A Performance Evaluation on Channel Assignment based on Deep Reinforcement Learning in Heterogeneous Wireless Network with Unlicensed Bands

Bayarmaa Ragchaa and Kazuhiko Kinoshita

Tokushima University

1. Introduction

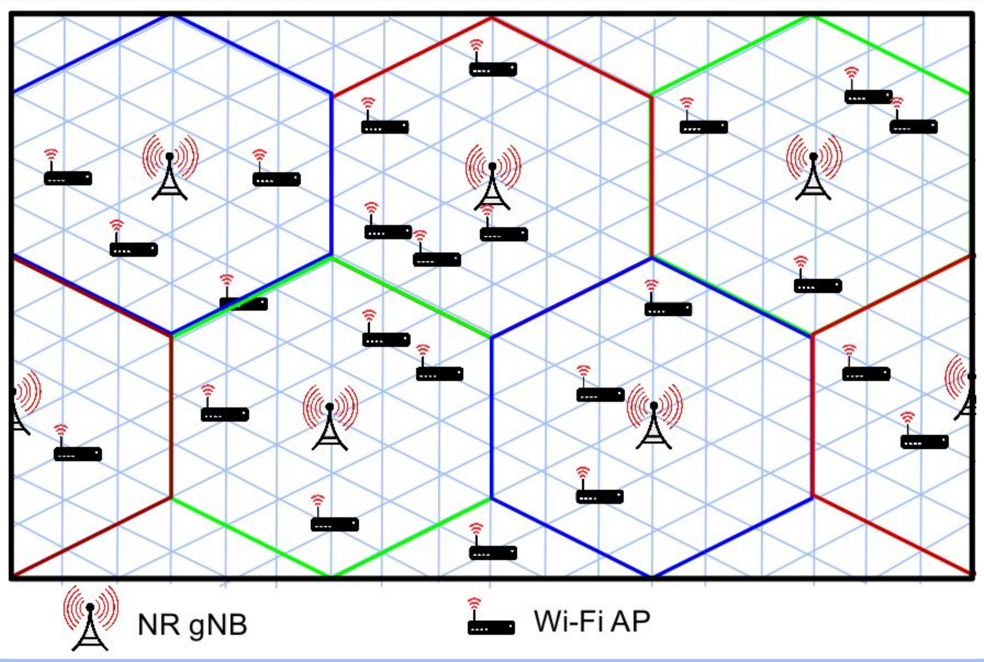
Extending cellular network to unlicensed spectrum

- Growth of mobile data traffic
- Scarcity of licenced band
- Inefficient usage of unlicensed band

Limitation of traditional methods for RRAM

- LBT (LAA R.13, 3GPP)
- CSAT/CA (LTE-U R.10-12, LTE-U forum)
- ABS (LTE R.10 3GPP)

