Knowledge-Defined Networking using In-band Network Telemetry

Jonghwan Hyun, James Won-Ki Hong

Department of Computer Science Engineering, POSTECH, Korea

{noraki, jwkhong}@postech.ac.kr

Sept. 27, 2017

Contents

- Introduction
- Background
- Proposed Architecture
- Use cases
- Conclusion

Introduction

❖ Complexity of Network Management

- Network traffic volume is growing [CISCO VNI 2017]
  - Will reach 3.3 ZB by 2021
  - Will increase nearly threefold over the next 5 years
- The number of connected devices is growing [CISCO VNI 2017]
  - 3.5 networked devices per capita by 2021
  - Was 2.3 in 2016
- Network changes dynamically
  - e.g., hosts moving around, contents dynamically moving in the network, dynamic bandwidth allocation caused by SLA and QoS

- Make it hard to manage network in real-time by human operators
- The need for the closed-loop network management solution is arising

Introduction

❖ Self-Driving Networks

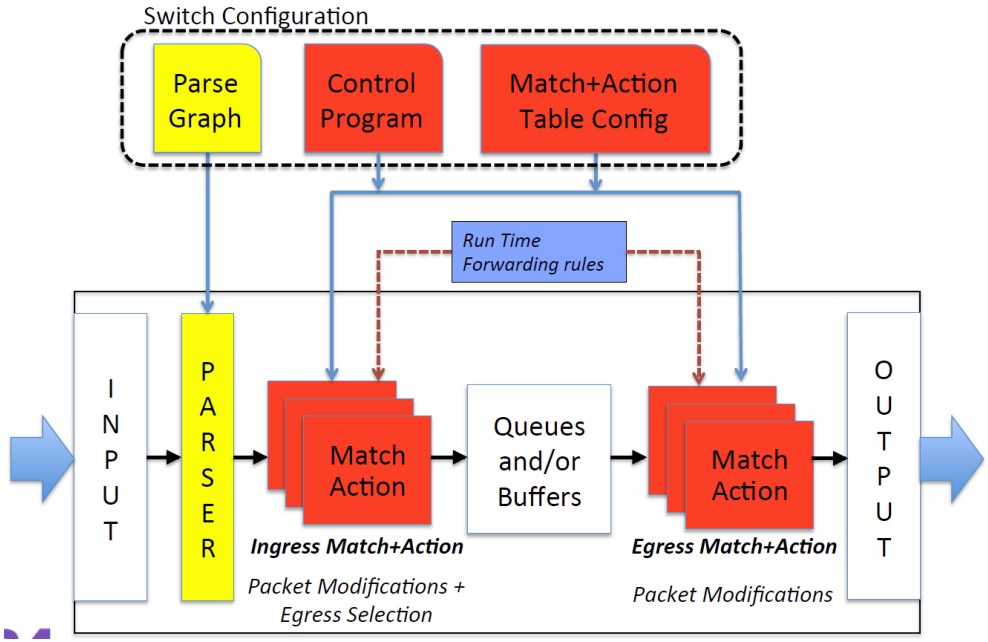
- Autonomous network that is predictive and adaptive to its environment
- Advantages
  - *Reduce operation complexity* by simplifying and abstracting networks
  - Enable customers to *deploy new network services faster*
  - *Improve capacity utilization* and *network resiliency*
- Technologies required
  - Telemetry, Automation, Intents, Decision making, Local and global views
- Building blocks
  - Knowledge-defined Networking (KDN)
  - Software-defined Networking (SDN)
  - In-band Network Telemetry (INT)

❖ In this paper

- Propose an architecture for collecting network telemetry and combining KDN to work with SDN and P4 INT
- Suggest several use-cases for the proposed architecture

Background

- **P4 (Programming Protocol-Independent Packet Processors)**
  - Domain specific language for programmable data plane
  - Based on a Match+Action forwarding model
  - Automatically generate APIs to manage the packet processing tables



Source: The P4 Language Specification Version 1.0.3, The P4 Language Consortium, Nov. 2016

