Machine Learning Driven UAV-assisted Edge Computing

Liang Zhang[1], Bijan Jabbari[1], Nirwan Ansari[2]

lzhang36@gmu.edu

1. Communications and Networks Laboratory,

Department of ECE, George Mason University, Fairfax, VA 22030 USA

2. Advanced Networking Laboratory,

Department of ECE, New Jersey Institute of Technology, Newark, NJ 07102 USA

Outline

UAV-aided MEC for 6G

- Billions of Internet of Things (IoT) devices are connected to the Internet, and many IoT devices have limited power, computing and storage resources.
- Mobile edge computing (*MEC*) is proposed to provide computing and storage services to these IoT devices.
- UAV-aided MEC has been used to improve the performance of MEC, which has high mobility, high maneuverability and flexibility of deployment.
- 6G will bring a full-fledged framework for connected things and automation systems from autonomous cars to UAVs with high reliability, low latency, high data rate, and high energy efficiency.
- The number of deployed UAVs will increase to 1.6 millions by 2024 according to Federal Aviation Administration (FAA).

Source: Mohammad Mozaffari, Xingqin Lin, Stephen Hayes. " Towards 6G with Connected Sky: UAVs and Beyond", preprint, Jul. 2021. https://doi.org/10.48550/arXiv.2103.01143

Prototypes of UAV Communications

- Nokia had developed a 4G UAV base station weighing only 2Kg in 2016, which was successfully mounted on a commercial quad-copter to provide coverage over a remote area in Scotland.

- Several projects by the industry have already been initiated, such as Project Aquila by Facebook, Cell on Wings (COW) by ATT, and Google projects such as SKYBENDER that are designed for UAV-based internet services.

Nokia and EE trial mobile base stations floating on drones to revolutionise rural 4G coverage

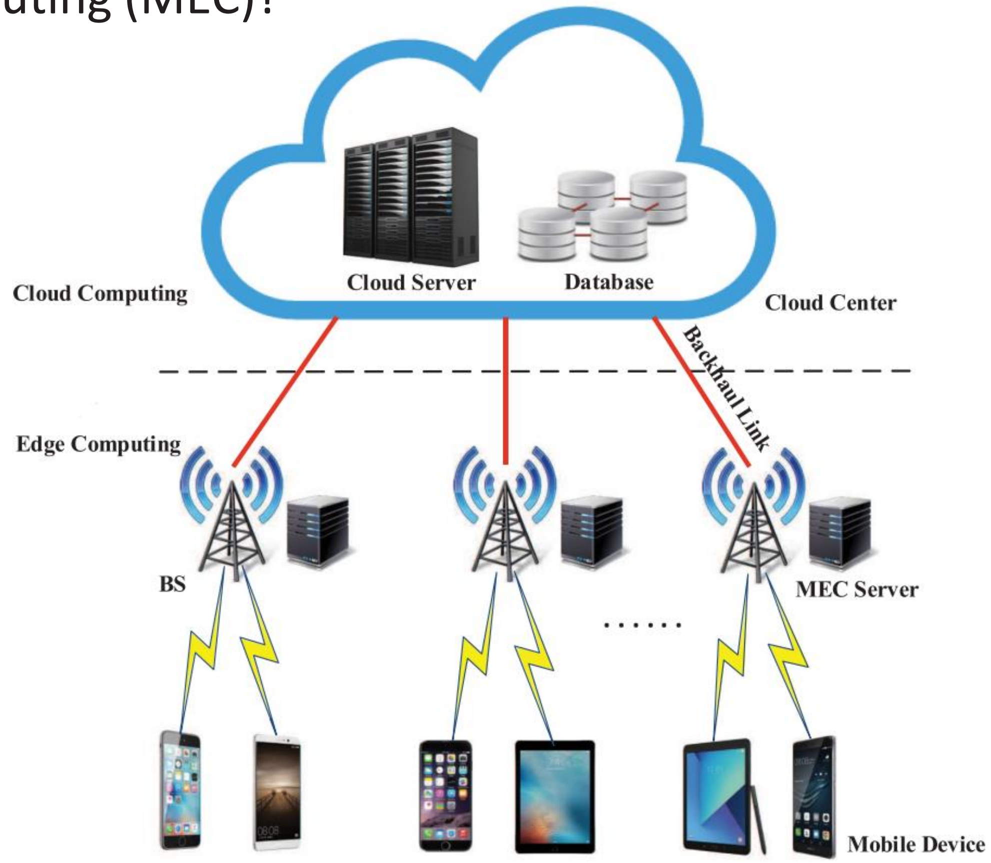
Nokia and EE test putting small cells on drones to provide temporary 4G coverage in hard-to-reach areas.

By Mary-Ann Russon
Updated August 15, 2016 11:19 BST

Source: I. B. Times, “Nokia and EE trial mobile base stations floating on drones to revolutionise rural 4G coverage,” url: https://www.ibtimes.co.uk/nokia-ee-trial-mobile-base-stations-floating-drones-revolutionise-rural-4g-coverage-1575795, 2016.
Source: D. Bharadia, E. McMilin, and S. Katti, “Full duplex radios,” in Proc. *ACM SIGCOMM*, pp. 375–386, Aug. 2013.

Why Do We Need UAV-aided Edge Computing?

- Why do we need Mobile Edge Computing (MEC)?
- Why do we need UAV-aided MEC?
- UAV-aided MEC:
  line-of-sight communications,
  high mobility, high flexibility,
  and high maneuverability.