2. Assumed environment



Small areas: covered by one or more APs

Coverage area: same size for AP and BS, hexagonal

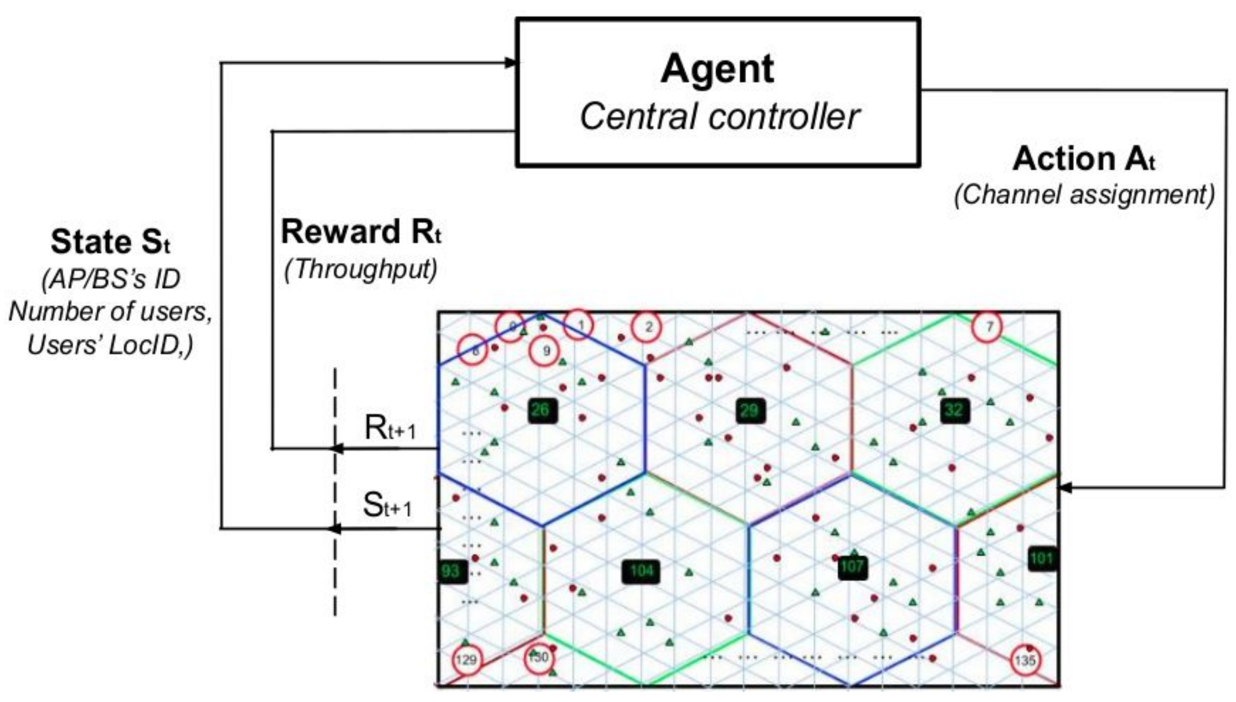
- **Two types of users:**
  - Wi-Fi only users: use the Wi-Fi network only
  - LTE/Wi-Fi combined users: use LTE and Wi-Fi
- **LAA LBT:** Cat 4, Modeled by Markov chain, 5GHz unlicensed

3. Proposed Method

DRL: channel assignment problem →MDP, (S, A, p, R)

- **State space:** # of users and their location ID, channel state
- **Action space:** $A_t \in A=\{0,1,2,3\}$
- **Transition probability:** $p(S_{t+1}, R_t | S_t, A_t) = P_r\{S_t = S_{t+1}, R_t = R_{t+1} | S_{t-1} = S_t, A_{t-1} = A_t\}$
- **Reward function:** $R_t(S_t, A_t)$

Interaction process between an agent and the environment



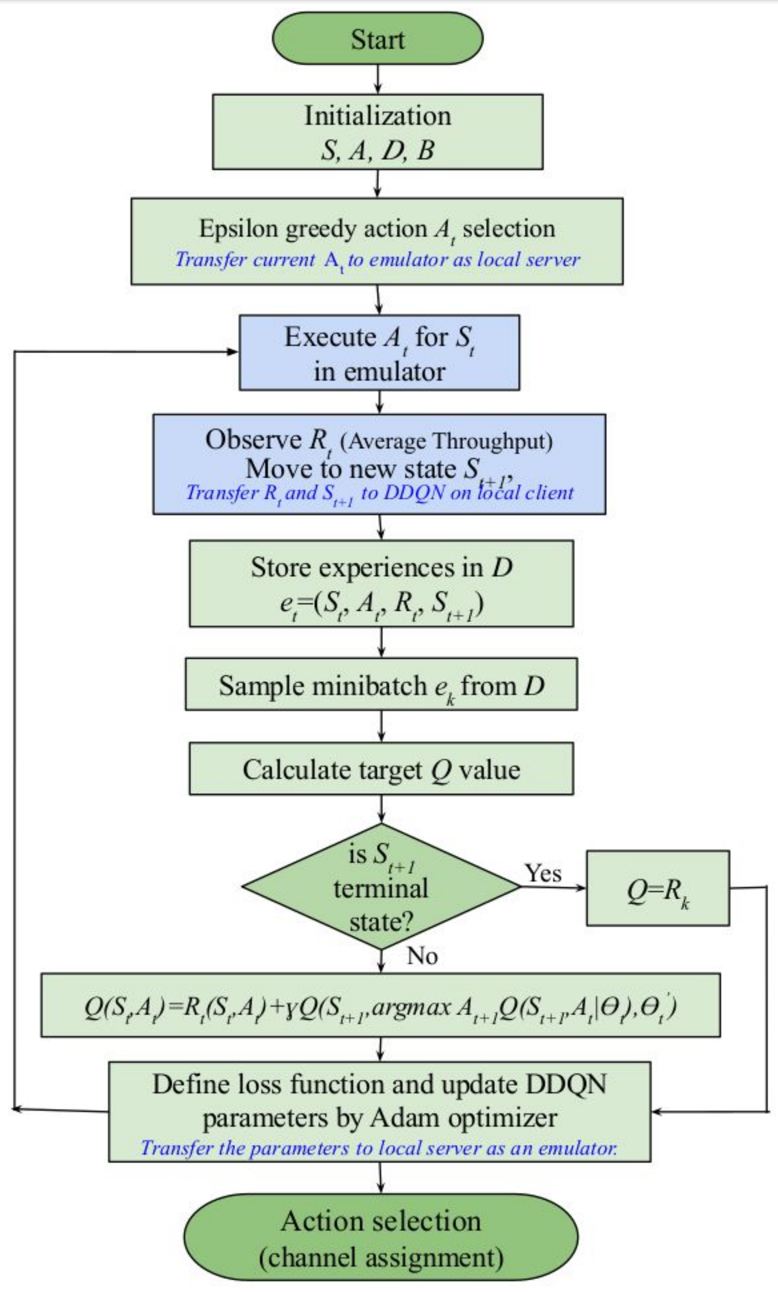
Agent (local client): central controller/ decision maker