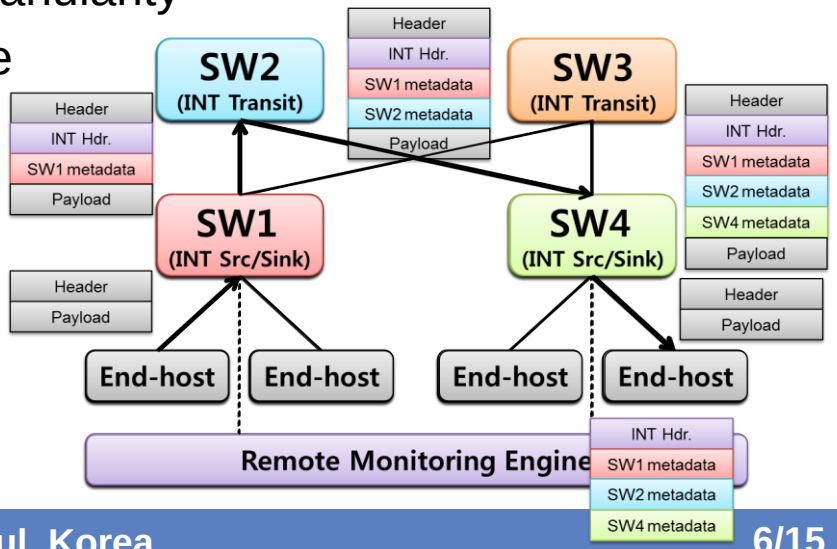
Background

❖ In-band Network Telemetry (INT)

- Framework for **collecting and reporting network state by the data plane**, without requiring intervention or work by the control plane
- Examples of network state (metadata)
  - Switch ID, timestamp, Ingress/Egress Port ID
  - Link Utilization, Hop Latency
  - Egress Queue Occupancy
  - Egress Queue Congestion Status
- Advantages
  - Provide metadata at packet-level granularity
  - Collect data in real-time, polling-free

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																													
Ver	Rep	C	E	R	R	R	R	R	**Ins Cnt**							**Max Hop Cnt**								**Total Hop Cnt**							
Instruction Bitmap																Reserved													
0	**INT Metadata Stack (Varying number of fixed-size values)**																												
0	…																												
1	**Last INT Metadata**																												