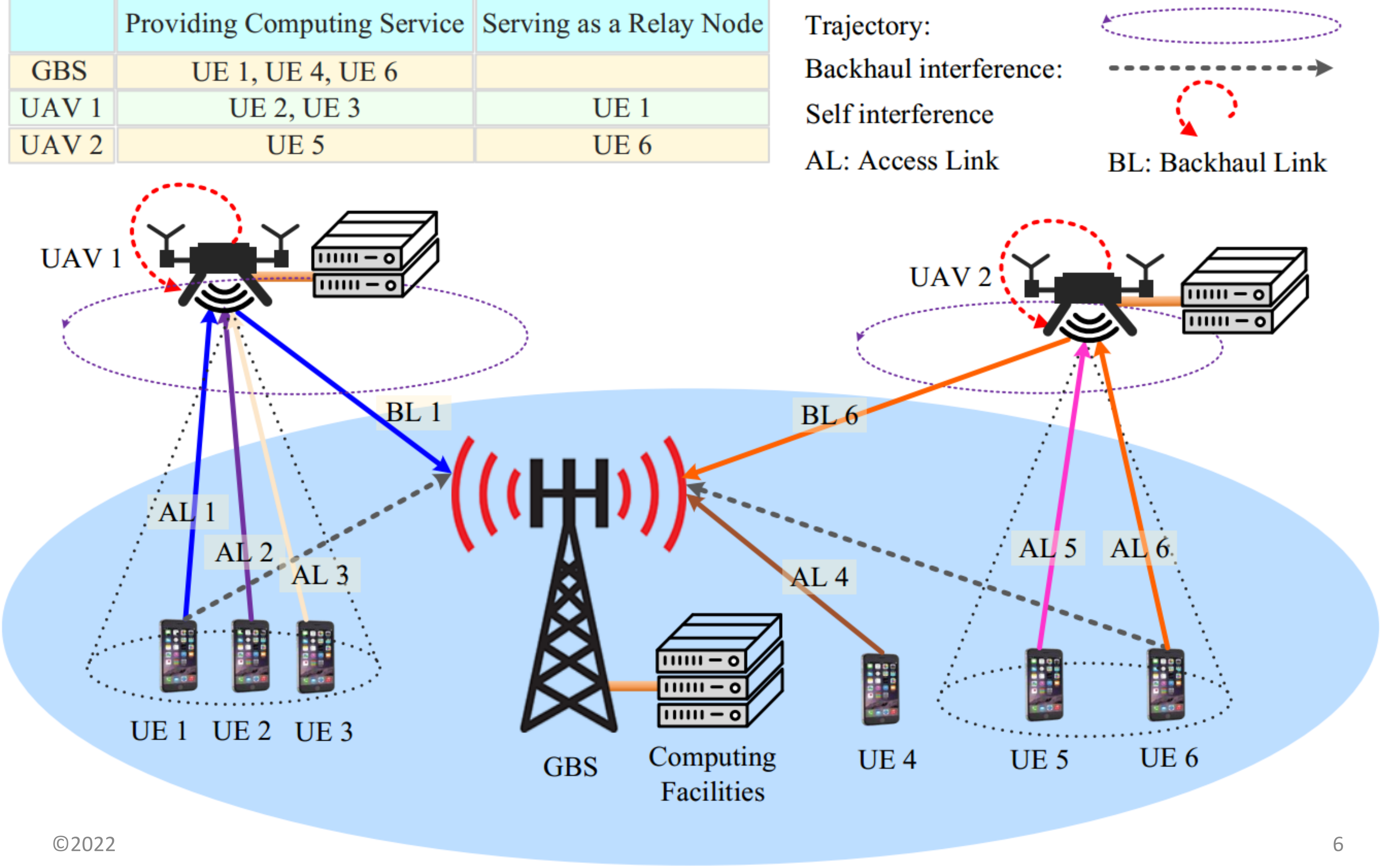
Source: J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," IEEE Trans. Veh. Technol., vol. 68, no. 5, pp. 5031–5044, May 2019.
Source: Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao and G. Y. Li, "Joint Offloading and Trajectory Design for UAV-Enabled Mobile Edge Computing Systems," in *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879-1892, Apr. 2019.

Outline

- Background of UAV-aided MEC
- System Model and Problem Formulation
- Algorithm and Analysis
- Evaluation Results
- Conclusions

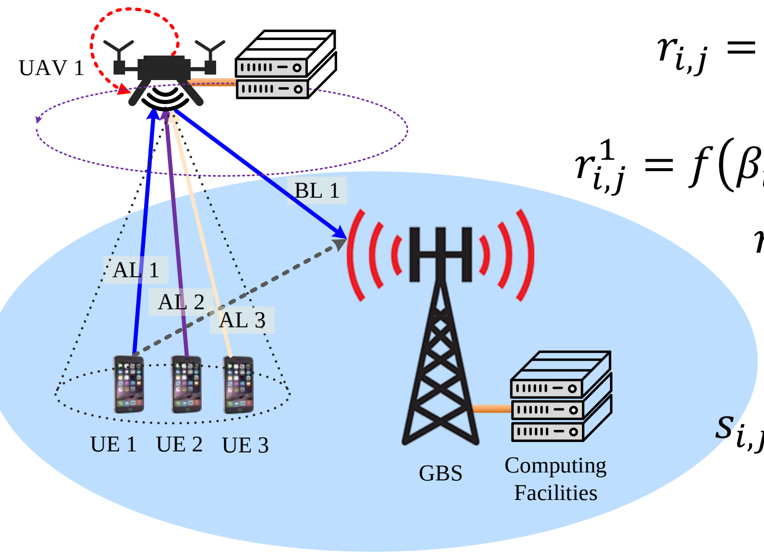
A UAV-aided Hierarchical Network for Edge Computing

