Traffic measurement, traffic classification and network modeling in

Software-Defined Networks

ADVANCED BROADBAND COMMUNICATIONS CENTER (CCABA)

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

José Suárez-Varela Maciá

Email: jsuarezv@ac.upc.edu

Departament d'Arquitectura de Computadors

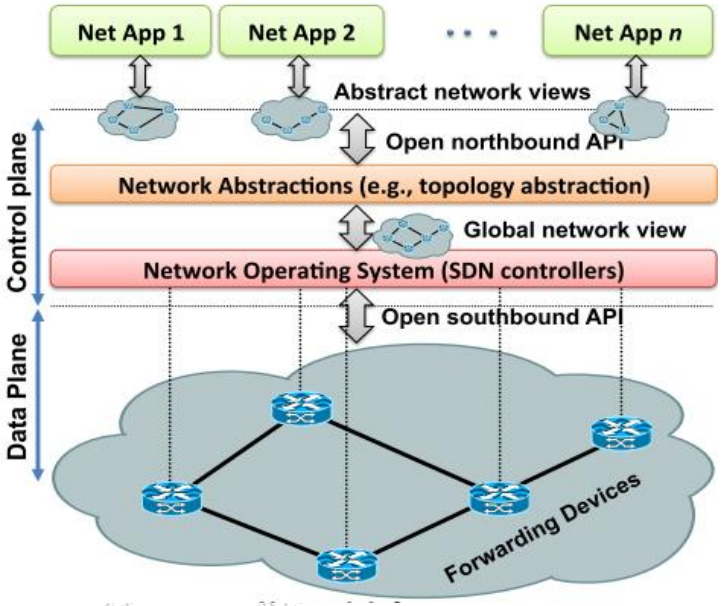
Table of contents

- Reviewing SDN
- Traffic measurement
- Traffic classification
- Deep Learning for network modeling

Table of contents

Reviewing SDN

- Physical separation of the network control plane and the data (forwarding) plane
- Open (standard) interfaces for the communication between planes:
  - Southbound API
    (e.g., OpenFlow, P4)
  - Northbound API
    (e.g., OpenDaylight controller)
- Abstractions in different levels:
  - Target and vendor-independent configuration of the data plane
  - Management from a global view of the network state (traffic measurements, link failures…)



Source: Kreutz, D et al. "Software-defined networking: A comprehensive survey", *Proceedings of the IEEE.*

Reviewing SDN

Motivation

- Large support by academia, industry (Cisco, Google, HP, NEC, Juniper…) and standardization organizations (ONF, IETF)
- Real-world SDN-based network deployments (e.g., Google's B4 [4], VMware [5])
- Programmable (software) networks → **Flexibility on network management** (flow-based managment)

Reviewing SDN

Motivation

- Global view of the state of the network → **Fine-grained management and network function virtualization (NFV)**

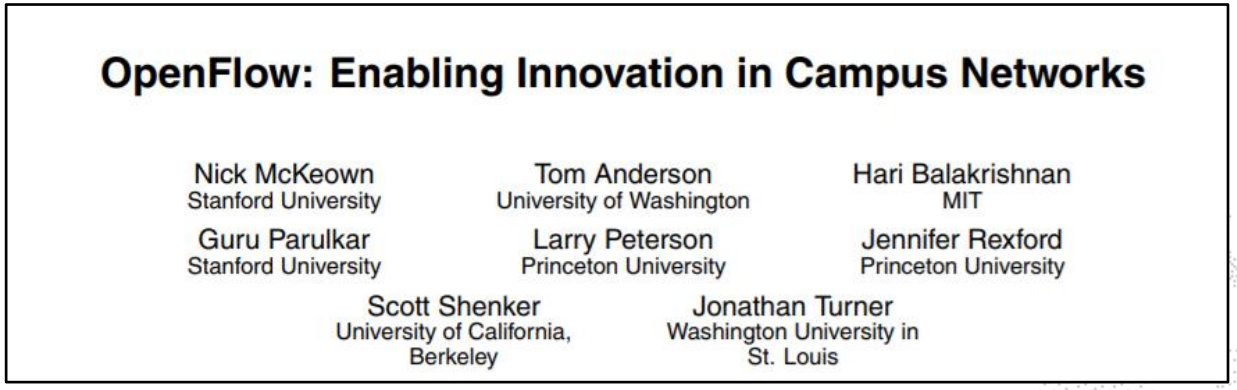
- Towards generic forwarding devices → **Flexibility on the underlying infrastructure**

There is still a lot of work to be done!

Reviewing SDN

OpenFlow protocol

- First proposal for SDN-based networks in 2008



OpenFlow: Enabling Innovation in Campus Networks

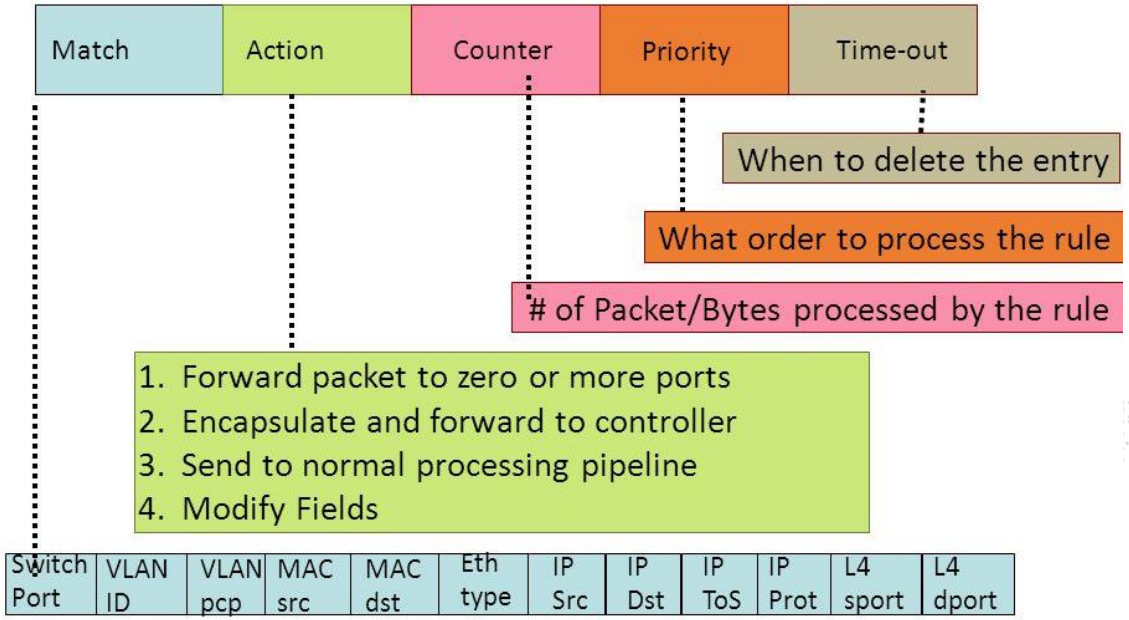
Nick McKeown, Stanford University
Tom Anderson, University of Washington
Hari Balakrishnan, MIT
Guru Parulkar, Stanford University
Larry Peterson, Princeton University
Jennifer Rexford, Princeton University
Scott Shenker, University of California, Berkeley
Jonathan Turner, Washington University in St. Louis

- Milestone in the networking field
- Standard interface to communicate the control and data planes (target-independent protocol)
- Abstraction of network devices as flow tables to define the forwarding behavior

Reviewing SDN

OpenFlow protocol

- Flow tables populated with flow entries

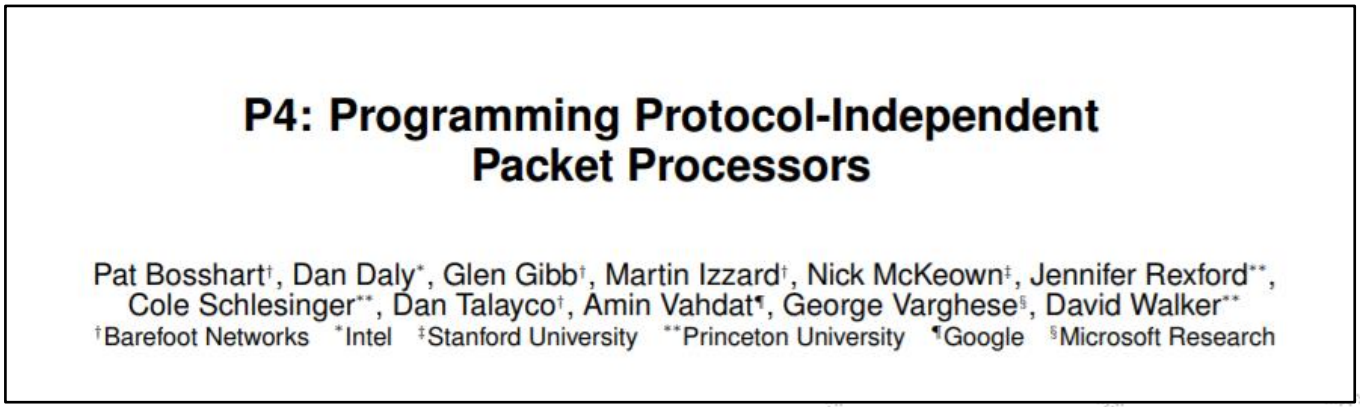


Source: Theophilus Benson, “CPS 590: Software-Defined Networking”, https://slideplayer.com/slide/5722212/

- Proactive and reactive installation

P4: A new proposal for SDN

- **P4** (Programming Protocol-Independent Packet Processors)



P4: Programming Protocol-Independent Packet Processors

Pat Bosshart[†], Dan Daly[*], Glen Gibb[†], Martin Izzard[†], Nick McKeown[‡], Jennifer Rexford[**], Cole Schlesinger[**], Dan Talayco[†], Amin Vahdat[¶], George Varghese[§], David Walker[**]