Environment (local server): simulator/testbed of agents

$R_t(S_t, A_t)$

Flowchart of DDQN based channel assignment

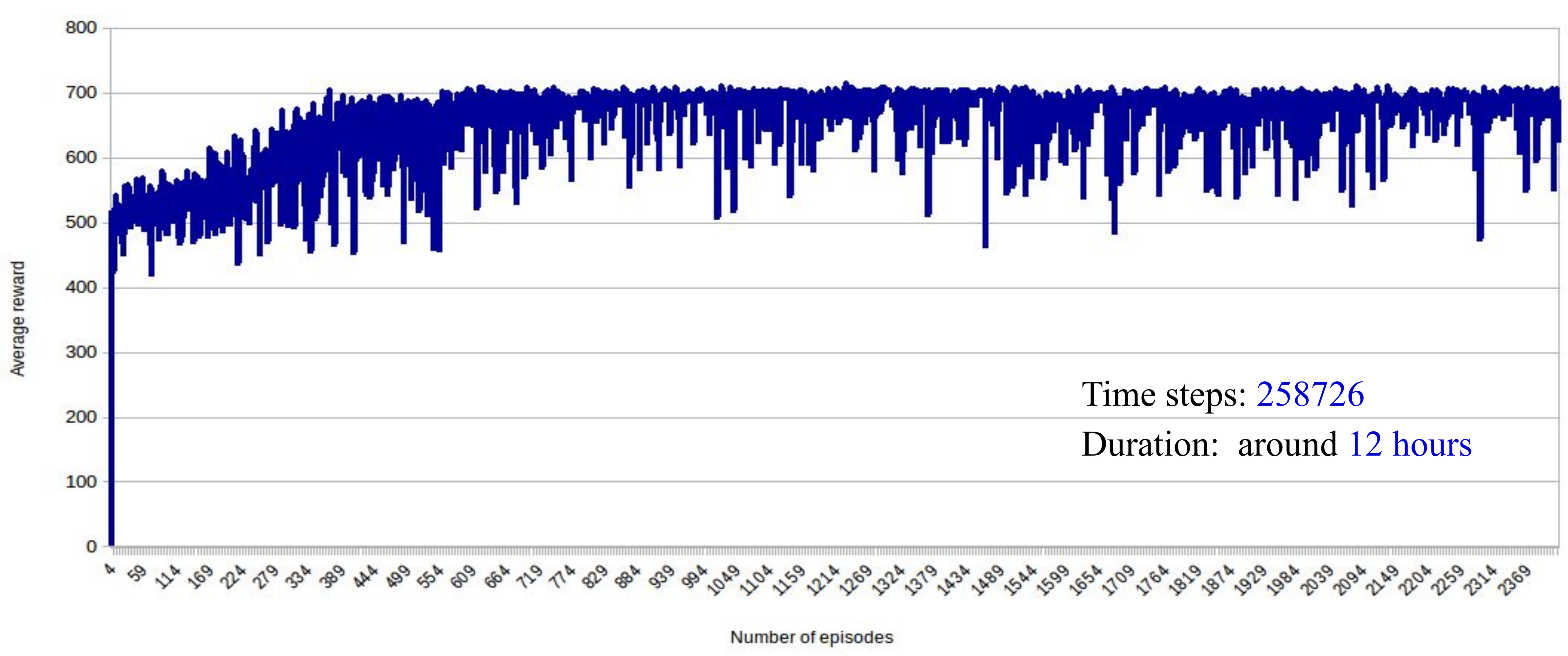
- Time steps: 107
- Epsilon greedy algorithm: $A_t \in A=\{0,1,2,3\}$
- Transfer A_t from DQN (client) to environment (server)
- Calculate $R(S_t, A_t)$
- Transfer R_t, S_{t+1}
- Store $e_t = (S_t, A_t, R_t, S_{t+1})$ in D
- Calculate Q value
- Loss function: Huber
- Optimizer: Adam



