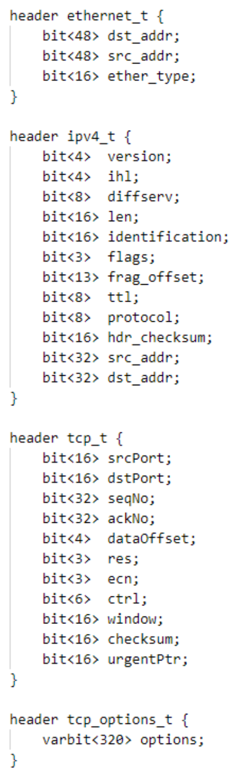
- Convenient TCP/IP
- Custom protocol stacks
- Parsing incoming traffic
- Meters and Counters for Traffic
- Queue and buffer information for traffic
- Host monitoring
- Stateful data processing
- Hash functions
- Packet modifications as per need
- Port Information
- Control blocks as per need
- Different packet re-generation methods
- Tables



Convenient updates on any protocol stack modifications

Custom protocol header For management

Parsing functionalities for further managements



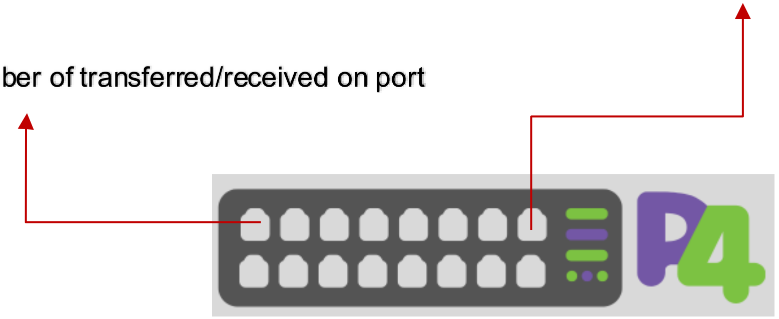
```
header ethernet_t {
    bit<48> dst_addr;
    bit<48> src_addr;
    bit<16> ether_type;
}

header ipv4_t {
    bit<4>  version;
    bit<4>  ihl;
    bit<8>  diffserv;
    bit<16> len;
    bit<16> identification;
    bit<3>  flags;
    bit<13> frag_offset;
    bit<8>  ttl;
    bit<8>  protocol;
    bit<16> hdr_checksum;
    bit<32> src_addr;
    bit<32> dst_addr;
}

header tcp_t {
    bit<16> srcPort;
    bit<16> dstPort;
    bit<32> seqNo;
    bit<32> ackNo;
    bit<4>  dataOffset;
    bit<3>  res;
    bit<3>  ecn;
    bit<6>  ctrl;
    bit<16> window;
    bit<16> checksum;
    bit<16> urgentPtr;
}

header tcp_options_t {
    varbit<320> options;
}
```

We are talking about how P4 is improving the whole system

P4 Use Cases: Traffic filtering and anomaly detection

Traffic Collection with P4

- Convenient TCP/IP
- Custom protocol stacks
- Parsing incoming traffic
- Meters and Counters for Traffic
- Queue and buffer information for traffic
- Host monitoring
- Stateful data processing
- Hash functions
- Packet modifications as per need
- Port Information
- Control blocks as per need
- Different packet re-generation methods
- Tables



Setting the port rate limit with Meters

Counting number of transferred/received on port

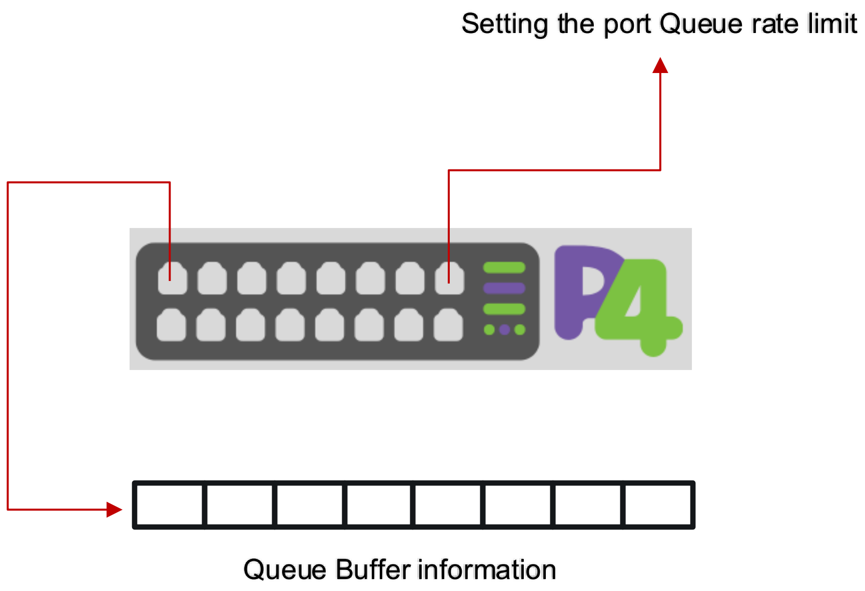
```
counter(MAX_PORTS, CounterType.packets_and_bytes) trsfr_counter;
counter(MAX_PORTS, CounterType.packets_and_bytes) rcved_counter;
```

We are talking about how P4 is improving the whole system

P4 Use Cases: Traffic filtering and anomaly detection

Traffic Collection with P4

- Convenient TCP/IP
- Custom protocol stacks
- Parsing incoming traffic
- Meters and Counters for Traffic
- Queue and buffer information for traffic
- Host monitoring
- Stateful data processing
- Hash functions
- Packet modifications as per need
- Port Information
- Control blocks as per need
- Different packet re-generation methods
- Tables



We are talking about how P4 is improving the whole system

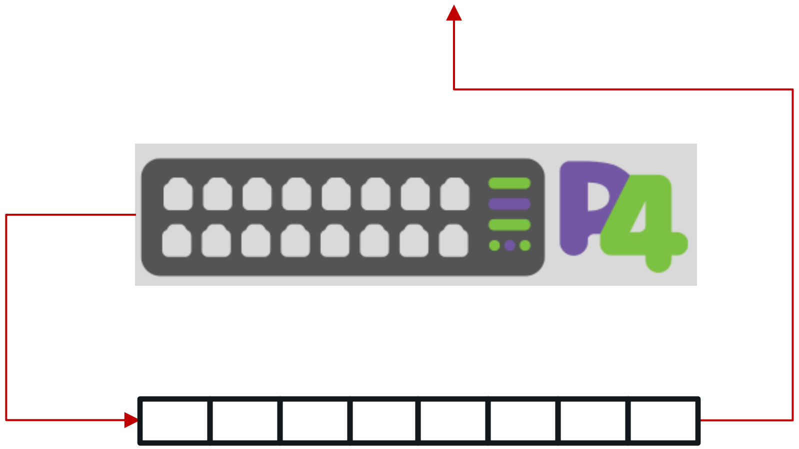
P4 Use Cases: Traffic filtering and anomaly detection

Traffic Collection with P4

- Convenient TCP/IP
- Custom protocol stacks
- Parsing incoming traffic
- Meters and Counters for Traffic
- Queue and buffer information for traffic
- Host monitoring
- Stateful data processing
- Hash functions
- Packet modifications as per need
- Port Information
- Control blocks as per need
- Different packet re-generation methods
- Tables