[†]Barefoot Networks [*]Intel [‡]Stanford University [**]Princeton University [¶]Google [§]Microsoft Research

- Proposed in 2014 by some authors of OpenFlow
- OpenFlow:
  - Protocol-dependent
    OpenFlow v1.0 → 12 match fields
    OpenFlow v1.4 → 41 match fields!!
  - Scalability problems

P4: A new proposal for SDN

- Leverage recent advances in chip design to build custom ASICs (Application-Specific Integrated Circuit) for networking
- Custom ASICs can process traffic at terabit speeds (1000 Gbps)
- Programmable networks (run small programs in switches)
  e.g., monitoring sketches
- More flexibility to program the behavior of switches
  - Protocol-independent (definition of new headers)
  - Processing pipeline to perform complex actions

Table of contents

Traffic measurement

- Introduction
  - Traffic volume estimation
  - Performance evaluation (e.g. end-to-end latency, link utilization)
  - Traffic anomalies → Detect security threats and vulnerabilities
  - Enables better network operation and management:
    - Troubleshooting
    - Traffic engineering (routing)
    - Efficient VNF allocation
    - Network planning
    - …

Traffic measurement

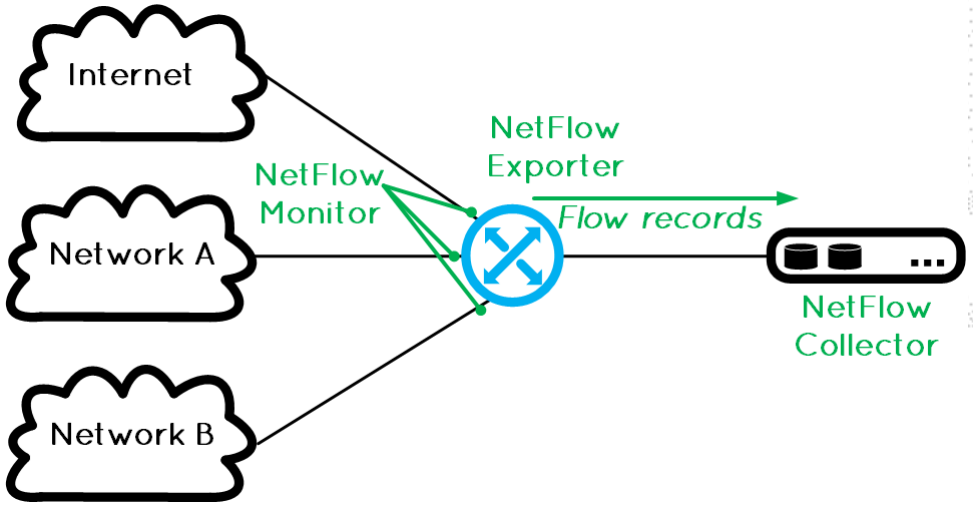
- Passive traffic measurement techniques in traditional networks:
  - SNMP (Simple Network Management Protocol):
    - Simple traffic measurements (e.g., traffic per interface)
    - Memory, CPU load in network devices
    - Traps (alarms triggered by some thresholds)
  - Network Probes:
    - Detailed (custom) traffic measurements
    - High cost of deployment
  - Flow-level traffic aggregation:
    - Definition of traffic flows (e.g., 5-tuple → {IPs, ports, IP proto})
    - Detailed measurements per flow
    - Enables to identify hosts and applications consuming the bandwidth

Traffic measurement

- Traffic measurement in traditional networks:
  - Flow-level traffic aggregation (cont.)

NetFlow/IPFIX protocol

- Flow-level measurements maintained in network devices
- Flow reports are sent to an external collector (low traffic overhead)
- Use of timeouts (hard or idle)
- Scalable → Traffic sampling (packet or flow)
- Standard widely deployed!!



Traffic measurement in SDN

How to perform traffic measurement in Software-Defined Networks?

Traffic measurement in SDN

OpenFlow

- Dominant protocol for the southbound interface in SDN:
  - **Widely implemented** in current off-the-shelf SDN switches
  - Mechanisms to maintain measurements in switches:
    - Flow entries with counters in the switches
  - Interface to retrieve measurements
  - Scalability issues:
    - The SDN controller is prone to become a bottleneck in the network
    - The memory in switches is limited and small

Traffic measurement in SDN

Retrieve traffic statistics with OpenFlow

Pull-based statistics retrieval

- The SDN controller requests traffic statistics to the switch

Field Name	Bits
type	16
flags	16
pad	32
table_id	8
pad	24
out_port	32
out_group	32
pad	32
cookie	64
cookie_mask	64
match	-

Multipart request message

Field Name	Bits
type	16
flags	16
length	16
table_id	8
pad	8
duration_sec	32
duration_nsec	32
priority	16
idle_timeout	16
hard_timeout	16
pad	48
cookie	64
packet_count	64
byte_count	64
match	-
action[]	-

Multipart reply message

Traffic measurement in SDN

- Retrieve traffic statistics with OpenFlow

Push-based statistics retrieval

- Asynchronous retrieval when flow entries expire

Name	Bits
match	32
cookie	64
command	16
idle_timeout	8
hard_timeout	8
priority	8
buffer_id	32
out_port	16
flags	16
action[]	32

Flag	
	SendFlowRem
	CheckOverlap
	Emerg

Flow_mod_add message

Name	Bits
match	32
cookie	32
priority	16
reason	8
pad	8
duration_sec	32
duration_nsec	32
idle_timeout	16
pad	8
pad	8
packet_count	64
byte_count	64

Field	Name
Reason	IdleTimeOut
	HardTimeout
	Delete

Flow_removed message

Use case: traffic measurement in SDN

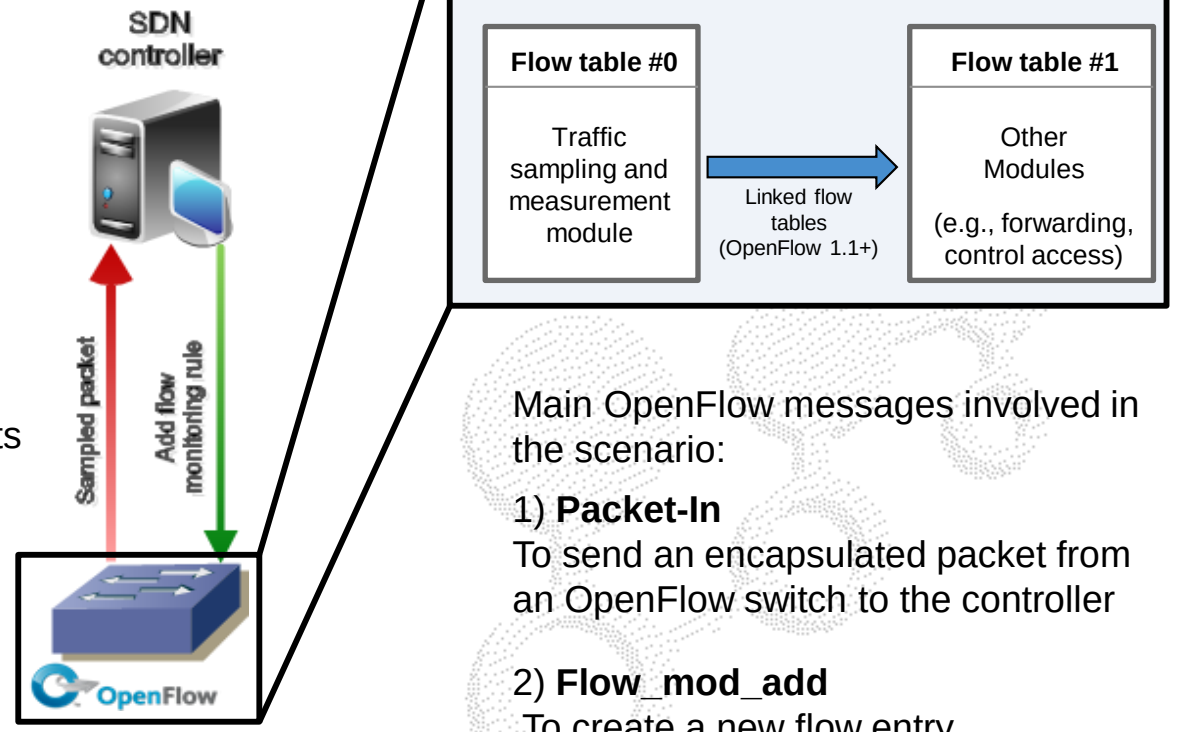
- Collect flow-level measurement reports as those of NetFlow/IPFIX in OpenFlow-based networks

3 stages

1) Add sampling rules proactively

2) Add flow (5-tuple) monitoring rules reactively with timeouts

3) Asynchronous (push-based) retrieval of flow measurement reports



Main OpenFlow messages involved in the scenario:

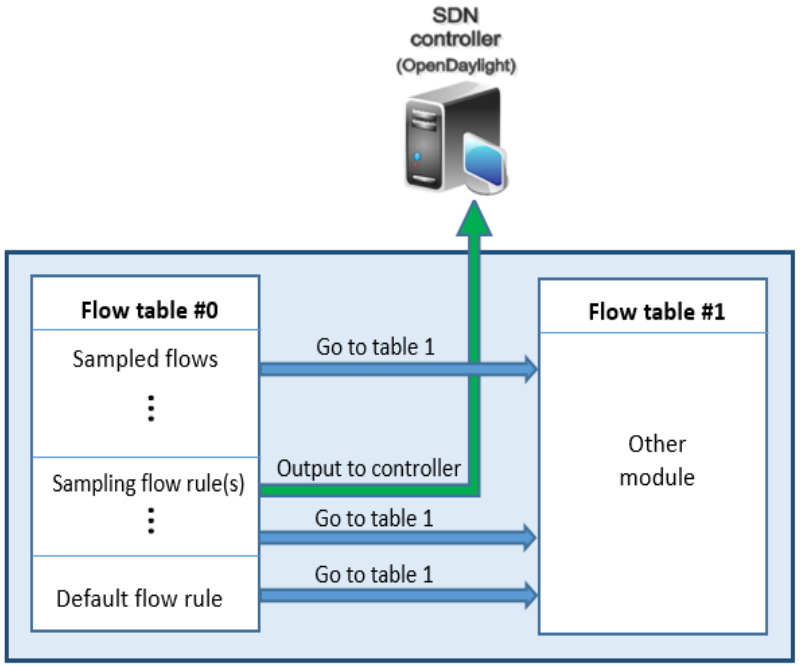
1) **Packet-In**
To send an encapsulated packet from an OpenFlow switch to the controller