	Providing Computing Service	Serving as a Relay Node
GBS	UE 1, UE 4, UE 6	
UAV 1	UE 2, UE 3	UE 1
UAV 2	UE 5	UE 6



Communication and Computing Model

- Let $s_{i,j}$, $\beta_{i,j}$ and $r_{i,j}$ be the signal to interference plus noise ratio (*SINR*), the assigned bandwidth, and the received data rate of user i towards the edge node j, respectively.



$$r_{i,j} = \begin{cases} \max\left(r_{i,j}^{1}, r_{i,j}^{2}\right), & j = 1 \\ r_{i,j}^{1}, & j > 1 \end{cases}$$

$$r_{i,j}^{1} = f\left(\beta_{i,j}, s_{i,j}\right) = \beta_0 \beta_{i,j}\, log_2(1 + s_{i,j})$$

$$r_{i,j}^{2} = \min\left(r_{i,j'}^{A}, r_{j',1}^{B}\right)$$

$$s_{i,j} = \begin{cases} \dfrac{P^E \Gamma_{i,j}^{G}}{\sigma_{i,j}^{2}}, & j = 1 \\ \dfrac{P^E \Gamma_{i,j}}{\sigma_{i,j}^{2}} & j > 1 \end{cases}$$

- Denote $\tau_{i,j}^{total}(t)$, $\tau_{i,j}^{C}(t)$ and $\tau_{i,j}^{T}(t)$ as the total delay, the computing time and the transmission delay of offloading the task from user i.

$$\tau_{i,j}^{total}(t) = \tau_{i,j}^{C}(t) + \tau_{i,j}^{T}(t)$$

Problem Formulation

- We formulate the joint **Resource Allocation** and UAV **Trajectory** planning for **Edge** computing (**RATE**) problem with the objective of maximizing the average total QoE of all users in all time slots.

$$
\begin{aligned}
\mathscr{P}_0 : \quad & \max_{\omega_{i,j}(t),\alpha_{i,j}(t),\beta_{i,j}(t),v_j(t)} \quad \frac{1}{|\mathscr{T}|}\sum_i \sum_t q_i(t) \\
s.t. : \quad & \\
& C1 : \sum_j \omega_{i,j}(t) \le 1, \quad \forall i \in \mathscr{U}, t \in \mathscr{T}, \\
& C2 : \sum_i \omega_{i,j}(t)\beta_{i,j}(t) \le \beta_j^{max}, \quad \forall j \in \mathscr{B}, \\
& C3 : \sum_i \alpha_{i,j}(t) \le C_j, \quad \forall j, \\
& C4 : \tau_{i,j}^{total}(t) \le L_i^{th}(t), \forall i, j, t, \\
& C5 : v_j \in \Theta, \quad \forall j \in \mathscr{B}, j > 1, \\
& C6 : v_j(t) \cap v_{j'}(t) = \varnothing, \quad \forall j, j' > 1, \\
& C7 : f(v_j(t), v_j(t+1)) \le v^{max}\Delta t, \quad j > 1, \\
& C8 : \omega_{i,j} \in \{0,1\}, \quad \forall i, j. \qquad (5)
\end{aligned}
$$

Outline

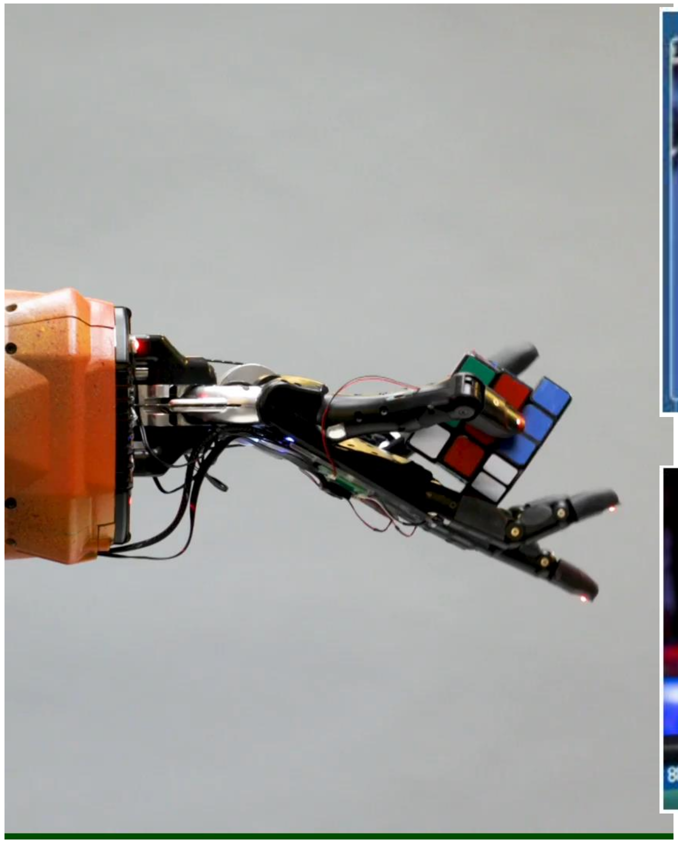
Problem Analysis

- The RATE problem is NP-hard even for one time slot because it is a non-convex, nonlinear and mixed discrete optimization problem.

- Machine learning is a convenient technique in solving the dynamic optimization problems, and we use a deep reinforcement learning method to obtain sub-optimal solutions for the RATE problem.

What is Machine Learning?

