Utilitarian Server Selection

T. Khoa Phan

Joined work with **David Griffin**, **Elisa Maini** and **Miguel Rio**

University College London

Next Generation Networking workshop, 2nd July 2015

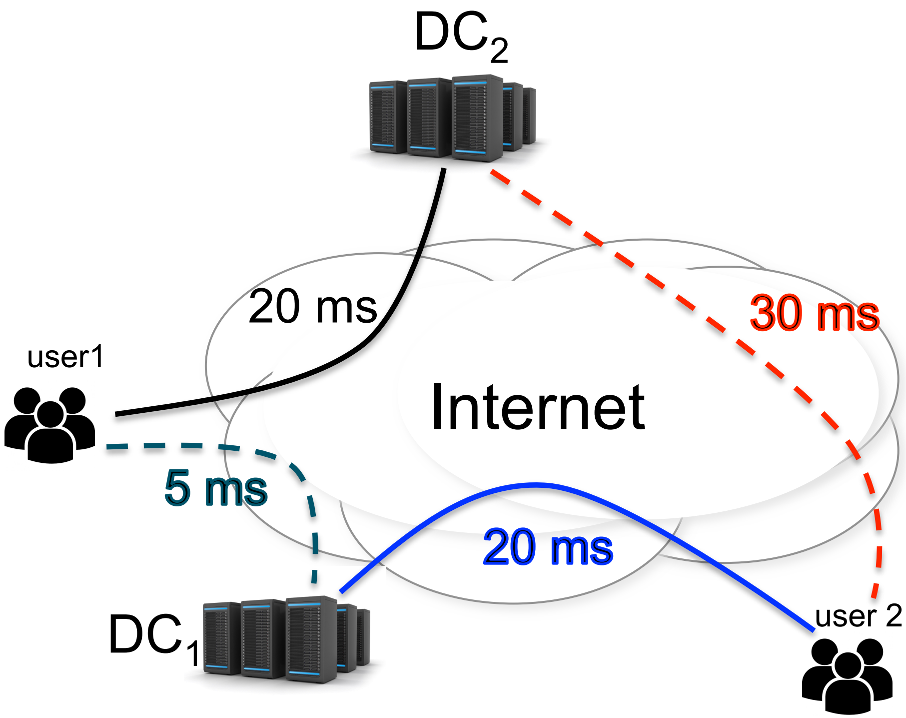
Utilitarianism

"it is the greatest happiness of the greatest number that is the measure of right and wrong”