Register<bit<32>>(64) register;

Stateful data processing using Registers



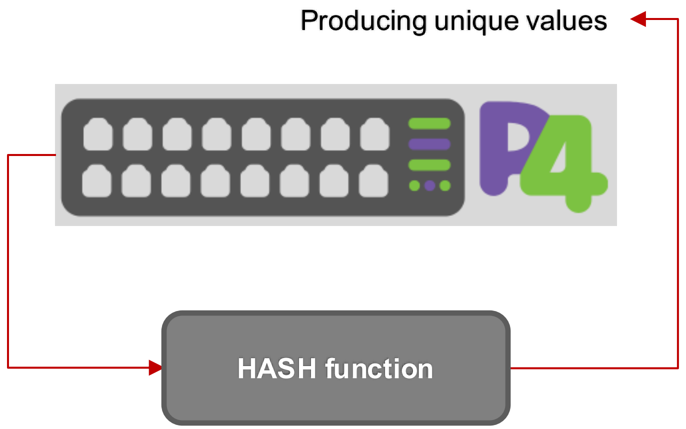
Host Monitoring using the Registers

We are talking about how P4 is improving the whole system

P4 Use Cases: Traffic filtering and anomaly detection

Traffic Collection with P4

- Convenient TCP/IP
- Custom protocol stacks
- Parsing incoming traffic
- Meters and Counters for Traffic
- Queue and buffer information for traffic
- Host monitoring
- Stateful data processing
- Hash functions
- Packet modifications as per need
- Port Information
- Control blocks as per need
- Different packet re-generation methods
- Tables



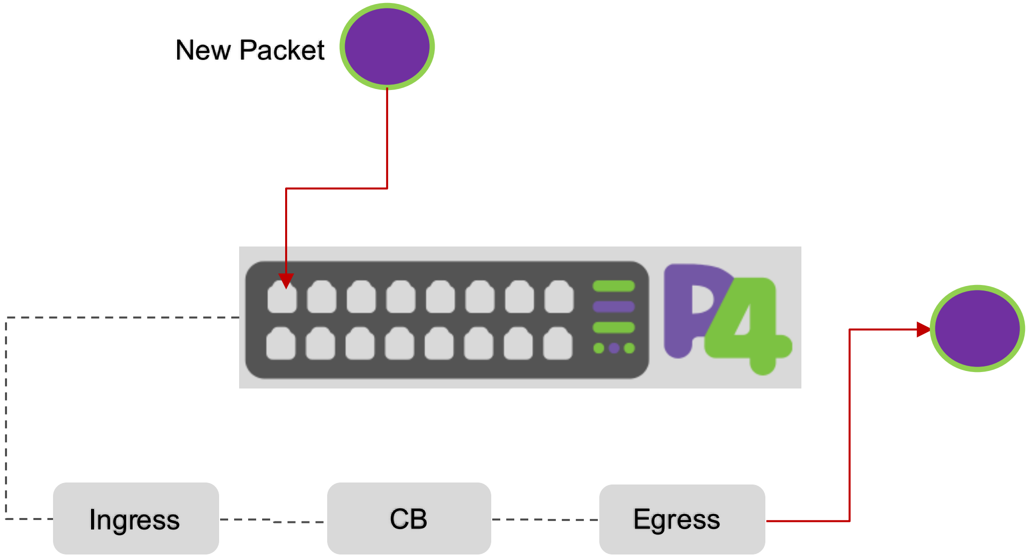
```
hash(var, HashAlgorithm.crc16, (bit<32>)0, {hdr.ethernet.src_addr}, (bit<32>)31);
```

We are talking about how P4 is improving the whole system

P4 Use Cases: Traffic filtering and anomaly detection

Traffic Collection with P4

- Convenient TCP/IP
- Custom protocol stacks
- Parsing incoming traffic
- Meters and Counters for Traffic
- Queue and buffer information for traffic
- Host monitoring
- Stateful data processing
- Hash functions
- Packet modifications as per need
- Port Information
- Control blocks as per need
- Different packet re-generation methods
- Tables



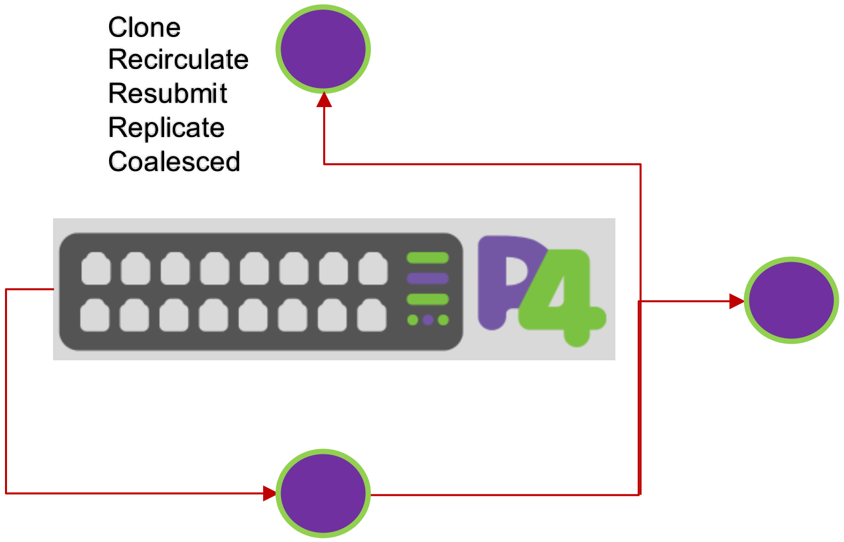
Default and Custom Control Blocks

We are talking about how P4 is improving the whole system

P4 Use Cases: Traffic filtering and anomaly detection

Traffic Collection with P4

- Convenient TCP/IP
- Custom protocol stacks
- Parsing incoming traffic
- Meters and Counters for Traffic
- Queue and buffer information for traffic
- Host monitoring
- Stateful data processing
- Hash functions
- Packet modifications as per need
- Port Information
- Control blocks as per need
- Different packet re-generation methods
- Tables



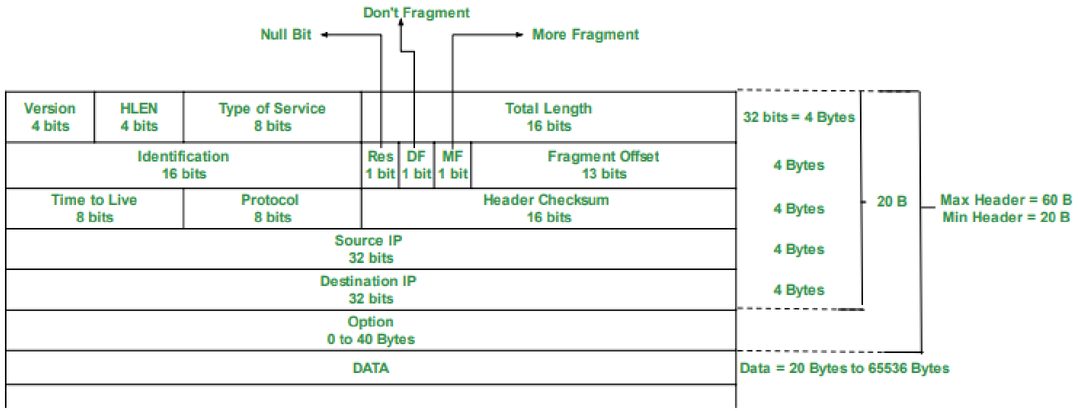
```
clone(CloneType.I2E, CLONE_ID);
clone(CloneType.E2E, CLONE_ID);
```

