NetAIgym: Democratizing “Network AI” Research & Development via Simulation-as-a-Service

Jing Zhu and Menglei Zhang

Intel Labs, June 2023

Contributor: Pinyarash Pinyoanuntapong (UNCC)
University Collaborators: Kun Yang (UVA) and Momin Haider (UCSB)
Acknowledged: Shu-ping Yeh
Motivation: Where is “Data”?

- Network AI Developer Challenges (Why NetAIGym?)
  - real-world dataset controlled by network operator, difficult to acquire, not aligned with specific usage or requirement
  - “dataset” by itself not enough, also need “environment” to train/test AI models, e.g. Reinforcement Learning, etc.
  - network simulation tools (e.g. ns3, etc.) often very complex and difficult to use, especially for Network AI researcher & developer

Gap: lack of common “Simulation” environment with simple “APIs” to develop, evaluate, and benchmark “Network AI” models & algos
Related Work: OpenAI Gym / Gymnasium

Gymnasium is a standard API for reinforcement learning, and a diverse collection of reference environments.

Gymnasium is a maintained fork of OpenAI's Gym library. The Gymnasium interface is simple, pythonic, and capable of representing general RL problems, and has a compatibility wrapper for old Gym environments.

```python
import gymnasium as gym
env = gym.make('LunarLander-v2', render_mode='human')
observation, info = env.reset(seed=42)
for _ in range(1000):
    action = env.action_space.sample()  # this is where you would insert your policy
    observation, reward, terminated, truncated, info = env.step(action)
    if terminated or truncated:
        observation, info = env.reset()
env.close()
```

Source: https://gymnasium.farama.org/

“Environment” is as important as “Data” (if not more) for Network AI R&D.
NetAI Gym: An Open “Network AI” Simulation-aaS Framework

Network Intelligence Controller (Algorithm)
(Traffic Management, RAN slicing, Energy Saving, etc.)

Rule/Policy based Models/Algos
Reference ML/AI Models/Algos
3rd party ML/AI Models/Algos

Configuration (use-case, topology, traffic, etc.)
Action
Observation/Data (State, Reward)

① Open Toolkit (Library, Models, Examples, etc.) for Network AI algorithm development
- NetAI Client

② Open API for collecting data and interacting with the simulated network environment
- NetAI Gym API

③ Open Network Simulator
Leverage open-source network simulation tools, e.g. ns3; Enhance it with customized capabilities & use-cases, e.g. Traffic Steering, Network Slicing, Distributed Compute, Dynamic QoS, Energy Saving, etc.
- NetAI Server
- NetAI Sim
NetAI Gym PoC: “Sim-aaS” E2E Infrastructure via vLab

NetAI Client 1
AI Agent
Stable-baselines3 RL model

NetAI Client 2
AI Agent
User customizable RL model (CleanRL)

NetAI Client 3
None-AI Agent
System default algorithm

NetAI Client 4
None-AI Agent
User customizable algorithm

NetAI Gym Open API

NetAI Server
Internal API

ZeroMQ Socket
https://zeromq.org/

Env 1
NetAISim
https://www.nsnam.org/

Env 2
NetAISim

Env 3
NetAISim

Env 4
Other Sim

vLab @ Intel Labs

NetAIClient 5
...
...

NetAIClient Database
WanDB
https://wandb.ai/

NetAIClient Repo: https://github.com/pinyaras/GMAClient

AI Agent
User customizable

Env 2
NetAISim

AI Agent
System default

Env 3
NetAISim

AI Agent
User customizable

Env 4
Other Sim

Env 1
NetAISim
https://www.nsnam.org/
NetAI-Gym Sim-aaS E2E Workflow

1) **NetAI-Client** sends a JSON configure file to **NetAI-Server** to launch a ns-3 simulation.

2) **NetAI-Sim** collects and sends measurement metrics to **NetAI-Server**.

3) **Algorithm Agent** computes an action based on the measurements, and stores data in **WandB**.

4) **NetAI-Client** transmits the action to the **NetAI-Sim** via the **NetAI-Gym Open API**.

5) The measurement history can be visualized via the **Web-based WandB**.
## A List of Supported Measurement Metrics