Jeremy Bentham

Spiritual father of UCL

Motivation

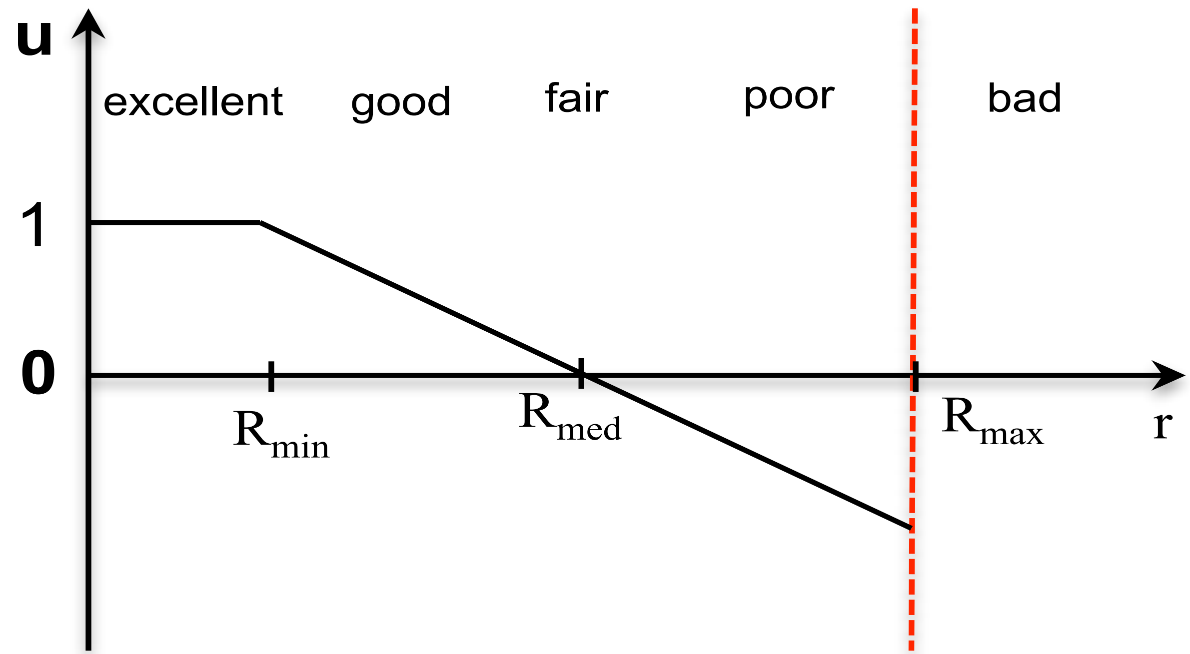


- Server selection using **network** and **server capacity.**
- Utilitarian server selection:
  - User 1 selects DC$_2$
  - User 2 selects DC$_1$

Summary

- Application provider publishes service utility function and replicates service in Execution Zones (EZ) in the Internet
- Service replicas send updates on session slots
- Network providers run resolvers that choose which replicas clients should use (e.g. via DNS) based on:
  - Utility function
  - Network conditions and policies
  - Availability of session slots

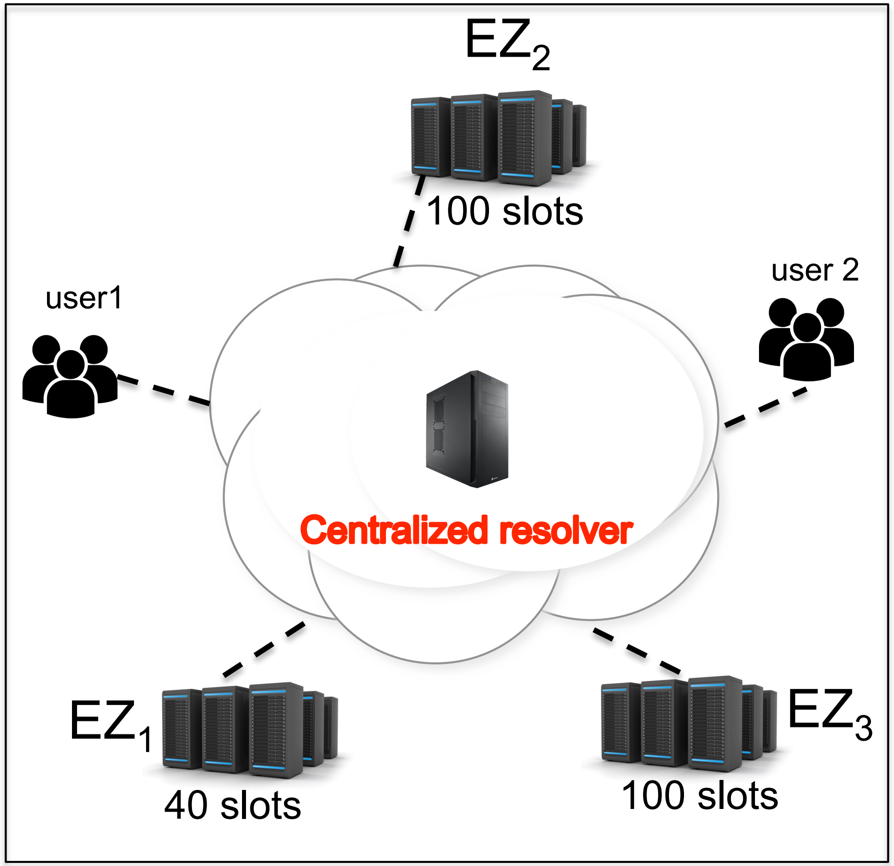
Utility Function



$$u_{ij}(r) = \begin{cases} 1 & \text{if } r \leq R_{min} \\ \frac{-r + R^{ij}_{med}}{R^{ij}_{med} - R^{ij}_{min}} & \text{if } R_{min} < r \leq R_{max} \\ U_b & \text{otherwise} \end{cases}$$