<INT Header Format>



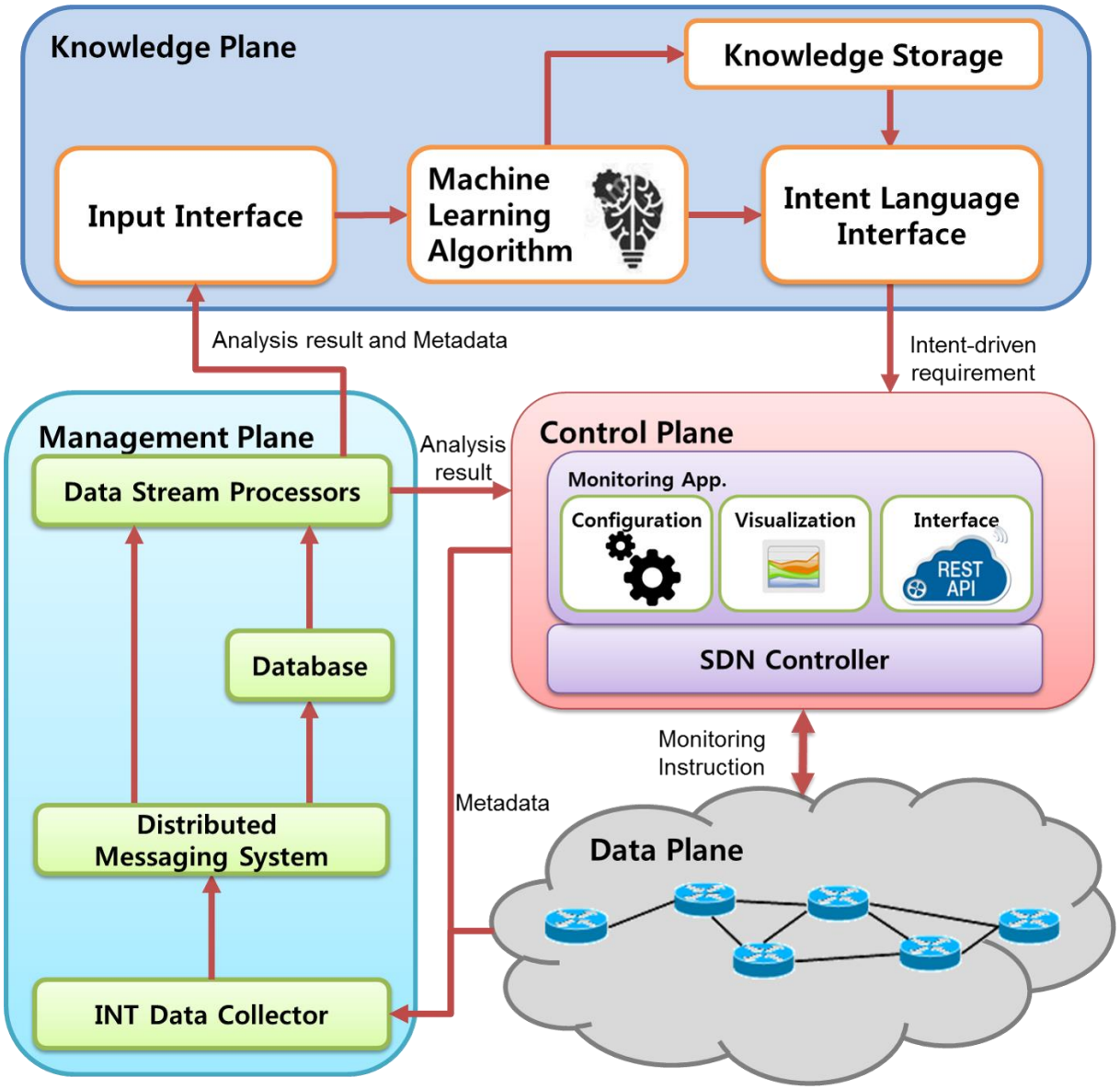
Background

❖ Knowledge-Defined Networking (KDN)

- Originally defined at the early stage of the Internet [Clark, ACM '03]
  - Suggested a concept of Knowledge Plane
  - Adopted Artificial Intelligence (AI) and cognitive system to *build a model for the network*
  - Limitation
    - Each network node only has a local view
    - Computing power is limited

    → Complex to learn on nodes
- Re-defined as the combination of knowledge plane, network telemetry, SDN and network analytics [Mestres, arXiv'16]
- When KDN adopts the programmable data plane, richer set of actions can be provided

