An Intelligent Approach to Zero Touch Edge & Cloud networking

ML-powered Zero-Touch approach to Optimised Service placement and Resource Allocation

Dr Xenofon Vasilakos,
Department of Electrical & Electronic Engineering, School of Computer Science, Electrical and Electronic Engineering, and Engineering Mathematics of the University of Bristol, UK
Problem and Motivation

Networks and their services are highly complex systems

• Many different entities: software modules and physical or virtualised devices
• Legacy administration:
  • Active human involvement + policy-based management has been vital to simplifying, (semi-)automating administration

Things changed:

• Mid-2000s and on: Cloud Computing, Fog and Edge computing
• Game changer: Software Defined Networking (SDN) and Network Function Virtualisation (NFV)
• Radically changed the needs of Network and Service Management and Orchestration (MANO).
Problem and Motivation

Radically different MANO needs:
✓ Special-purpose hardware equipment (e.g. routers or firewalls) replaced by fully softwarised Virtual Network Functions (VNFs) running over sharable general-purpose hardware.
✓ Reduce Capital expenditures (CAPEX) and Operating expenses (OPEX), and (in part) to avoid dependencies on vendor hardware.
✗ Resource allocation and maintenance complexity

MANO complexity & difficulties
• Fine-tunned allocation of sharable resources even within the same hardware implies multi-objective optimisation problems
  • generally hard to address, NP-Hard [7,8,10]
  • Even worse: instantiation and cooperation of multiple VNF instances over locations (service chaining)
• 5G/6G era - Unprecedented operational agility:
  • (i) personalised user needs posing their own demand pattern
  • (ii) a high dynamicity (e.g., intense user & service mobility, unexpected traffic & volumes)
  • (iii) within very short timescales (10-50ms) (e.g., online mobile video gaming [ZeroDEAM]).
Extreme range of requirements

➢ Seemingly infinite resource capacity
➢ Imperceptible latency
➢ Ultra-high reliability
➢ Personalized services (customer-experience)
➢ Global web-scale reach
➢ Massive machine-to-machine communication
Zero touch network & Service Management
ETSI: https://www.etsi.org/technologies/zero-touch-network-service-management
Outline

1. The M-Plane: Macroscopic autonomous VNF placement

2. The μ-Plane: microscopic monitoring and autonomous profiling

3. Extensions towards an Integrated ML-Powered ZSM Framework
   • Fitting everything together
   • Future work
M-Plane - Macroplane

Autonomous VNF placement


M-Plane - Macroplane

We approach VNF placement “macroscopically”
• consider **e2e service-level performance statistics** and predictions after placement actions
• macroscopic monitoring input to **distributed ML models** rather than traditional system-local resource availability.

What we’ve done:
• validated and evaluated over a real testbed and use case [a]
• a comprehensive comparative study of different VNF ML prediction models
  • Supervised Learning (SL)
  • Online learning with Reinforcement Learning (RL)
  • Hierarchical adaptation of RL for multi-objective optimization purposes.

[a] 5gINFIRE Sginfire.eu, University of Bristol 5G Testbed – 5GinFIRE, 2018-2020, Available online: https://5ginfire.eu/university-of-bristol-5g-testbed/
M-Plane - Macroplane

Testbed and Use Case

- Sensor: 360° camera model Ricoh Theta V.
- Source Node: Raspi model 3B running Raspbian Jessi.
- Compute/Controller Node 1: CORSAIR ONE PRO, INTEL I7-7700K, 8-core processor, 16GB RAM, 800GB HD, running Ubuntu 16.04, KVM, Openstack Queen (Controller and Compute), OSM MANO release 4.
- CN 2: Same as Node 1, except from: 80GB HD, Openstack Queen (Compute).
- CN 3 & CN 4: IBM x3455, AMD Opteron, 4-core processor, 8GB RAM, 70GB HD, running Ubuntu 16.04, KVM, Openstack Queen (Compute).
- VNF video transcoder: Virtual Display Unit (VDU) of a 2-core processor, 3GB RAM, 4GB HD, running Ubuntu 16.04.

[a] 5gINFIRE 5ginfire.eu, University of Bristol 5G Testbed – 5GinFIRE, 2018-2020, Available online: https://5ginfire.eu/university-of-bristol-5g-testbed/
M-Plane - Macroplane

Management network

Provider network

End user(s)

Rasbi source(s)

VNF monitoring agent

Node monitoring agent

Placement agent

Prediction agent

M-Plane
Supervised Learning models

- Decision Tree (DT),
- Random Forest (RF),
- Linear Regression (LR),
- Support Vector Regression (SVR),
- Kernel Ridge Regression (KRR),
- Lasso Regression (LAR),
- and K-Nearest Neighbors Regression (K-NNR);

Vs.