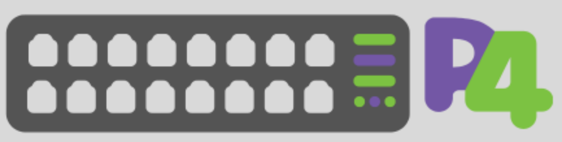
We are talking about how P4 is improving the whole system

P4 Use Cases: Real-time packet inspection and classification

Packet inspection with P4

- Deep packet Inspections
- Protocol header value
- Standard metadata values for management
- Custom metadata values for even more management
- Default and Custom Control blocks
- Utilization of basic timestamps
- Available metadata for better management
- Custom Tables for necessary managements





We are talking about how P4 is improving the whole system

P4 Use Cases: Real-time packet inspection and classification

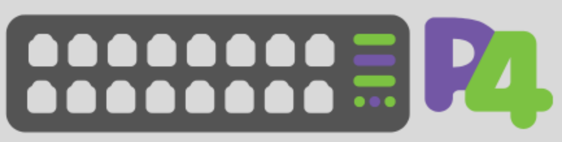
Packet inspection with P4

- Deep packet Inspections
- Protocol header value
- Standard metadata values for management
- Custom metadata values for even more management
- Default and Custom Control blocks
- Utilization of basic timestamps
- Available metadata for better management
- Custom Tables for necessary managements

Inspecting each packet's characteristic to extract packet-level information

Packet

1. Parsing each packet
2. Standard meta data
3. Custom meta data
4. Header field values
5. Custom data types
6. Custom Tables
7. Passing information between control Blocks for management
8. Checksum

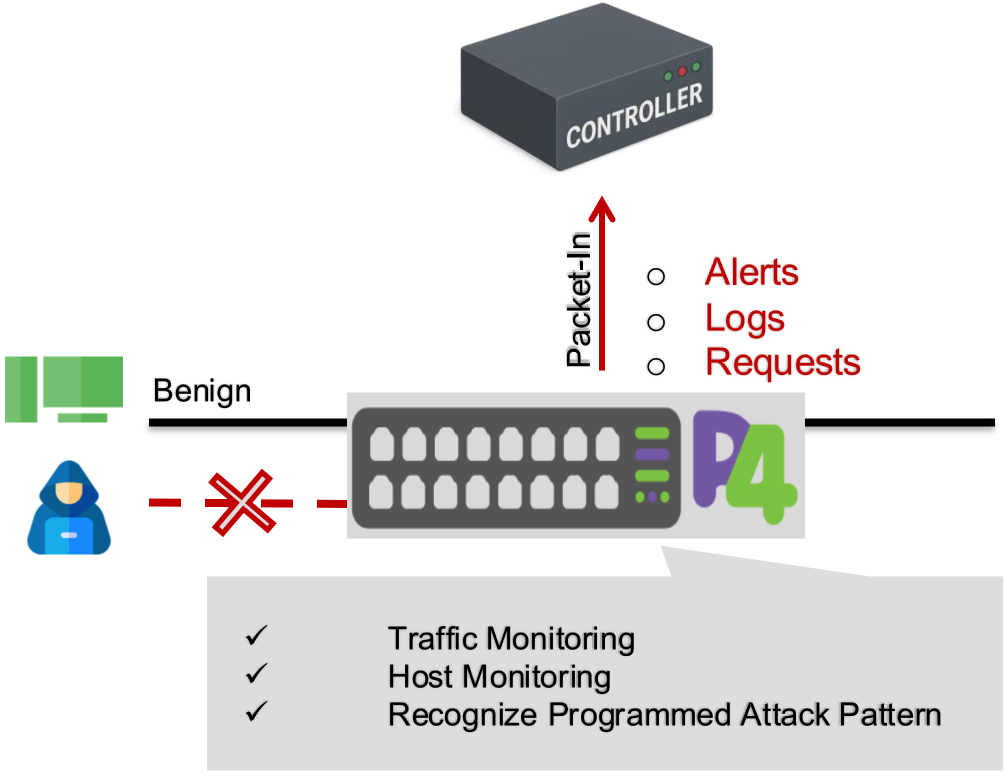


We are talking about how P4 is improving the whole system

P4 Use Cases: Real-time packet inspection and classification

Anomaly Detection & Alerting with P4

- Custom Actions serving as functions
- Programmability of new attack patterns
- Statistical analysis through mathematical operations
- Appropriate Logs for each switch
- Fast reactions with low processing overhead for the controller
- Convenient communication with the SDN controller

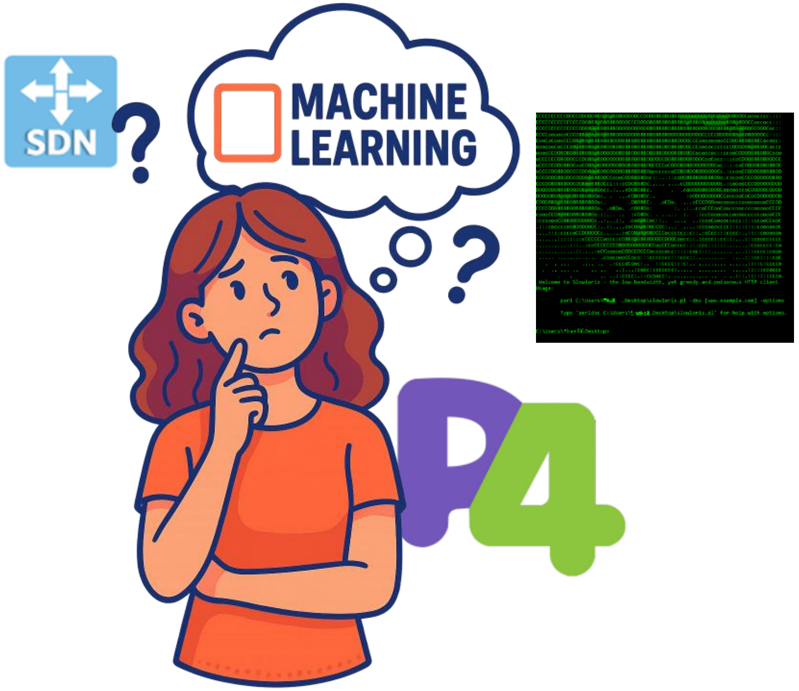


We are talking about how P4 is improving the whole system

P4 Use Cases: Dynamic rule updates for mitigation

We cannot depend on P4 solely

- We still need dynamic solutions
  - P4 -> overcome limitations
  - P4 -> Innovation in solutions
  - P4 -> more control
- Attacks adapt and evolve
  - We still need Dynamic solutions!



```
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client
Usage:

        perl C:\Users\[redacted]\Desktop\slowloris.pl -dns [www.example.com] -options

        Type 'perldoc C:\Users\[redacted]\Desktop\slowloris.pl' for help with options.


C:\Users\[redacted]\Desktop>
```

How can we improving it even further?

