On the Analysis of Spatially Constrained Power of Two Choice Policies

Nitish K. Panigrahy, Prithwish Basu, Don Towsley, Ananthram Swami and Kin K. Leung

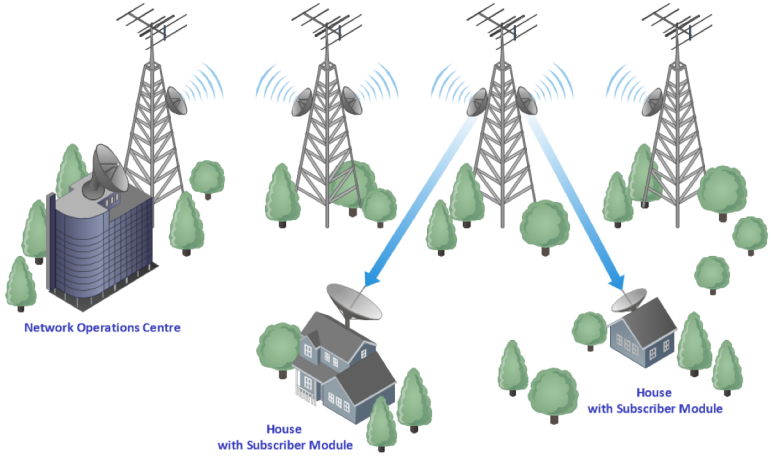
Performance 2020

Talk Outline

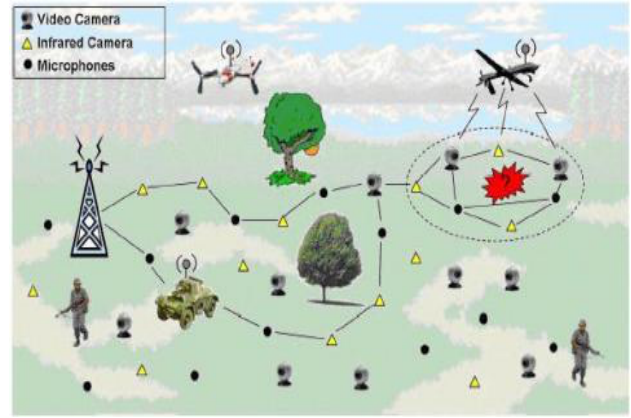
1. Motivation
2. Server Allocation Policies in 2D
3. Load and Request Distance Tradeoff
4. Two new policies: dPOT and k-sPOT
5. Future work

Distributed Service Network

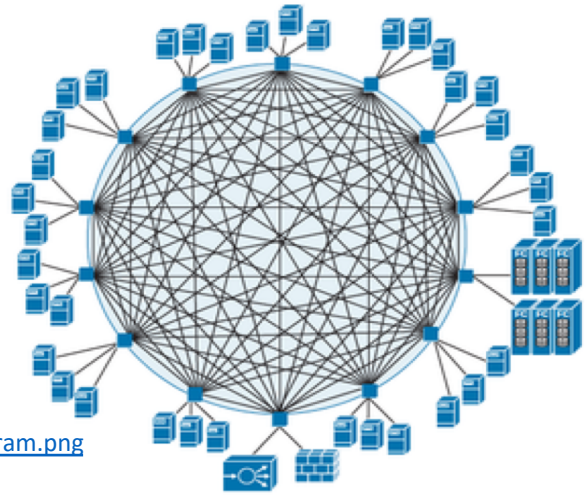
- **Geographically** distributed smart devices, servers and end-users
- Effect of user and resource spatial distributions on performance



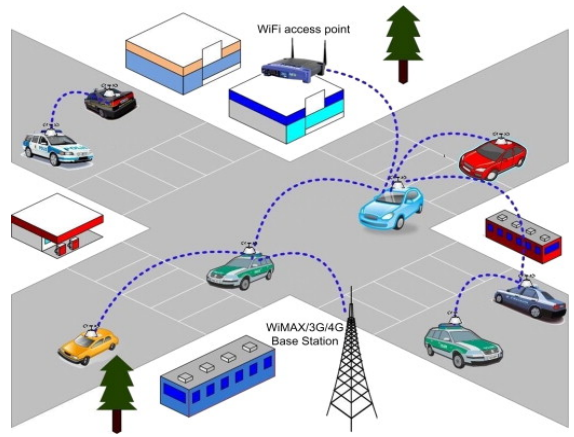
Wireless Network[1]



Sensor Network[2]



Data Center[3]



VANET[4]

1. https://www.conceptdraw.com/How-To-Guide/picture/Wireless-broadband-network--layout-diagram.png
2. https://www.mistralsolutions.com/articles/building-wireless-sensor-network-using-smartphones/
3. https://www.networkcomputing.com/data-centers/why-large-data-centerss-need-overlay-networks
4. https://link.springer.com/article/10.1186/s13638-019-1503-4

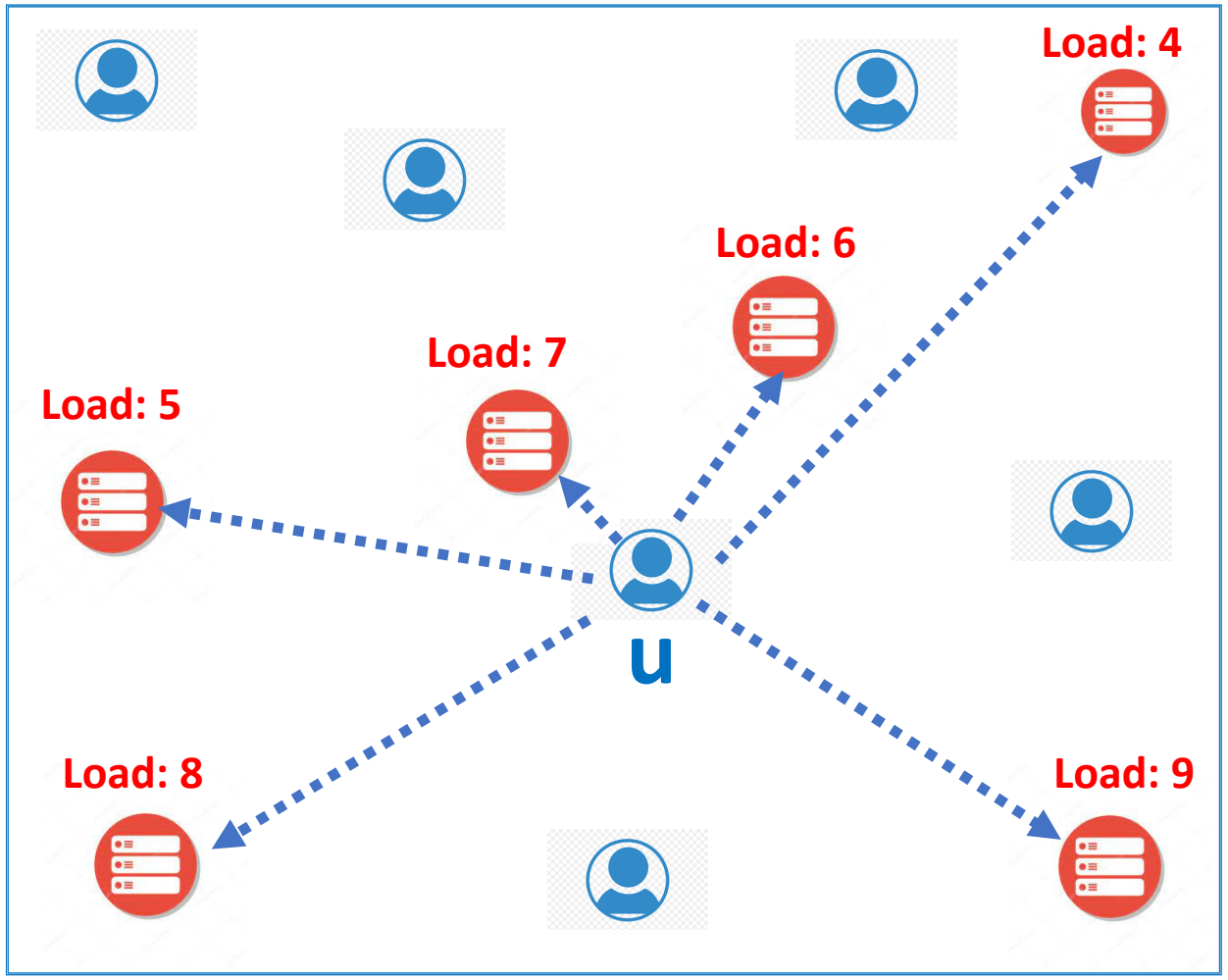
Questions:

