Self Programming Networks
Is it possible for to “Learn” the control planes of networks and applications?

Operators specify what they want, and the system learns how to deliver
CAN WE LEARN THE CONTROL PLANE OF THE NETWORK?

Control & Orchestration Plane automatically synthesized by learning from telemetry data

Northbound Application Interfaces

Southbound Infrastructure Interfaces

Operator Policies & Intents & SLA

Real-time Network Telemetry

Applications with Access to Network Control

Infrastructure Control Points

You specify what you want, and the network figures out how to deliver it!
Classical Machine Learning

1. Build Rules-based Algorithm / Model
2. Deploy
3. Evaluate Results
4. Tune / Update Algorithm / Model

Deep Reinforcement Learning

1. Learning Agent
2. Environment
3. State
4. Reward
5. Actions

Application and Network state variables

Reward values to achieve desired outcomes

Controllable actions

Millions of Training Cycles per hour
DEEP RL IS A GREAT FIT FOR NETWORK CONTROL

- **LEARN DIRECTLY FROM EXPERIENCE**
  - No models, brittle assumptions
  - Policies adapt to actual environment & workload

- **OPTIMIZE A VARIETY OF OBJECTIVES END-TO-END**
  - Provide “raw” observations of network conditions
  - Express goals through reward
  - System does the rest

- **NETWORK CONTROL DECISIONS ARE OFTEN HIGHLY REPETITIVE**

- **LOTS OF TRAINING DATA → SAMPLE COMPLEXITY LESS OF A CONCERN**
**INFERENCES**

- Understand how known network input variables affect network KPI (e.g. throughput)
- Leverage traditional **supervised** machine learning and statistical analysis (random forest, etc.) to develop function \( f(X_{t+1}) \) to be used as input for deep learning agent

\[
X_t = f(a, b, c)
\]

- **X** = Average Throughput per user, per cell
  - Current and past **collisions** per cell \( a \)
  - Current and past **number of users** per cell \( b \)
  - Current and past **signal strength** per user, per cell \( c \)
Network State Variables

Extracting features from Training Data (offline)

\(a, b, c\)

Input Features from realtime data feed

Prediction Neural Network (real-time)

\(X_{t+1}\)

Throughput KPI Prediction
CONTROL

Inference → Prediction → Control

Prediction
Neural Network (real-time)

Deep Learning (unsupervised, reinforced learning)

Network State Variables

Operator Policies
Client Conditions
App specific inputs

Control Decision
USE CASE: VIDEO STREAMING

Network Assisted Mobile Video Streaming

Optimization Objective:
Maximize video quality and Minimize Stall Time in challenging network scenarios

Policy:
Maintain minimum average throughput of 500Kbps per user for non-video traffic

Data:
Real-time Network State Data Feed
Real-time Mobile Application State Data

Control:
Next Chunk Adaptive Bit Rate (ABR)

Result:
50-75% Higher Video Quality
75%-100% Lower Stall Time
Video QoE depends on user throughput.

What does user throughput depend on?

User performance can be modeled fairly accurately using features extracted from real time network data feeds.
PREDICT

Predicting the network state variables and then throughput

Test for seasonality and trend, learning ARIMA using a neural network
WHAT BEHAVIOR DOES THE RL AGENT FOR CONTROLLING VIDEO ABR LEARN?

ON DEMAND VIDEO (buffer limit up to 4 minutes)

Build a safe buffer, then safely play the highest quality without stalling

LIVE VIDEO (buffer limit less than 30s)

Track throughput predictions to play as high quality as possible without stalling

Network predictions become more valuable as buffer limit grows smaller
USE CASE: CONTENT PREPOSITIONING

Content Pre-Positioning

Optimization Objective:
Maximize content downloads, while maintaining operator defined minimum throughput requirements

Policy:
Maintain minimum average throughput of 500Kbps per user

Data:
Real-time Network State Data Feed

Control:
When and how much data to download

Result:
More than double data downloaded on existing infrastructure with deterministic impact on other subscribers
USE CASE: MOBILE NETWORK LOAD BALANCING

RAN Load Balancing

**Optimization Objective:**
In cases where signal strength is sufficient on two or more cells, manage connectivity to balance load across available cells.

**Policy:**
Maintain minimum average throughput of 500Kbps per user

**Data:**
Real-time Network State Data Feed

**Control:**
When and which user to handoff to neighboring cell?
CONCLUSION & LESSONS LEARNED

• Network control is a “Big Control” problem
  • Learning techniques can lead to automated control and intent based system design

• Infer + Predict + Control is a general framework
  • Applies to systems beyond mobile networks

• Streaming analytics pipeline latencies are hard to predict right now \(\rightarrow\) Direct implication on hardness of prediction

• Analytics pipeline system performance requires careful manual tuning
  • Hard to iterate and keep performance predictable

• Online learning is still an unsolved problem, both in learning techniques as well as system design