04

AI & attack Detection in SDN.

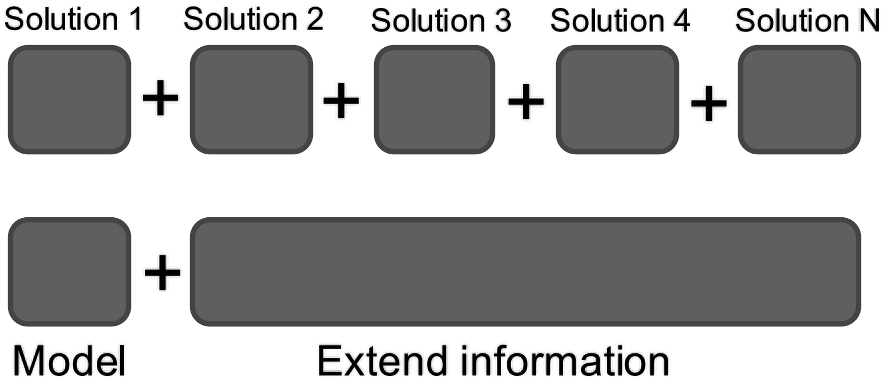
Aim of this Introduction

In this section, we will provide a brief overview of AI concepts and solutions, aiming to address key questions about their necessity and what we seek to achieve through their application.

AI & attack Detection : ML-based vs. traditional rule-based detection

Why AI solutions are better?

- Why do we need dynamic AI approach?
  - Detections based on behavior not the Approach
  - Focusing on info instead of proposing new solution



We are talking about AI solutions in general terms

AI & attack Detection : Types of AI

Artificial Intelligence

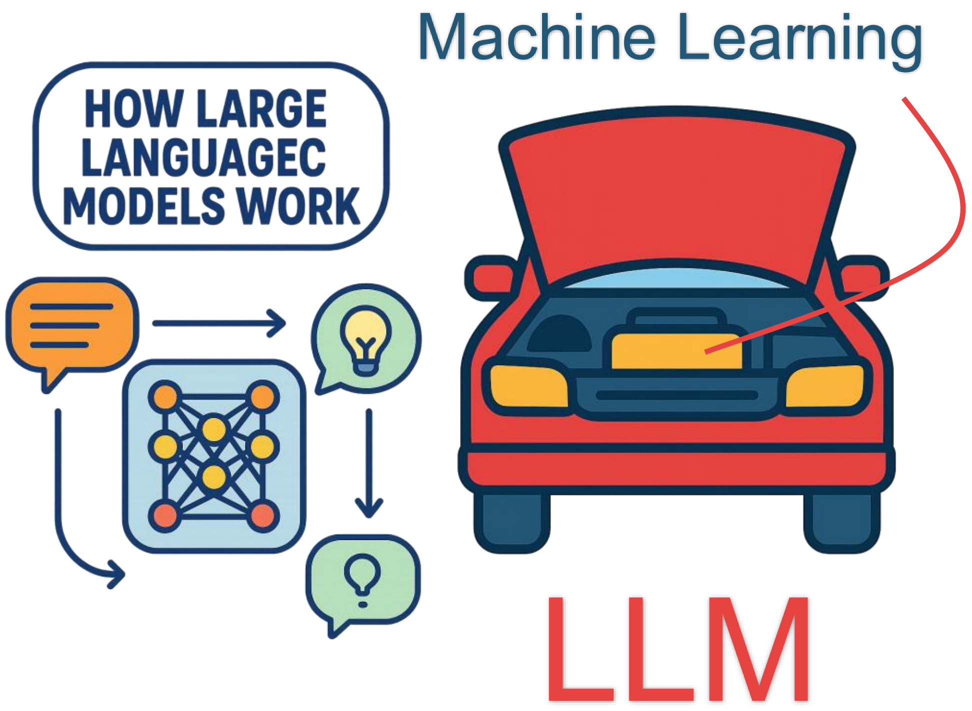
- What is AI?
- Types of AI in learning
- What is a model?
- AI nowadays
  - Large Language Models
  - Predict patterns
- Foundation: Machine Learning
  - Lower resources
  - Least cost
  - Customizable

Supervised Learning
Unsupervised Learning
Reinforcement Learning
Semi-Supervised Learning
Self-Supervised Learning
Online Learning

We are talking about AI solutions in general terms

LLM and ML

- What is AI?
- Types of AI in learning
- What is a model?
- AI nowadays
    - Large Language Models
    - Info or predict
- Foundation: Machine Learning
    - Lower resources
    - Least cost
    - Customizable

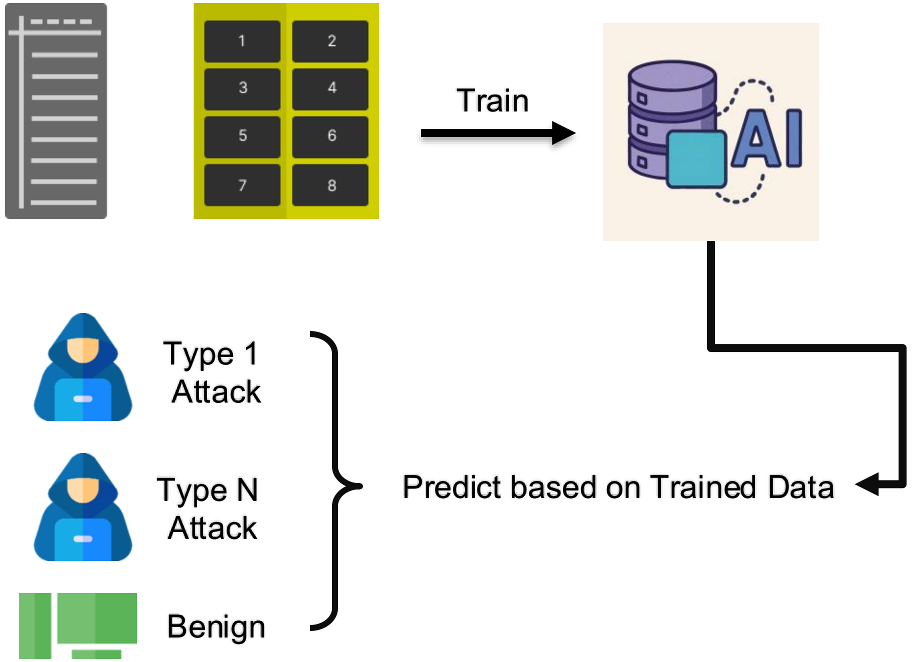


Machine Learning is the heart of AI tools nowadays

AI & attack Detection : Types of AI

Training Process and Accuracy

Aspect	Machine Learning	Large Language Models
Usage	Anomaly detection	Threat intelligence gathering
Data Type	Structured data (network traffic, flow data)	Unstructured data (logs, emails, social media, reports)
Strength	Real-time detection, automated defense	Contextual understanding
Scalability	Scales well with large datasets	Requires extensive computational resources for real-time use
Adaptability	Can be trained on evolving data, but slower to adapt	Can handle novel, evolving attacks via unstructured data learning
Computational Complexity	Requires significant computation, especially for deep models	Extremely high for training
Challenges	Requires large labeled datasets, may struggle with novel attacks	Struggles with low-level network analysis, privacy concerns