1. How to allocate **Users/apps** to **Servers/Resources**?
2. Performance Metrics?
   Minimize Load? Request Distance?

# of users allocated to a server

Distance b/w a user and its allocated server

Distributed Load Balancing Policies in 2D



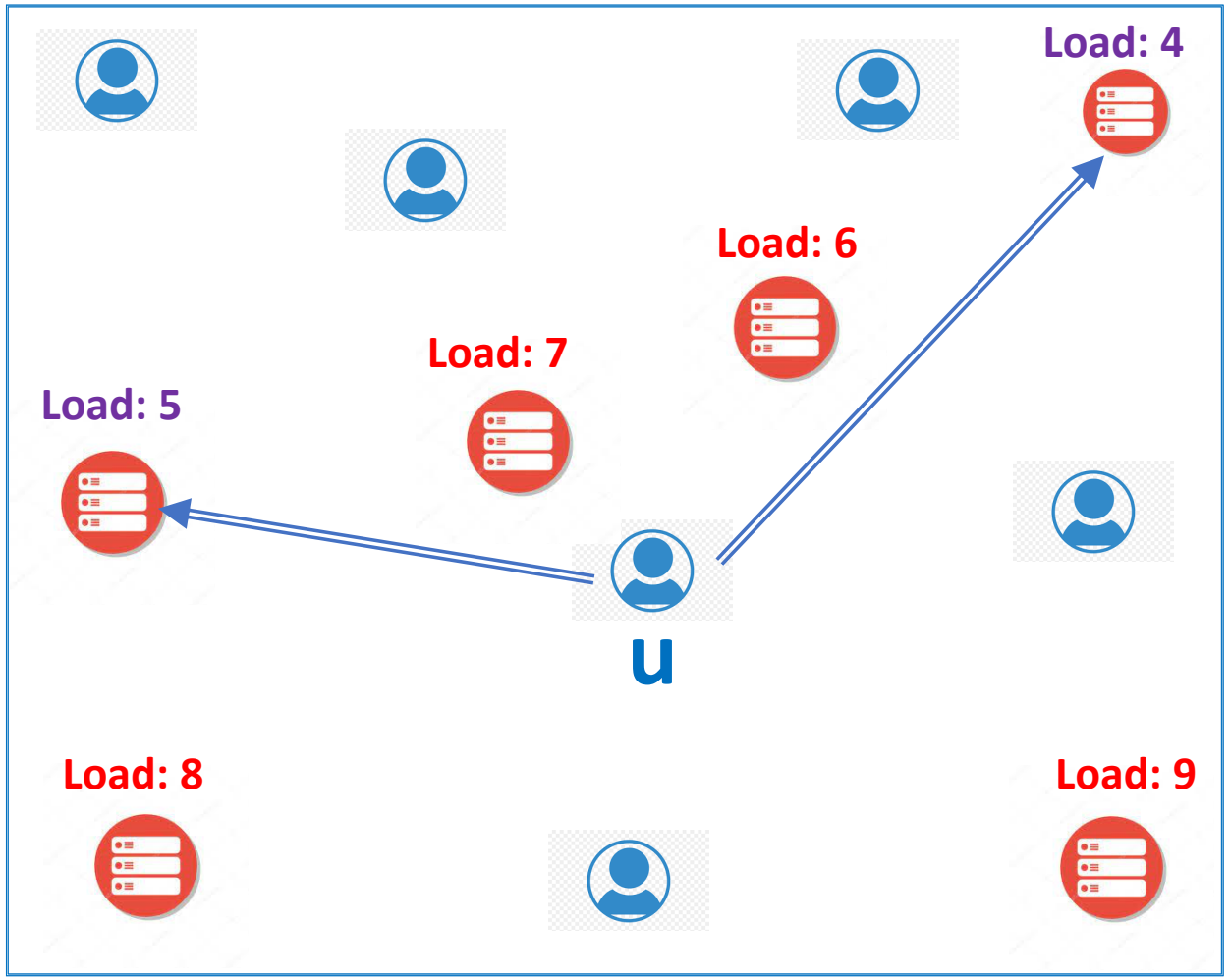
User (U)

Server (S)

Candidate Servers

Power of Two (POT) [Azar et al, STOC' 94]: **Allocate to leastly loaded server among two servers sampled uniformly at random**

Distributed Load Balancing Policies in 2D



User (U)

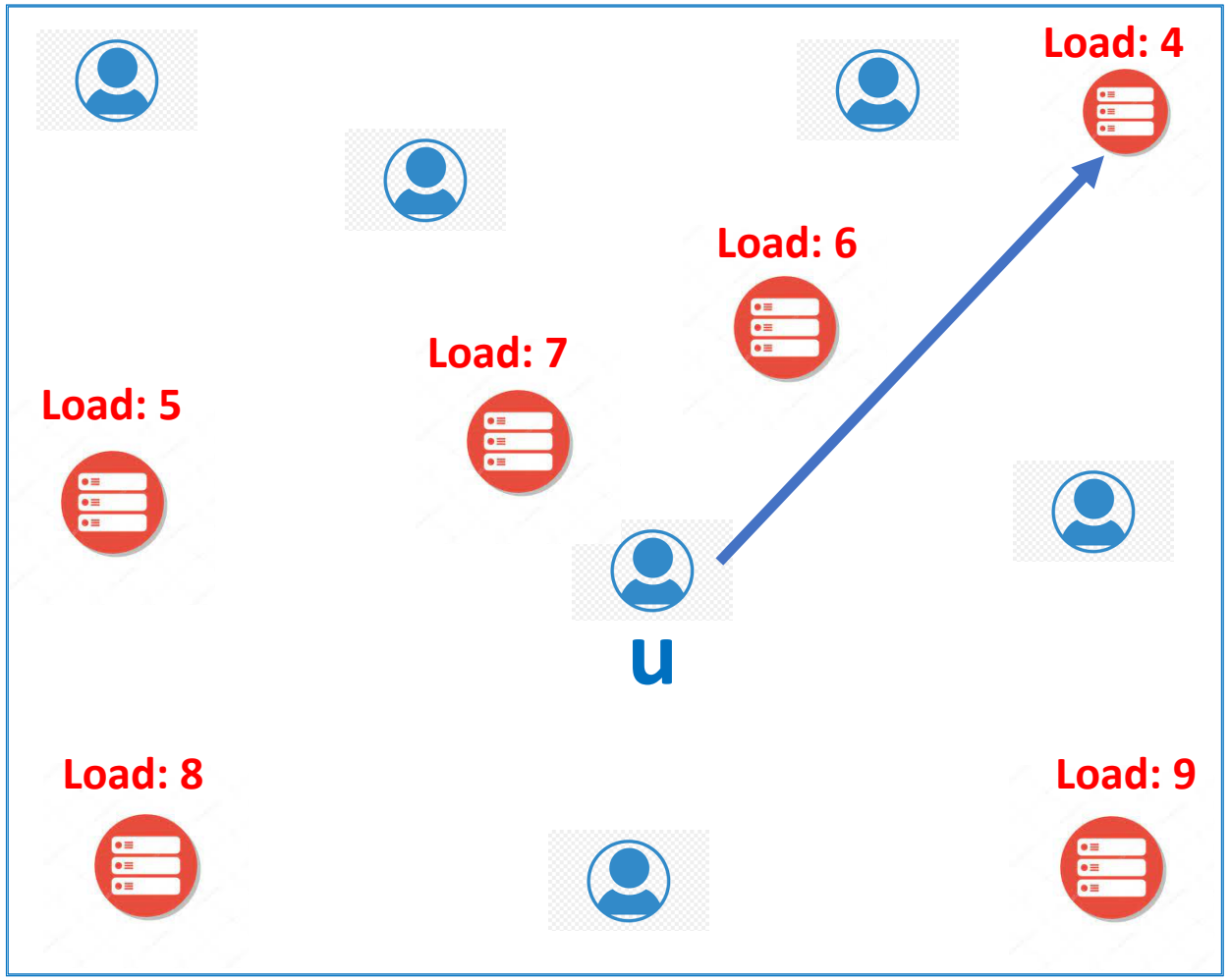
Server (S)

Candidate Servers

Sampled Servers

Power of Two (POT) [Azar et al, STOC' 94]: **Allocate to leastly loaded server among two servers sampled uniformly at random**

Distributed Load Balancing Policies in 2D



User (U)

Server (S)

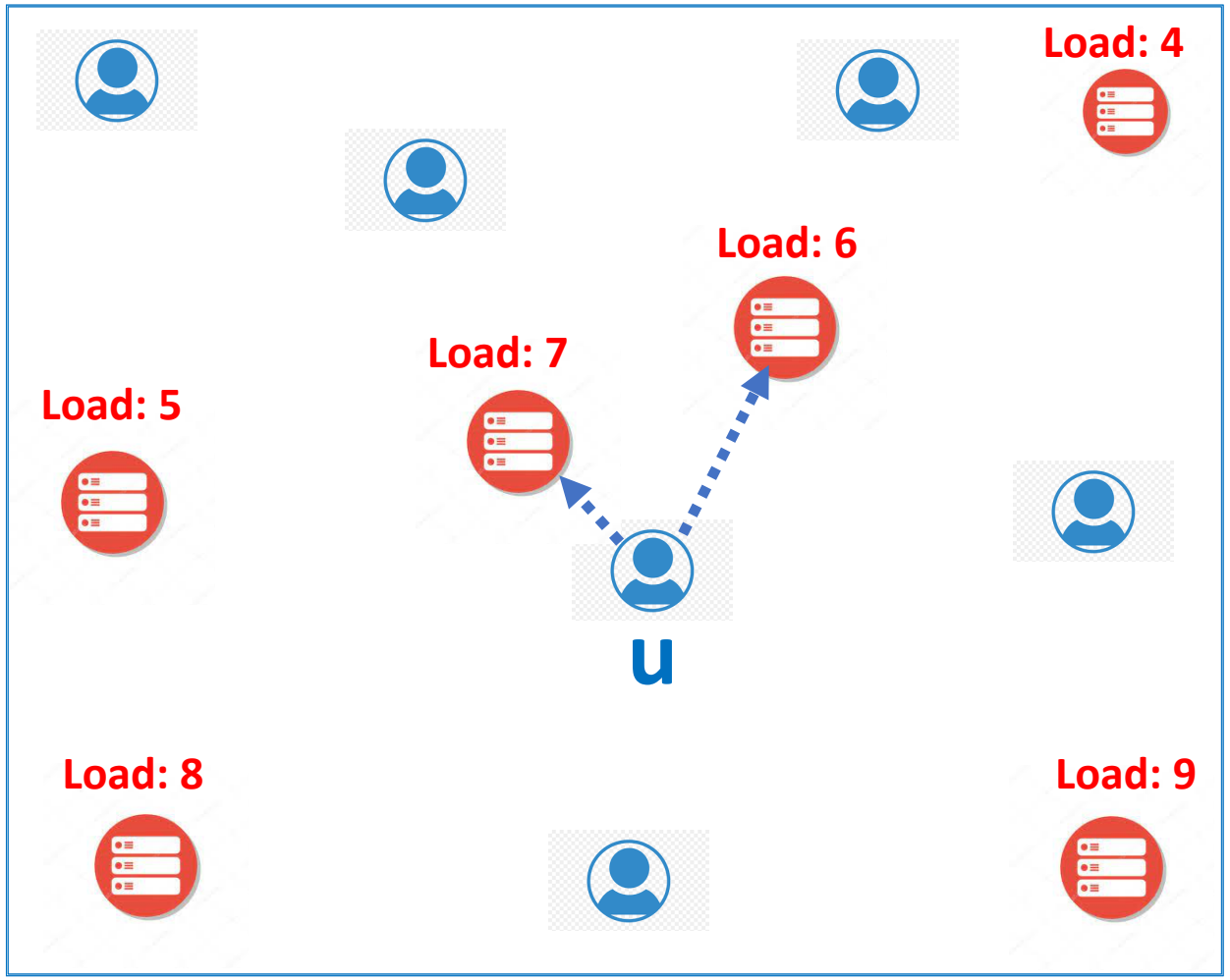
Candidate Servers

Sampled Servers

Allocated Server

Power of Two (POT) [Azar et al, STOC' 94]: Allocate to leastly loaded server among two servers sampled uniformly at random