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_rate</td>
<td>mbps</td>
<td>LTE/Wi-Fi link capacity measured by each user</td>
</tr>
<tr>
<td>load</td>
<td>mbps</td>
<td>input traffic throughput measured by each user</td>
</tr>
<tr>
<td>rate</td>
<td>mbps</td>
<td>output traffic throughput measured by each user, including LTE, Wi-Fi, and ALL</td>
</tr>
<tr>
<td>qos_rate</td>
<td>mbps</td>
<td>output traffic throughput that meets the QoS requirement, including LTE, Wi-Fi, and ALL</td>
</tr>
<tr>
<td>owd</td>
<td>ms</td>
<td>one-way delay measured by each user, including LTE, Wi-Fi, and ALL</td>
</tr>
<tr>
<td>tsu</td>
<td></td>
<td>traffic split ratio measured by each user, including LTE, Wi-Fi</td>
</tr>
<tr>
<td>ap_id</td>
<td></td>
<td>access-point/cell ID measured by each user, including LTE and Wi-Fi</td>
</tr>
<tr>
<td>slice_id</td>
<td></td>
<td>(LTE) slice ID measured by each user</td>
</tr>
<tr>
<td>rb_usage</td>
<td>%</td>
<td>(LTE) resource block usage measured by each user</td>
</tr>
<tr>
<td>delay_violation</td>
<td>%</td>
<td>one-way delay violation percentage (%) measured by each user</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A List of Supported Use Cases and Actions

<table>
<thead>
<tr>
<th>Use-Case</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Access (MX) Traffic Splitting</td>
<td>Traffic Splitting</td>
<td>update traffic split ratio of a flow over Wi-Fi and LTE</td>
</tr>
<tr>
<td>QoS-aware MX Traffic Steering</td>
<td>Traffic Steering</td>
<td>steer traffic over Wi-Fi or LTE for a flow</td>
</tr>
<tr>
<td>Cellular RAN Slicing</td>
<td>Resource Allocation</td>
<td>update LTE resource block allocation ratio for a slice</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
NetAI Gym Example: Multi-Access (MX) Traffic Splitting
Multi-Access Virtualization Framework: GMA

Apps
- IoT App (TSN, etc.)
- eMBB App
- URLLC App
- mMTC App

Edge
- Virtual Interface (IP, Ethernet)
- Generic Multi-Access (GMA) convergence

Access
- Programmable Data-Plane (HQoS, INT, AQM, etc.)
- Cellular (LTE, 5G)
- Wi-Fi
- Radio X

Traffic Management API [1]:
app/flow/pkt QoS requirements (delay, latency, loss, etc.)

Radio Network Info API [2] [3]:
link quality / measurement info, queue status, load/utilization, etc.

Seamless Information Exchange across layers (Apps, Edge, Access)

Multi-Access Traffic Management (Control-Plane)

Enable Multi-Access Convergence over ANY Access for ANY Apps

INT: In-band Network Telemetry
AQM: Active Queue Management
HQoS: Hierarchical Quality of Service

[2]: ETSI/MEC RNIS API (https://www.etsi.org/deliver/etsi_gs/MEC/001_099/012/02.01.01_60/gs_mec012v020101p.pdf)
[3]: ETSI/MEC WLAN API (https://www.etsi.org/deliver/etsi_gs/MEC/001_099/028/02.01.01_60/gs_MEC028v020101p.pdf)
GMA 1.0 Network Reference Architecture

Virtual Connection to Integrate Multiple Physical Connections

IETF RFC & Drafts on GMA Framework & Protocols:

GMA 1.0 Software Release:
- GMA client: https://github.com/IntelLabs/gma
Traffic Splitting Scenario Config

- 1 LTE Cell: 5MHz(UL) + 5MHz(DL)
- 2 Wi-Fi APs(11ac): 20MHz + 20MHz
- Downlink traffic: TCP Cubic
- UE Number: 4
  - with random deployment
- UE Speed: 1m/s (left and right)
- Evaluation Metrics: Throughput, Delay
  - Baseline: GMA
  - Online RL algorithm: PPO and DDPG
Per User Throughput Comparison

**Goal**: maximize throughput and minimize delay

- **Reward**: throughput - delay.
- **States**: LTE and Wi-Fi Max rate

![Graphs showing throughput comparison for UE0_rate, UE1_rate, UE2_rate, and UE3_rate for DDPG, PPO, and GMA.](image-url)
Total Throughput and Delay Comparison

**Goal:** maximize throughput and minimize delay

➢ **Reward:** throughput - delay.

➢ **States:** LTE and Wi-Fi Max rate
Summary

- NetAIGym – an open “Network AI” Simulation-as-a-Service framework
  - NetAISim: ns3-based network simulator with enhanced capabilities, e.g. multi-access, RAN slicing, etc.
  - NetAlServer: the NetAIGym server application software to manage connection and interaction between an NetAIGym client and the NetAISim worker
  - NetAIClient + API: the NetAIGym client application software to configure the simulation and run the “Network AI” algorithms together with the simulation through open API

- A PoC/Trial system available for experiment, support three use-cases: multi-access traffic splitting, QoS-aware traffic steering, and (cellular) RAN slicing
  - limited access available upon request

- How to collaborate and contribute?
  - NetAISim: ns3 modules for new use-cases or capabilities
  - NetAIClient: AI algorithms & models for the existing use-cases

“NetAIGym” is a Use-Case driven “Network AI” Sim-aaS framework, Open for Contributions from the Community