2) **Flow_mod_add**
To create a new flow entry

Use case: traffic measurement in SDN

Scheme of OpenFlow flow tables

- 3 Different blocks of flow entries ordered by their priority field:
  1. Sampled flow records
  2. Sampling flow rules
  3. Default rule



- All the packets are eventually directed to a next table to be processed by other network modules
- 2 alternative sampling methods based on the OpenFlow features available in switches

Use case: traffic measurement in SDN

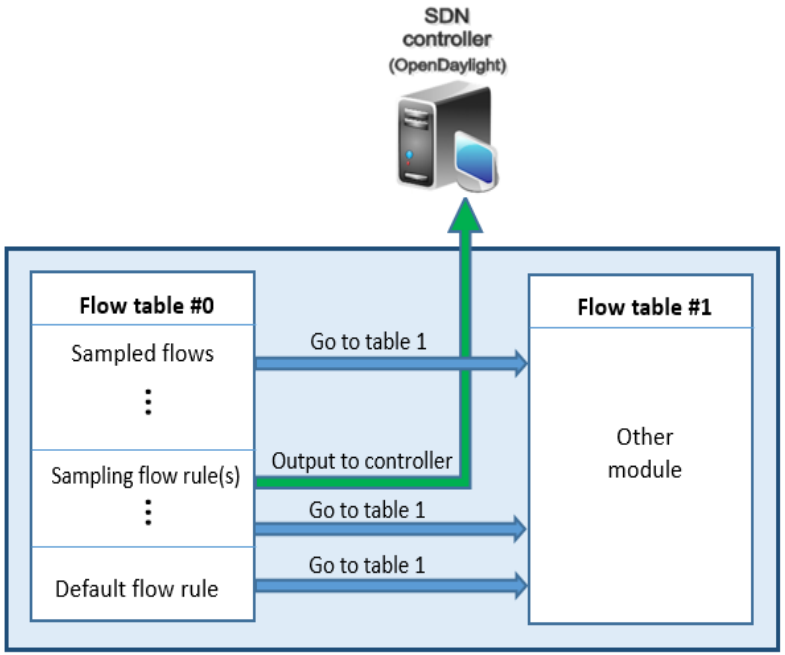
Traffic sampling methods

1) Sampling based on IP suffixes

- Check the last 'n' bits in the IP addresses to apply a sampling rate
- Single or pair of IP suffixes

$$sampling\ rate = \frac{1}{2^m \cdot 2^n}$$

- IP suffixes periodically changed



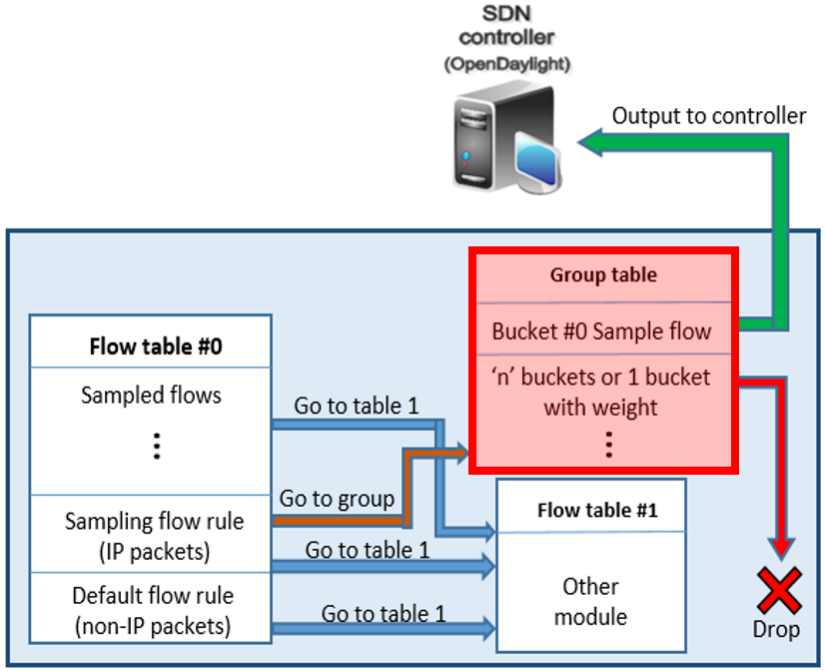
- Requirements:
  - Support of arbitrary masks to check the last bits of an IP address

Use case: traffic measurement in SDN

Traffic sampling methods

2) Hash-based flow sampling

- Compute a hash function on the 5-tuple fields of the packet header
- OpenFlow group tables (originally designed for load balancing)
- Weights for different buckets to apply a specific sampling rate



- Requirements:
  - Support for group tables with buckets of type select
  - Accurate algorithm to properly balance the load among buckets (out of the scope of the OpenFlow specification)

Use case: traffic measurement in SDN

Features of the system:

1) Scalable

- Measurements are maintained in switches and asynchronously collected (push-based retrieval)
- Traffic sampling methods to reduce:
  - Processing overhead in SDN controllers
  - Number of flow entries maintained in switches

2) Transparent operation

- Decouple measurement module from other network tasks (e.g., forwarding)

Alternative measurements proposals for SDN

- OpenSketch architecture for virtual (software) switches (2013)

Software Defined Traffic Measurement with OpenSketch

Minlan Yu[†] *Lavanya Jose*[*] *Rui Miao*[†]
[†] *University of Southern California* [*] *Princeton University*

- In-band Telemetry based on the P4 specification (2015)
  - Encoding information within packet headers (in-band):
    - Timestamps → per-hop and end-to-end delay measurements
    - Hops → Check packet routes (troubleshooting)

In-band Network Telemetry via Programmable Dataplanes

Changhoon Kim[*], Anirudh Sivaraman[**], Naga Katta[***], Antonin Bas[*], Advait Dixit[*], Lawrence J Wobker[*]
[*]Barefoot Networks, [**]Massachusetts Institute of Technology, [***]Princeton University
{chang, antonin, adixit, ljw}@barefootnetworks.com, anirudh@csail.mit.edu, nkatta@cs.princeton.edu

Table of contents

Traffic classification

Introduction

- Identification of applications generating traffic flows
- Protocol or application-level classification (e.g., for web-based traffic)
- Combination of flow-level measurements with classification labels
- Applications:
  - Security (e.g., malicious applications)
  - QoS provisioning
  - Accounting, billing
  - …

Traffic classification

- **Classification techniques in traditional networks:**
  - Port-based classification:
    - List of well-known ports (e.g., port 80 = HTTP, port 22 = SSH)
    - Very low processing cost
    - Classification at the protocol-level
    - Easy to bypass by applications (e.g., P2P apps)
  - Statistical behavior models:
    - Find patterns on communications
    - Usually host-based patterns:
      e.g., hybrid P2P networks (Skype)
      A host connects to a general server and then to a peer
    - Accurate for specific applications (e.g., P2P vs non-P2P traffic)
    - Not generic models

Traffic classification

- **Classification techniques in traditional networks:**
  - Deep Packet Inspection (DPI):
    - Inspection of the packet payload
    - Signature-based (apps) or protocol-based field inspection
    - Classification at the application-level
    - Pros:
      - High accuracy (very few false positives)
      - It is not easy to bypass
    - Cons:
      - High processing cost (possibility of inspecting first few bytes)
      - Necessity of creating continuously signatures for new applications
      - Privacy concerns

Traffic classification

- **Classification techniques in traditional networks:**
  - Machine Learning (**ML**)
    - Use of ML-based models (decision trees, random forests, neural networks…)
    - Trained with application labels obtained by accurate classifiers (e.g., DPI tools)
    - Very low computational cost compared to DPI
    - Typical features:
      - IP proto and ports (TCP/UDP)
      - Size (# bytes) and duration of flows
      - Size (# bytes) of first few packets
      - Inter-packet arrival times
      - …

Traffic classification

- **Classification techniques in traditional networks:**
  - Machine Learning (**ML**)
    - Feature selection problem (measurements available)
    - The accuracy depends on the features selected and the training dataset
    - Necessity of re-training with new applications
    - Less privacy issues (the packet payload is not inspected)

Traffic classification in SDN

Motivation

- OpenFlow-based environments
- Classification leveraging some particularities of SDN:
  - Flow-based redirection of traffic (e.g., DNS traffic) to controllers
  - First few packets of flows to perform DPI in controllers
  - Flow measurements records (machine learning using L4 features)
- Combination of DPI and ML-based techniques
- Find the tradeoff between accuracy and computational cost

Use case: Traffic classification in SDN

Use case description

– Classification of flows sampled at two different levels:

1) All the flows are classified by application protocol

e.g., RTP, SSH, HTTP…

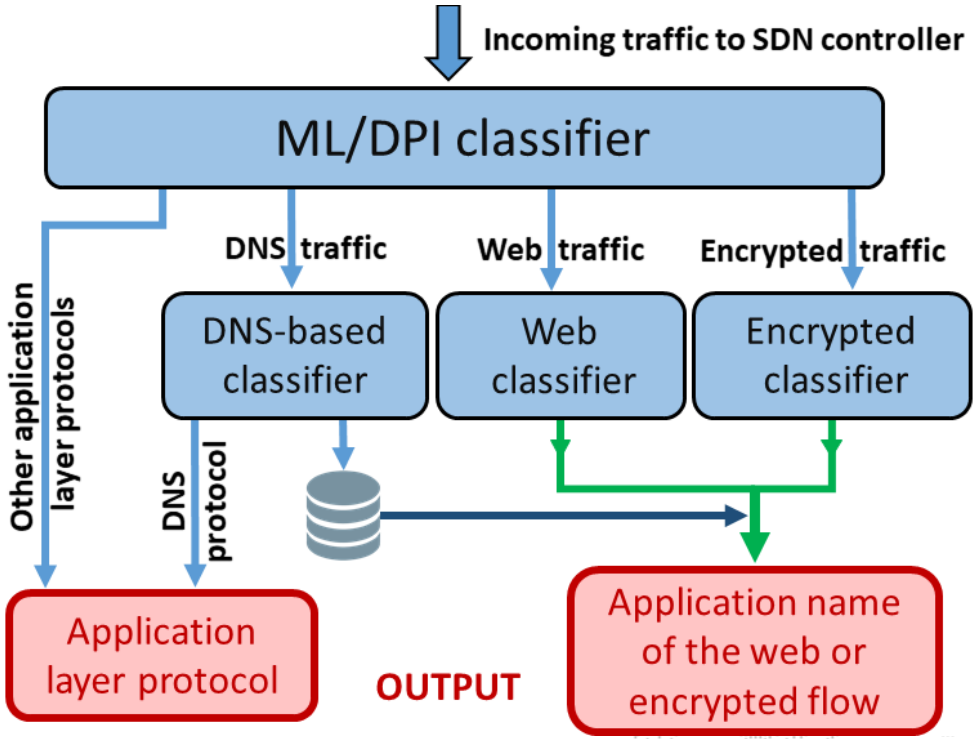
2) Web-based traffic is classified by applications (domain names)

e.g., netflix.com, facebook.com, dropbox.com

- Use of information from:
  - HTTP headers (Host field)
  - HTTPS → SSL/TLS certificates
  - DNS traffic

Use case: Traffic classification in SDN

- Architecture



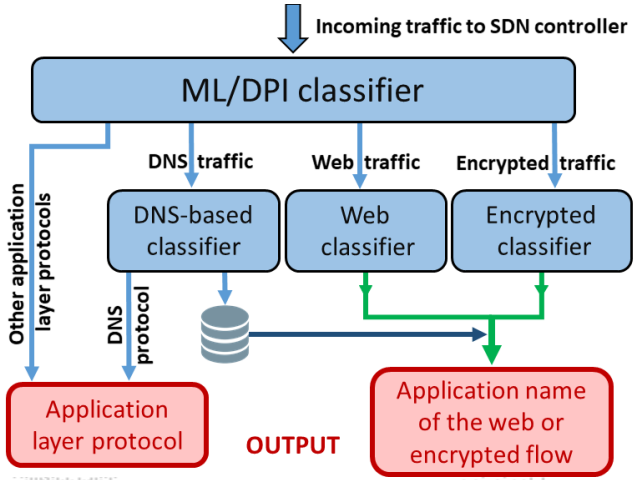
Use case: Traffic classification in SDN

Architecture

Modules:

1) DPI/ML module:

- Classifies all the flows except for HTTP and HTTPS traffic
- Tradeoff between accuracy and cost (DPI or ML)



2) DNS module:

- Information from client DNS requests (DNS records)
- Associate domain names to IP addresses

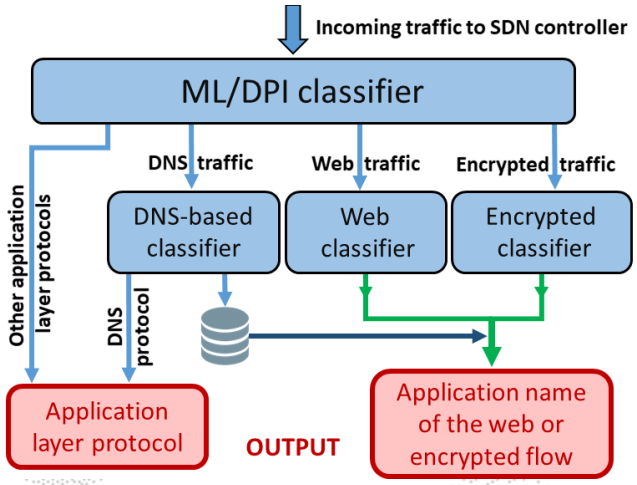
Use case: Traffic classification in SDN

Architecture

- **Modules:**

  3) Web module (DPI):

  - Field ‘host’ in HTTP headers



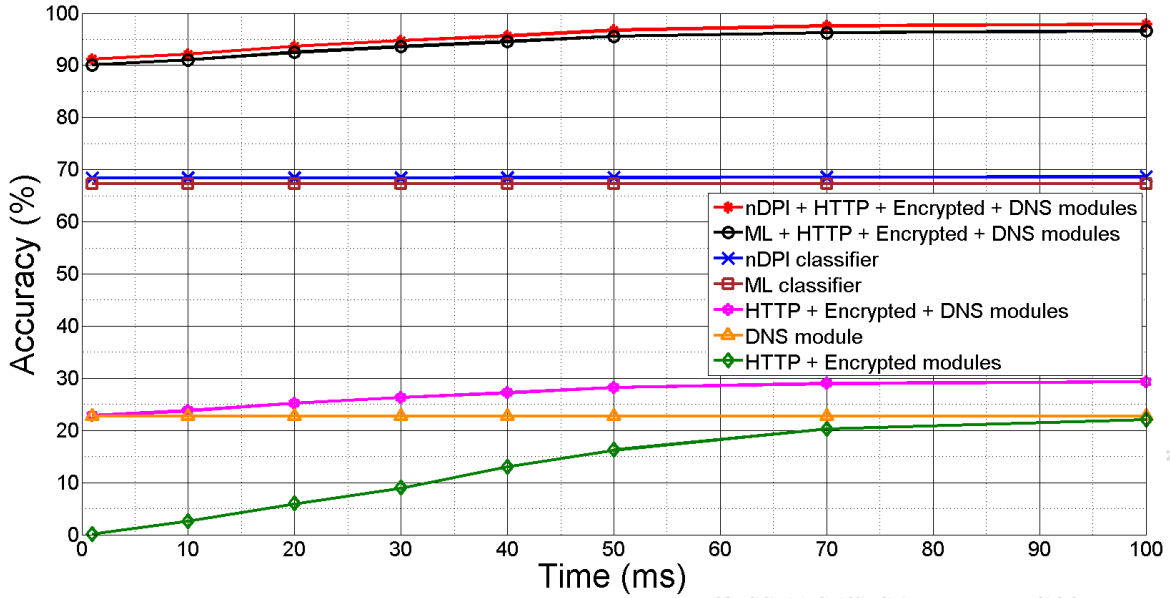
  4) Encrypted traffic module (DPI):

  - Field ‘Server Name Indication’ (SNI) in SSL/TLS certificates

- This information is usually present in the first few packets of flows (HTTP get or SSL initial handshake)

Use case: Traffic classification in SDN

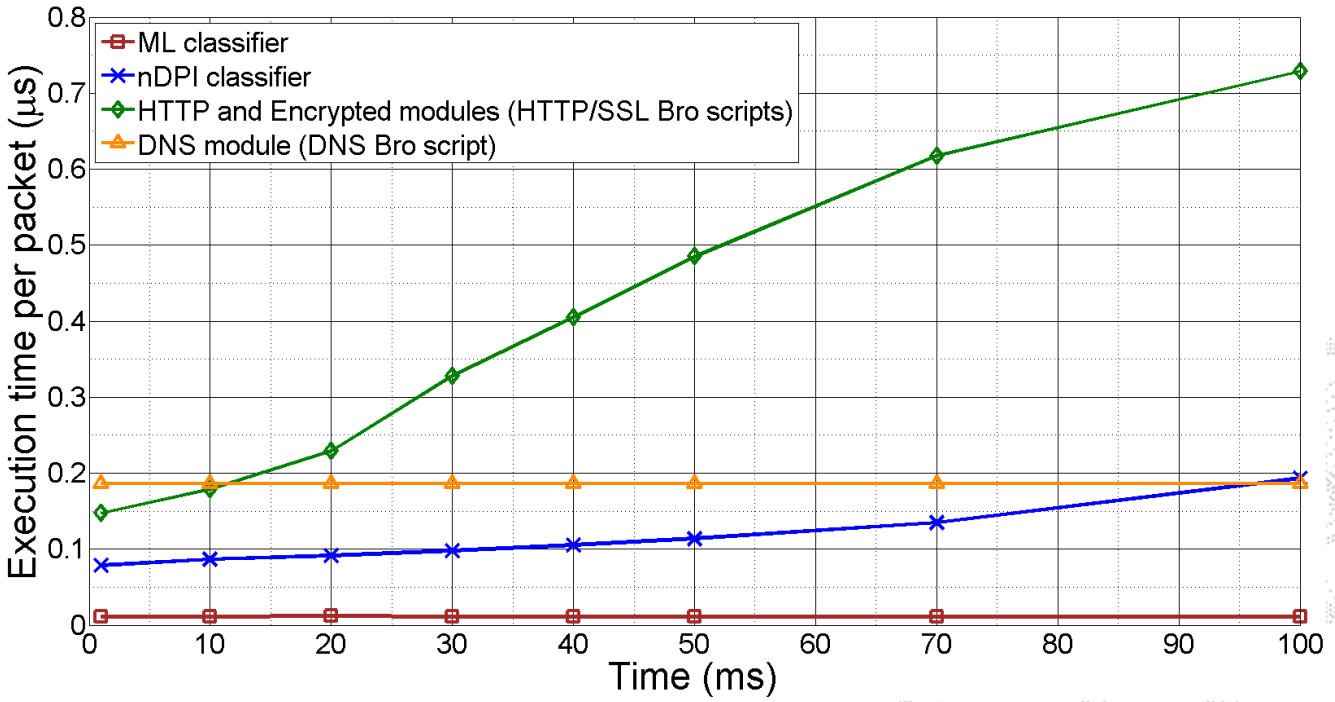
- **Evaluation accuracy**



  - ML vs DPI (protocol-level classification)
  - DNS, HTTP and encrypted traffic modules to identify applications on web-based traffic