Proposed Spatial Policy



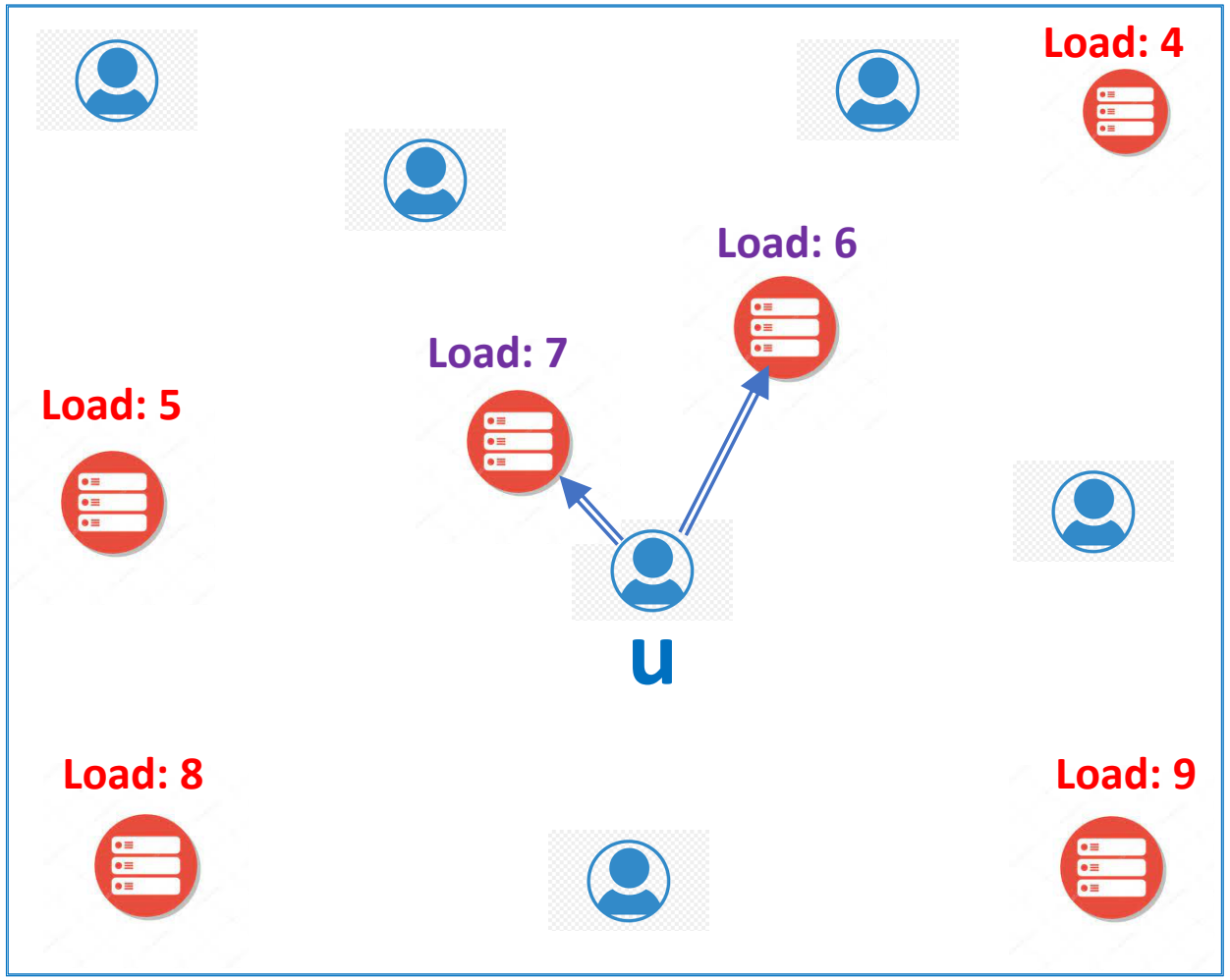
User (U)

Server (S)

Candidate Servers

Spatial Power of Two (sPOT): Allocate to leastly loaded server among two nearest servers

Proposed Spatial Policy



User (U)

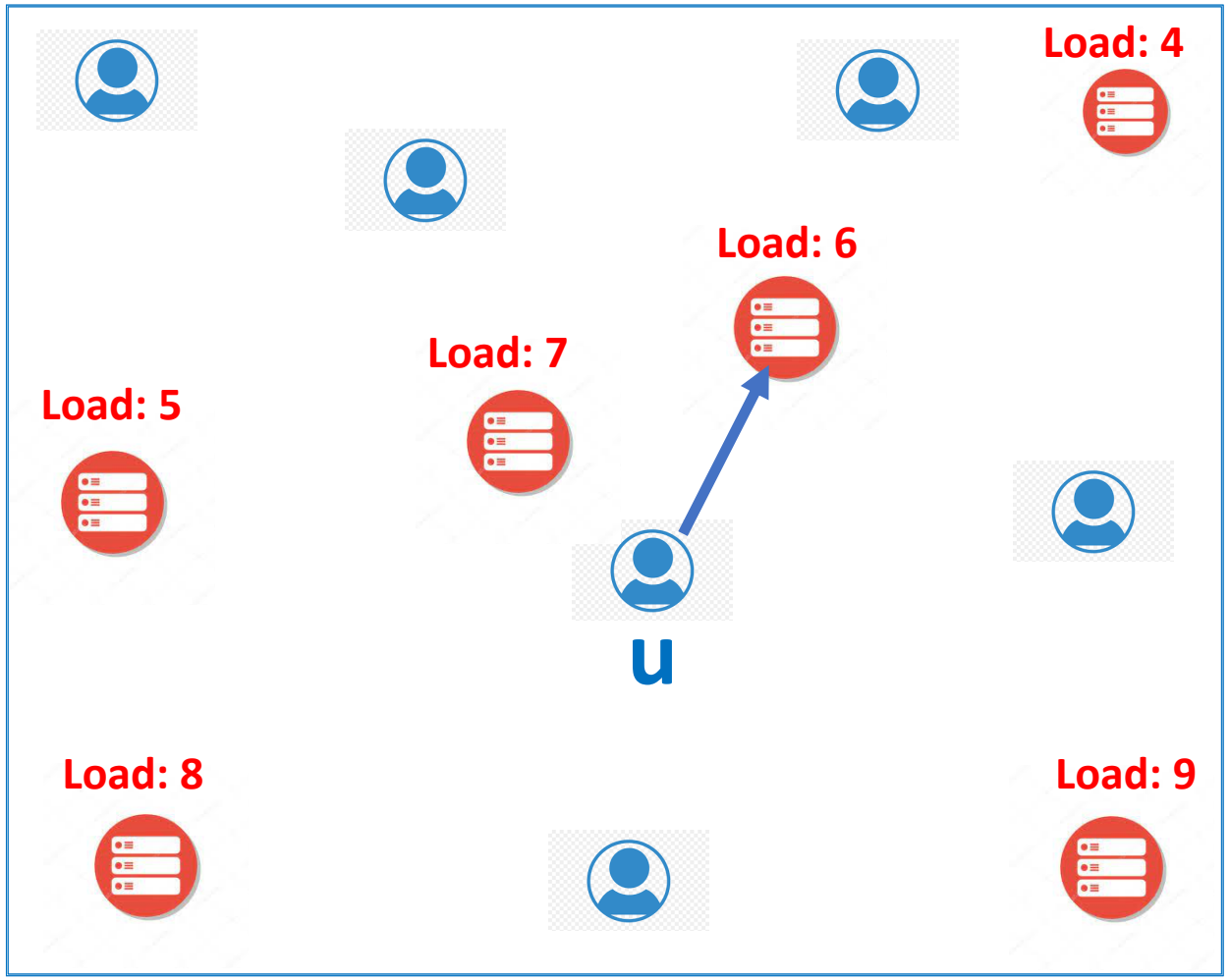
Server (S)

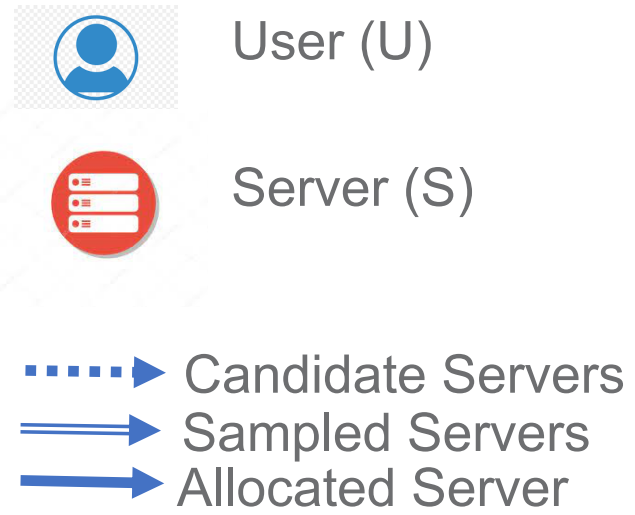
Candidate Servers

Sampled Servers

Spatial Power of Two (sPOT): Allocate to leastly loaded server among two nearest servers

Proposed Spatial Policy





Spatial Power of Two (sPOT): Allocate to leastly loaded server among two nearest servers

Maximum Asymptotic Load Behavior

- Power of one (POO): Max. asymptotic load $O(\log |S| / \log \log |S|)$
- Power of two (POT) : Max. asymptotic load $\boldsymbol{O(\log \log |S|)}$ [Azar et al, STOC' 94]

POT benefits

$|S|$: # of servers

Question: Does sPOT provide POT benefits?