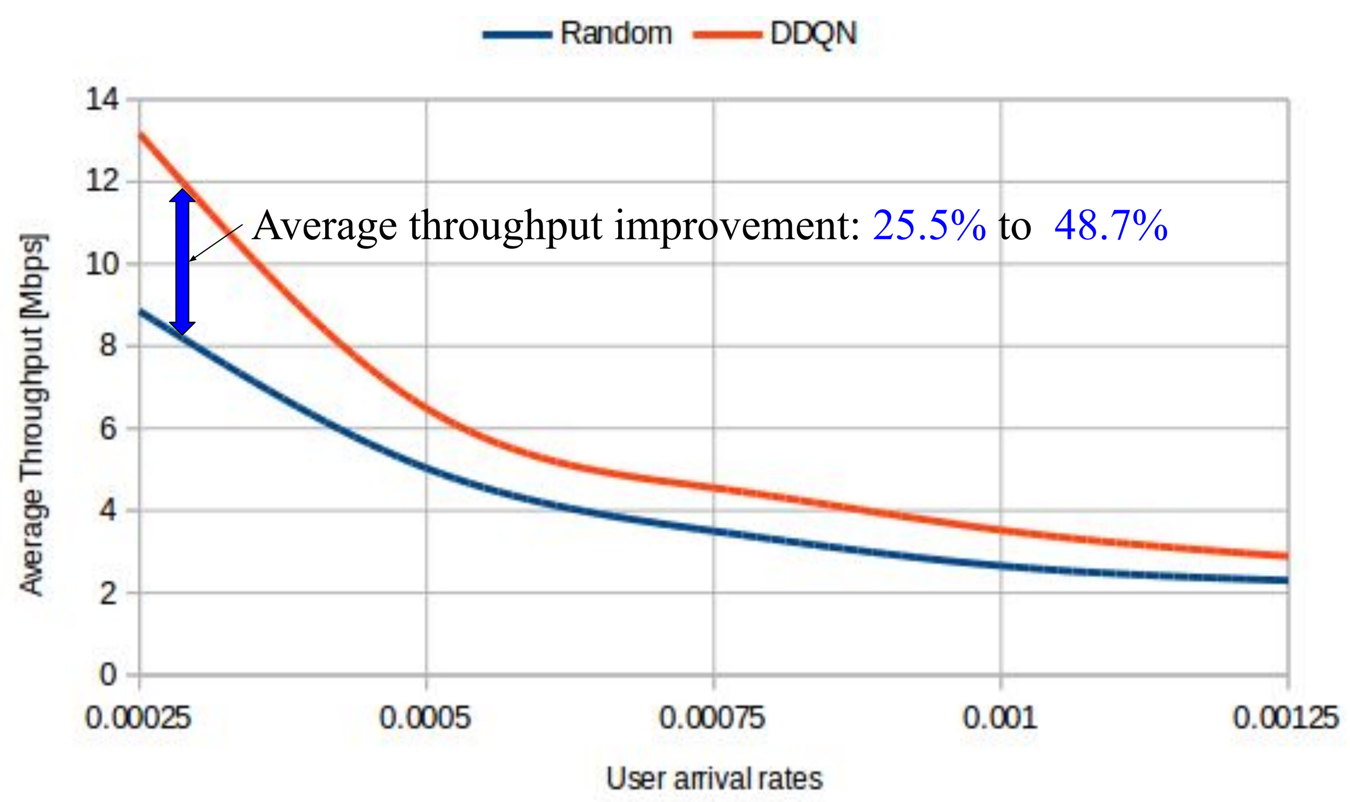
4. Performance Evaluation

Simulation model			
Minimum area:	288	# of channels for AP:	4
Placeable area:	136	# of channels for BS:	3
Coverage area:	54	User arrival rate:	λ (Poisson arrival process)
# of BS:	7	Arrival ratio Wi-Fi and LTE+Wi-Fi users:	1:1
# of AP:	100	Communication time:	300sec (exponential distribution)