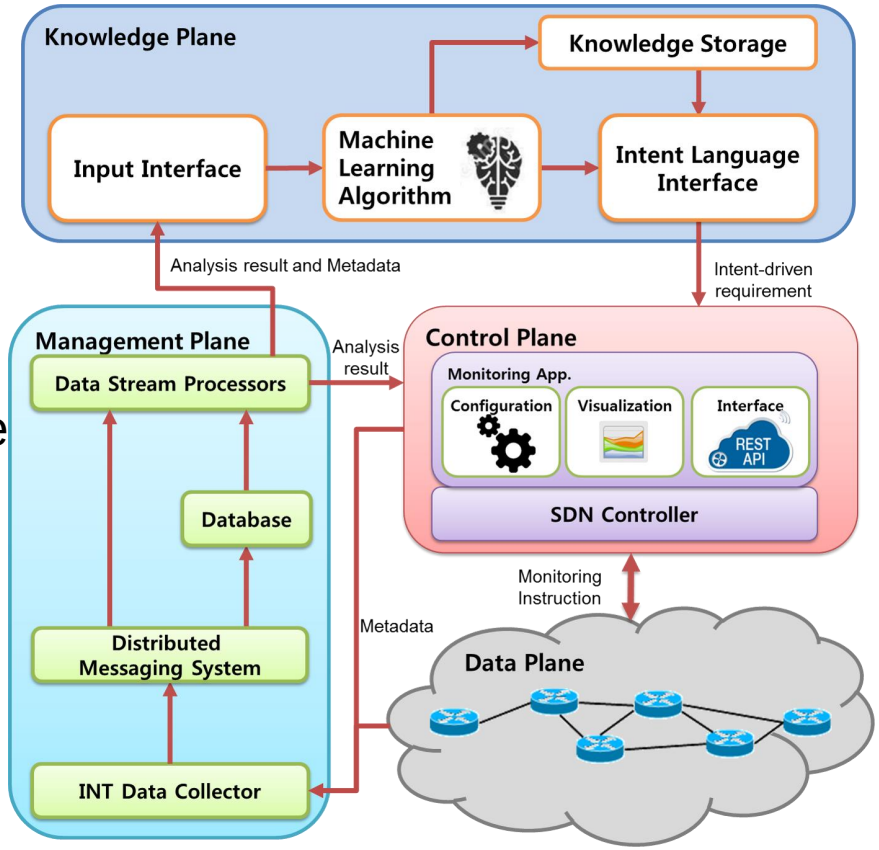
Proposed Architecture



Proposed Architecture

❖ Control Plane

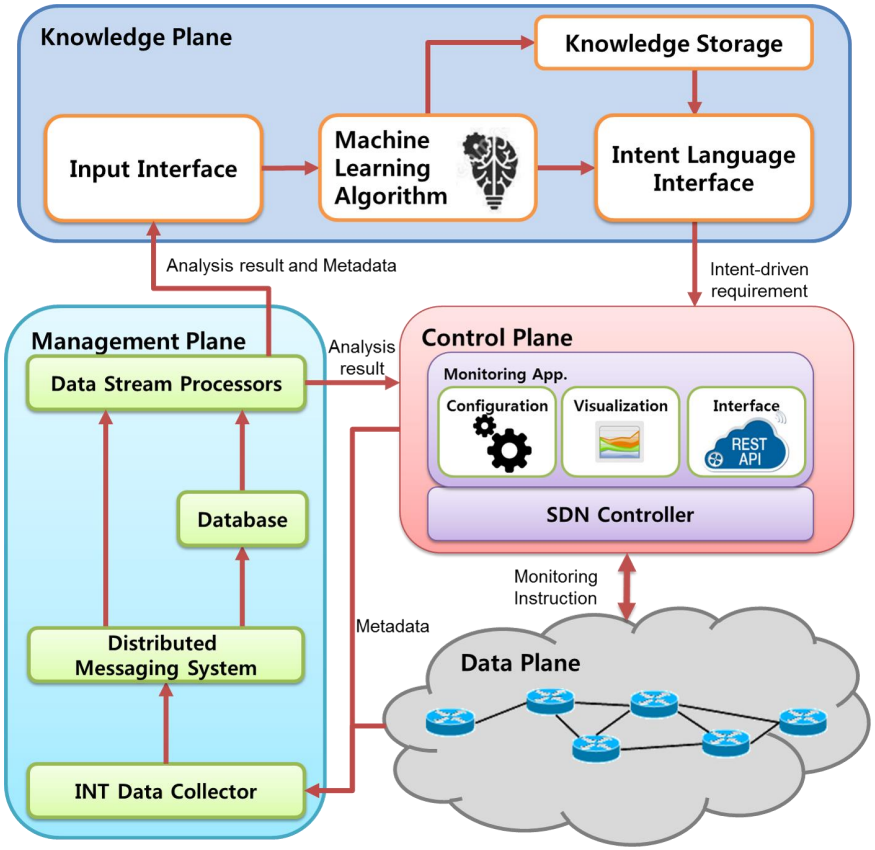
- SDN controller works as a control plane
  - Deploy INT functionality to programmable switches
  - Control the condition of network monitoring using INT
  - Convert the intent-driven requirements from the knowledge plane into the specific network policy