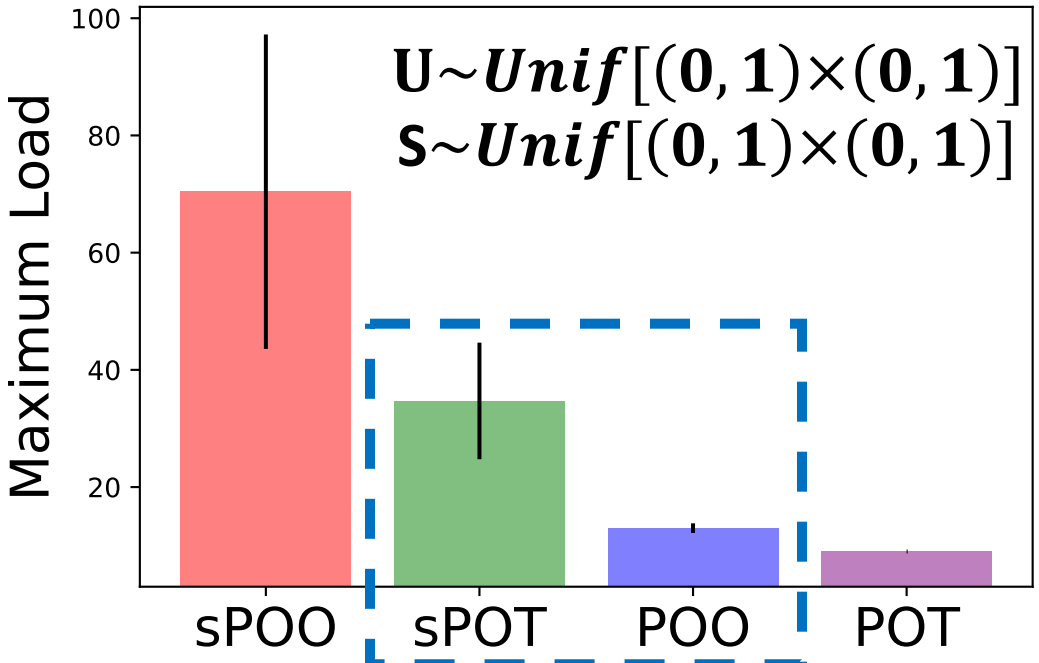
Our Result: sPOT does not provide POT benefits

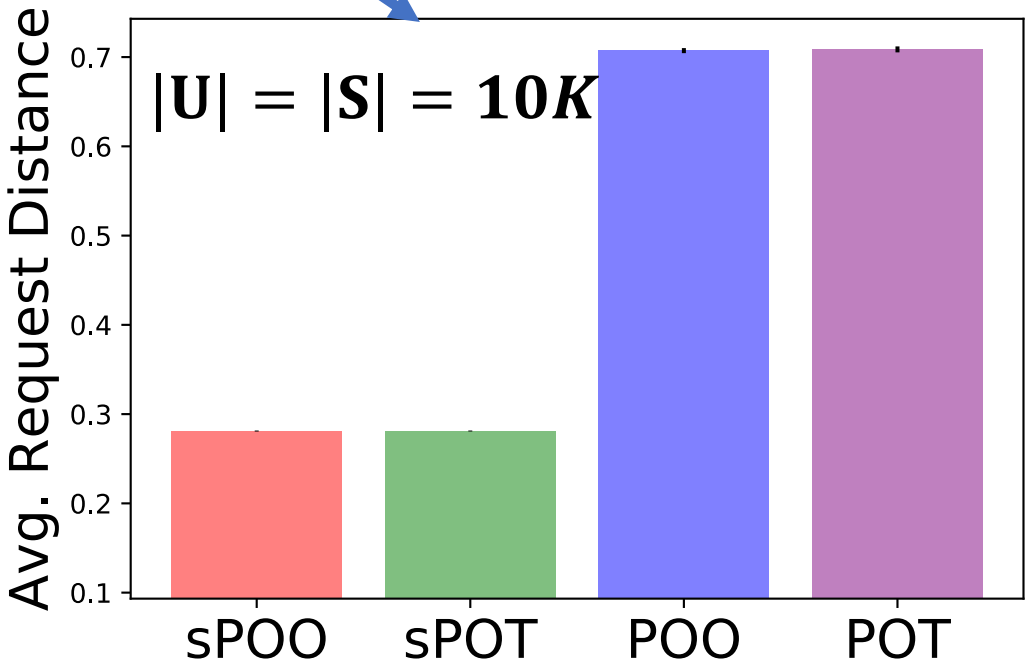
Theorem

Suppose users and servers are placed uniformly at random on a 2D Euclidean plane. Under **sPOT** policy, the maximum load over all servers is at least $\Omega(\log |S|/\log \log |S|)$ with high probability, i.e., we **do not get POT benefits**.

Load vs Request Distance Trade-off

POT performs best wrt Load but worst wrt req dist



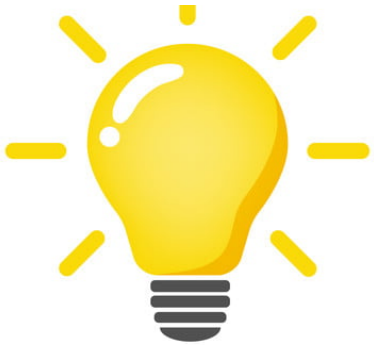


Spatial POO

sPOT performs worse than **POO** wrt Load

Question:

1. Benefits of both the worlds? POT like load behavior and sPOT like req dist profile?



Solution:

1. **Random** sampling to reduce load
2. **Distance based sampling** to reduce req dist

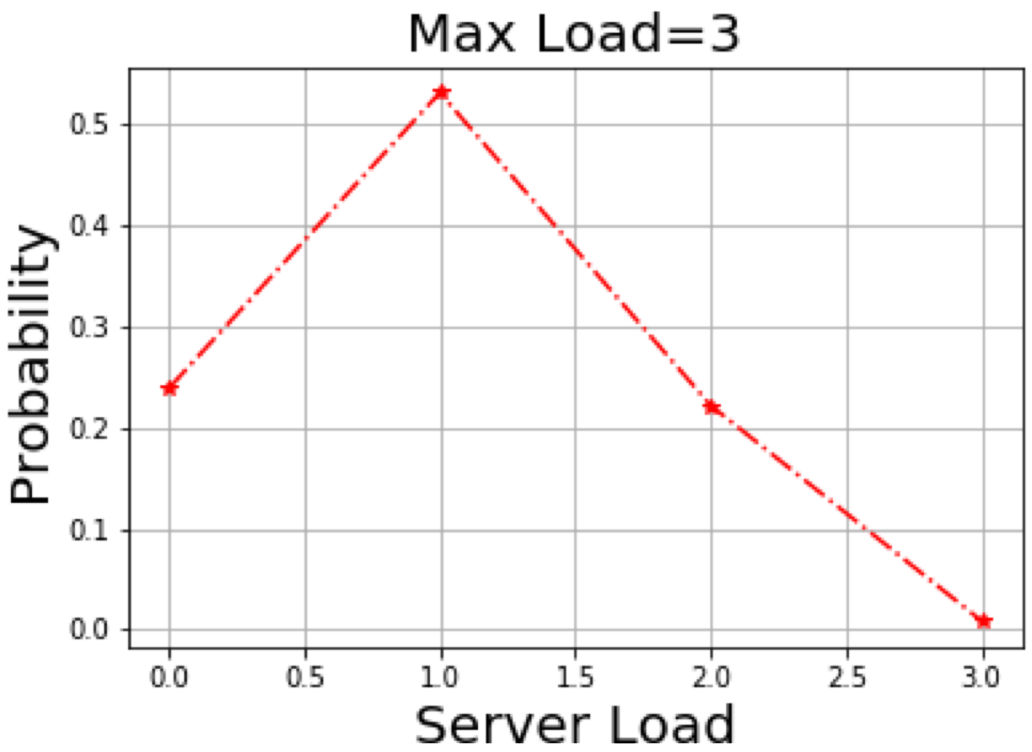
Two New Policies: d-POT and k-sPOT

d-POT: User samples two servers from S, each with probability proportional to $1/d^2$; Allocate to server with least load.

k-sPOT: User samples two servers uniformly from k nearest servers; Allocate to server with least load.

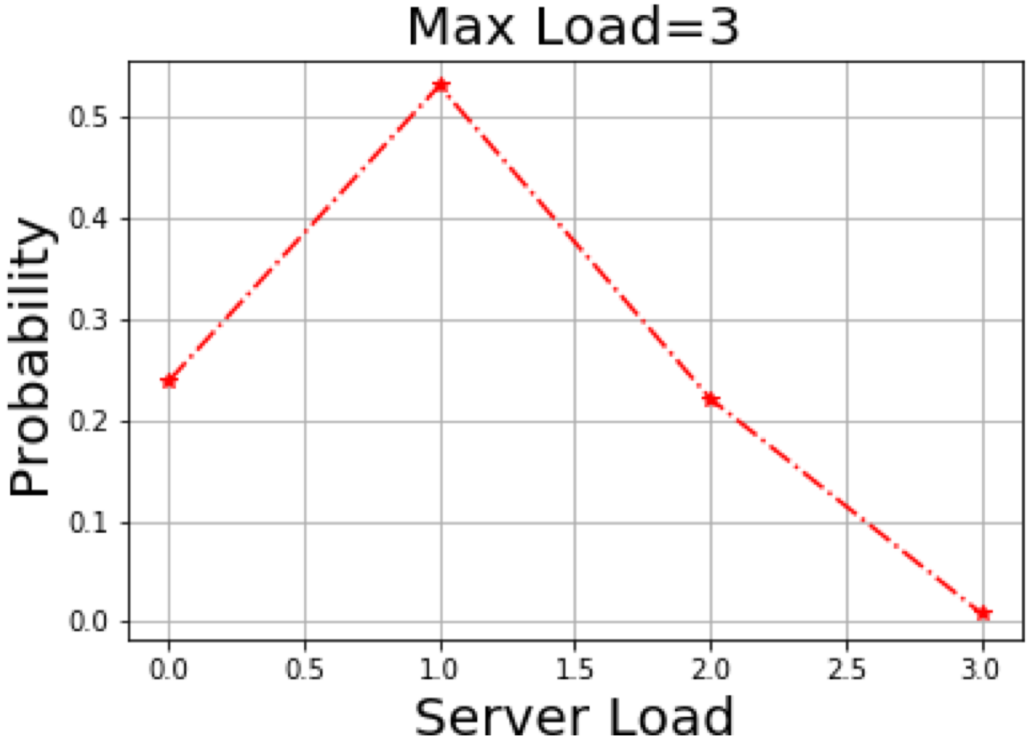
dPOT achieves POT like load behavior!!