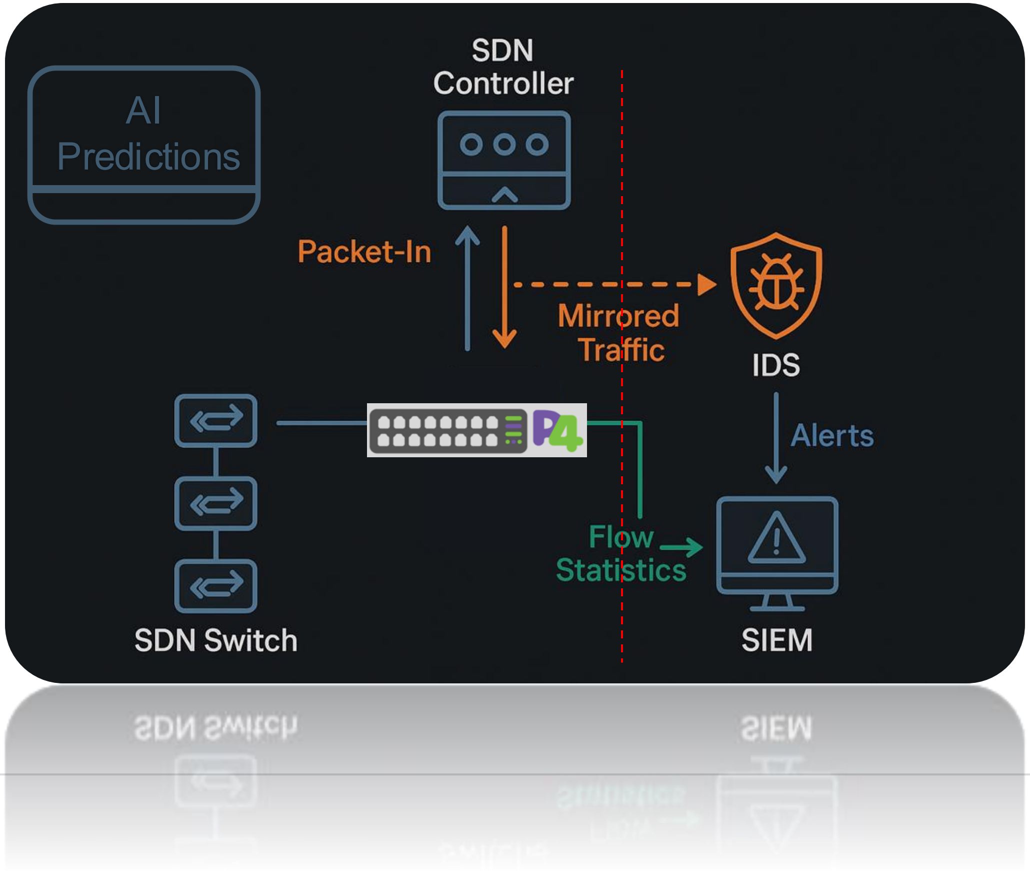


We are talking about LLM vs ML approach

AI & attack Detection : Role of ML in identifying DDoS attack patterns

Network Attacks, P4 and, AI solutions

- AI -> Security Predictions
- P4 -> Security Network Information



AI solution + P4 data plane programmability

05

Experience of P4 + AI.

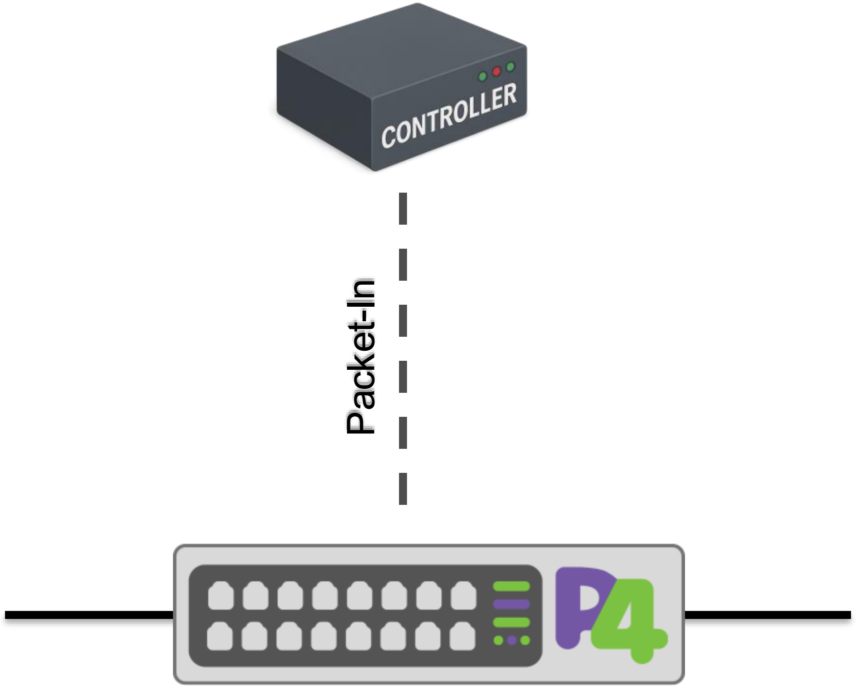
Aim of this Introduction

Add a brief introduction of your section here: Let’s dive in and get to know some interesting facts about animals!

AI + P4: Experience of P4 along with AI solution

Integration of AI and P4 solution

- Information + Prediction
- System's representation
  - Scenario 1
  - Scenario 2
- Advantages of this system
  - All-In-One solution through the programming
- Limitations
  - P4's limitations