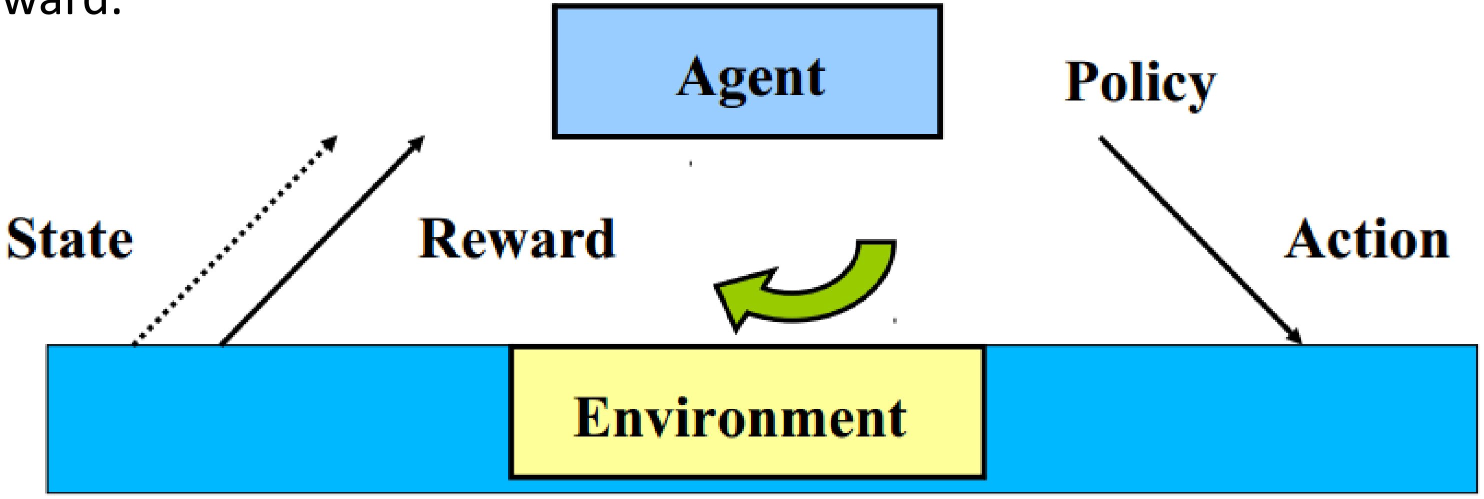
- Machine learning is a branch of artificial intelligence and computer science, which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy*.



[*] Source: https://www.ibm.com/cloud/learn/machine-learning

Elements of Reinforcement Learning

- Agent: Intelligent programs
- Environment: External condition
- A typical fully connected neural network includes three layers: the input layer, the hidden layer and the output layer.
- The output (action $a(t)$) is determined by the input (state $s(t)$), the reward ($r(t)$) is determined by the output, and the weights are updated based on the reward.



The framework of DDPG

- State: $s(t) = \{X\big(U(t)\big), Y\big(U(t)\big), D(t), W(t), V(t)\}$
- Action: $a(t) = \{X\big(V(t)\big), Y\big(V(t)\big)\}$
- Reward: $g(t) = \sum_i q_i(t)$

- **Actor network**: input is s, output is a
- **Critic network**: input is (s, a), output is Q(s,a)
- **Target actor network and target critic network:**

  input is actor network and critic network

  it is used to calculate the loss function of the critic network

The framework of DDPG

- The critic network is updated based on the loss function:

$$\begin{cases} L(\theta^Q) = \frac{1}{K}\sum_{k=1}^{K}(g(k) - Q(s(k), a(k)|\theta^Q + \gamma A_1)^2, \\ A_1 = Q(s(k+1), a(k+1)|\theta^{Q-})). \end{cases} \quad (6)$$

- The actor network is updated based on the gradient policy as follows.

$$\begin{cases} \nabla_{\theta^\varphi} J(\theta^\varphi) = \frac{1}{K}\sum_{k=1}^{K}(A_2 \cdot A_3), \\ A_2 = \nabla_a Q(s, a|\theta^Q)\big|_{s=s(k), a=\varphi(s(k))}, \\ A_3 = \nabla_{\theta^\varphi} \varphi(s|\theta^\varphi)\big|_{s=s(k)}. \end{cases} \quad (7)$$

- Then, the target networks are updated as:

$$\begin{cases} \theta^{\varphi-} \leftarrow \eta\theta^\varphi + (1-\eta)\theta^{\varphi-}, \\ \theta^{Q-} \leftarrow \eta\theta^Q + (1-\eta)\theta^{Q-}. \end{cases} \quad (8)$$