Use case: Traffic classification in SDN

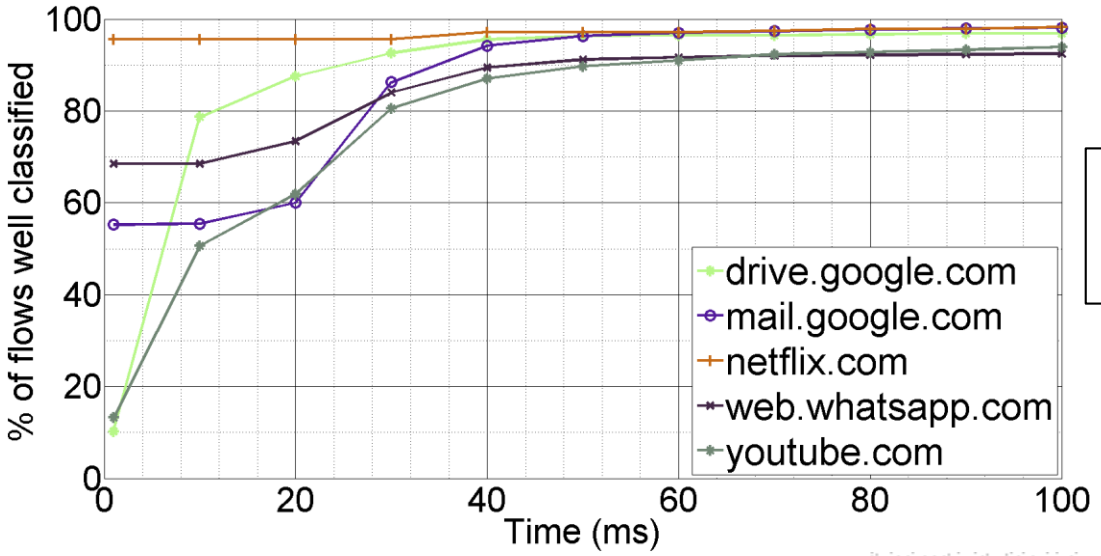
- **Evaluation overhead**



  - In real-world traffic, the DNS traffic involves <0.1% of the total bytes (use of DNS caches)

Use case: Traffic classification in SDN

- Evaluating specific apps in encrypted traffic



How RTT affects on the classification accuracy of different apps

Application name	Domain name
Google Drive	drive.google.com
Google Gmail	mail.google.com
Netflix	netflix.com
Whatsapp web	web.whatsapp.com
YouTube	youtube.com

Table of contents

- Reviewing SDN
- Traffic measurement
- Traffic classification
- Deep Learning for network modeling

Deep Learning for Network Modeling

- Traditionally networks have been **distributed** systems
  - Partial view and control
- Beyond programmability, SDN provides **centralization**:
  - **Full control** over the network
- **Network Analytics (NA)** and **Telemetry** provides **full view** of the network
- **SDN** and **NA** are **enabling technologies** for Machine Learning techniques applied to networks

Evolution in other fields

Hardware → Software → Artificial Intelligence

?

Deep Learning (DL) applied to networks

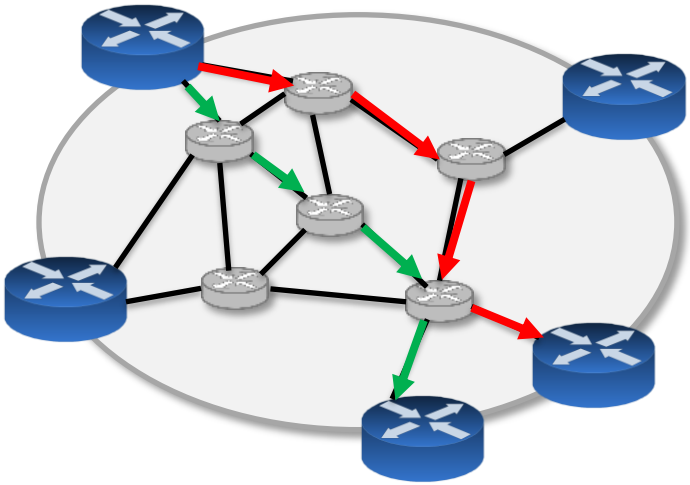
- Apply DL techniques to Networking:
  - Control (fast dynamics)
    - *e.g., routing, resource allocation (NFV/SFC), congestion detection*
  - Management (slow dynamics)
    - *E.g., network planning, load estimation*
  - Recommendation systems
- Towards **self-driving networks**

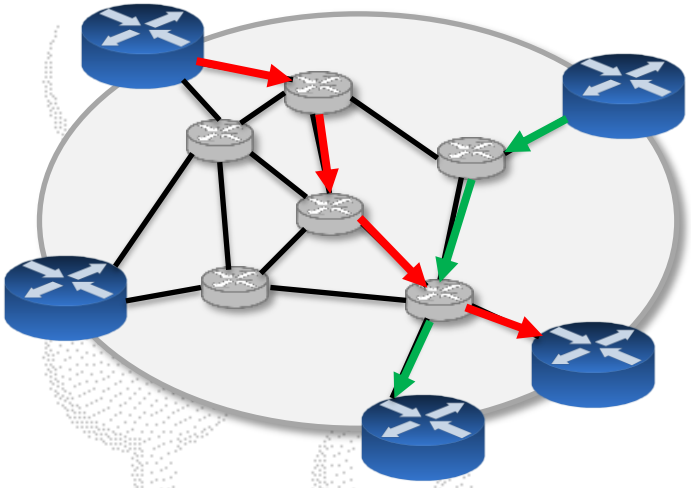
Use-cases and applications

- Network routing optimization
  - Performance modeling (e.g., delay, jitter)
- Knowledge extraction from network logs
  - Unsupervised log processing
- Network planning
  - Long-term traffic modeling and forecasting
- 5G mobile communication networks
  - Multi-layer network optimization

Use-cases: Routing optimization

- What is the optimal routing to minimize the end to end delay among all nodes?