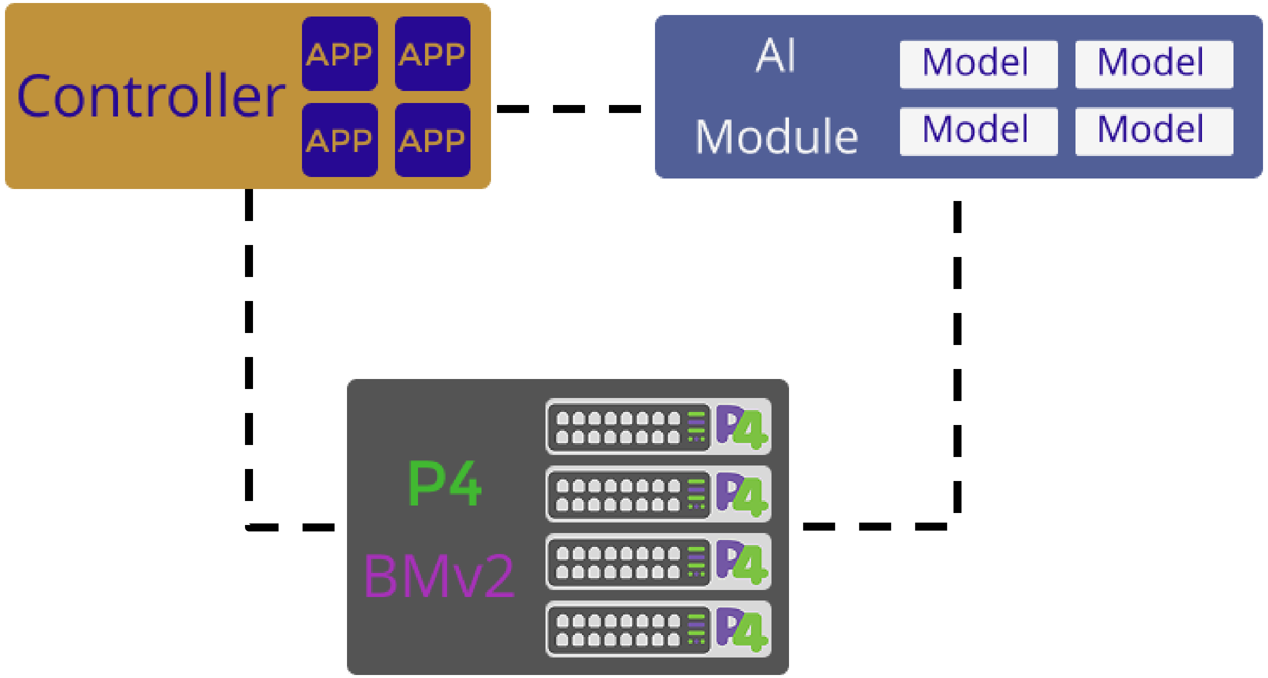
Extract Network Information

Experience on the combination of P4+AI for SDN security

AI + P4: Experience of P4 along with AI solution

Integration of ML with P4 switches

- Information + Prediction
- System's representation
  - Scenario 1
  - Scenario 2
- Advantages of this system
  - All-In-One solution through the programming
- Limitations
  - P4's limitations

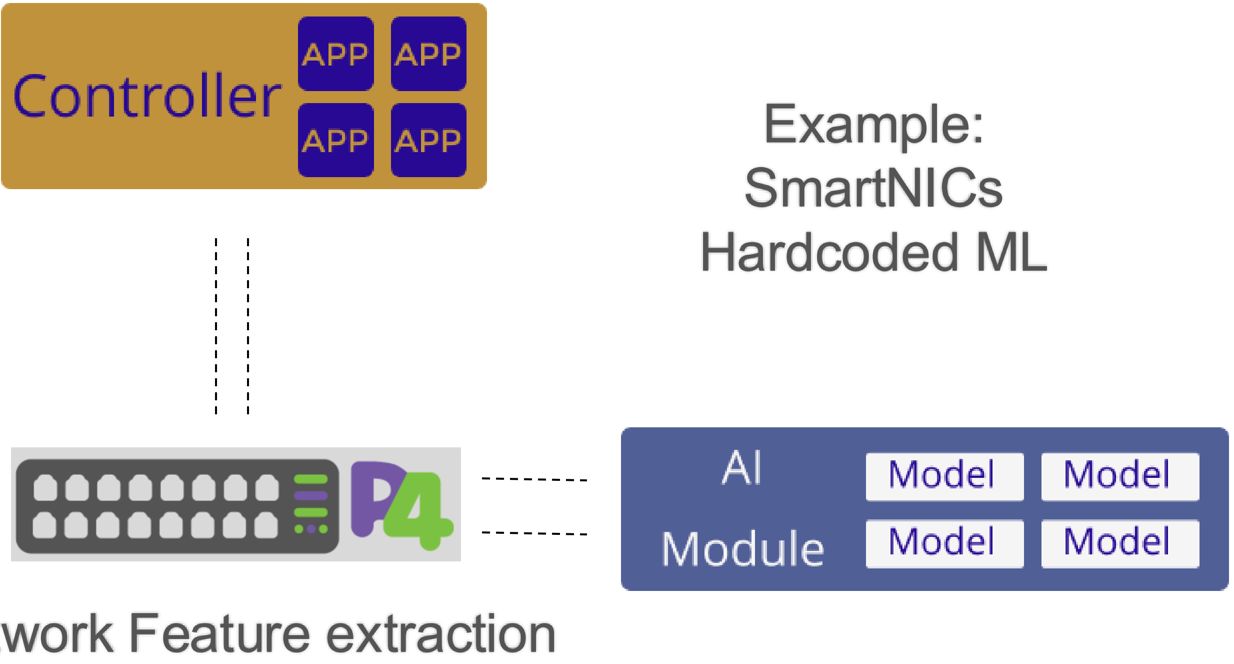


Experience on the combination of P4+AI for SDN security

AI + P4: Experience of P4 along with AI solution

Integration of ML with P4 switches

- Information + Prediction
- System's representation
  - Scenario 1
  - Scenario 2
- Advantages of this system
  - All-In-One solution through the programming
- Limitations
  - P4's limitations

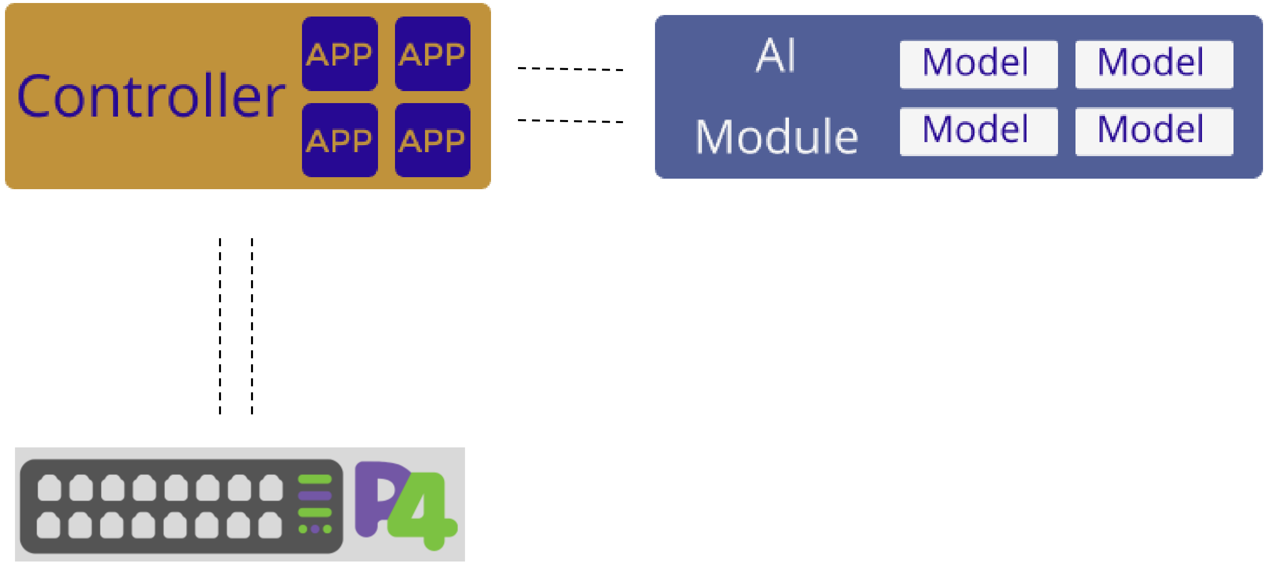


Experience on the combination of P4+AI for SDN security

AI + P4: Experience of P4 along with AI solution

Integration of ML with P4 switches

- Information + Prediction
- System's representation
  - Scenario 1
  - Scenario 2
- Advantages of this system
  - All-In-One solution through the programming
- Limitations
  - P4's limitations