$$|\mathbf{U}| = |\mathbf{S}| = 50K$$



dPOT

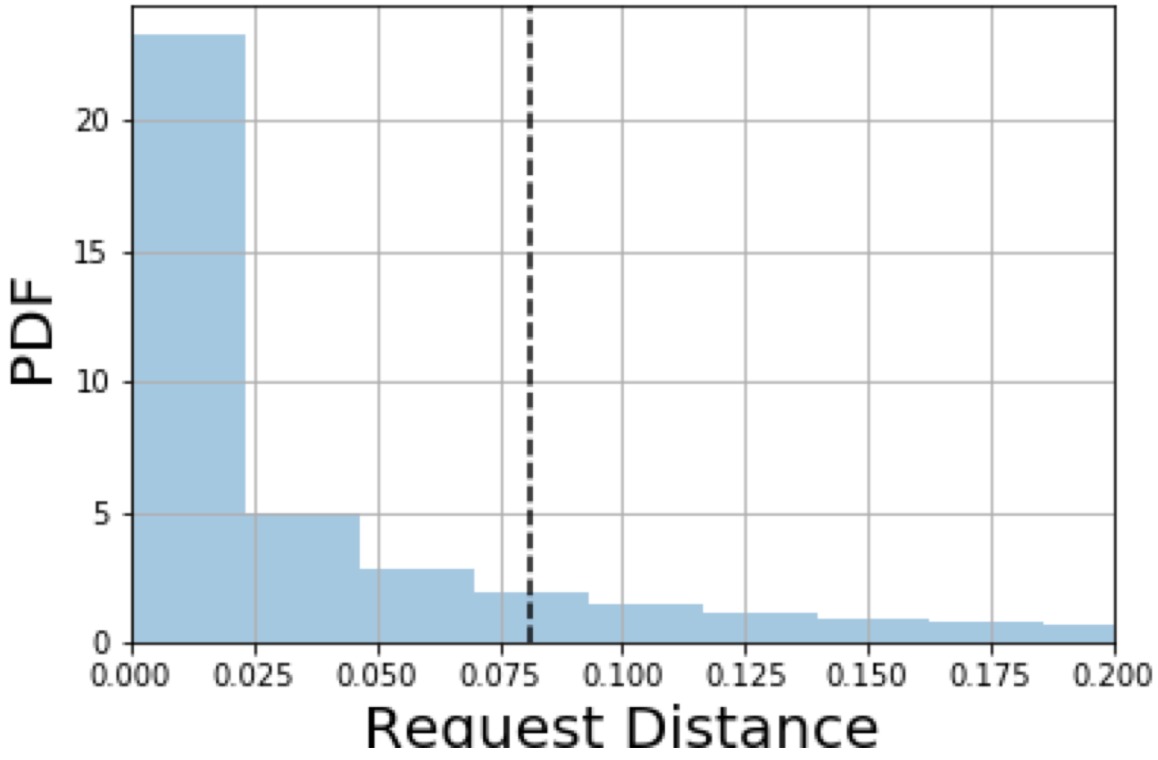
$$\mathbf{U} \sim Unif[(0,1)\times(0,1)]$$
$$\mathbf{S} \sim Unif[(0,1)\times(0,1)]$$



POT

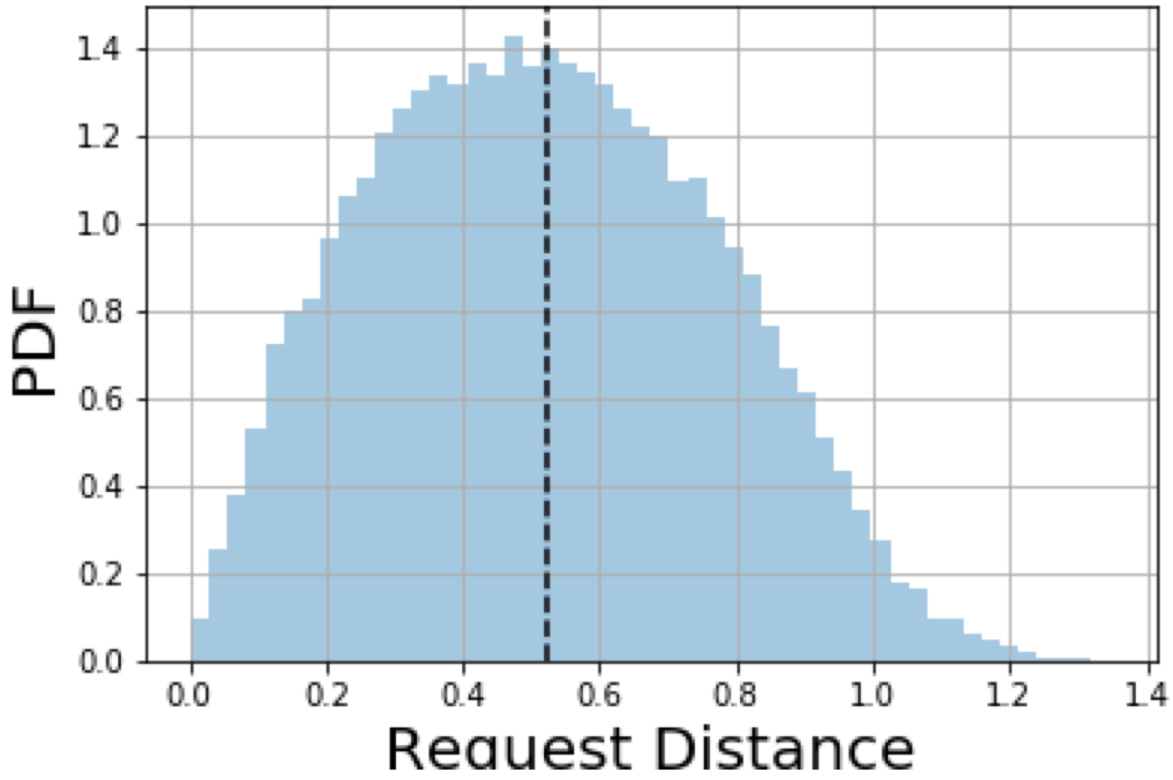
dPOT achieves pretty low req dist!!

$$|\mathbf{U}| = |\mathbf{S}| = \mathbf{50}\boldsymbol{K}$$



dPOT

$$\mathbf{U} \sim \boldsymbol{Unif}[(\mathbf{0},\mathbf{1}) \times (\mathbf{0},\mathbf{1})]$$
$$\mathbf{S} \sim \boldsymbol{Unif}[(\mathbf{0},\mathbf{1}) \times (\mathbf{0},\mathbf{1})]$$

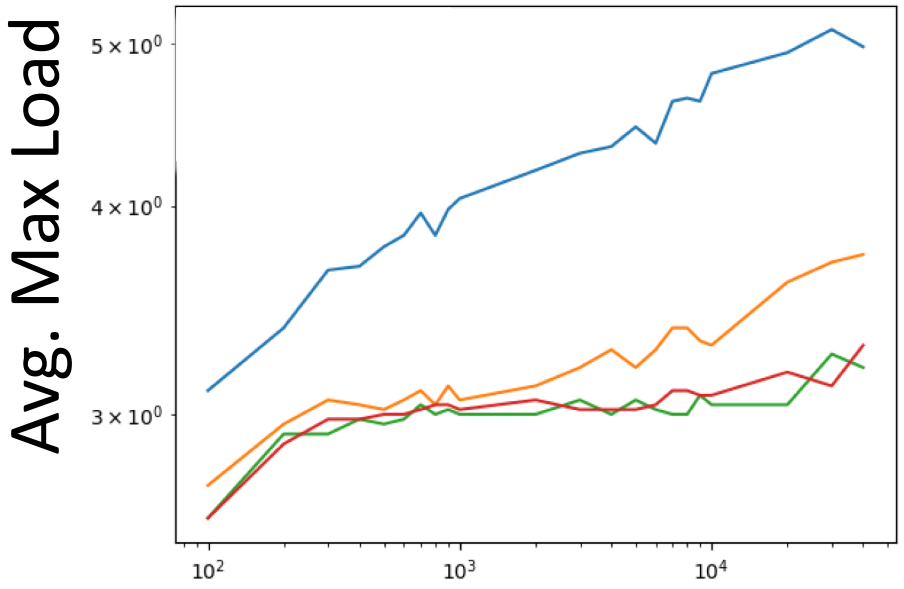


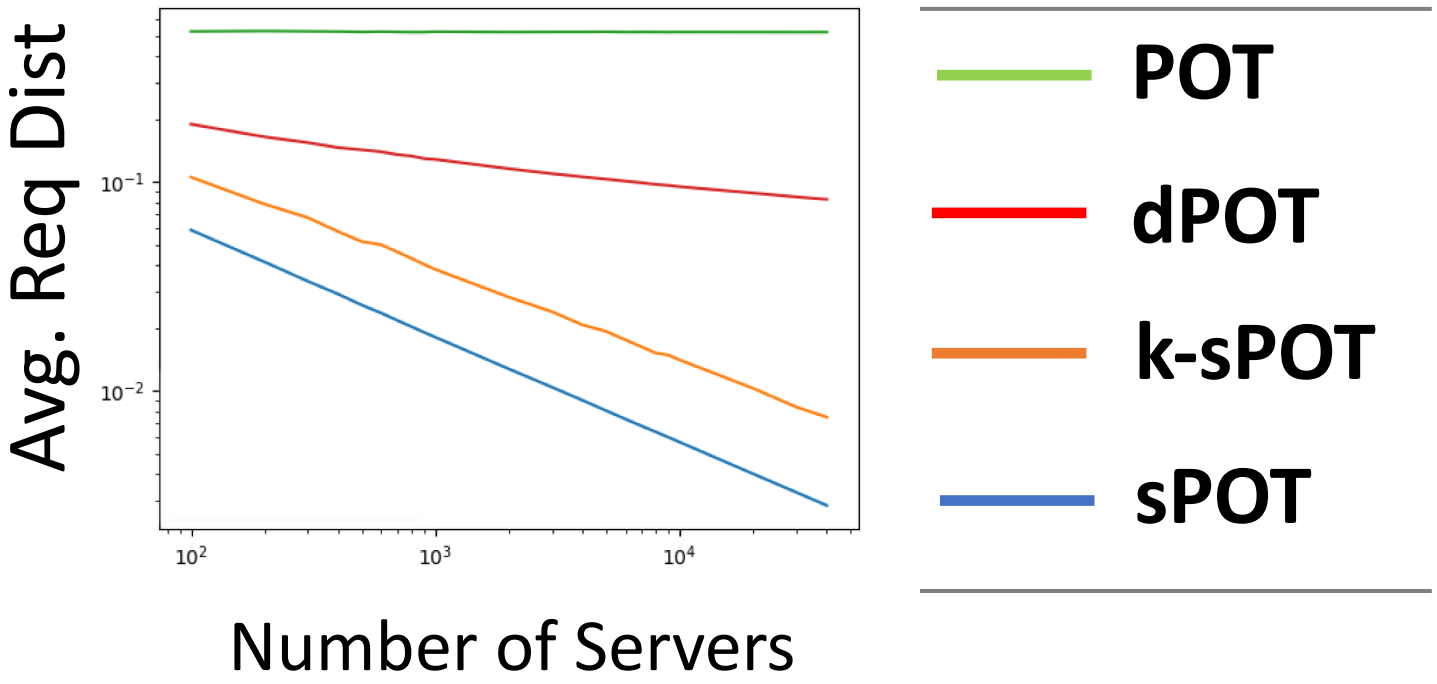
POT

k-sPOT achieves low req dist and low load

$$|\mathbf{U}| = |\mathbf{S}| = \mathbf{100} - \mathbf{40}\boldsymbol{K}$$
$$\mathbf{k} = \mathbf{log}|\mathbf{S}|$$