Performance of the obtained DDQN model



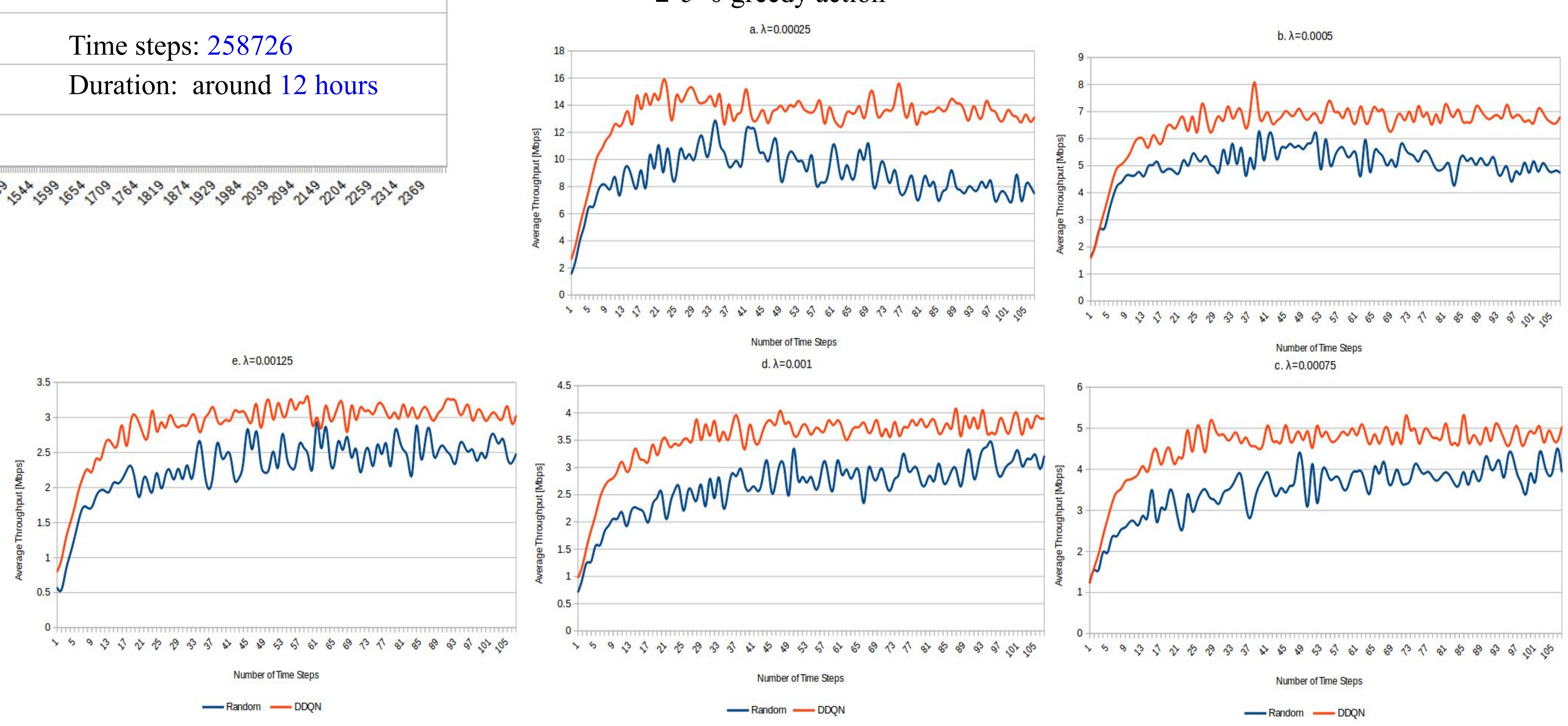
Comparison of average throughput



Validation results in different user arrival rates

Epsilon greedy algorithm: $A_t = \pi(S_t)$; $\pi^* = \max R$

- e=1 random action,
- e=0 greedy action



Network architecture

- Two hidden layer: (tried from 2 to 5)
- Number of nodes per layer (from 8 to 288)

Layer (type)	Output Shape	Total Params	Trainable params	Non-trainable params
dense (Dense)	(None, 60)	6540	8494	0
dense_1 (Dense)	(None, 30)	1830		
dense_2 (Dense)	(None, 4)	124		