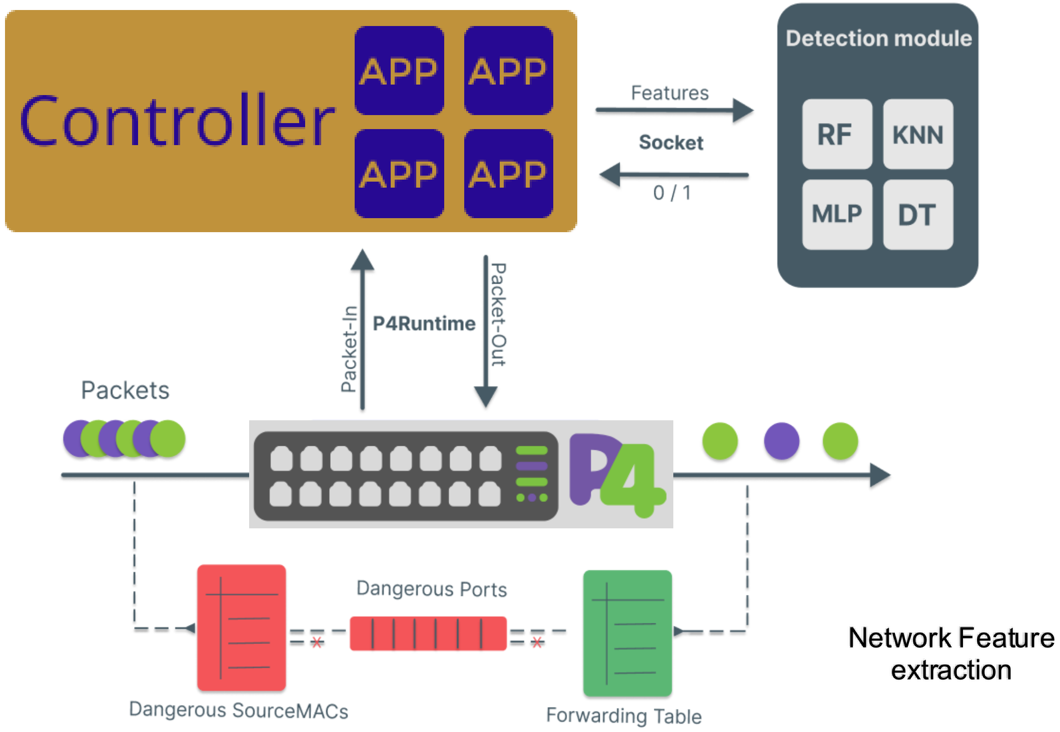


Network Feature extraction

AI + P4: Experience of P4 along with AI solution

Integration of ML with P4 switches

- Information + Prediction
- System's representation
  - Scenario 1
  - Scenario 2
- Advantages of this system
  - All-In-One solution through the programming
- Limitations
  - P4's limitations

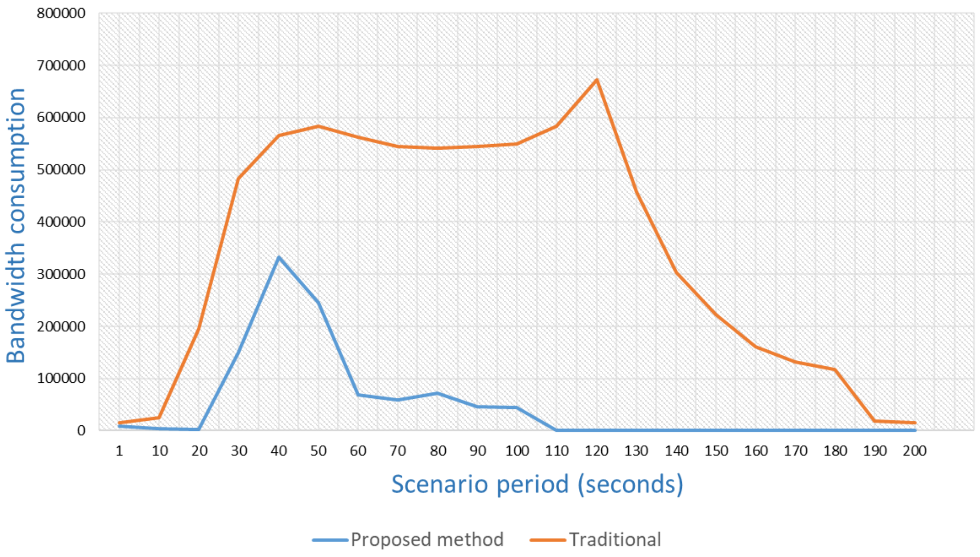


Experience on the combination of P4+AI for SDN security

AI + P4: Experience of P4 along with AI solutions

Attack bandwidth consumption

- My Experience
  - Accurate Attack Handling
  - Less bandwidth occupation
  - Accurate for management
  - Less processing overhead for the Controller
  - Programmed IDS + SEIM

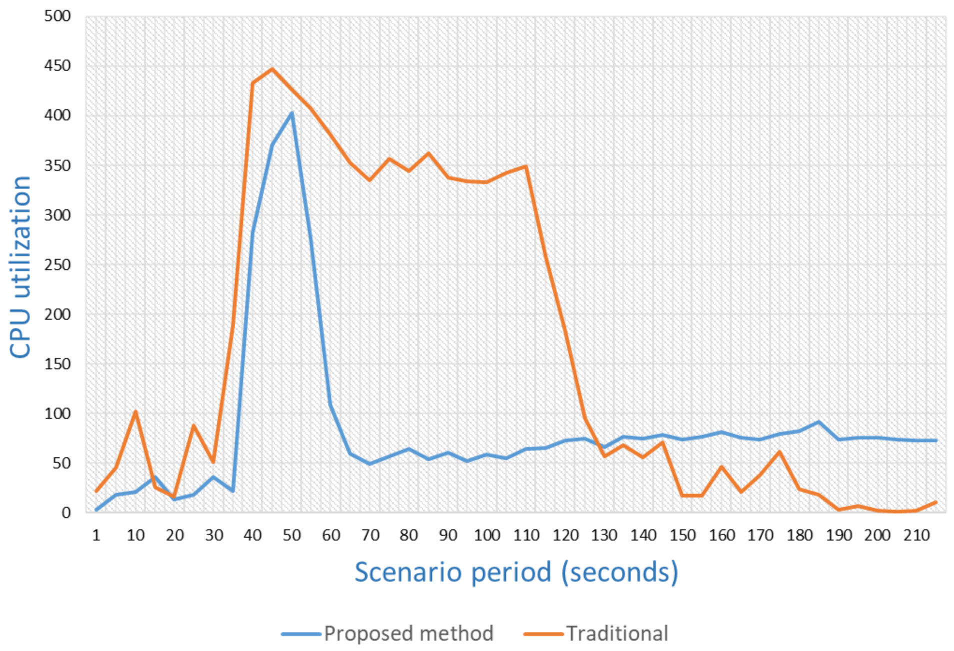


My Experience on the combination of P4+AI for SDN security

AI + P4: Experience of P4 along with AI solutions

Performance evaluation

- My Experience
  - Accurate Attack Handling
  - Less bandwidth occupation
  - Accurate for management
  - Less processing overhead for the Controller
  - Programmed IDS + SEIM



My Experience on the combination of P4+AI for SDN security

06

Final Thougths.