The DDPG-RATE Algorithm

```
Algorithm 1: DDPG-RATE
Input  : $\mathscr{B}$, $\mathscr{U}$, $u_i(t)$, $\varphi(s(t)|\theta^{\varphi})$, $Q(s(t),a(t)|\theta^{Q})$,
         $\varphi(s(t)|\theta^{\varphi-})$ and $Q(s(t),a(t)|\theta^{Q-})$;
Output: $q_i(t)$;
1  for time slot $t$ in $T$ do
2      Initialize the actor network $\varphi(s(t)|\theta^{\varphi})$ and the critic
         network $Q(s(t),a(t)|\theta^{Q})$ with parameters $\theta^{\varphi}$ and $\theta^{Q}$;
3      Initialize the target actor network $\varphi(s(t)|\theta^{\varphi-})$ and the
         target critic network $Q(s(t),a(t)|\theta^{Q-})$ with parameters
         $\theta^{\varphi-}=\theta^{\varphi}$ and $\theta^{Q-}=\theta^{Q}$;
4      Initialize the replay buffer $R$;
5      for epoch $e$ in $E$ do
6          Initialize $s(t)$, $a(t)$ and $g(t)$;
7          Initialize the position of all UAVs;
8          for training slot $t_1$ in $T_1$ do
9              Obtain $X(\mathscr{U}(t_1))$, $Y(\mathscr{U}(t_1))$, $D(t_1)$, $W(t_1)$, and
                 $V(t_1)$;
10             Get $s(t_1+1)$;
11             Record the sample in the replay buffer
                 $(s(t_1),a(t_1),g(t_1),s(t_1+1))$;
12             if $R(t_1) \geq R^b$ then
13                 Update $\theta^{Q}$ of the critic network by Eq. (6);
14                 Update $\theta^{\varphi}$ of the actor network by Eq. (7);
15                 Update target networks by Eq. (8);
16             Generate the action $a(t_1+1)$;
17             Add noise to action $a(t_1+1)+=a_0$;
18             Calculate $g(t_1+1)$ based on action $a(t_1+1)$;
19     Initialize $s(t)$, $a(t)$ and $g(t)$;
20     Generate action $a(t+1)$ and $g(t+1)$;
21     Calculate $v_j(t+1)$ based on $v_j(t)$ and $a(t+1)$;
22     Obtain $\omega_{i,j}(t)$ based on the least resource provision ;
23     Compute $\beta_{i,j}(t)$ and $\alpha_{i,j}(t)$;
24     calculate $q_i(t+1)$.
```

