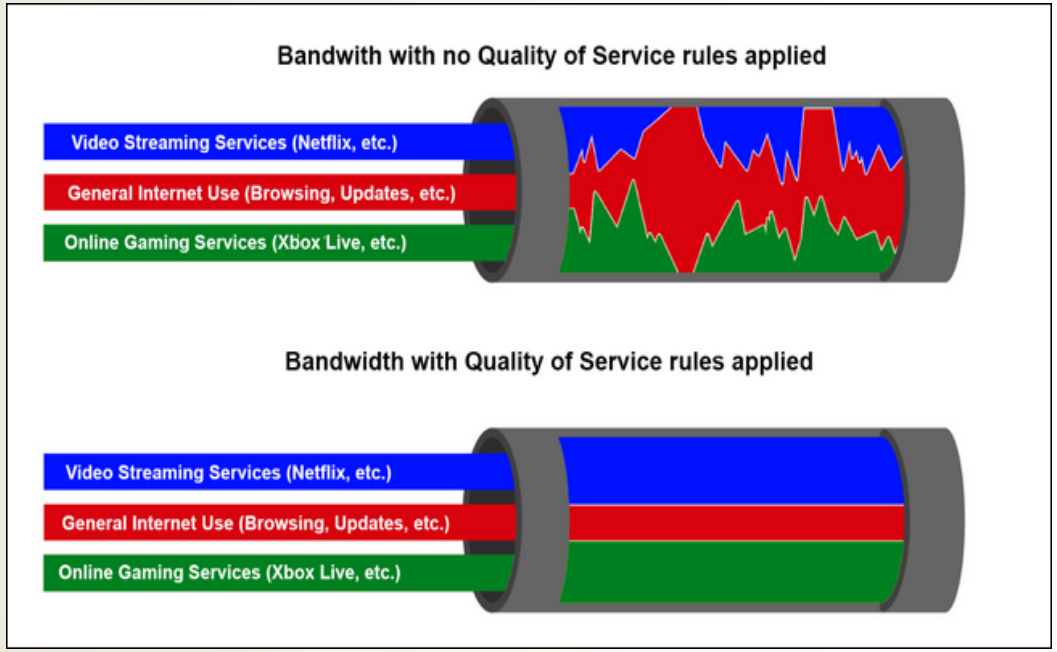
PIQoS: A Programmable and Intelligent QoS Framework

Udaya Lekhala and Israat Haque

Computer Science

Dalhousie University, Halifax

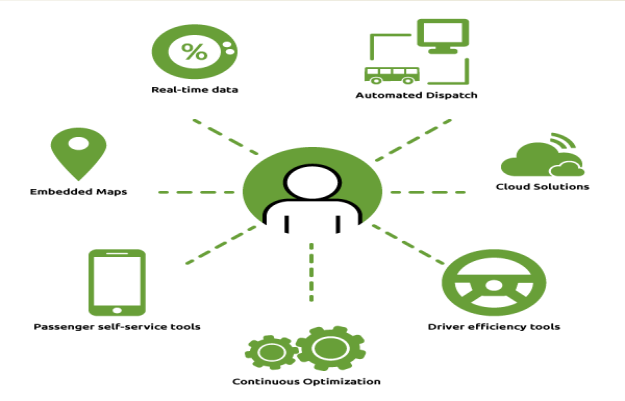
Importance of Quality of Service (QoS)



- Different applications (e.g., VoIP, file-transfer) demand different **quality factors** (e.g., delay, throughput).
- Service providers must satisfy users as per the QoS demand of their applications.
- Demand-based resource allocation is a must to offer QoS.

Dynamic and Intelligent QoS

- **According to CISCO VNI 2018 report**
  - **Broadband speed** will nearly double by 2022.
  - **IP video traffic** will globally be 82% of all IP traffic by 2022.
  - **Internet gaming traffic** will grow nine fold from 2017 to 2022.



