Special Topics: Machine Learning (ML) for Networking

COL867
Holi’25

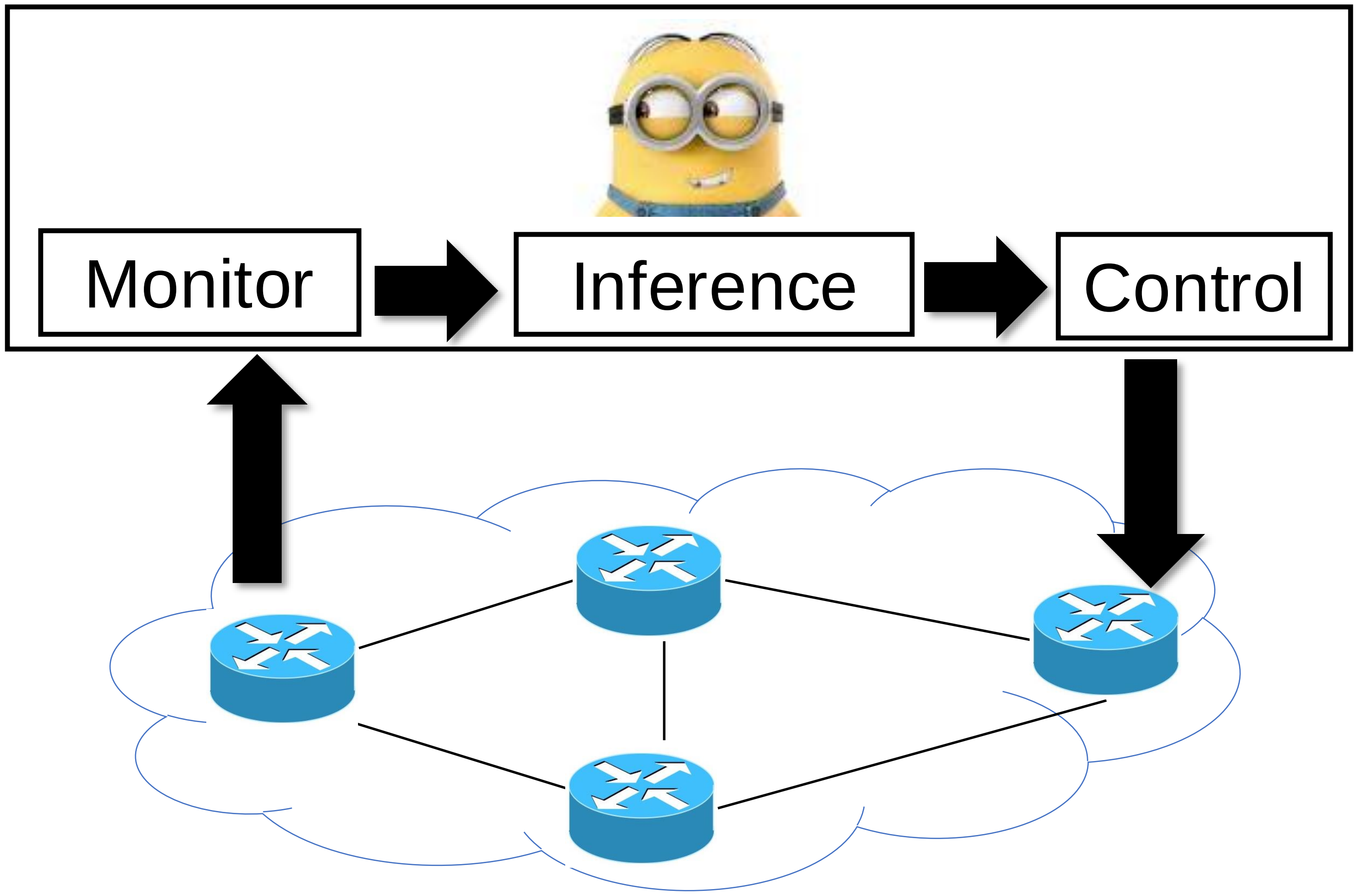
Lecture 2

Tarun Mangla

Lecture Outline

- What is ML for Networking (MLNet)
- Why MLNet Now
- Ideals of MLNet
- Intro to Networking Data