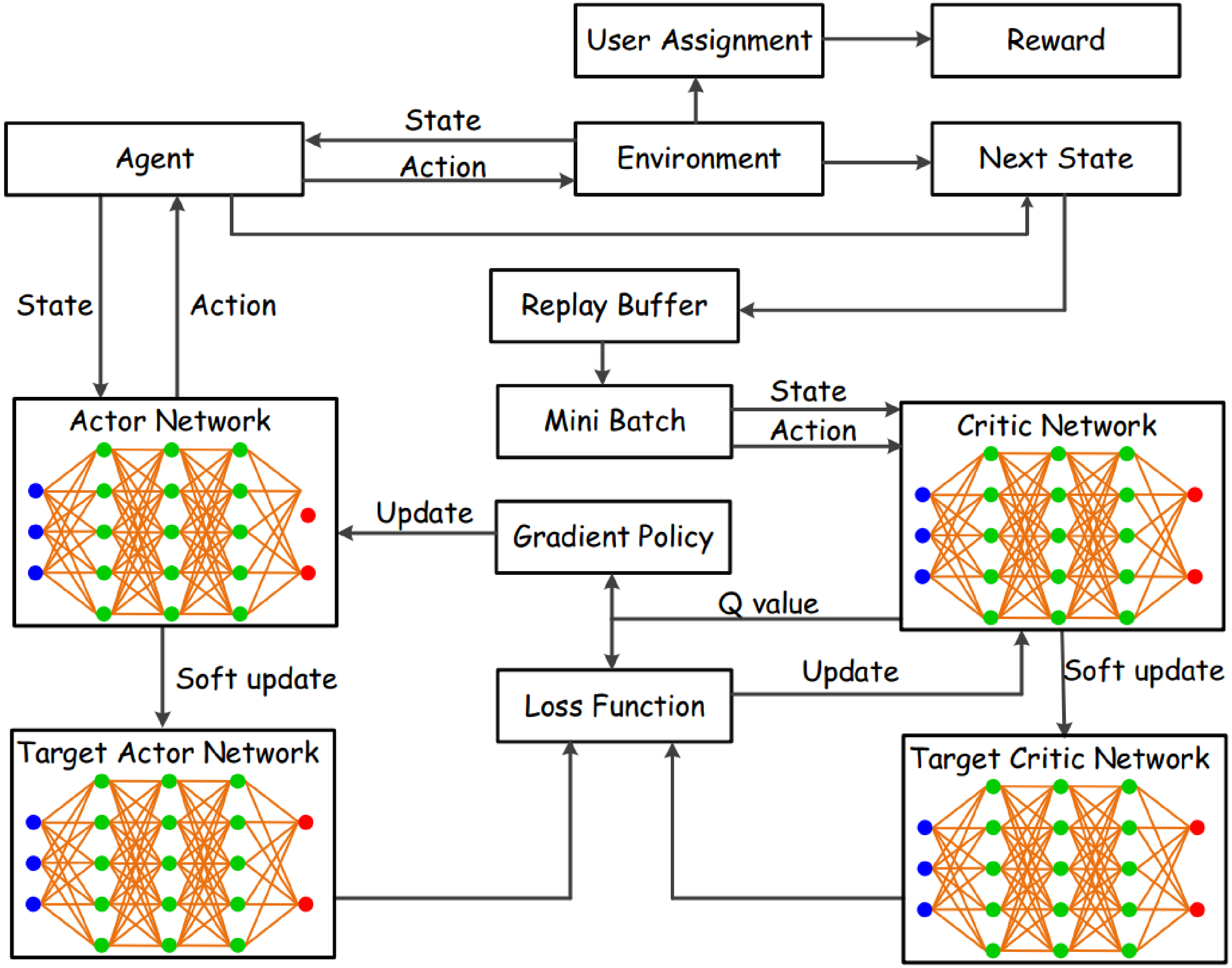
Update Critic network

Update Actor network

Update Target Actor network

Update Target Critic network

The Diagram of the DDPG-RATE Algorithm

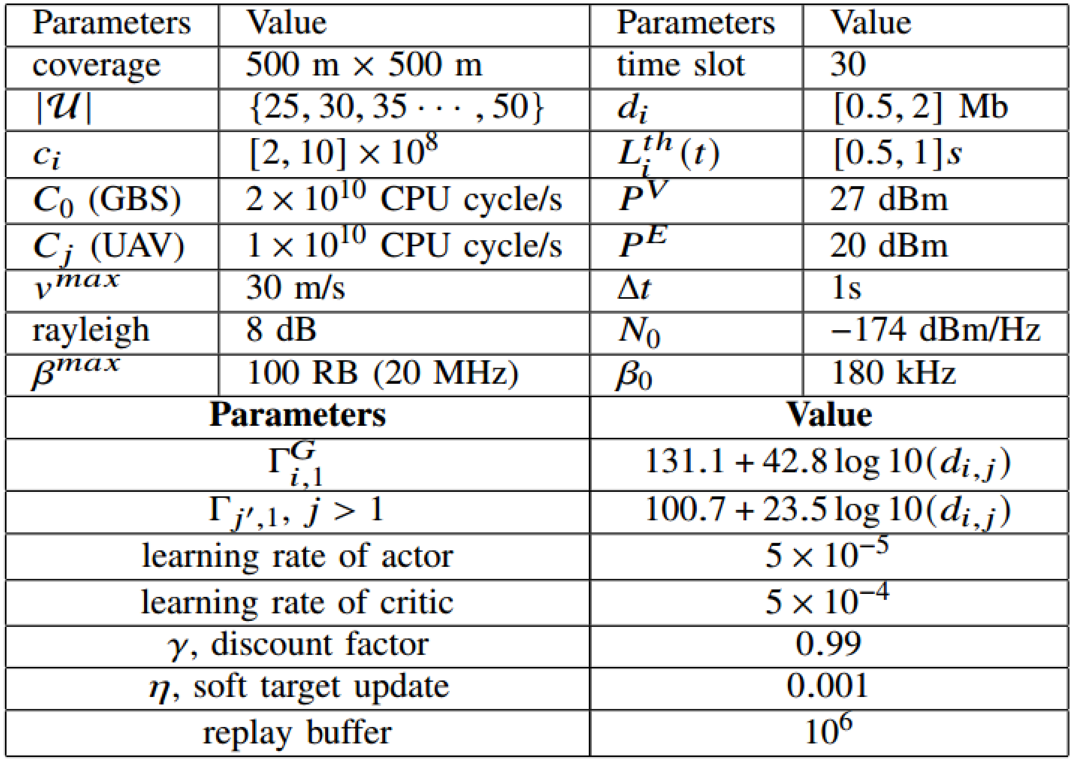


Outline

Simulation Settings

- We use Python3.7 and Pycharm 2020.3.2 to run our simulations, and utilize Tensorflow 2.4 (optimizer is tf.keras.optimizers.Adam).
- Two baseline algorithms are utilized to evaluate the performance of the DDPG-RATE algorithm: Fixed-UAV-RATE and No-UAV-RATE.