- The explosion of applications and traffic dynamics pose challenges on QoS
  - Congestion, high bandwidth demand, dynamically changing network conditions, applications' requirements, and failures

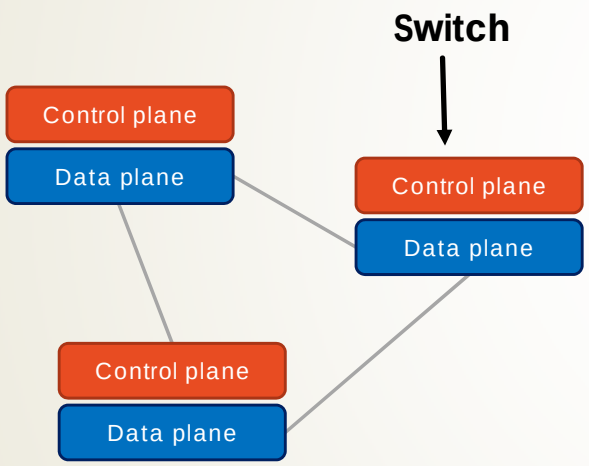
- Automated **error detection** and **root cause analysis** can lead to an efficient QoS provisioning
  - ***We need a programmable and intelligent QoS framework***

Road Map of PIQoS

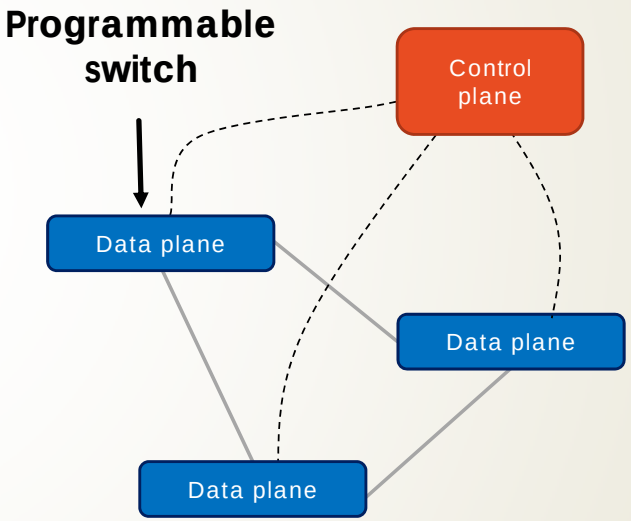
- Background
- PIQoS architecture
- Experimental setup
- Model evaluation data
- Discussion on results
- Conclusions

Software Defined Networking (SDN)

SDN decouples control plane (control logic) from data plane (forwarding hardware)