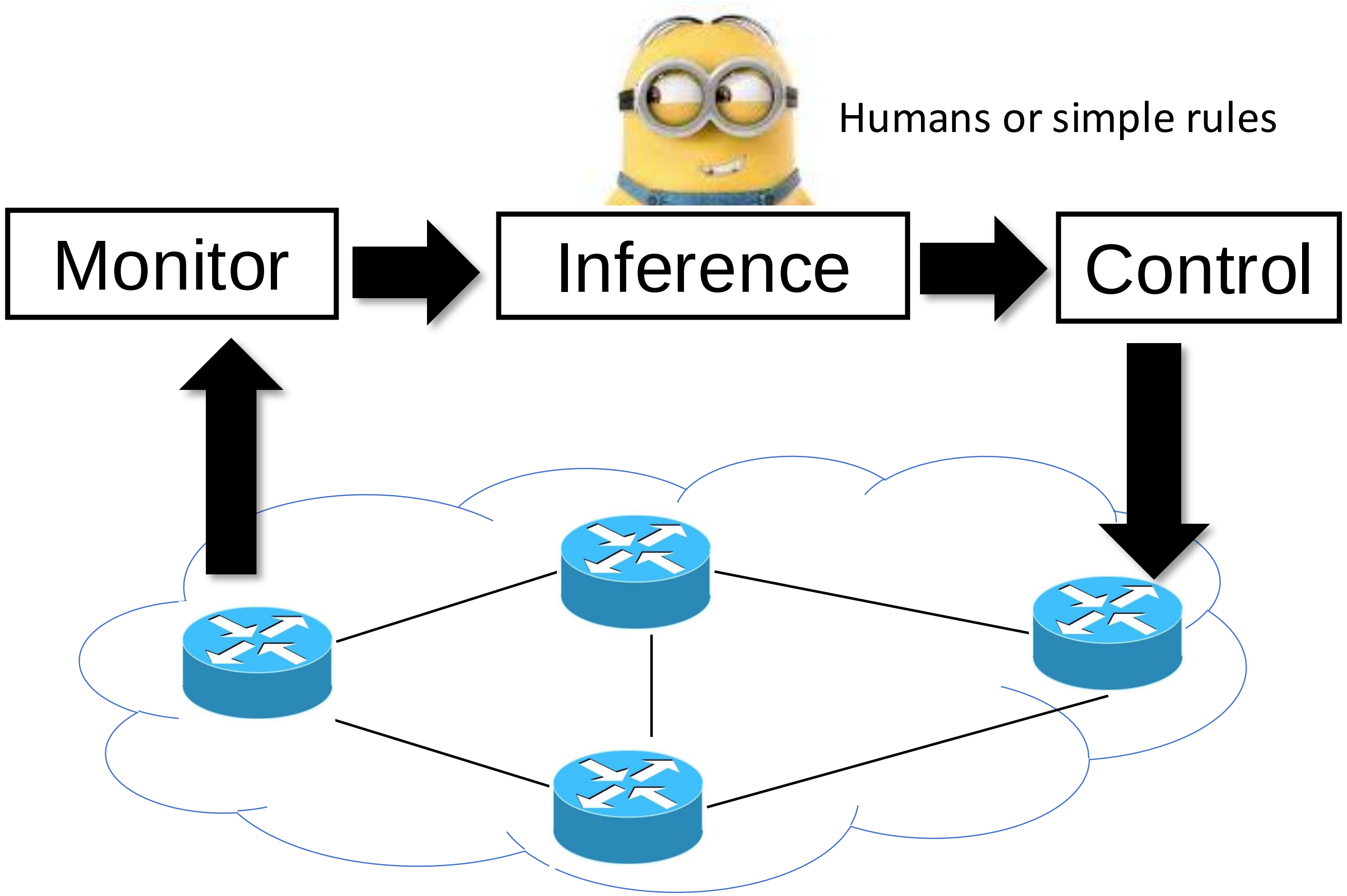
Network Management Control Loop



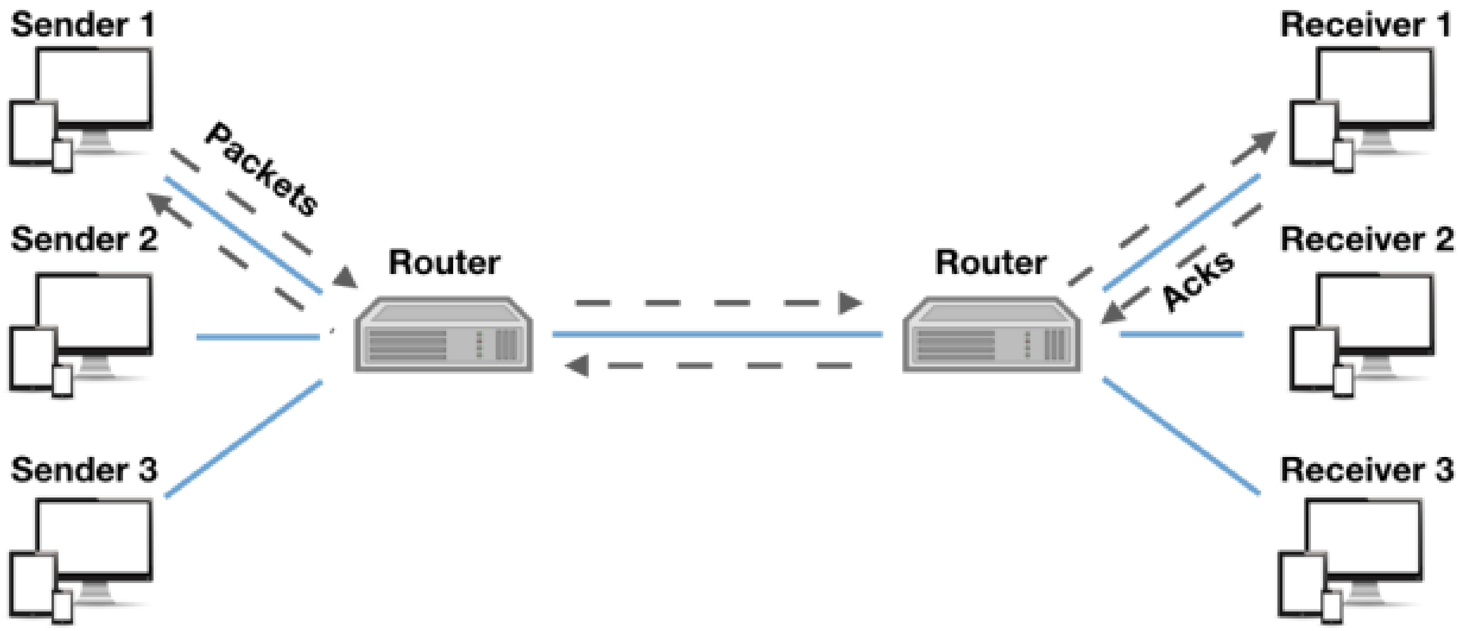
Learning Problems for Networks

- ***Learning* for networks is not new**
  - Learning is intrinsic to networking
  - Attributable to inherent partial visibility into network's state
  - Entails multiple complex closed-loops across different spatial and temporal granularities
- **Examples**
  - For TCP, end-hosts need to infer available bandwidth
  - For video streaming, client needs to infer future network bandwidth
  - For network security, the operator needs to infer benign/malicious traffic
  - …

Existing Approaches are Heuristics-based



Learning Problems for Networks (TCP Congestion Control)



Learning Problems for Networks (TCP)