$$\mathbf{U} \sim \boldsymbol{Unif}[(\mathbf{0},\mathbf{1}) \times (\mathbf{0},\mathbf{1})]$$
$$\mathbf{S} \sim \boldsymbol{Unif}[(\mathbf{0},\mathbf{1}) \times (\mathbf{0},\mathbf{1})]$$





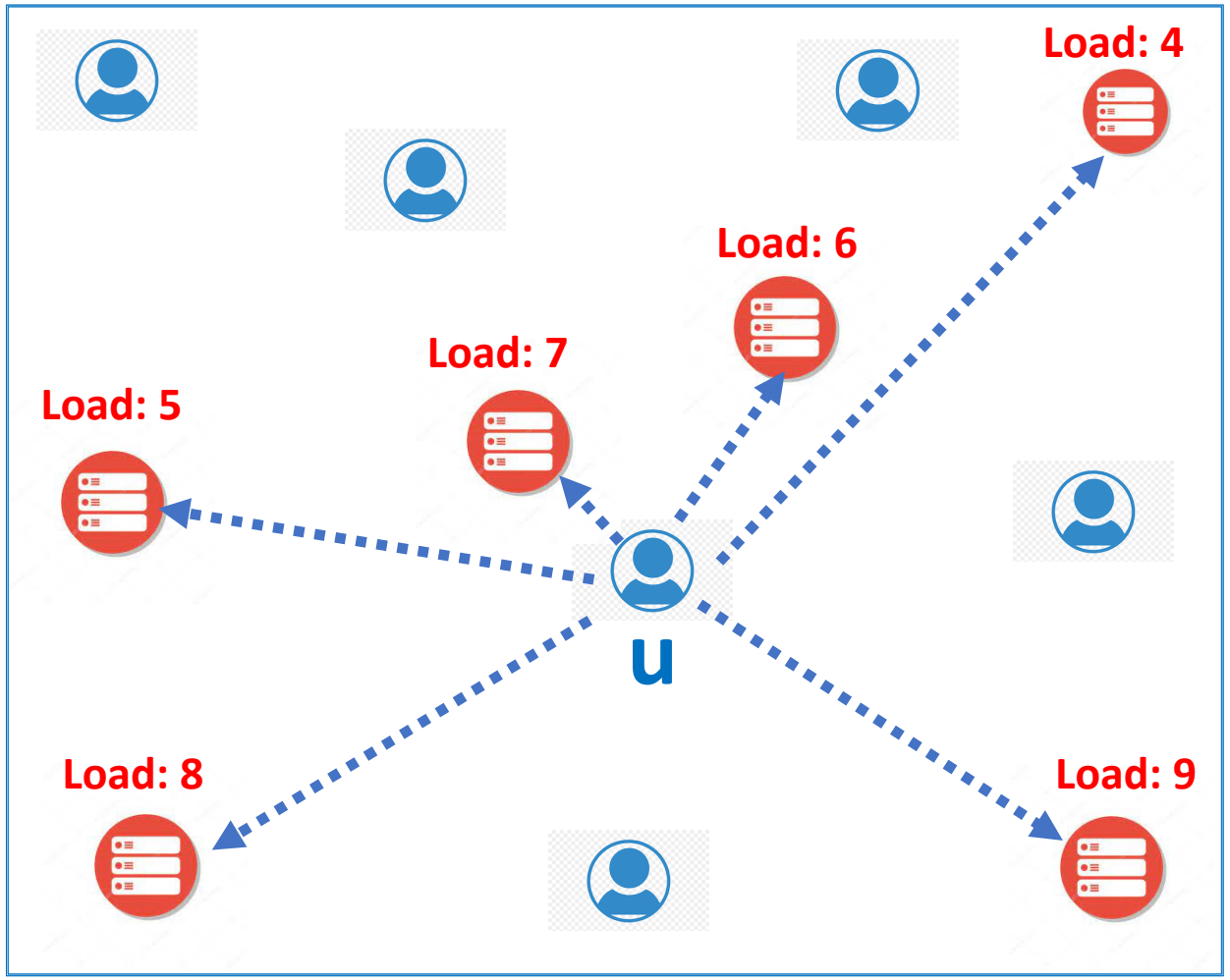
Future Directions

- Dynamic arrival of users
  - Users arrive and leave the system over time
  - Map to supermarket model
- Other topologies for server placement

Questions

Bonus Slides

Distributed Load Balancing Policies in 2D



User (U)

Server (S)