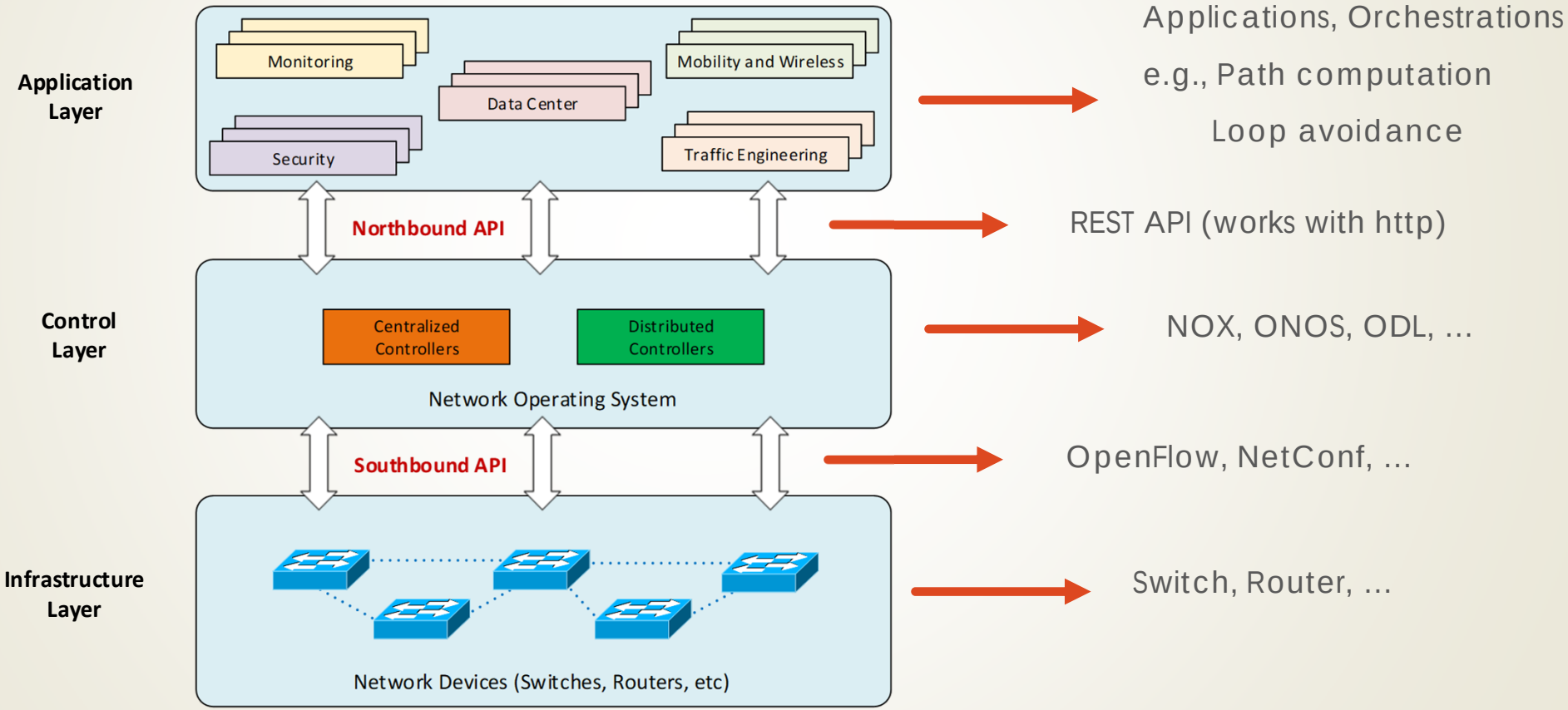
Traditional networking



Software-Defined networking

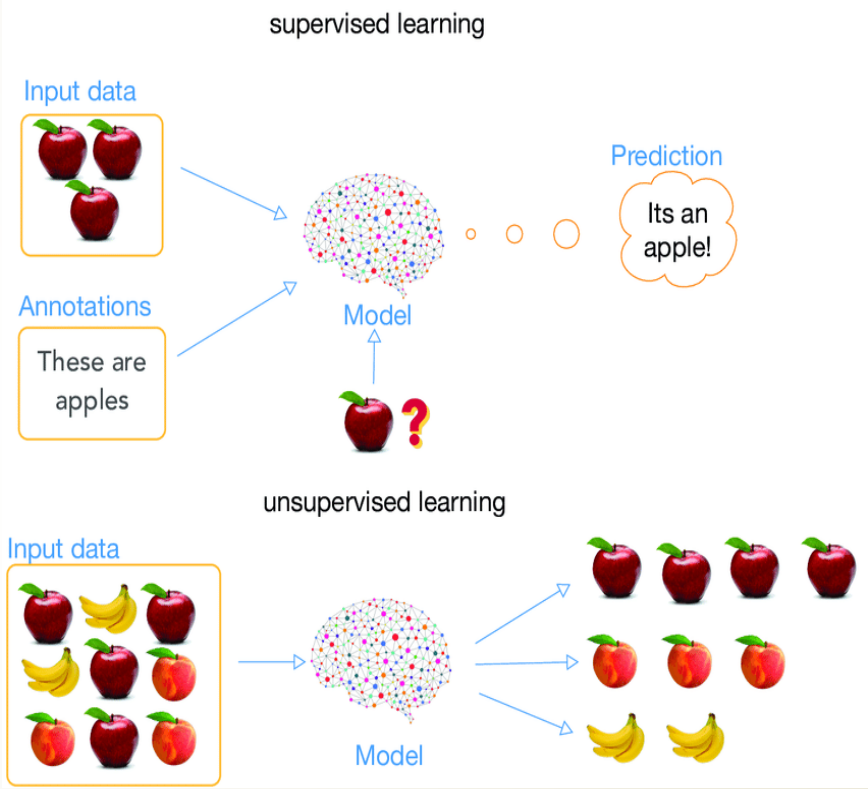
SDN Architecture

SDN Logical Layers and Architecture (IETF RFC 7426)