- **Monitor**
  - **Indicators of congestion:** Packet loss, delays
- **Infer**
  - **Available bandwidth:** TCP aims to estimate the available bandwidth on the path between the sender and receiver.
- **Control**
  - **Window size**: Update window size so as to achieve a balance between high utilization and low delay while ensuring fairness and stability

Approaches to Solve Learning Problems

- **Simple (transparent, explainable) heuristics**
  - E.g., current breed of congestion control and ABR algorithms employ simple rule-based approach to infer network state

Changing Requirements

Monitor → Inference → Control

Need to handle tasks that are:

~~simpler~~ (complex),

~~small~~(large)-scale,

~~infrequent~~ (frequent)

Approaches to Solve Learning Problems

- **Simple (transparent, explainable) heuristics**
  - E.g., current breed of congestion control and ABR algorithms employ simple rule-based approach to infer network state
- **Machine Learning for Networks**
  - Recent works have shown ML-based solutions can better detect subtle and complex patterns in data → better decisions
  - E.g., ex Machina (TCP), Pensieve (ABR), …

Learning Problems for Networks (Security)

Scaling Intrusion Detection System (IDS) and Firewalls

- Require flexible packet processing
- Specialized h/w or user-space processing (**expensive**)

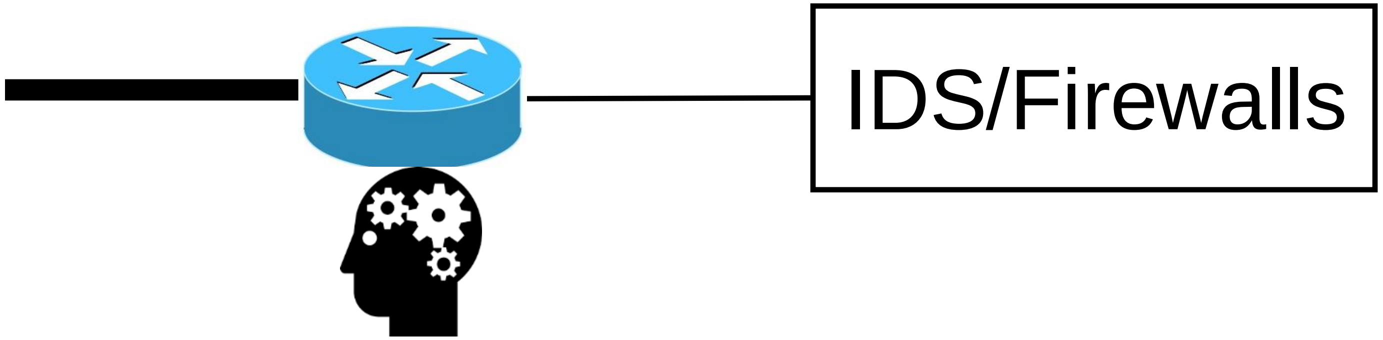
Learning Problems for Networks (Security)