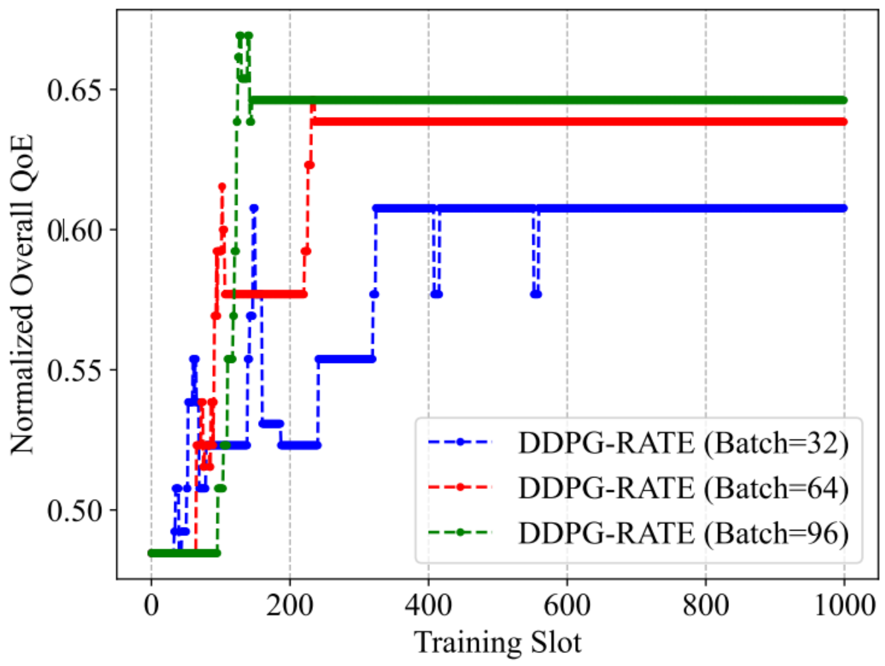
Table I: Simulation Parameters



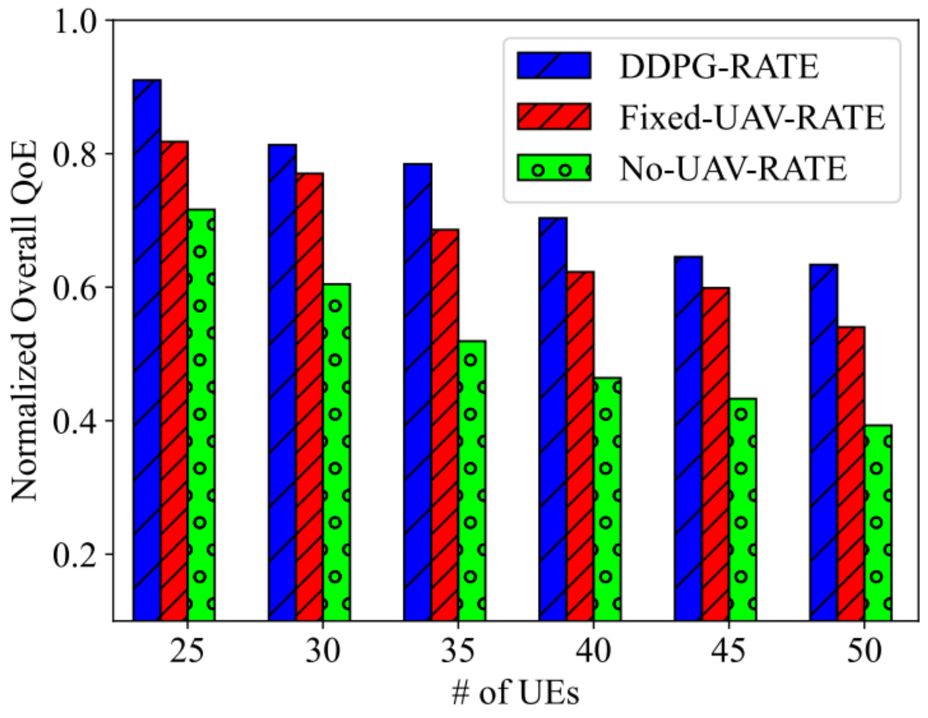
Parameters	Value	Parameters	Value
coverage	500 m $\times$ 500 m	time slot	30
$\lvert\mathcal{U}\rvert$	$\{25, 30, 35 \cdots, 50\}$	d_i	$[0.5, 2]$ Mb
c_i	$[2, 10] \times 10^8$	$L_i^{th}(t)$	$[0.5, 1]s$
C_0 (GBS)	2×10^{10} CPU cycle/s	P^V	27 dBm
C_j (UAV)	1×10^{10} CPU cycle/s	P^E	20 dBm
v^{max}	30 m/s	Δt	1s
rayleigh	8 dB	N_0	-174 dBm/Hz
β^{max}	100 RB (20 MHz)	β_0	180 kHz

Parameters	**Value**
$\Gamma_{i,1}^G$	$131.1 + 42.8 \log 10(d_{i,j})$
$\Gamma_{j',1}$, $j > 1$	$100.7 + 23.5 \log 10(d_{i,j})$
learning rate of actor	5×10^{-5}
learning rate of critic	5×10^{-4}
γ, discount factor	0.99
η, soft target update	0.001
replay buffer	10^6

Evaluation Results



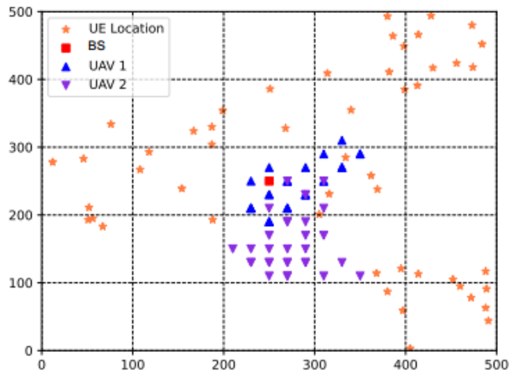
Convergence results versus different batch size.



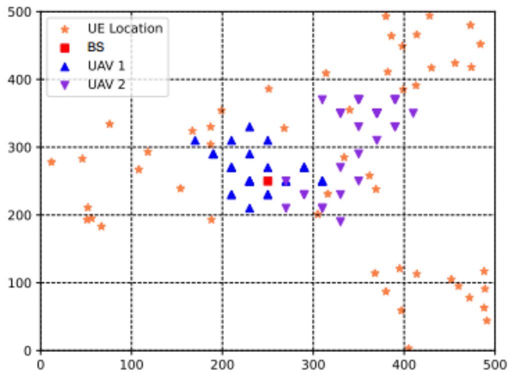
Average QoE versus number of users.

Evaluation Results (Cont'd)

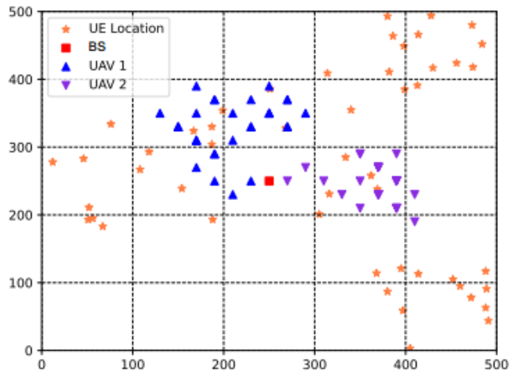
Trajectory Results



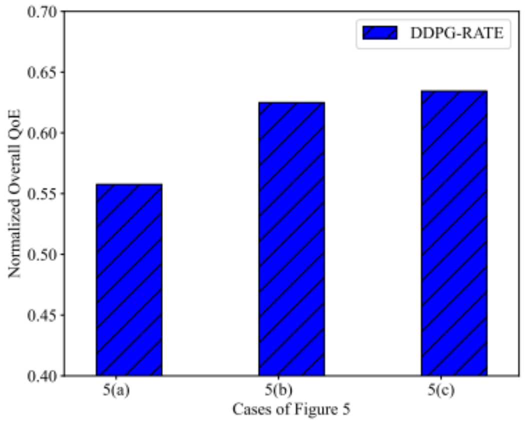
(a) $T_2 = 1$ and $T_1 = 200$.



(b) $T_2 = 2$ and $T_1 = 200$.