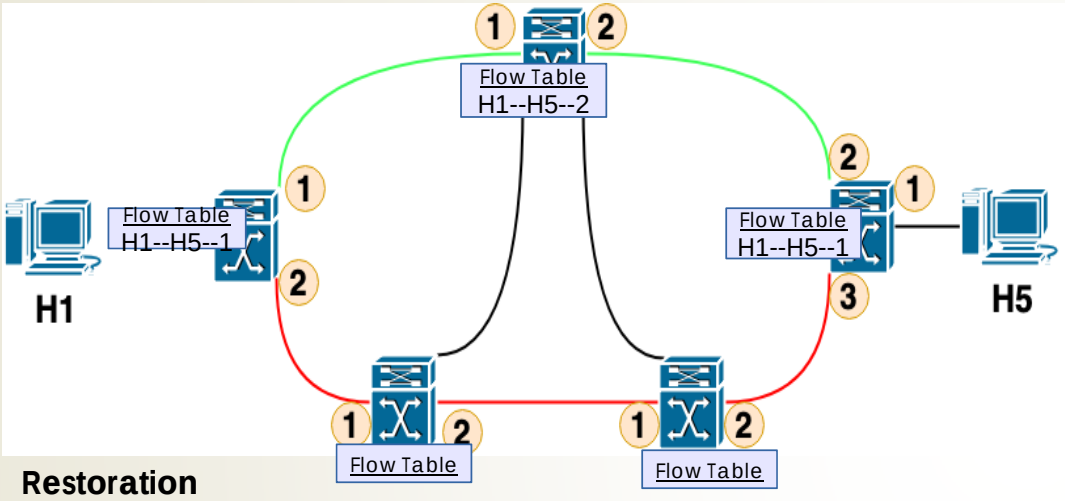
Supervised vs. Unsupervised Machine Learning Algorithms

- Supervised algorithms use labeled data sets to create models, while unsupervised learning uses unlabeled data.
- Each of these algorithms is better suited for a particular type of application
  - classification, regression or clustering
- Different learning algorithms can be used to solve the same problem.
- The best choice depends on the requirements of each scenario
  - performance and accuracy constraints

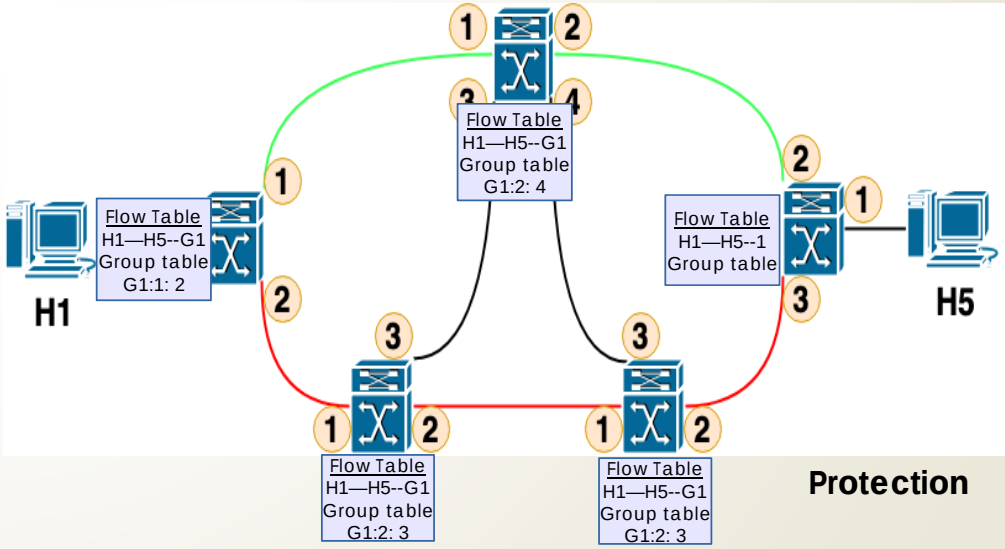


Source:https://www.researchgate.net/publication/329533120_Background_Augmentation_Generative_Adversarial_Networks_BAGANs_Effective_Data_Generation_Based_on_GAN-Augmented_3D_Synthesizing