- **Observation**

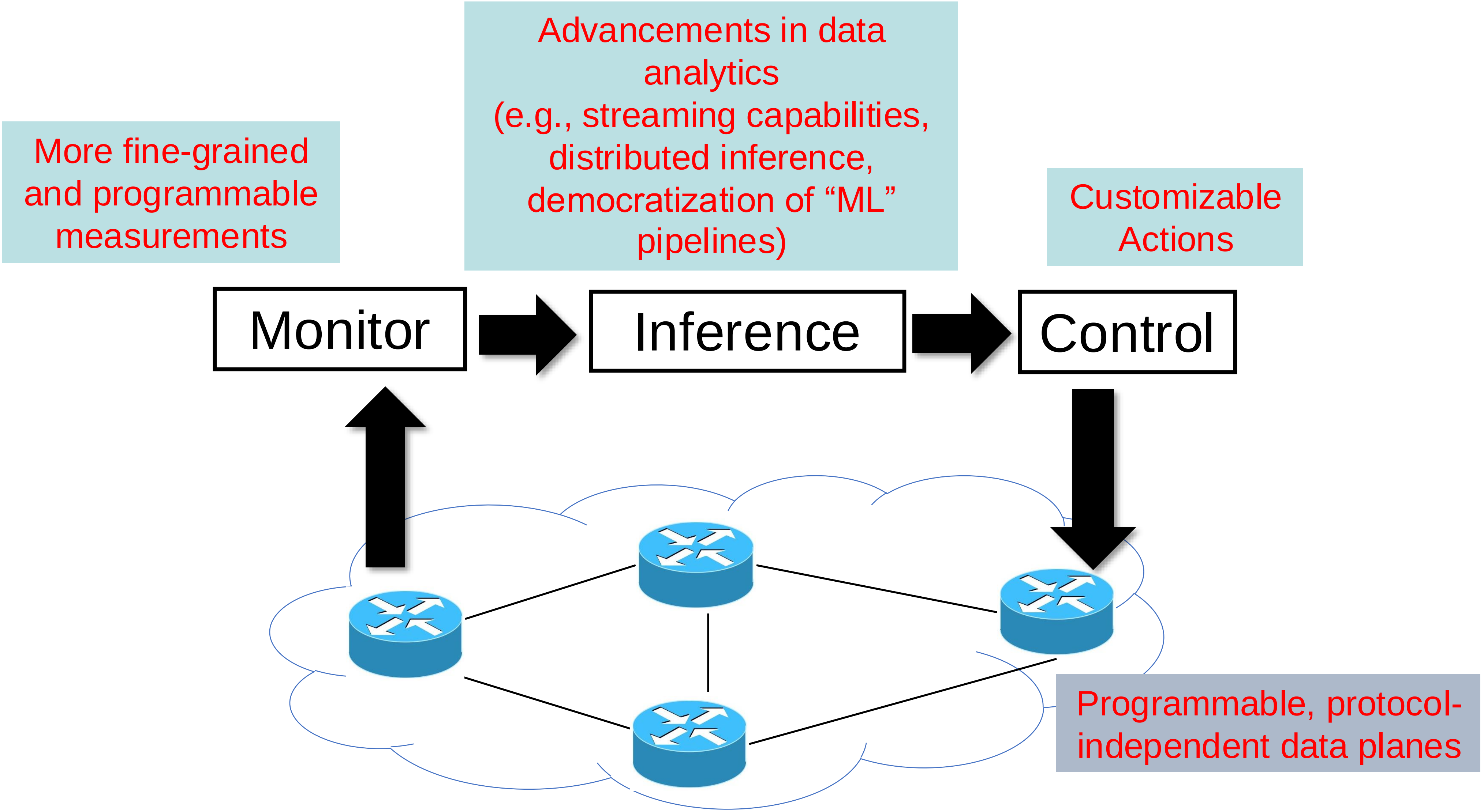
  Not all traffic needs to go the the IDS boxes

- **Approach**

  Use ML model(s) to detect benign traffic



Why ML for Networking "Now"?



The State of the NetML: Where We Are, Where We'd Like to Be

- **Monitoring**
  - **low-level metrics/aggregates**
  - …but not high-level application characteristics properties (e.g., QoE)
- **Inference**
  - **offline detection (security) and prediction (TE, provisioning)**
  - …but not in real time and not coupled w/control
- **Control**
  - **networks are more programmable**
  - … but not always scalable, distributed, or real time
  - … not always integrated across routing protocols, scheduling/shaping, job/task/cache placement, etc.