Proposed Architecture

❖ Data Plane

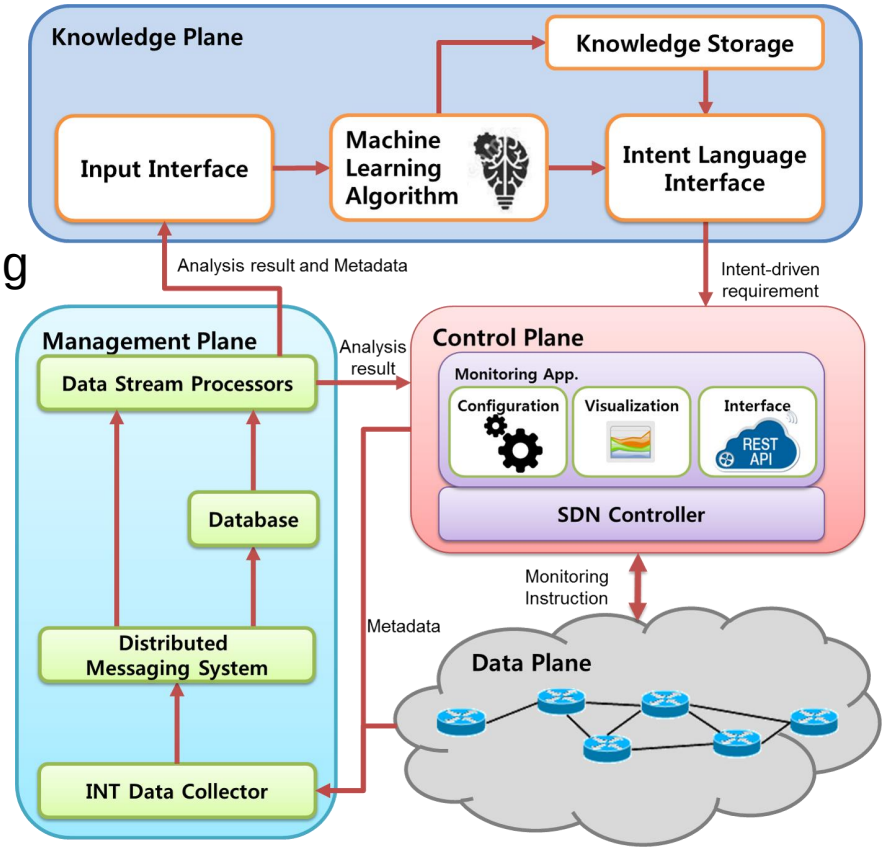
- INT metadata is generated, extracted and transmitted to the management plane
- INT operations
  - Source switch: insert INT header for target flows
  - Transit switch: identify INT header and insert metadata
  - Sink switch: Extract INT header and metadata, sending to the management plane



Proposed Architecture

❖ Management Plane

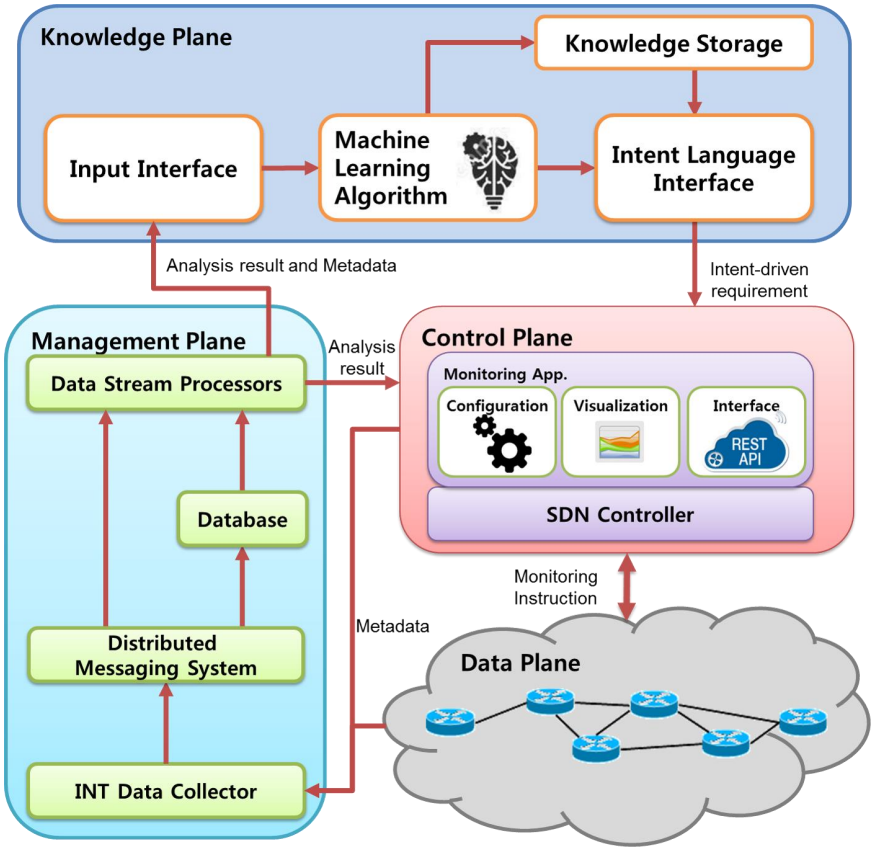
- Collect, store and aggregate INT metadata
- INT data collector
  - Collect the metadata
  - Send to the distributed messaging system
- Distributed messaging system
  - Distribute the metadata to one of the data stream processors
- Data stream processor
  - Provide basic analysis result and statistics per flow, per switch
  - Send the result to the control plane and knowledge plane