Time Series Forecasting (TSF) benchmark models:
- AutoRegression (AR);
- Exponential Smoothing (ES); and
- AutoRegressive Moving Average (ARMA).

Conclusion 1: ML intelligence matters for VNF MANO decisions.

Adopted Reinforcement Learning

- Adding intelligence to OSM MANO with SL models proved beneficial
- Next, we investigated online learning with adopted RL e2e service-level (Macroscopic) performance predictions
  - we assume an underlying Markov Decision Process, justifying our adoption of Q-learning

Conclusion 2: Resilience to dynamics & portability with RL
Hierarchical RL (HRL)

We try to capture

✓ different VNF types posing different and possibly opposing objectives
✓ allowing tunable (scalarized) Q-learning balancing of objectives.

1) LRL. A local predictions level comprising a Local RL (LRL) module per each candidate placement host, accurately predicting (i) e2e service-level delay or (ii) a CPU Load prediction model, or (iii) both for the case of multi-objective optimisation.

2) GRL. A single Global RL (GRL) module lies on top of LRLs. It captures system-wide dynamics by being continuously fed with delay and CPU Load predictions by all LRLs.
Hierarchical RL (HRL)

Conclusion 3: HRL allows tunable & distributed orchestration targeting multiple optimisation objectives.

Conclusion 4: HRL addresses both highly dynamic and static conditions.
μ-Plane - Microplane

Microscopic monitoring and autonomous profiling

μ-Plane - Microplane

We approach VNF placement “microscopically”

• Unlike the macroplane, ML models utilise system-local (hence, “microscopic”) monitoring input on resource and local VNF performance

• Powered by ML, capture the interplay and optimal trade-offs between
  • (i) VNF performance estimates or goals
  • (ii) the needed resources in terms of amount combinations of CPU, Memory, and Network.

Discover the Optimum Maximum Input Rate (Optimum MIR), i.e. the maximum traffic load handled by a VNF with a given resources while meeting with KPI goals.

[a] 5GIFIRE Sgifire.eu, University of Bristol 5G Testbed – 5GInfIRE, 2018-2020, Available online: https://5gfire.eu/university-of-bristol-5g-testbed/
(1) **Weighted Resource Configuration Selector**: produces the “VNF Performance Dataset”. Requests for traffic by Traffic Generator to the VNF instance.

(2) **Analyser and Post Processor.**
   - (1) passes to Weighted Resource Configuration Selector monitoring data. This helps to discover the Optimum MIR
   - (2) passes preprocessed data (sanitized, extracted features) to a data lake such as ElasticSearch

(3) **Predictor Manager**. The ML profile extraction process.
Reveals a dual profile relationship among
   - (1) possible resource configurations,
   - (2) service demand and
   - (3) performance targets.

Predictions can be fed directly to MANO or an intermediate submodule used to translate these predictions to suggested actions to MANO.
Extensions towards an Integrated ML-Powered ZSM Framework

Fitting everything together

Future work
Fitting everything together in a framework

(a) \(\mu\)-PLANE

- KPI Targets / SLA
- Profiling Parameters
- VNF Descriptor

(1) Weighted Resource Selector

VNF Performance Dataset

Traffic Generator

VNF Monitoring agent

Node monitoring agent

Placement agent

Prediction agent

Management network

Provider network

(b) \(M\)-PLANE

- KPI predictions
- Online eternal training
- Host/VNF predictions

Rasbi source(s)

End user(s)

VNF monitoring agent

Node monitoring agent

Placement agent

Prediction agent

VNF Traffic Generator

Records

Training data

NAP Profiler

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Performance Dataset

VM or Container host

NAP Profiler

VNF

(2) Analyser & Post Processor

(3) Predictor Manager

KPI Targets / SLA

Profiling Parameters

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Descriptor

KPI Targets / SLA

Profiling Parameters

VNF Descriptor
Future work

1. Use RL for profiling
   • Address more objectives for optimisation
   • Train in a sandpit, then continuous learning eternally online
   • Portable ML profiling to other environments
   • Resilient to changes

2. Move from RL to Deep RL (DRL) for orchestration
   • Use actor-critic approach [6]

3. Two integration scenarios for the planes:
   1. LRL models and profiling models cooperate
   2. LRL models adopt (replaced by) profiling ML approach

4. Profiling + congestion pricing [9]
   • Use profiler to compute performance + resource configurations
   • Use congestion pricing to valuate resource configurations
   • Assign resource based on a cost/benefit approach
   • → has extensions to auctioning and slice creation/maintenance

---


Thank you
References


References