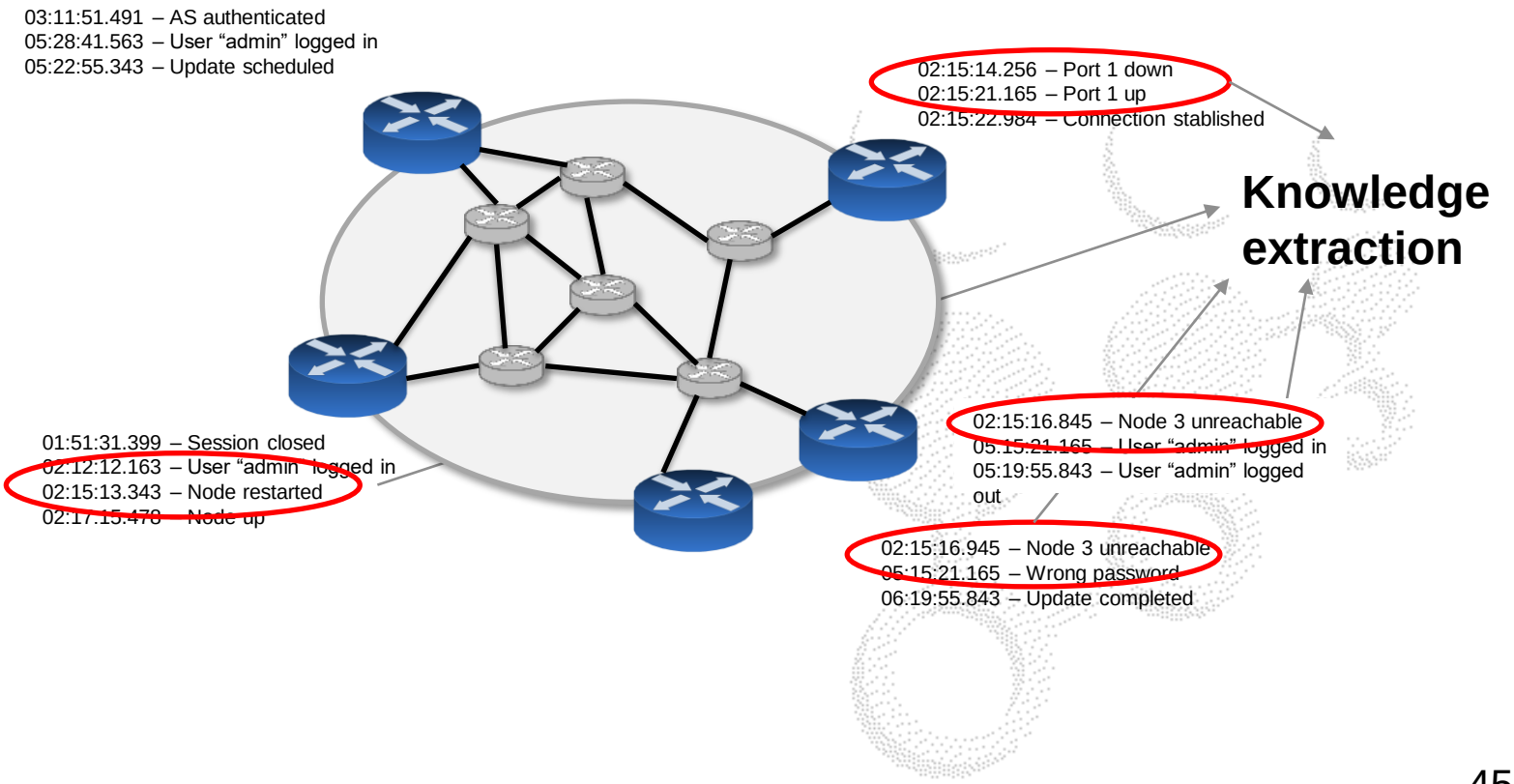




  - Dynamic routing as a function of the traffic

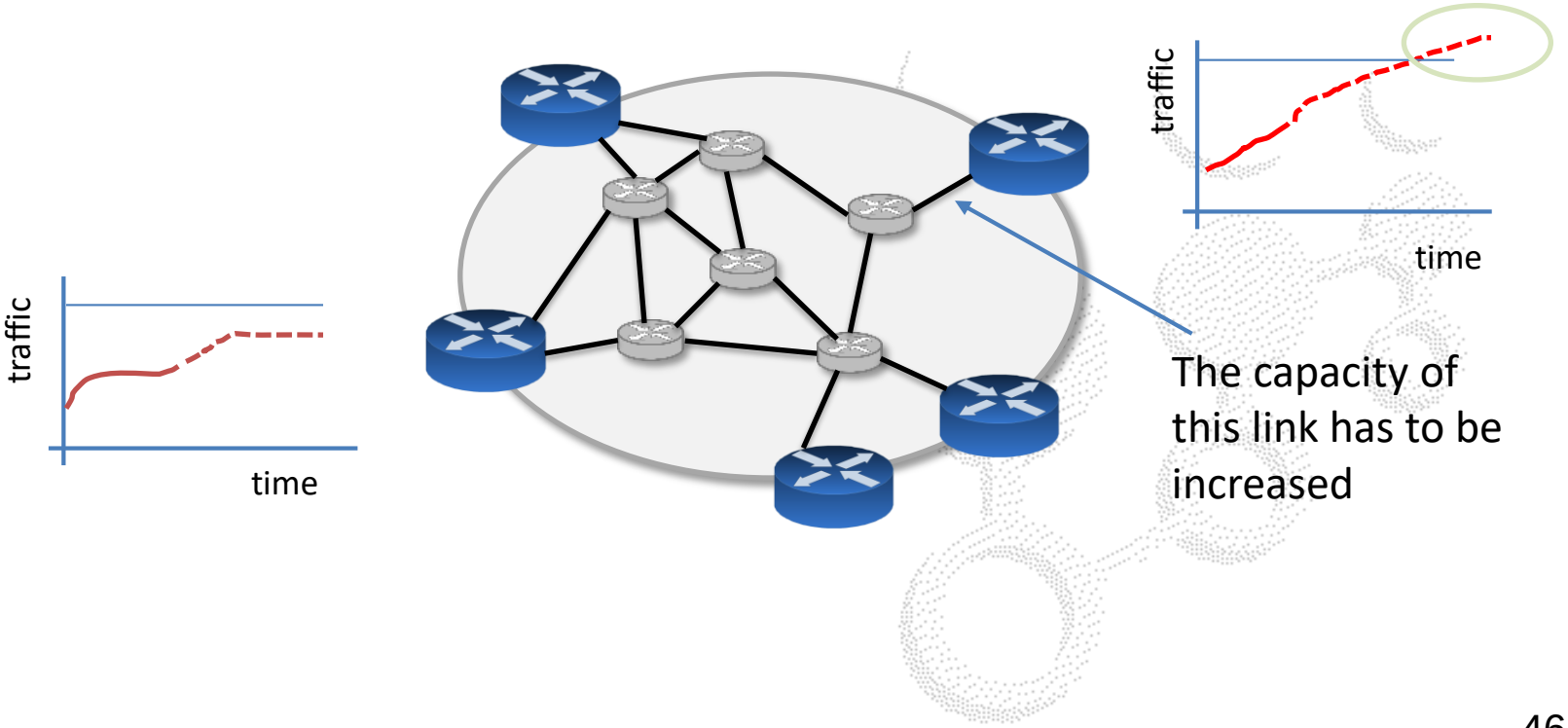
Use-cases: Knowledge extraction from network logs

- Is it possible to automatically process all logs in a network?



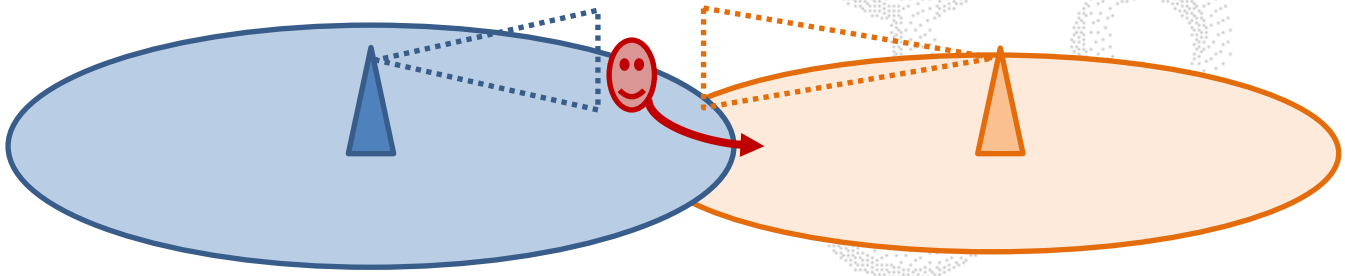
Use-cases: Network planning

- Is it possible to predict the growth of the traffic to anticipate network bottlenecks?



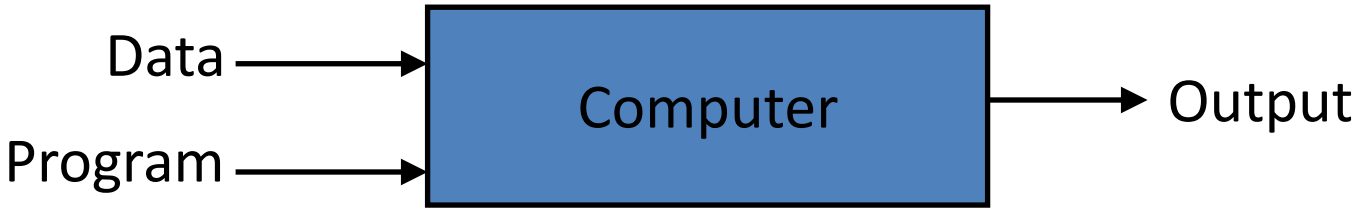
Use-cases: 5G mobile communication networks

- The 5G network is, by design, a Wireless SDN.
  - Multi-layer optimization scenarios
- Example: User mobility prediction.
  - Intelligent routing algorithms for a large number of mobile users (optimal access points)
  - Efficient beam-steering techniques.



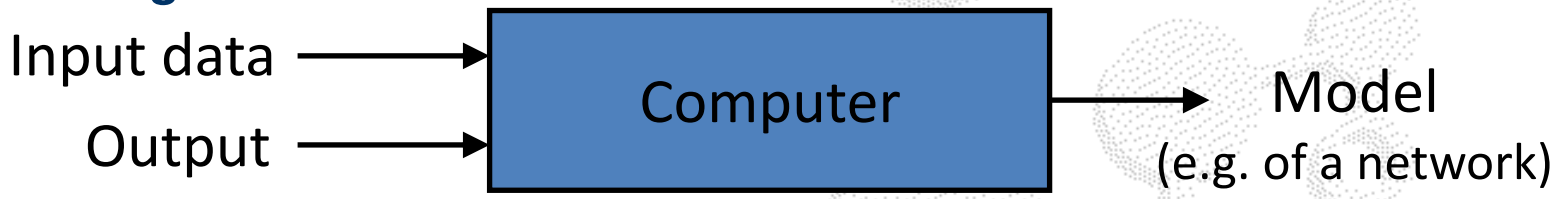
A brief introduction to Deep Learning

- **Traditional Programming**

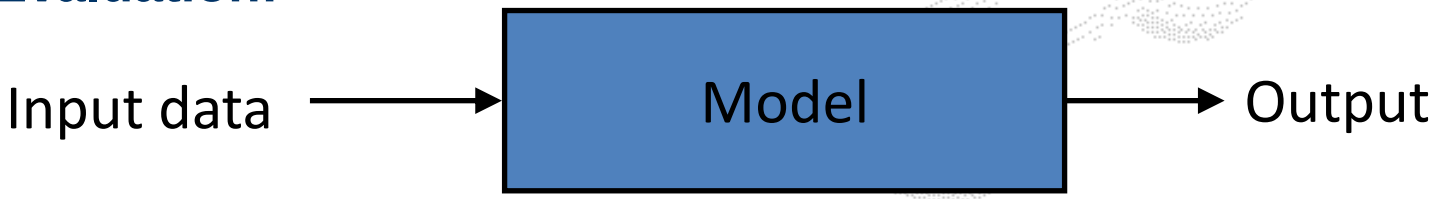


- **Deep Learning (supervised)**

Training:



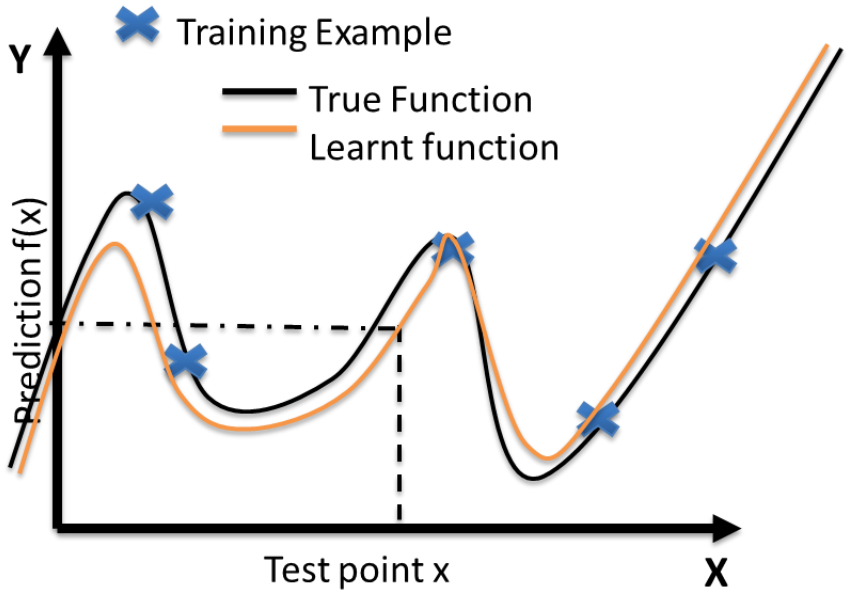
Evaluation:



A brief introduction to Deep Learning

DL for modeling problems

- Multi-dimensional function regression or fitting

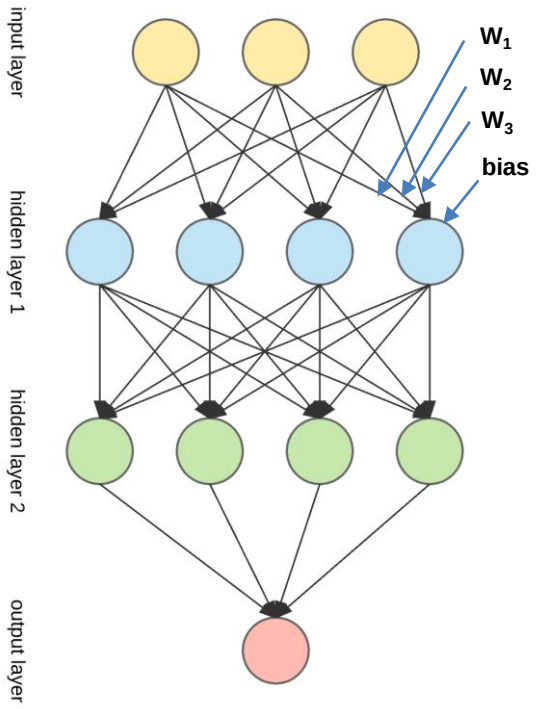


- With enough data DL will fit the true function
- Generalization capability:
  - Inter and extrapolation
- ML can predict unseen scenarios

A brief introduction to Deep Learning

Deep Learning

- Sub-field of ML that uses neural networks
- Based on the model of the human brain:
  - Neurons interconnected
  - Non-linear behavior
- Backpropagation to train the network:
  - Train: Define weights and biases in connections between neurons
  - Target: Minimize the loss (e.g., Mean Square Error) for all the samples in the training dataset
- The more hidden layers, the more knowledge abstraction capacity but it typically needs more training to converge

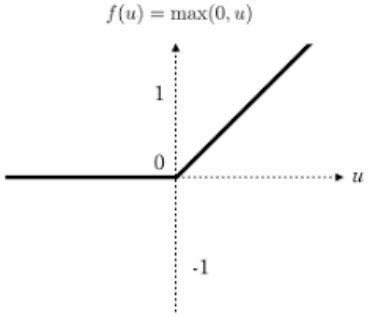


A brief introduction to Deep Learning

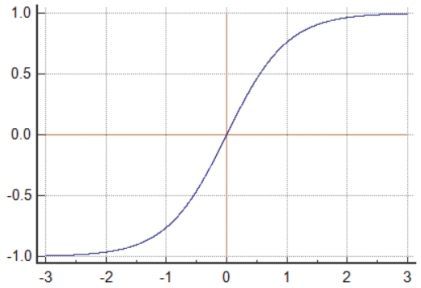
Deep Learning

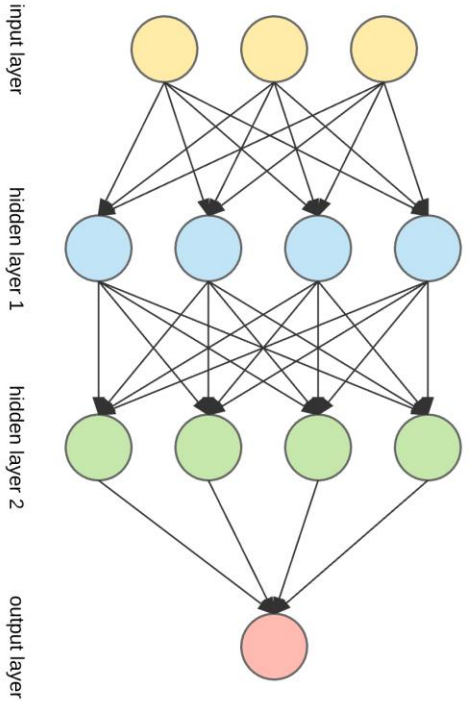
- Non-linear functions in neurons:

Rectified linear unit (ReLU)



hyperbolic tangent (tanh)





- Good to model scenarios with non-linear relationships between input features
- Output: continuous (e.g., bandwidth prediction) or discrete values (e.g., traffic classification)

Example: Traffic modeling and forecasting

- Objective
  - Is it possible to improve the performance of traditional forecasting techniques by using Deep Learning tools and the information of external events?

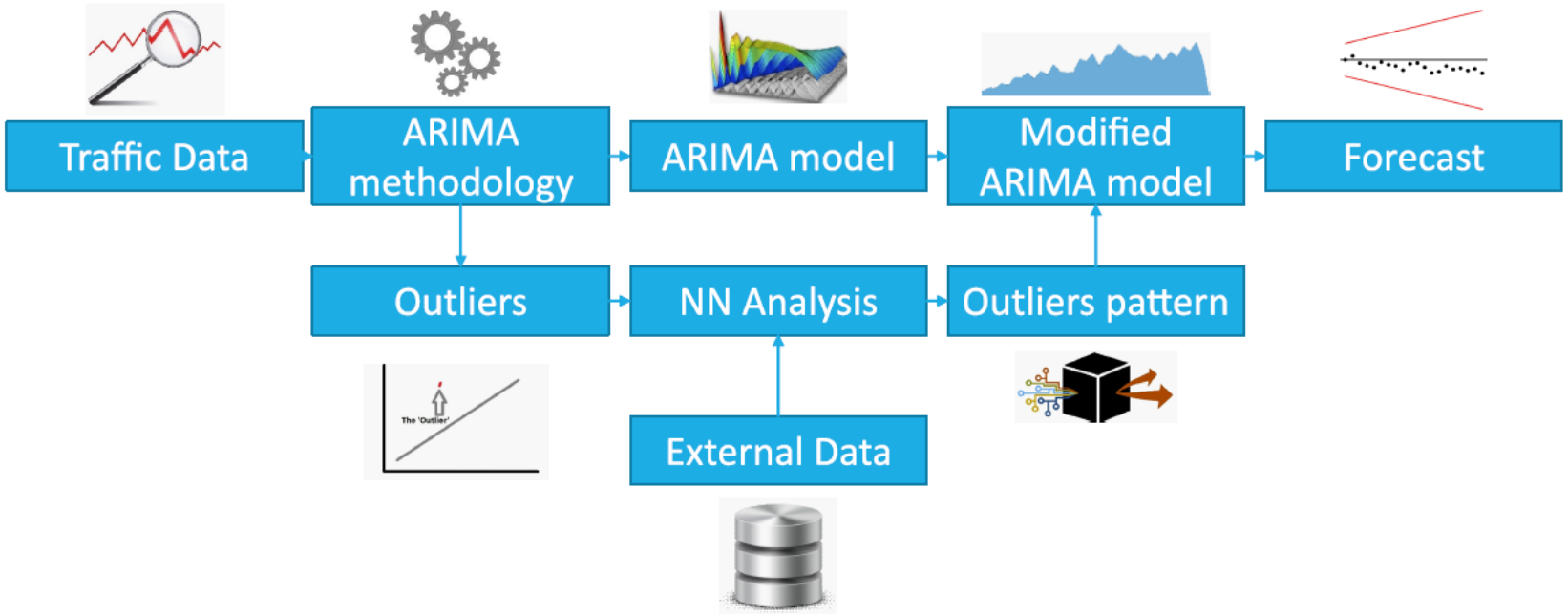
Problem statement

- Traffic forecasting is necessary to estimate the performance of the network
- Traditional approaches model the periodical behavior of the traffic (time series analysis):
  - ARIMA → Auto regression + integration + moving average
  - Some approaches use ML techniques to capture non-linearities in this behavior
- **External information** can be used to improve the performance of existing techniques