Security and Privacy Ideals

- **Monitoring:** Gather only what's needed
  - Use queries to drive data collection (IPFIX, packet traces, payload, DNS)
  - Trigger "deep dives" based on lightweight thresholds
- **Inference:** Detect attacks in real-time (not just offline traces)
  - "Compile" machine learning-based inference models to line-rate targets
  - Trigger gathering of more information as needed (networking meets active learning)
- **Control:** Automate (some) decisions
  - Rate limiting decisions (DNS response rate-limiting in-network)
  - Traffic redirection (e.g., to scrubbers, middleboxes)
  - On-the-fly placement of virtual middleboxes

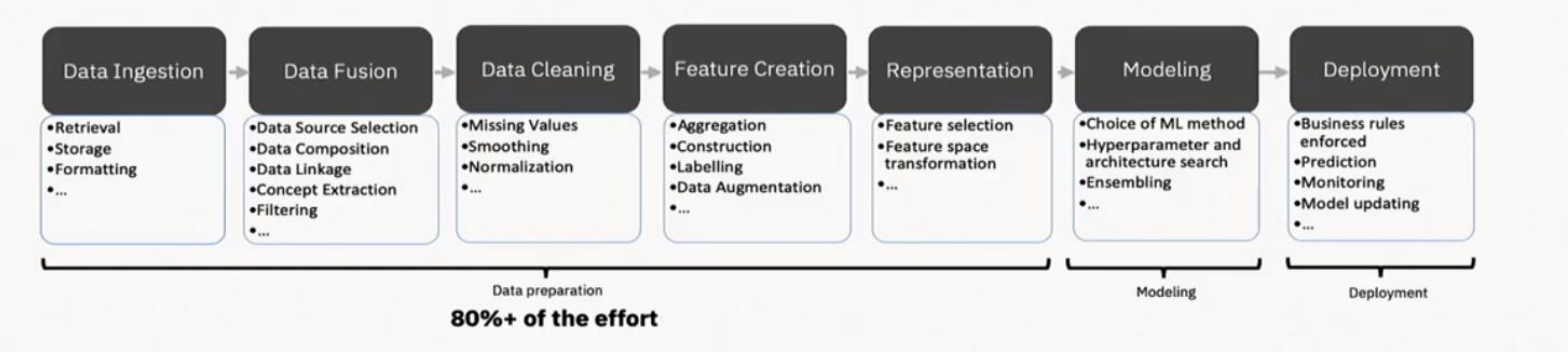
Summary

- Learning fundamental to networking
  - Closed loop: Monitor → Infer → Control

- Traditional networking learning characterized by heuristics. However, sub-optimal