Final Thoughts

Which features of P4 I personally found helpful in my work?

- But All of this was not going to happen without **stateful data processing and hash** functions!

- The **control** I have over the packet processing procedure!
  - I have been able to extract 33 network-related ML feature values from the flows! per host! using the P4!
  - Easy access to the header field values
  - Convenient functionalities implemented in the language for bit-level calculations
  - Detailed Logs
  - The Standard metadata

Final Thoughts

Is there anything about P4 that I found making my work more difficult?

- Lack of direct Timing Methods in the BMv2 switch
- Limits on the Division Operations
- Accessing Meters or counters associated with each port
- Protocol Hashed payloads
- Convenient access to the Payload values

Final Thoughts

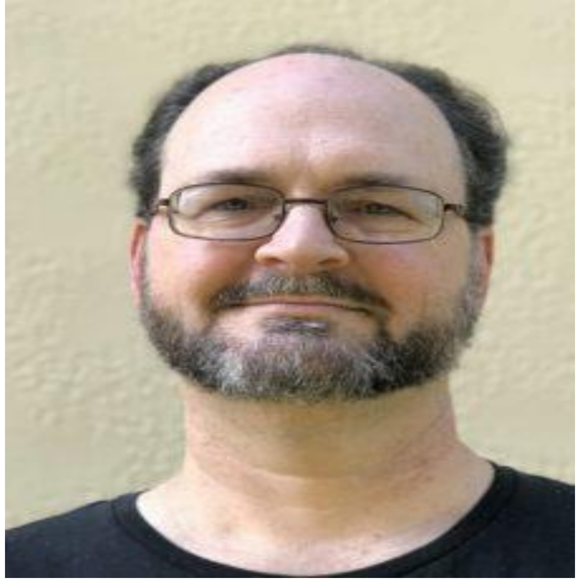
My thoughts on potential improvements on BMv2

- Implementation of Switch Time methods through programming
- Extending the operations to support direct divisions (even for limited equations)
- Adding APIs for external applications
- Accessing more direct functionalities of the switch through the code

My thanks to

Dr. Reza Mohammadi

My supervisor who introduced me to the P4



Mr. Andy Fingerhut

key role in sustaining P4, committing substantial time and effort to keep it moving forward

Mr. Antonin Bas

Provided Invaluable guidance and invested considerable time to improve P4

Credits.

- My sincere thanks to ***Denise Barton*** for her outstanding support and seamless coordination behind the scenes—your efforts have made a real difference and are truly appreciate

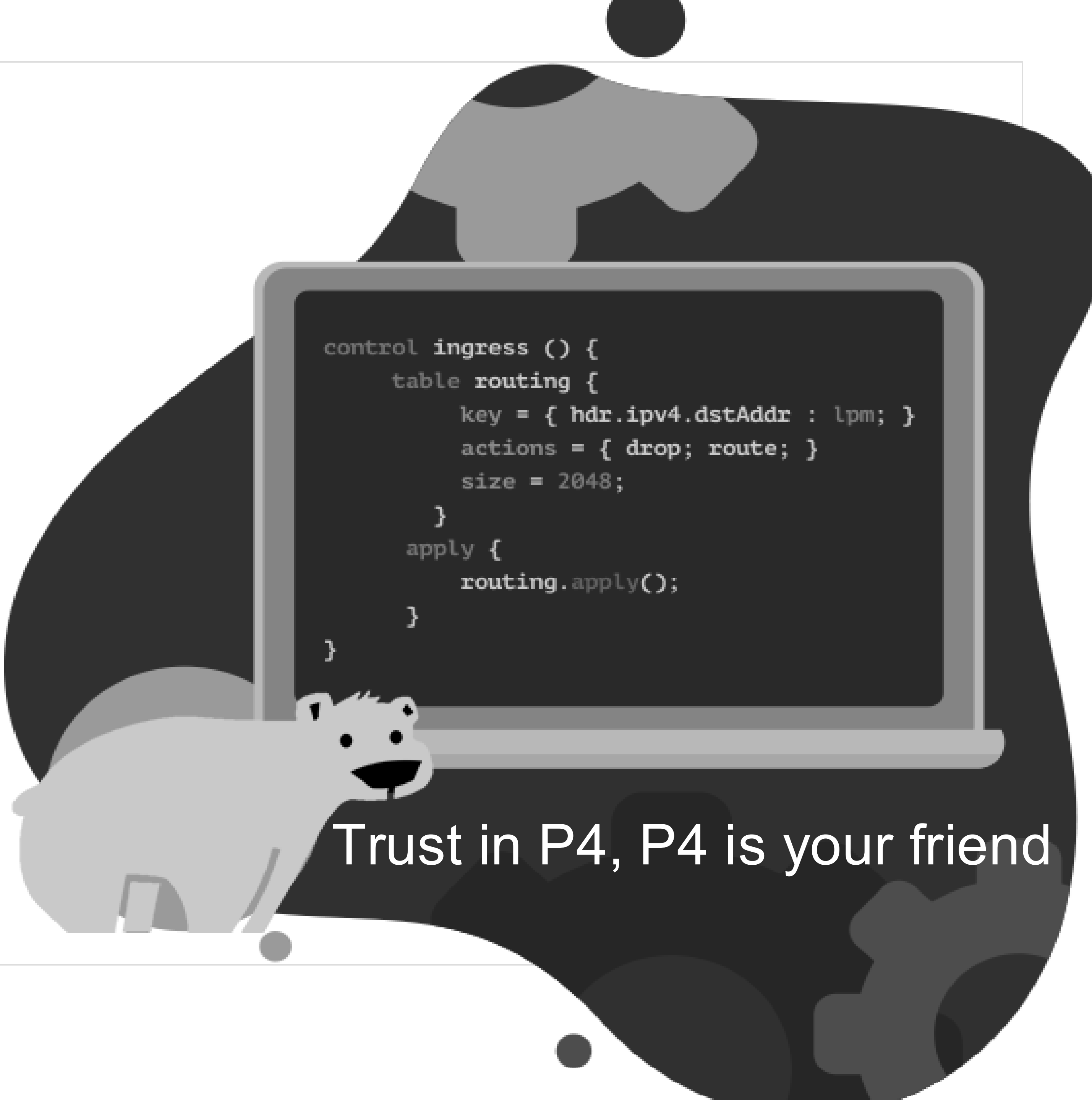
- Special thanks to ***Andy Fingerhut*** for hosting the session and guiding the discussion.

Presented by: Reza Fallahi Kapourchali

Thank You!

I hope you found the conversation engaging and walked away with something useful.

Reza Fallahi Kapourchali



```
control ingress () {
    table routing {
        key = { hdr.ipv4.dstAddr : lpm; }
        actions = { drop; route; }
        size = 2048;
      }
    apply {
        routing.apply();
    }
}
```

Trust in P4, P4 is your friend