Failure Recovery Scheme: **Restoration and Protection**

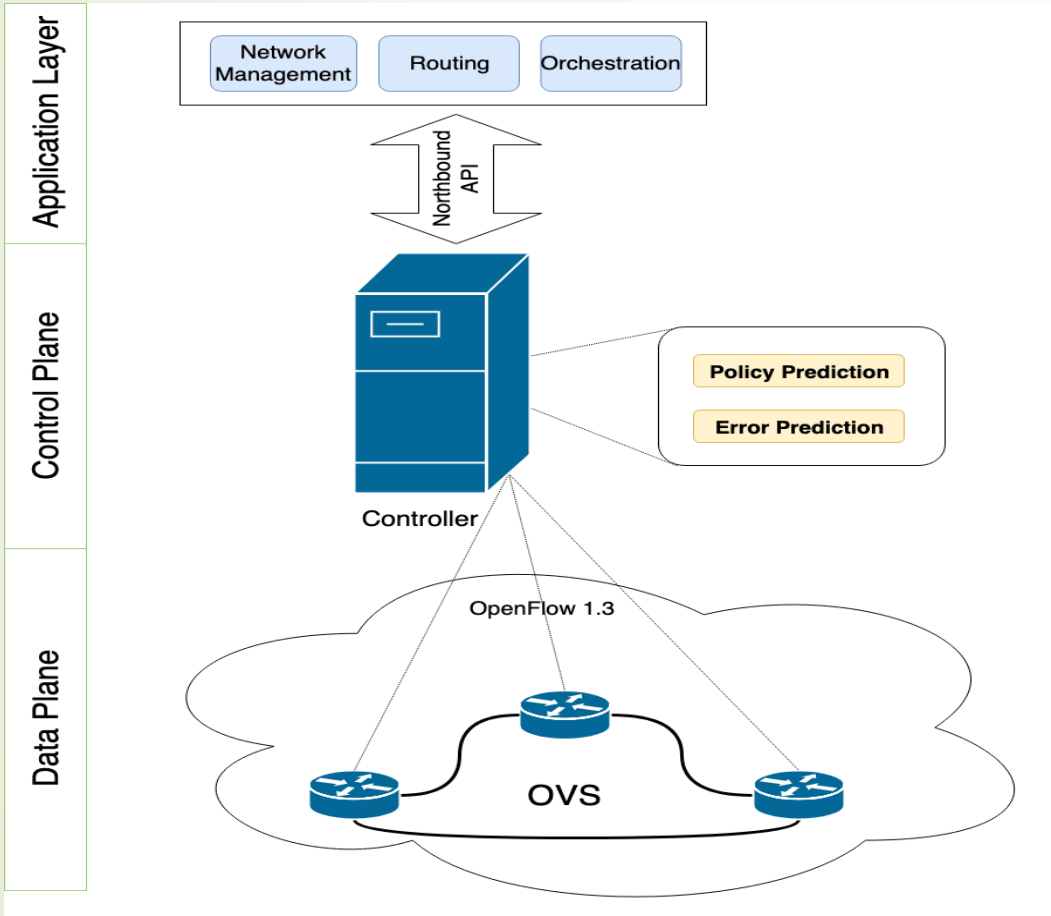


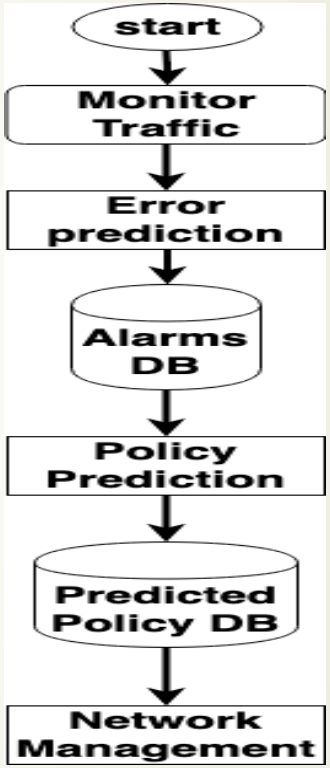
Restoration



Protection

PIQoS Architecture





workflow

Experimental Setup

Environment:

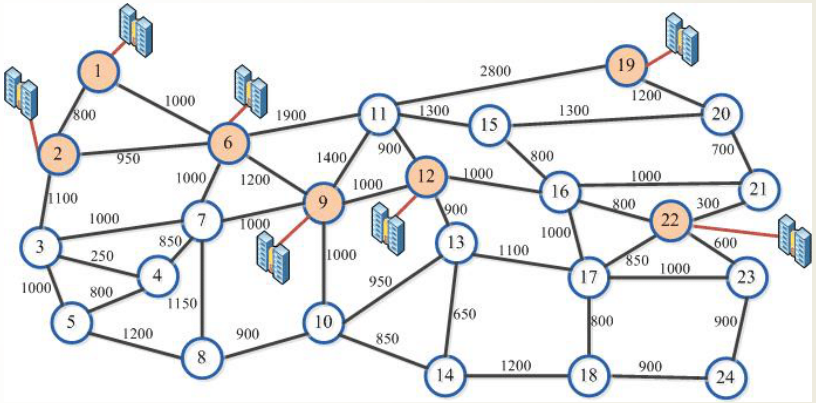
1) Emulator: Mininet
2) Data plane elements: Open vSwitch
3) Controller: Ryu
4) Scikit and Weka

Topologies:

1) Ten switch Mininet topology
2) USNET topology

Performance Metrics:

1) Failure recovery time
2) Throughput
3) Prediction accuracy



USNET Topology

Model Evaluation data

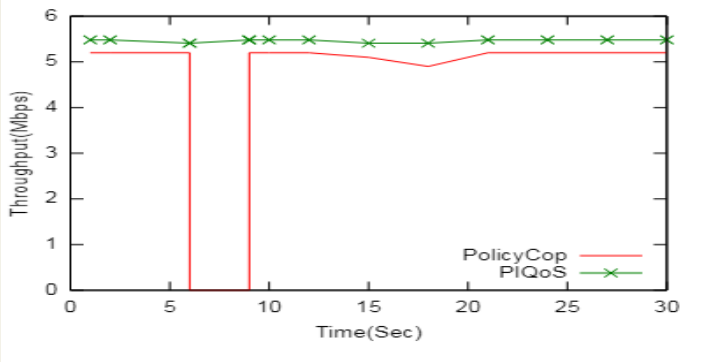
The list of feature used in PIQoS

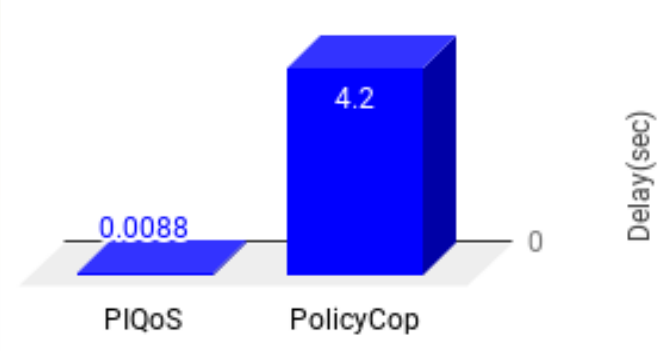
Feature	Explanation
Time stamp	When a flow starts at a link
Flow ID	ID of a flow or a path
Source Node	Source node of a link in a flow
Destination Node	Destination node of a link in a flow
Source Outport	Port from which a source node sends packets
Destination Inport	Port from which a destination ode receive packets
Packets	Total number of packets sent in a link
Bytes	Number of bytes sent in a link
Linkspeed (B/Sec)	Speed of a link
Source Output Capacity	Maximum capacity of a source output
Destination Inport Capacity	Maximum capacity of a destination inport
Source port Speed (B/s)	Speed at which a source outport sends packets
Destination port Speed (B/s)	Speed at shich a destination inport receives packets
Delay (ms)	Time taken for a packet to be transferred from a source to a destination in a flow
Error Prediction Classification	A flow has an issue or not (error/no error)
Policy Prediction Classification	Classification type (link failure/congestion)

The number of flows and links used for training and testing models

Model	Total flows	Total links	Training flows	Training links	Testing flows	Testing links
Error prediction	318	1020	225	714	93	306
Policy prediction	151	465	106	326	45	139

