Data Driven 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

Applications with Access to Network Control

Operator Policies & Intents & SLA

Real-time Network Telemetry

You specify what you want, and the network figures out how to deliver it!
DATA DRIVEN CONTROL LOOP

Operator Policies
Intent & SLAs

Real-time
Data

Network Telemetry

Service KPI

Prediction

Will Current Controls Meet KPI?

Y

N

Will Adjusted Controls Meet KPI?

Y

N

Meets Operator Policies?

Y

N

Apply Controls

Recommend Current Control Settings

Recommend Adjusted Control Settings

Apply Controls

Recommend Current Control Settings

Recommend Adjusted Control Settings

What-If Adjust Control Knobs

Meets Operator Policies?
TWO LEARNING APPROACHES:

Classical Machine Learning

- Requires extensive domain knowledge

Diagram:

- Operator Policies, Intent & SLAs
- Service KPI
- Real-time Data
- Network Telemetry
- Prediction
- Will Current Controls Meet KPI? Y N
- What-if Adjust Control Settings? Y N
- Apply Controls
- Recommend Current Control Settings
- Recommended Adjusted Control Settings
- Monitor Operator Performance

Deep Reinforcement Learning

- Largely blind, truly approximates intent based system design

Diagram:

- Operator Policies, Intent & SLAs
- Service KPI
- Real-time Data
- Network Telemetry
- Reward Function
- Neural Network
- Reinforcement Learning (Prediction & What-if)
- Recommend Current Control Settings
- Recommend Adjusted Control Settings
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
   - State
   - Reward
   - 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**
LEARNING & CONTROL PIPELINE

Inference

Machine Learning

Deep Learning

Prediction

Feature Extraction

Control

Neural Networks
**INFEERENCE**

**Inference**

- 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

$X = \text{Average Throughput per user, per cell}$

- 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 ($c$)

$X_t = f(a, b, c)$
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

Network State Variables ➔ Prediction Neural Network (real-time) ➔ Control Decision

- Deep Learning (unsupervised, reinforced learning)
- Control

- Operator Policies
- Client Conditions
- App specific inputs
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
SYSTEM ARCHITECTURE

Video Stream

ABR Request

LTE

Real-time Network Conditions

Network Assisted ABR

Video ABR recommendation for next chunk

ABR guidance request

Video Encoded for ABR Streaming
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
Predict + Control

Video buffer states

A3C ABR

LSTM thpt forecaster

Past b_app ts

Past b_seg ts

RSRP/RSRQ ts

CELT ts

conv1d

relu

relu

linear

thpt prediction

softmax

ABR Policy

Hidden layers

inputs
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
WHY FINE-GRAINED NETWORK PREDICTION MATTERS FOR QOE

1. No context awareness at the start. Huge startup time!

2. Cell congestion can spike transiently (~10s of seconds). Huge stalls!
Downlink interference can spike transiently (~10s of seconds). Huge stalls!

WHY FINE-GRAINED NETWORK PREDICTION MATTERS FOR QOE
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

**Result:**
More than double data downloaded on existing infrastructure with deterministic impact on other subscribers

**Control:**
When and how much data to download
**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?
PLATFORM ARCHITECTURE – DEEP LEARNING ENGINE

Dynamic Network Conditions

- CTR/CTUM Feed
- Over-the-Air

Uhana Managed Service (Cloud Infrastructure)

Real-time Per-User Fine-Grained Monitoring

Deep Learning Engine

- App Specific Neural Networks

Fine Grained Mobile Network Visibility

Operator App Specific Intents (Desired Outcomes)

Operator Policies (Constraints)

Varied based on:
- User, Subscription Level, Device Type, Content Type, etc.

Application Control Points

API

- App Control
- App Control

Over-the-Air
How will network metrics look in the future?

**INFER**
- What network metrics affect app QoE metrics?
- e.g., avg bitrate, stalls, switches

**PREDICT**
- How will network metrics look in the future?
- e.g., link quality, cell congestion etc.

**CONTROL**
- What should be the app control for the best app QoE?
- e.g., ABR for the next video chunk
- e.g., buffer, app throughput

**THE INFER-PREDICT-CONTROL PRIMITIVES**
PLATFORM FEEDBACK: PROCESSING LATENCY Dictates Prediction Requirements

Lower the latency of processing, lower the requirement on prediction horizon (easier the prediction)
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