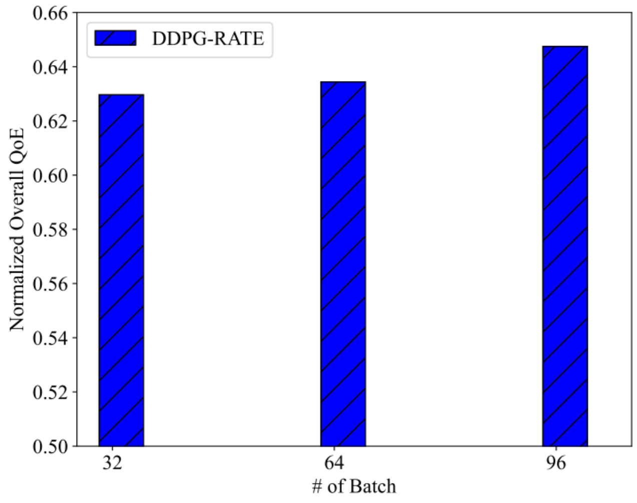


(c) $T_2 = 4$ and $T_1 = 400$.



(d) Results of different T_2 and T_1.

Different Batch Results



Outline

- Background of UAV-aided MEC
- System Model and Problem Formulation
- Algorithm and Analysis
- Evaluation Results
- Conclusions

Conclusions

- We have studied the RATE problem with consideration of the UAV path planning, UE assignment, and bandwidth and computing resource assignment, and the objective is to maximize the average total QoE of all users over multiple continuous time slots.

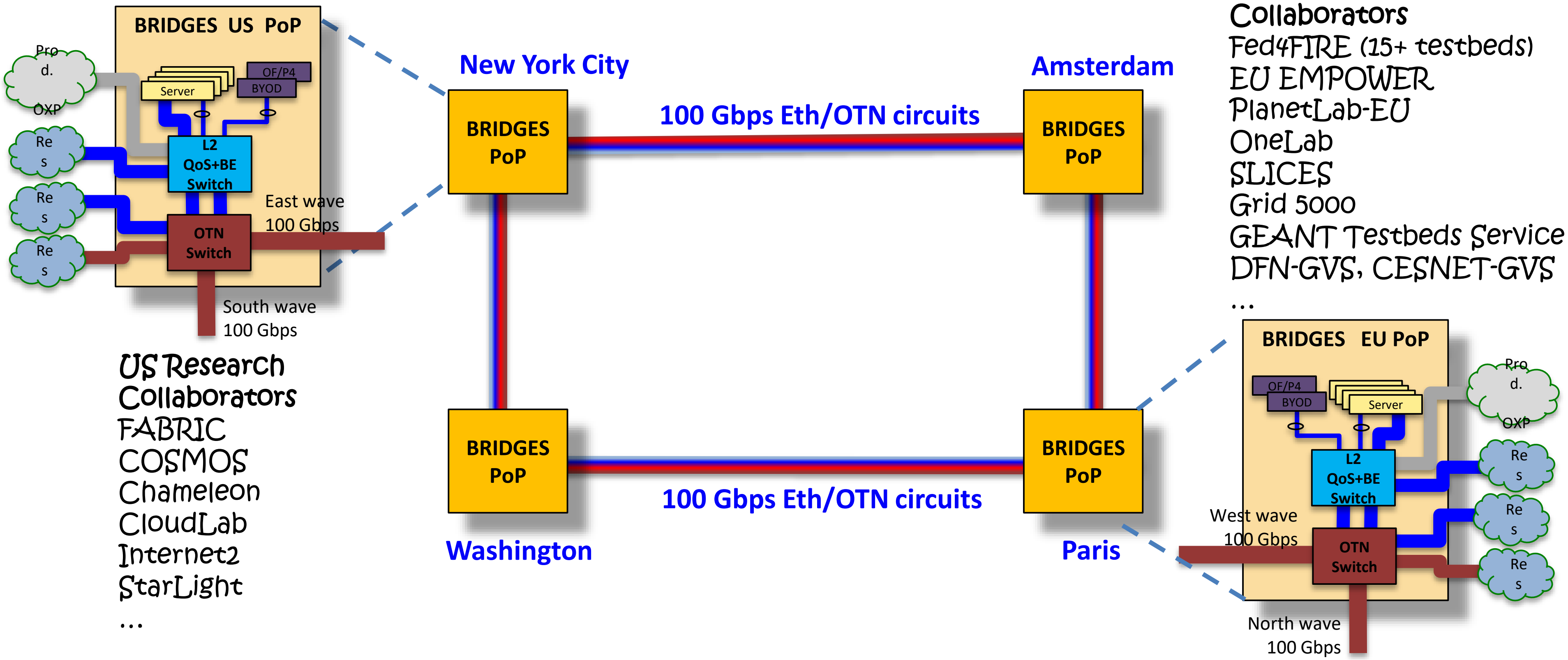
- A deep reinforcement learning approach, referred to as DDPG-RATE (deep deterministic policy gradient for RATE) algorithm, has been proposed to achieve the suboptimal solution.

- We demonstrate that the average total QoE result of the DDPG-RATE algorithm has increased up to 17.3% and 61.2% as compared to those of baseline strategies.

BRIDGES

A Programmable Cyber-System Funded By NSF: NSF OAC-2029221

EU Research Collaborators
Fed4FIRE (15+ testbeds)
EU EMPOWER
PlanetLab-EU
OneLab
SLICES
Grid 5000
GEANT Testbeds Service
DFN-GVS, CESNET-GVS
…



US Research Collaborators
FABRIC
COSMOS
Chameleon
CloudLab
Internet2
StarLight
…

For more information, please visit: https://cnl.gmu.edu/bridges