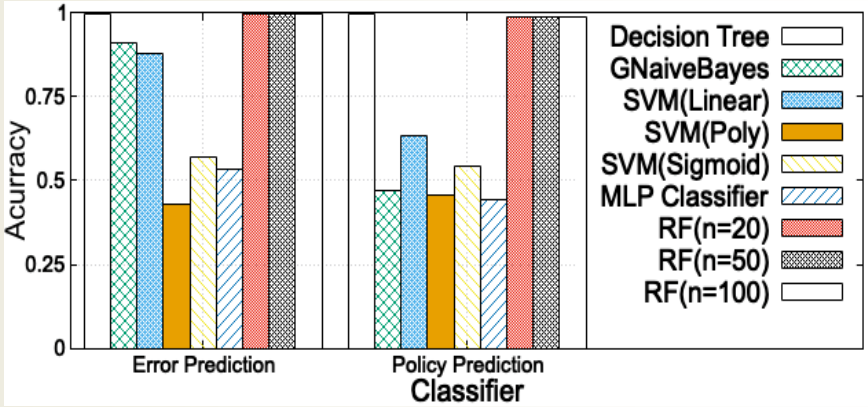
Resiliency of PIQoS





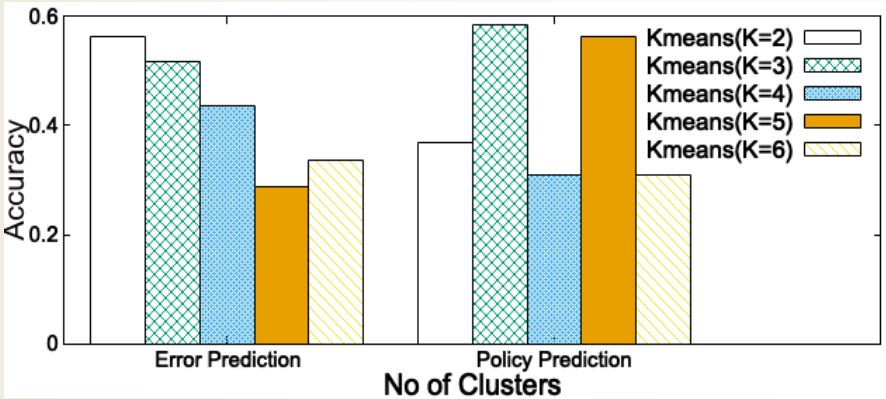
- PolicyCop: throughput drops and delay increases as it needs to contact the controller
- PIQoS: ***local recovery improves both the throughput and delay***
- ***Local recovery can furthermore offer scalability***

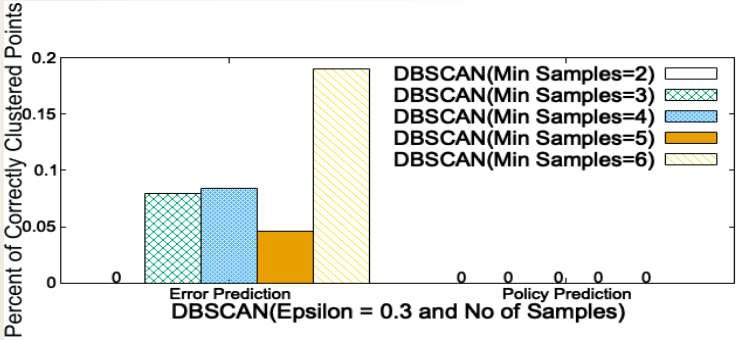
Intelligence of PIQoS (supervised algorithms)



- Naive Bayes and SVM have the worst accuracy.
- Random Forest and Decision Tree have the best accuracy.
- Decision Tree is less complex.
- ***Decision tree is the winner.***

Intelligence of PIQoS (unsupervised algorithms)





- Supervised algorithms need label on data, which may not scale with large networks.
- K-means offers best accuracy for two clusters.
- This accuracy is quite low compared to that of DT.
- ***There is a trade-off between accuracy and overhead.***

Conclusions

- We proposed a programmable and intelligent QoS framework, called PIQoS.
- PIQoS pushes link failure recovery at the data plane to improve the delay and throughput.
- We proposed two learning models for efficient network state diagnosis and selecting corresponding management policies.
- Implemented the proposed framework in a realistic simulation environment.
- **PIQoS can accurately predict link failure or network congestion and update policy accordingly.**