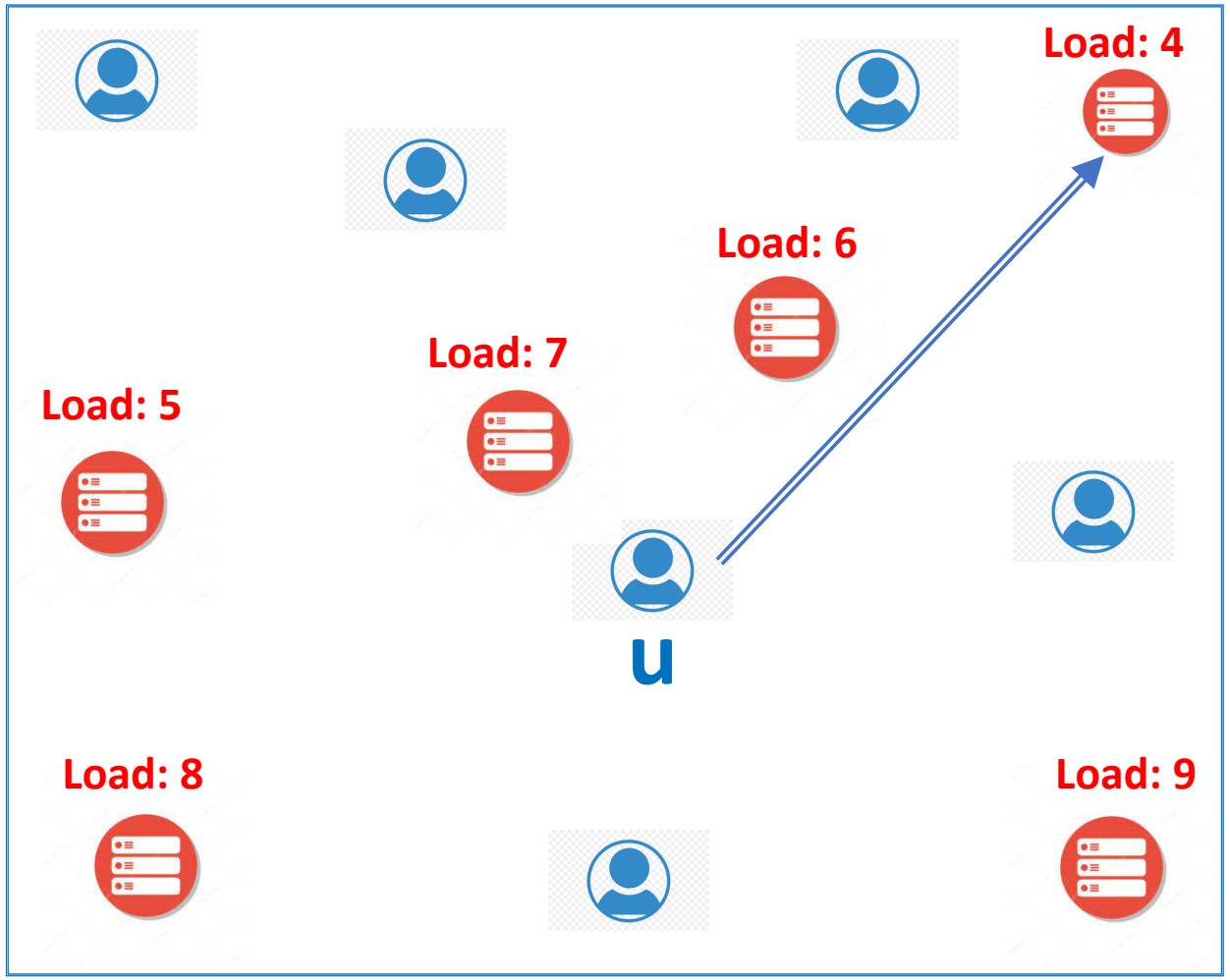
Candidate Servers

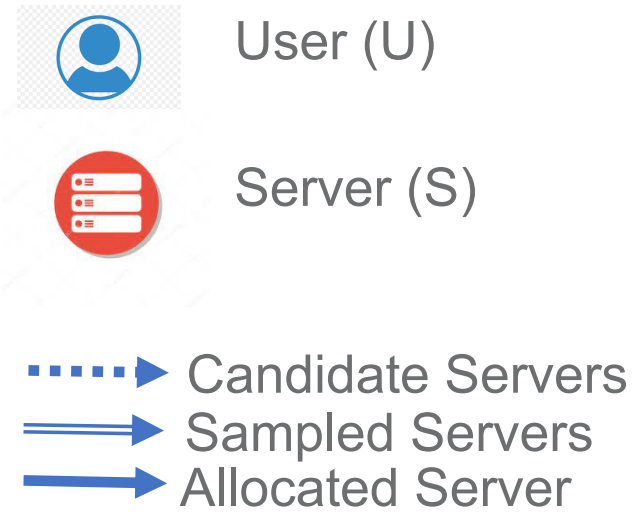
Sampled Servers

Allocated Server

Power of One (POO): Randomly allocate to a server

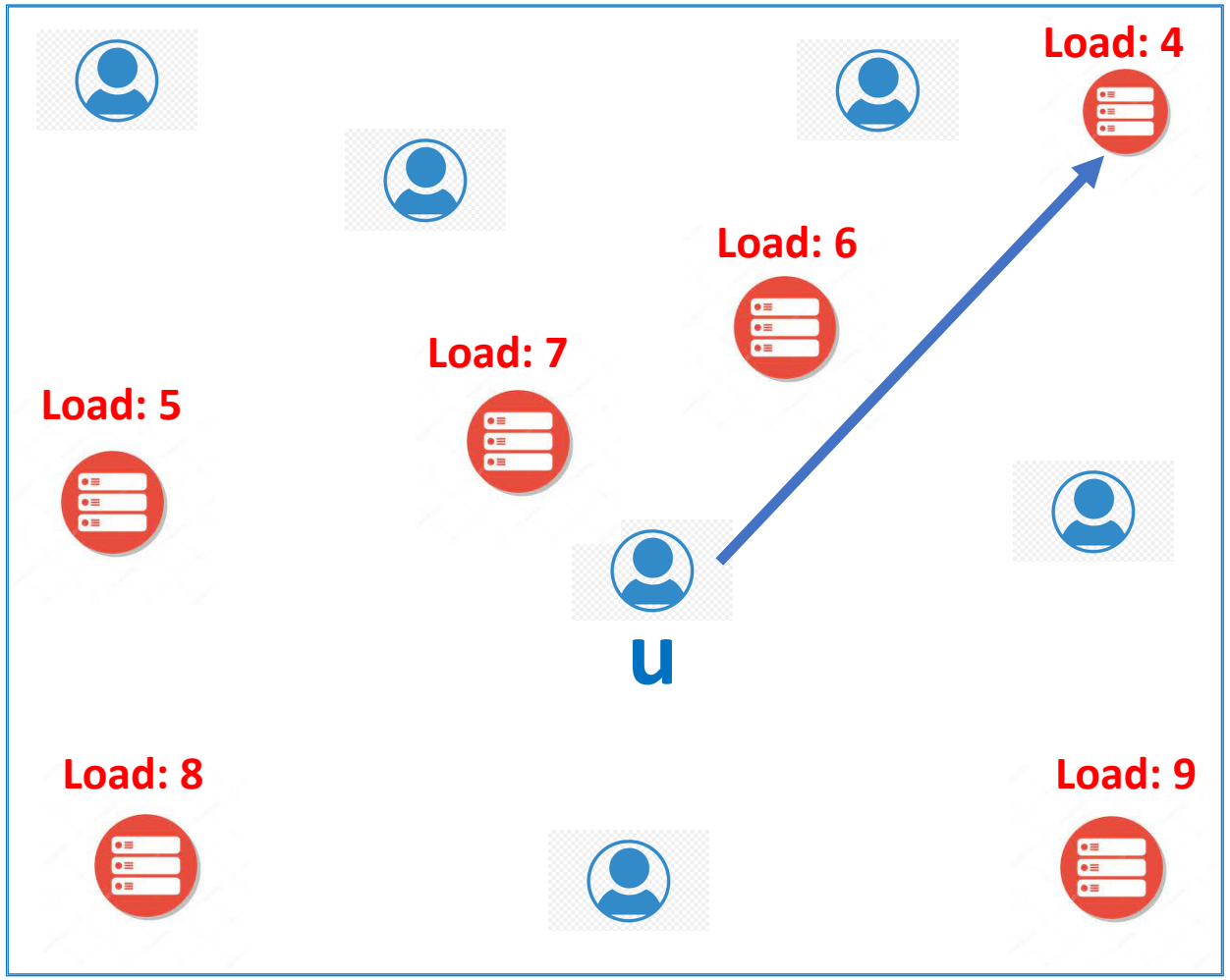
Distributed Load Balancing Policies in 2D

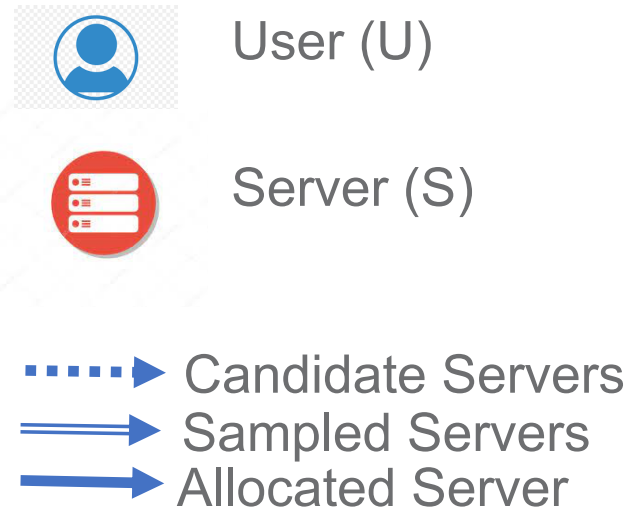




Power of One (POO): Randomly allocate to a server (Load Oblivious)

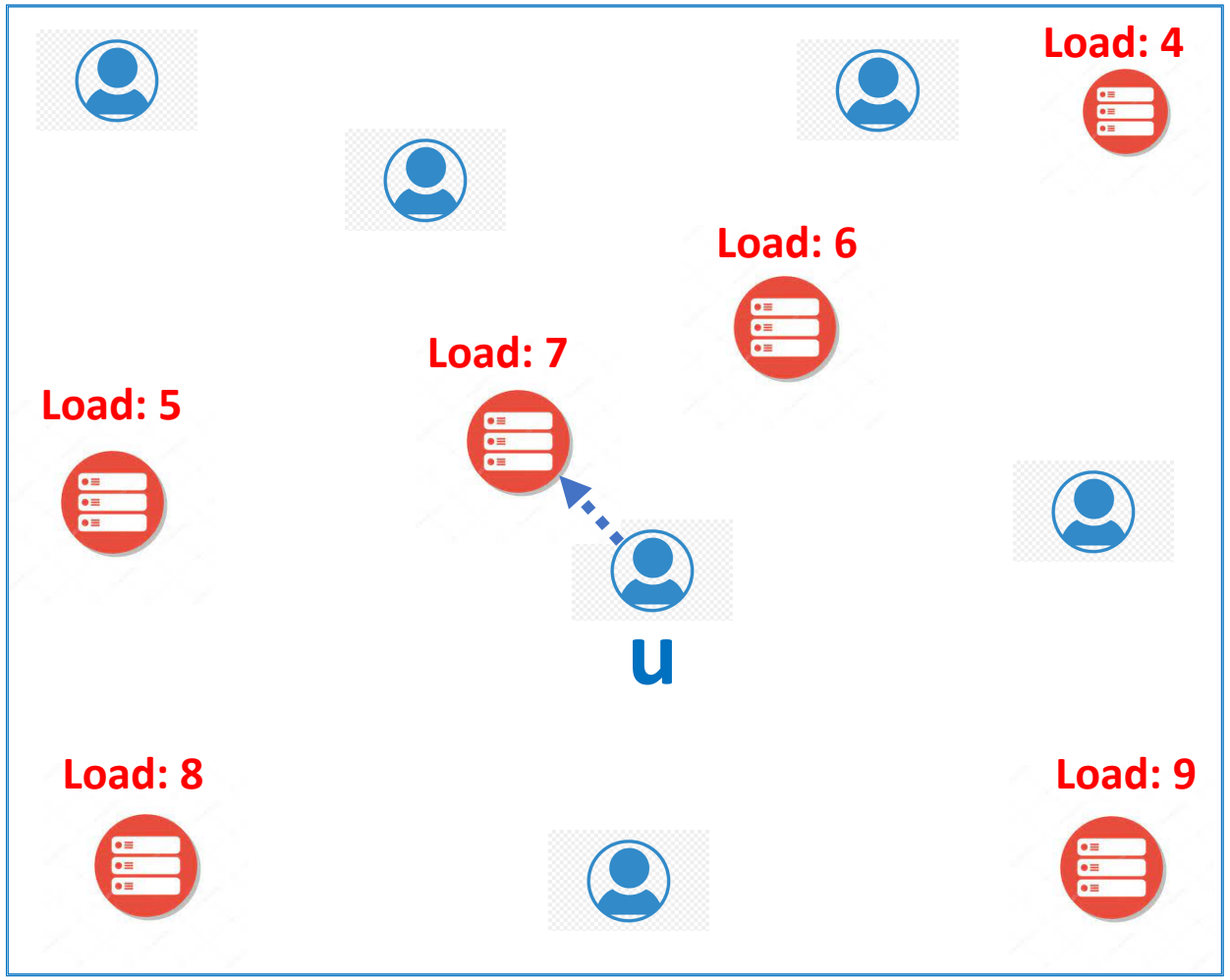
Distributed Load Balancing Policies in 2D

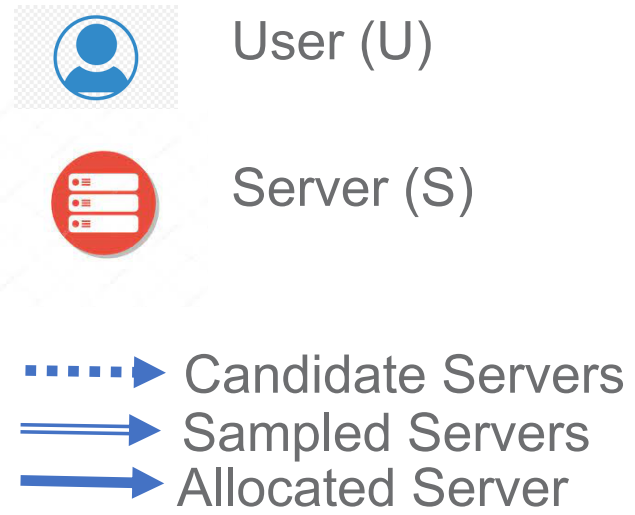




Power of One (POO): Randomly allocate to a server (Load Oblivious)