- Growing adoption of machine learning
  - Application pull and Technology push

Machine Learning Pipeline



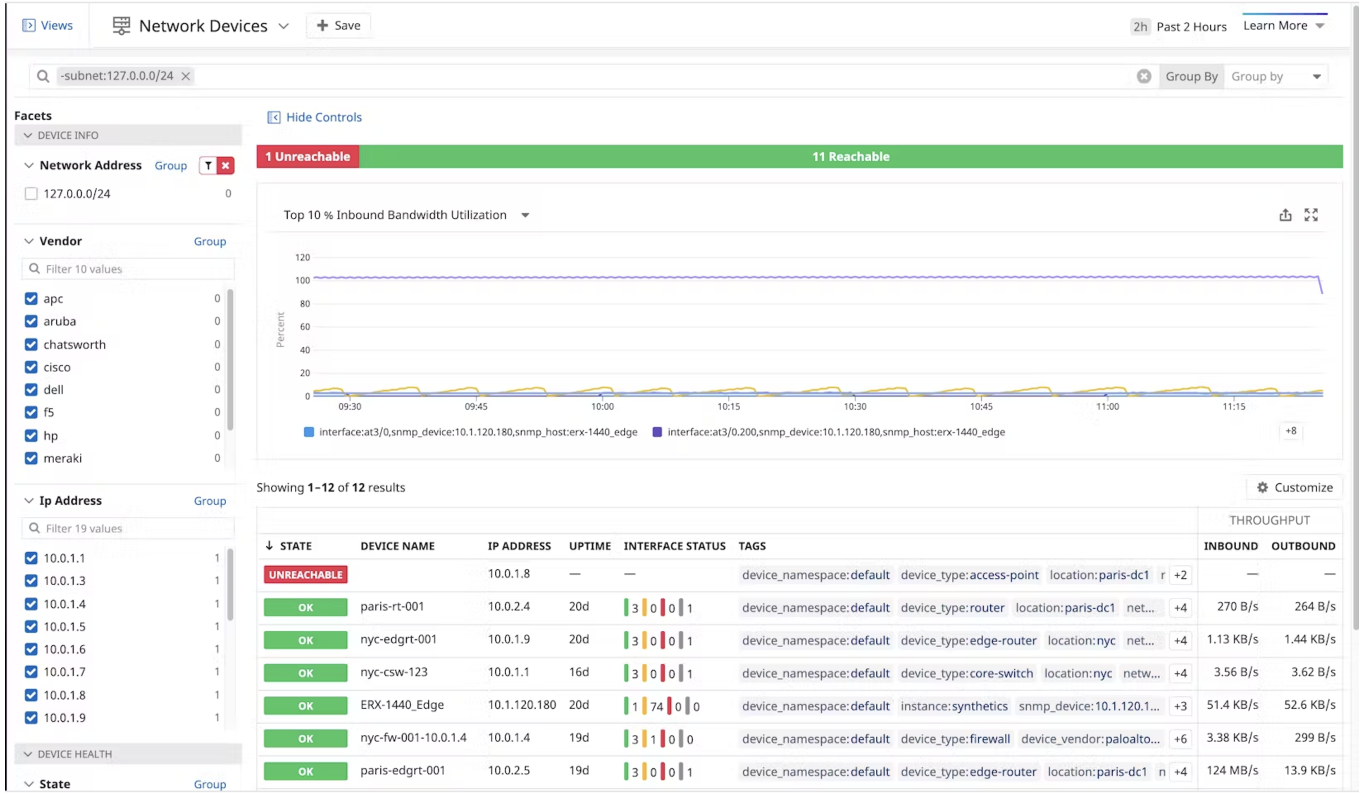
- What is the starting point?

- Let's look at the network data

Network Data

SNMP Logs

- Simple Network Management Protocol (SNMP) logs
  - CPU Usage
  - Throughput
  - Temperature
  - Uptime
  - ..