  e.g., weather, sports events, academic calendar…

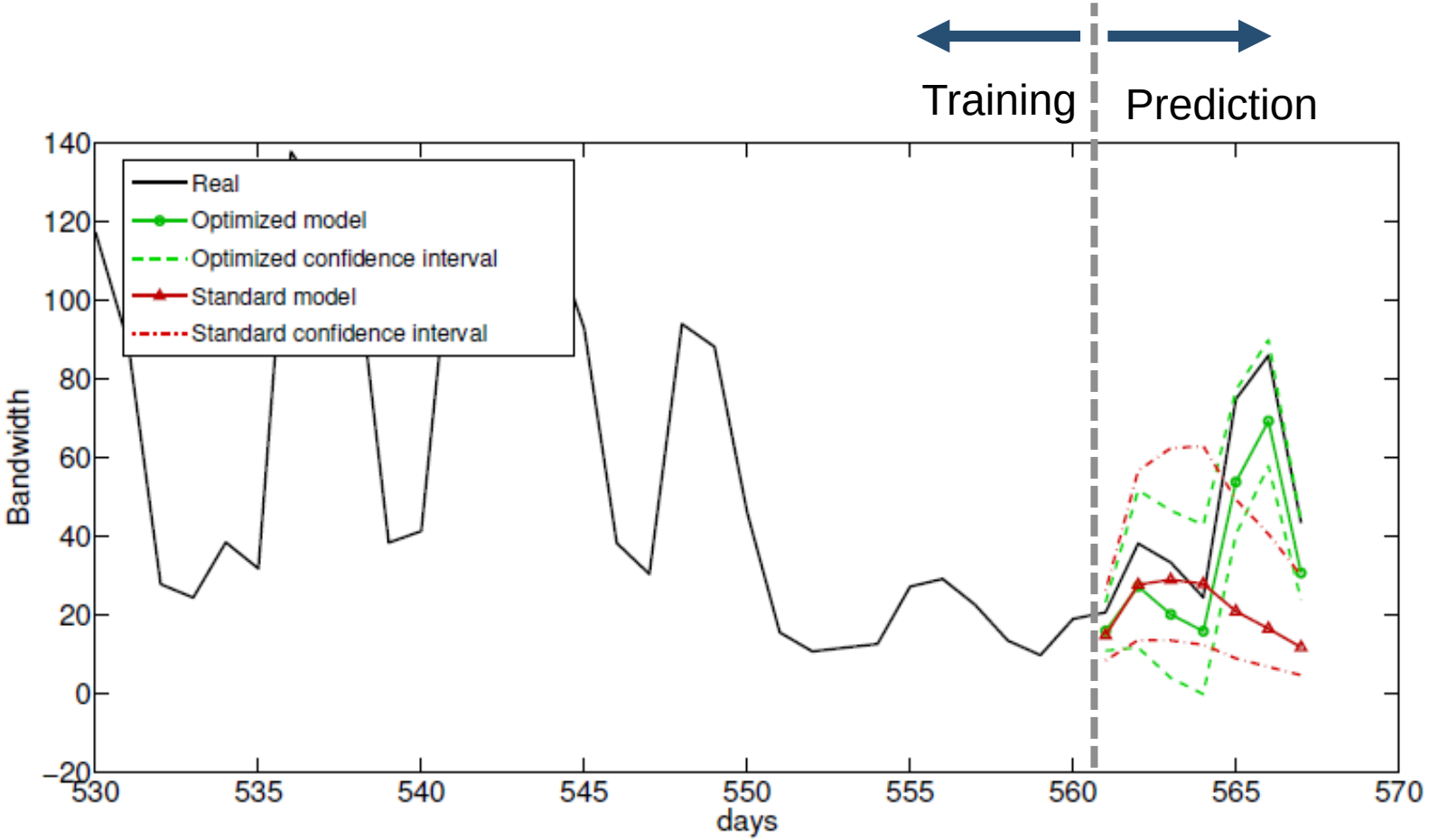
Methodology

- Based on outlier detection and prediction

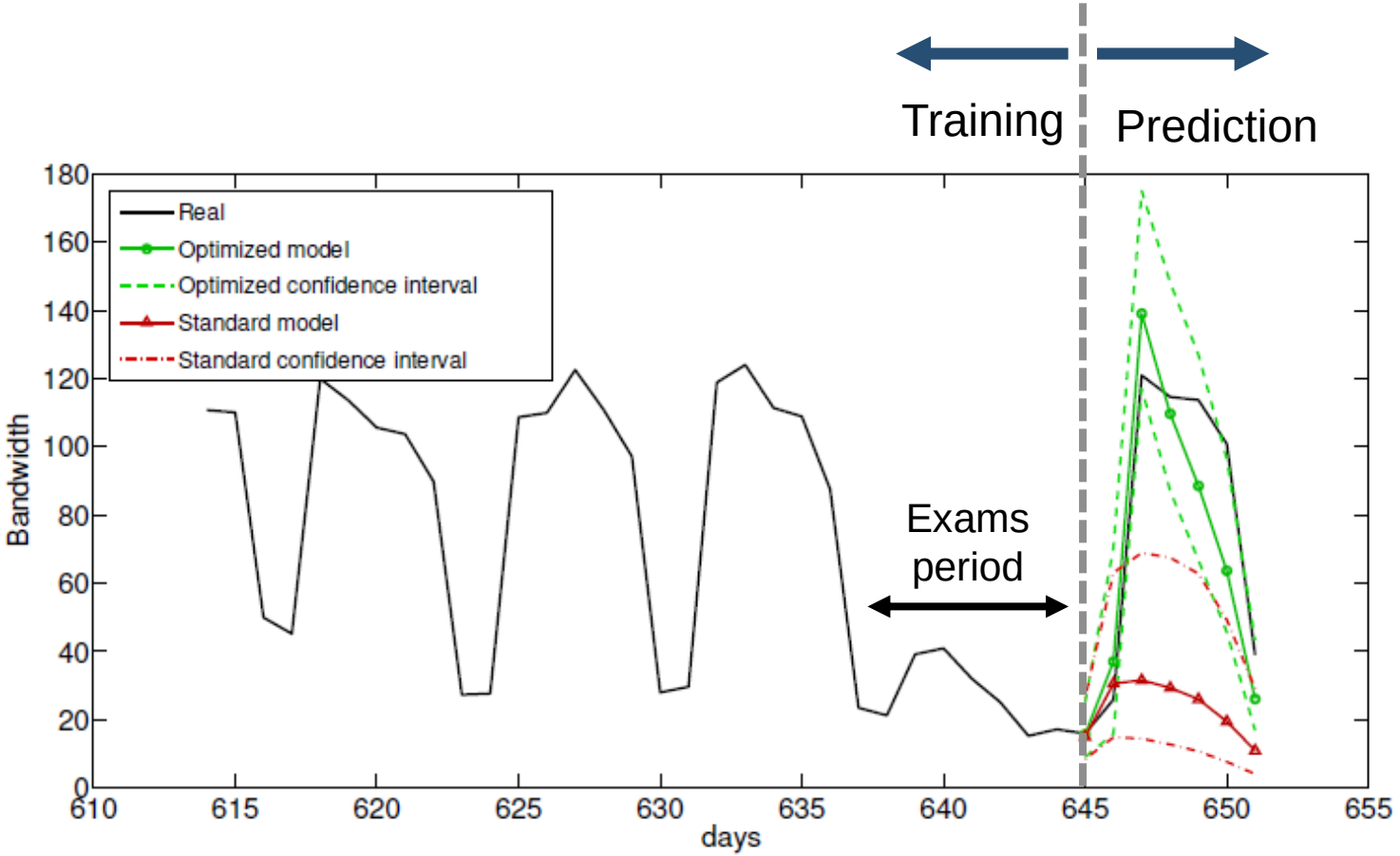


AO: Additive Outliers
TC: Transitory Changes

Results: Traffic forecasting



Results: Traffic forecasting



References

Software-Defined Networking (SDN)

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.

[2] Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE Vol 103, no. 1, pp. 14-76, 2015.

[3] Bosshart, Pat, et al. "P4: Programming protocol-independent packet processors." ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 87-95, 2014.

[4] Jain, Sushant, et al. "B4: Experience with a globally-deployed software defined WAN." ACM SIGCOMM Computer Communication Review. Vol. 43. No. 4, pp. 2-14, 2013.

[5] Koponen, Teemu, et al. "Network virtualization in multi-tenant datacenters." 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 203-216, 2014.

Traffic measurement

[4] N. L. Van Adrichem, C. Doerr, and F. A. Kuipers, “OpenNetMon: Network monitoring in OpenFlow software-defined networks,” in IEEE Network Operations and Management Symposium (NOMS), pp. 1-8, 2014.

[5] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, “OpenTM: traffic matrix estimator for OpenFlow networks,” in International Conference on Passive and Active Network Measurement, pp. 201-210, 2010.

References

Traffic measurement (cont.)

[6] M. Malboubi, L. Wang, C.-N. Chuah, and P. Sharma, “Intelligent SDN based traffic (de) aggregation and measurement paradigm (iSTAMP),” in Proceedings of IEEE INFOCOM, pp. 934-942, 2014.

[7] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, “Payless: A low cost network monitoring framework for software defined networks,” in Network Operations and Management Symposium (NOMS), pp. 1-9, 2014.

[8] José Suárez-Varela, and Pere Barlet-Ros. “Towards a NetFlow implementation for OpenFlow Software-Defined Networks.” In proceedings of the International Teletraffic Congress (ITC 29), 2017.

[9] Yu, Minlan, Lavanya Jose, and Rui Miao. "Software Defined Traffic Measurement with OpenSketch." NSDI. Vol. 13. 2013.

[10] Kim, Changhoon, et al. "In-band network telemetry via programmable dataplanes." in Proceedings of ACM SIGCOMM, 2015.

Traffic classification

[11] Nguyen, Thuy TT, and Grenville Armitage. "A survey of techniques for internet traffic classification using machine learning." IEEE Communications Surveys & Tutorials, vol. 10, no. 4, pp. 56-76, 2008.

[12] Dainotti, Alberto, Antonio Pescape, and Kimberly C. Claffy. "Issues and future directions in traffic classification." IEEE network 26.1 (2012).

References

Traffic classification (cont.)

[13] Mori, Tatsuya, et al. "Statistical estimation of the names of HTTPS servers with domain name graphs." Computer Communications, vol. 94, pp. 104-113, 2016.

[14] Trevisan, Martino, et al. "Towards web service classification using addresses and DNS." International Wireless Communications and Mobile Computing Conference (IWCMC), 2016.

[15] Suárez-Varela, José, and Pere Barlet-Ros. "Flow monitoring in Software-Defined Networks: Finding the accuracy/performance tradeoffs." Computer Networks vol. 135, pp.289-301, 2018.

Deep Learning for network modeling

[16] Mestres, Albert, et al. "Knowledge-defined networking." ACM SIGCOMM Computer Communication Review, vol. 47, no. 3, pp. 2-10, 2017.

[17] Mestres, Albert, et al. “Understanding the Modeling of Computer Network Delays using Neural Networks.” In proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, pp. 46–52, 2018.

[18] Mestres, Albert, et al. "A machine learning-based approach for virtual network function modeling." 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2018.

[19] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang. “Machine learning for networking: Workflow, advances and opportunities.” IEEE Network, vol. 32, no. 2, pp. 92–99, 2018.

Thank you for your attention!

ADVANCED BROADBAND COMMUNICATIONS CENTER (CCABA)

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

José Suárez-Varela Maciá

Email: jsuarezv@ac.upc.edu

Departament d'Arquitectura de Computadors