Thank you ☺

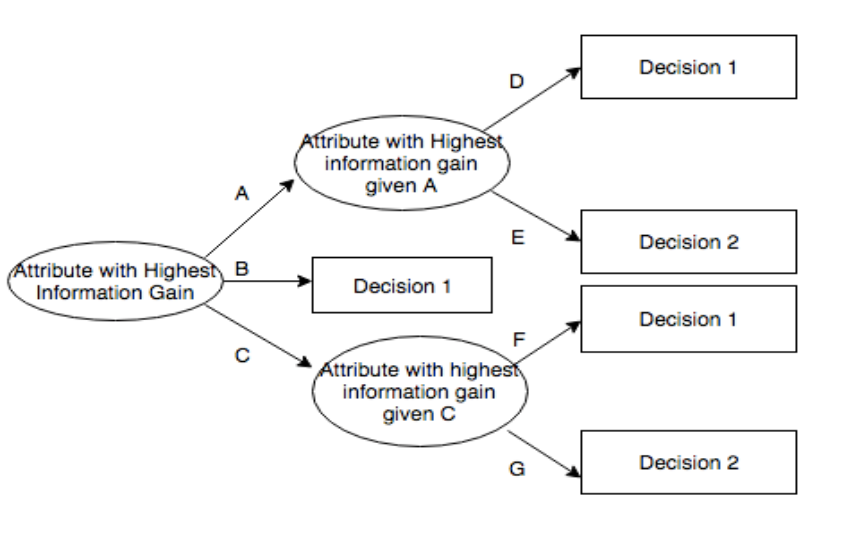
israat@dal.ca

Quality of Service (QoS)

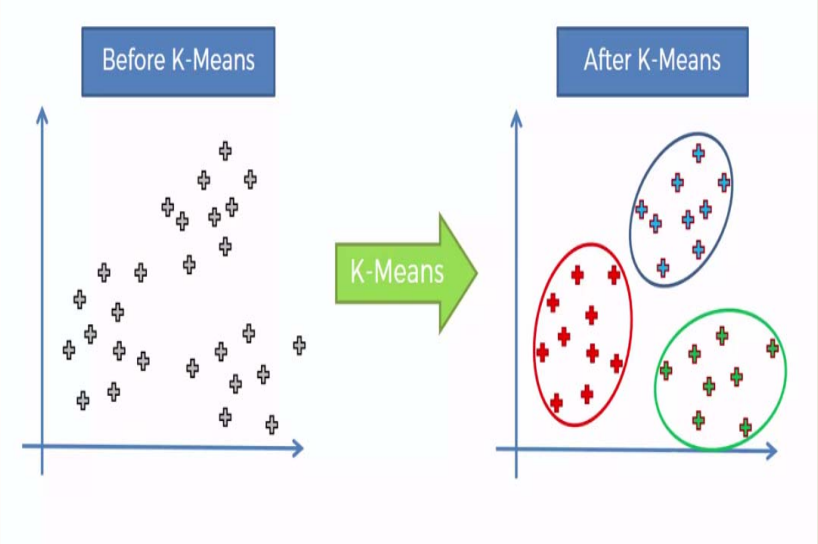
- An important factor for service providers
  - To offer consistent user experience
- Emerging service requirements, network traffic and their dynamics pose a new challenge
- Automated error detection and root cause analysis can lead to an efficient QoS provisioning

- ***How can we design a programmable and intelligent QoS framework***?

Decision Tree and K-means



Decision Tree



K-means

Source: https://en.wikipedia.org/wiki/ID3_algorithm

Source: https://towardsdatascience.com/k-means-clustering-identifying-f-r-i-e-n-d-s-in-the-world-of-strangers-695537505d

Decision Tree and K-means

Decision tree:

- Data is stored in a tree-like structure a node, branch, and leaf-node represent test, decision, and class label, respectively
- A decision tree path defines the data classification rules
- An attribute that best classify training data is chosen as the root and repeat the same process at each branch until all attributes are covered

K-means:

- Partitions data points into clusters
- Each cluster of data points has a centroid defined by those data points
- At the beginning of the clustering, we can randomly choose K centroids and iteratively optimize their positions