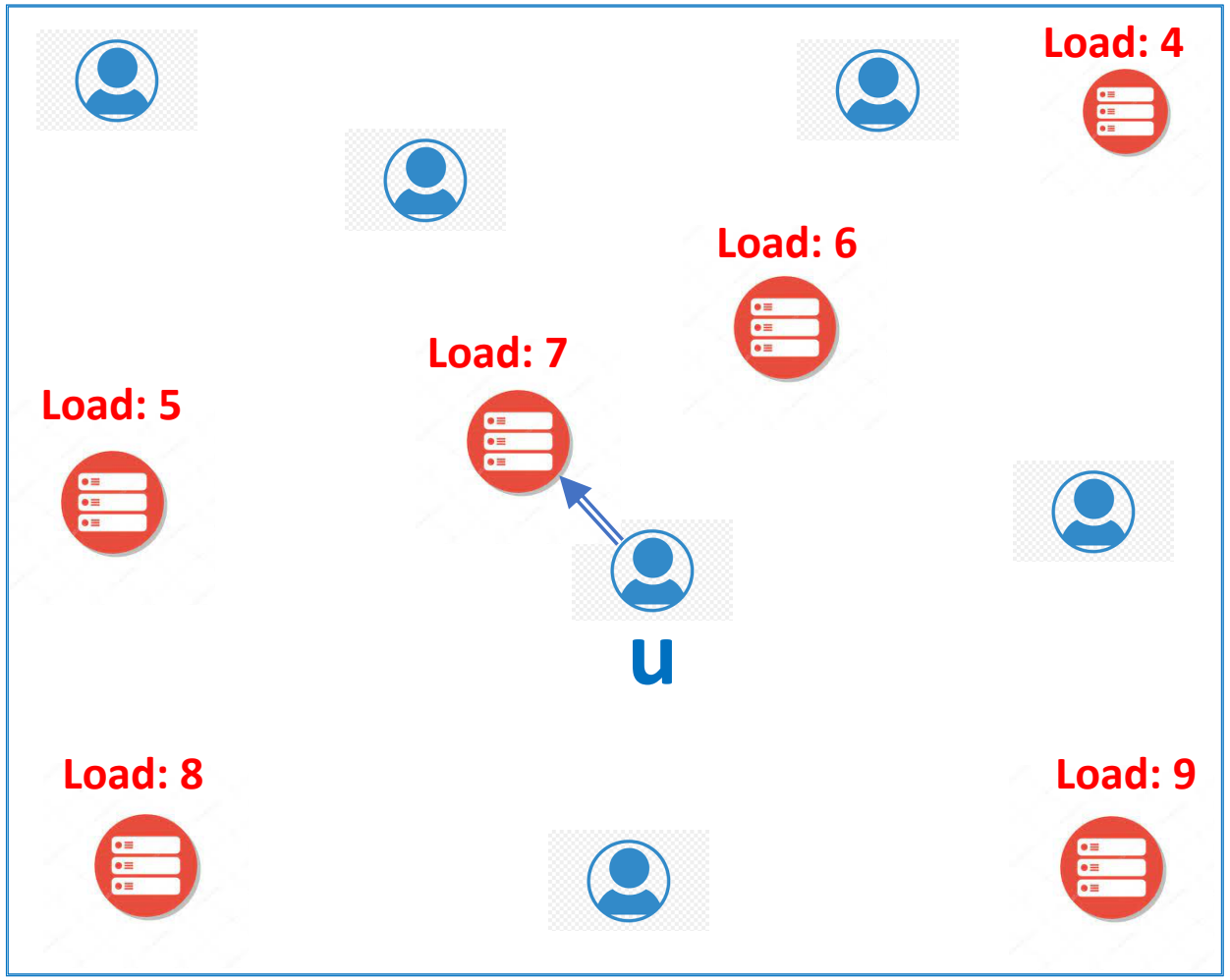
Spatial Policies

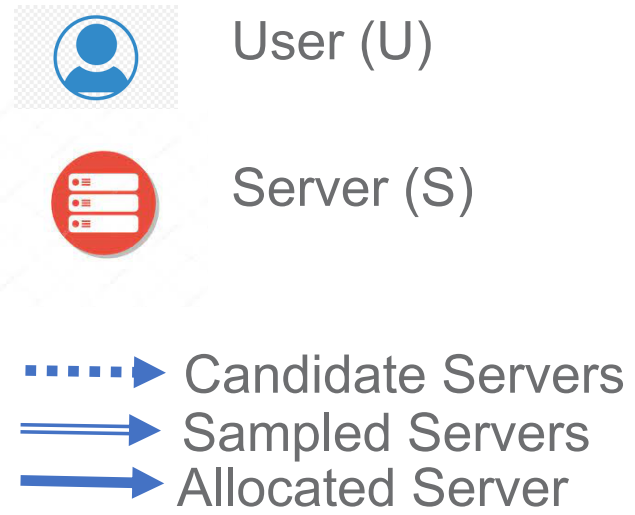




Spatial Power of One (sPOO): Allocate to nearest server

Spatial Policies

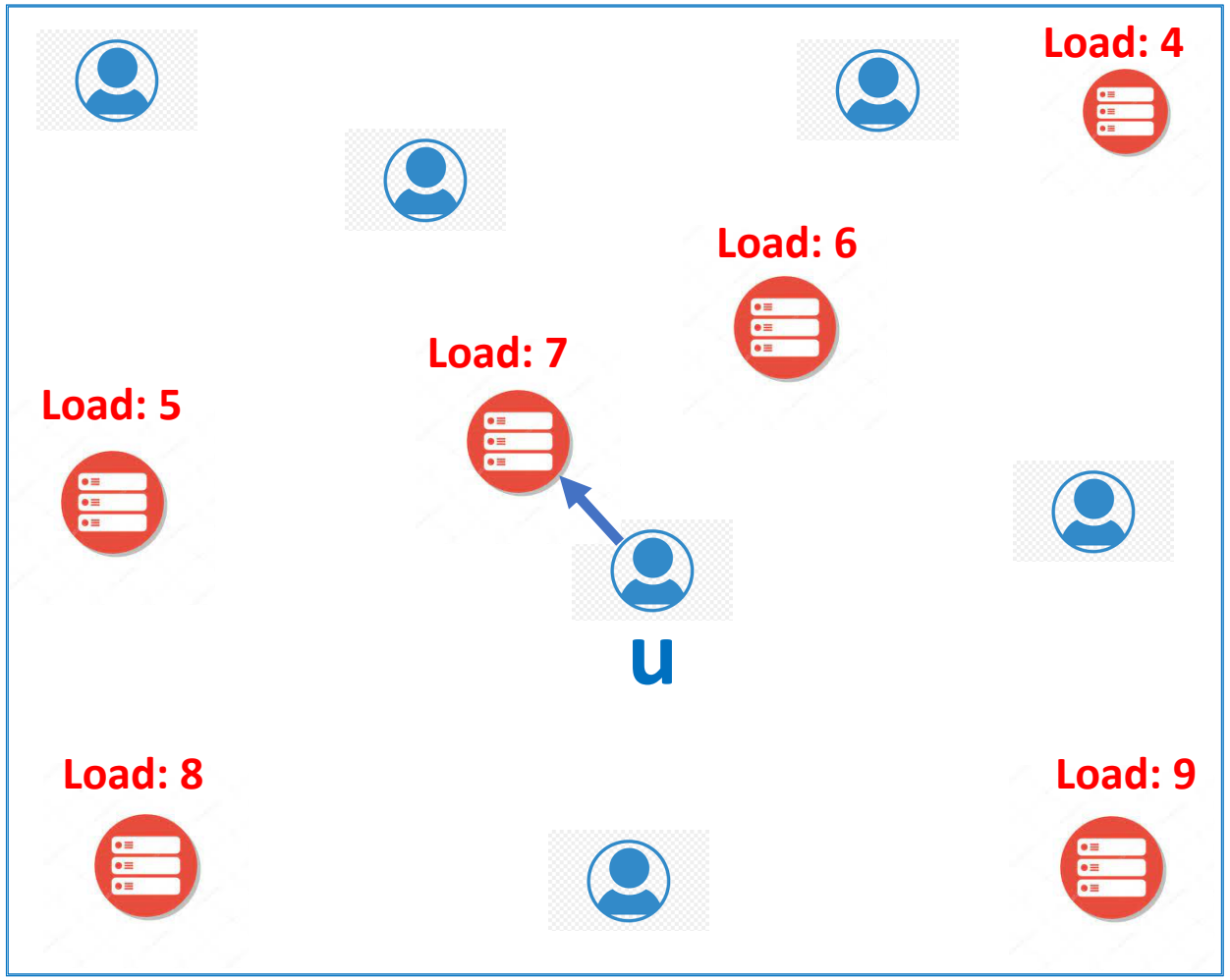


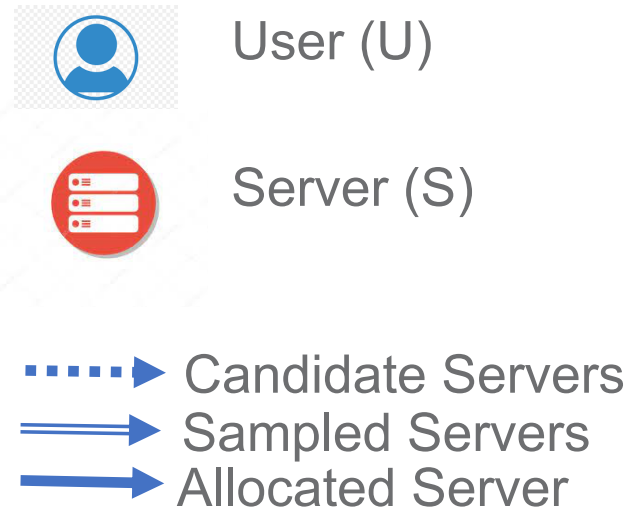


Spatial Power of One (sPOO): Allocate to nearest server

(Load Oblivious)

Spatial Policies





Spatial Power of One (sPOO): Allocate to nearest server (Load Oblivious)