Proposed Architecture

❖ Management Plane

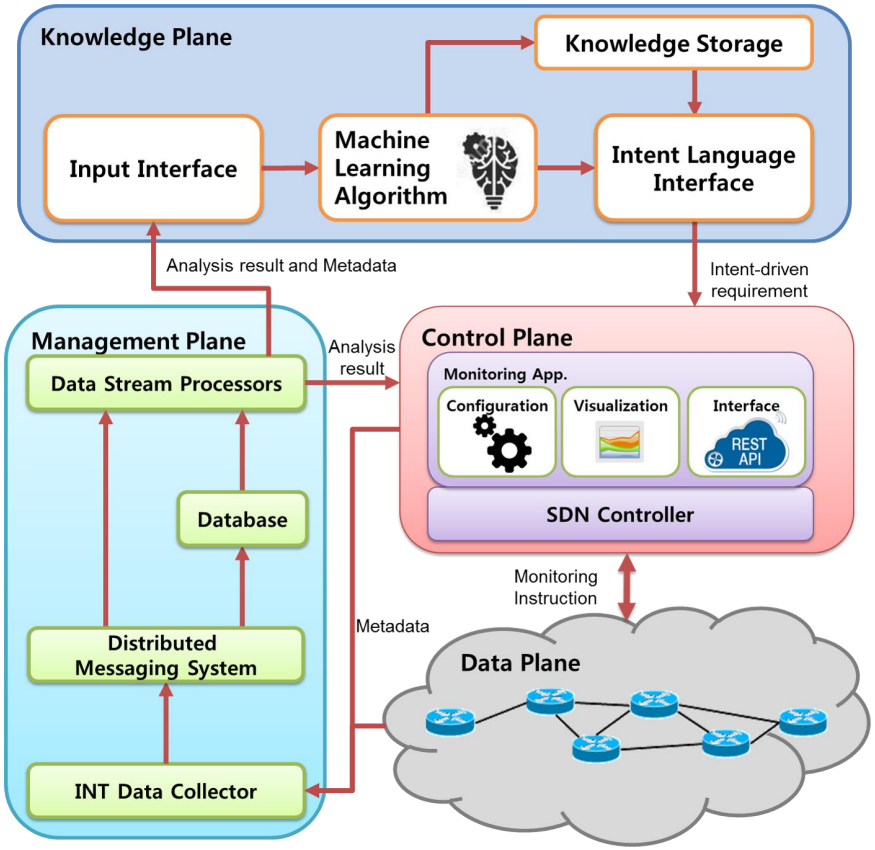
- Output
  - Contain the real path of the flow traversed in the network, latency for each hop, queue occupancy and congestion status for each switch
  - To the control plane: event which requires immediate action
    - e.g., link failure, black-hole or loop detection
  - To the knowledge plane: actions requiring knowledge
    - e.g., resource planning and optimization, performance management, verification



Proposed Architecture

❖ Knowledge Plane

- Take the data gathered in the management plane
- Feed the data to the ML algorithm
- Populate knowledge
- Convert the knowledge into the form of intent language
  - To describe the requirement to change the network configuration
- Need to be separated from control plane because of the compute-intensive ML algorithms



Use Cases

❖ Traffic Engineering (TE)

- Offload traffic information collection and analysis on the management plane, reducing the overhead to the SDN controller
- Collect fine-grained traffic information, deeper into the packet level
- Refine and cluster traffic information, reducing the number of input values of the TE algorithm

❖ Network Anomaly Detection

- Build network model for the normal condition
- Raise an alarm when the unexpected traffic pattern is detected
  - DDoS and port scan attacks can be easily detected in this way
- Involve knowledge plane to record and analyze the traffic generated in each host
  - To further detect an abnormal traffic pattern

Conclusion

❖ Summary

- Discuss the concept of self-driving network and its building blocks, P4 INT, SDN and KDN
  - P4 INT: collect packet-level network telemetry
  - KDN: bring the intelligence to the network management, using the telemetry data
  - SDN: manage and control the network according to the decision made by the knowledge plane
- Suggest two use cases: traffic engineering and anomaly detection

❖ Future work

- Implement the proposed architecture and proof of concepts for the use cases.


THANK
YOU
THANK