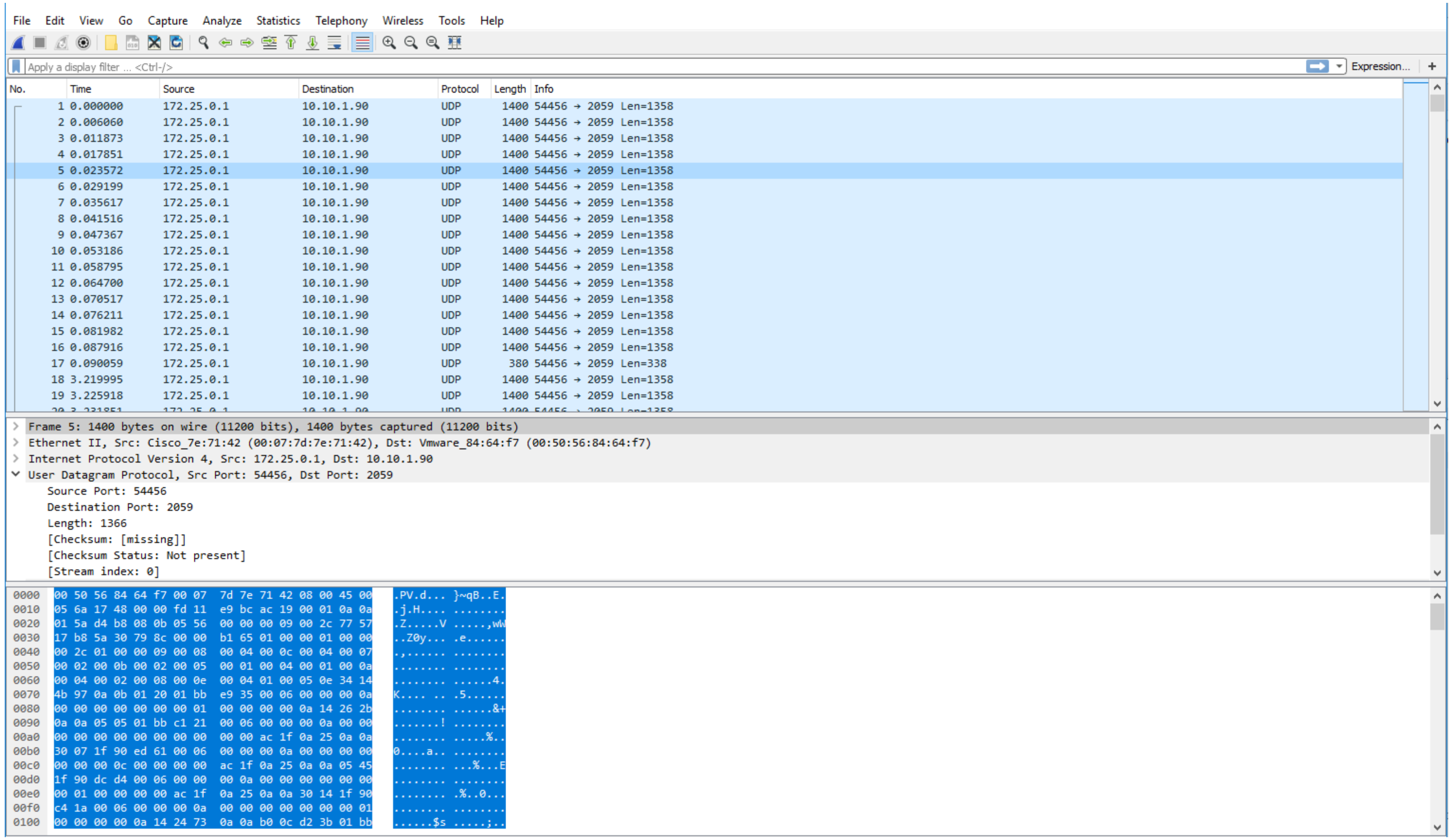
Flow-level Statistics

- Multiple protocols such as NetFlow, sFlow, IPFIX
- Sample NetFlow logs

Date flow start	Duration	Proto		Src IP Addr:Port			Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2011-12-27 14:59:51.000	0.000	TCP	SOH	[illegible]:80	->	SOH	[illegible]:61414	.A...F	0	1	54	0	0	54	1
2011-12-27 14:58:33.000	59.000	TCP	SOH	[illegible]:80	->	SOH	[illegible]:32812	.AP...	0	6	2236	0	303	372	1
2011-12-27 14:59:44.000	5.000	TCP	SOH	[illegible]:443	->	SOH	[illegible]:64067	.AP...	164	13	13144	2	21030	1011	1
2011-12-27 14:59:52.000	0.000	TCP	SOH	[illegible]:80	->	SOH	[illegible]:53103	.AP...	0	1	200	0	0	200	1
2011-12-27 14:59:57.000	0.000	TCP	SOH	[illegible]:80	->	SOH	[illegible]:46756	.AP...	0	1	522	0	0	522	1
2011-12-27 14:56:04.000	208.000	TCP	SOH	[illegible]:80	->	SOH	[illegible]:61787	.A....	164	475	719150	2	27659	1514	1
2011-12-27 14:59:54.000	0.000	TCP	SOH	[illegible].248.125.23:80	->	SOH	[illegible]:58525	.A....	0	1	1514	0	0	1514	1
2011-12-27 14:59:32.000	0.000	UDP	SOH	[illegible]:25165	->	SOH	[illegible]:53	......	0	1	89	0	0	89	1

Packet-level data

- Most detailed data
- Challenging to collect and store at scale



Exploring Packet-level Data

- Clone on your laptop: https://github.com/tarunmangla/col867-holi25/tree/master
- Open network-data-exploration.ipynb in jupyter-lab
- **Exercise**: Find the speed from the network trace