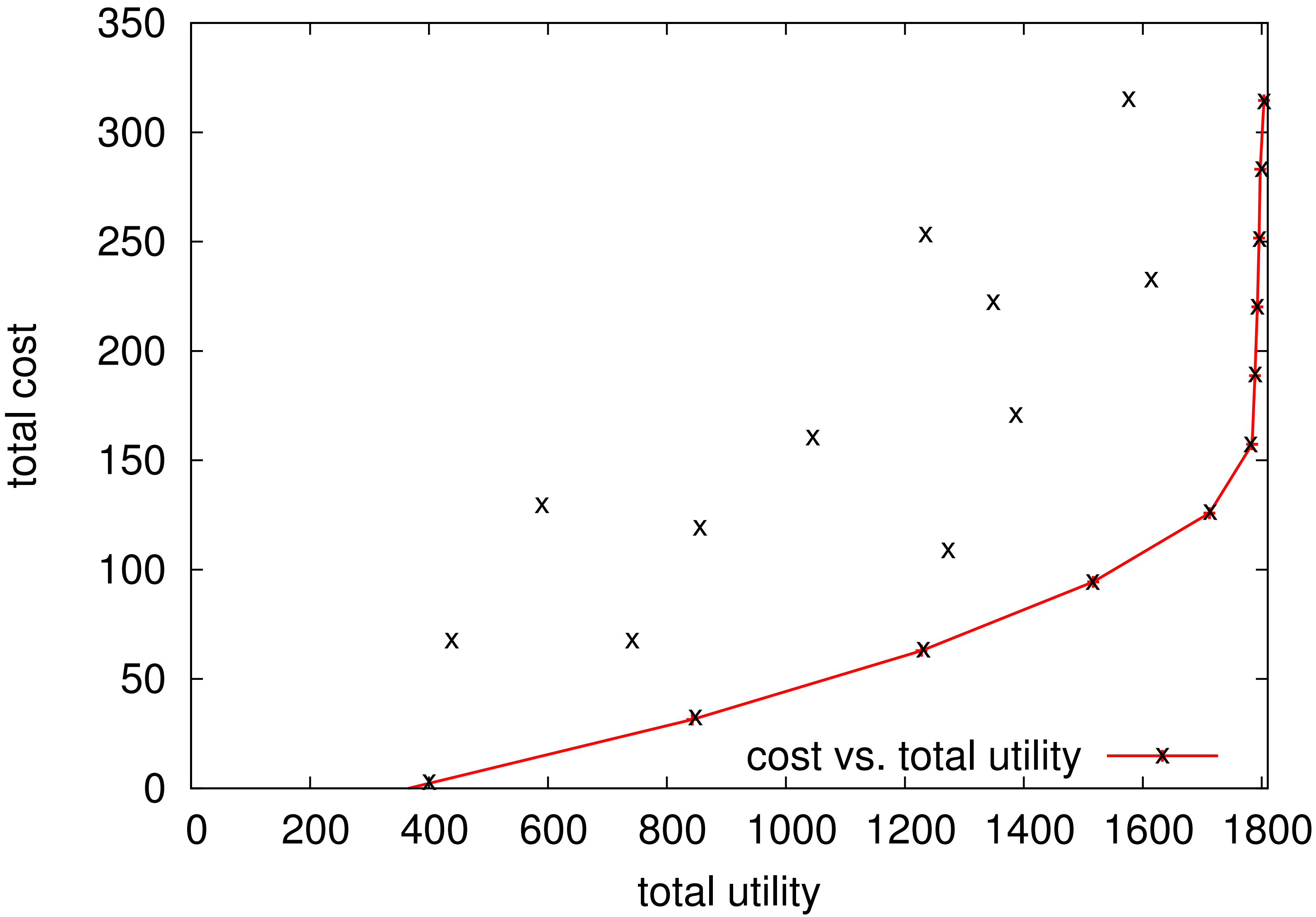
where *i* and *j* are user id and service id, respectively; *r* is response time; U_b has negative value.

Centralized model

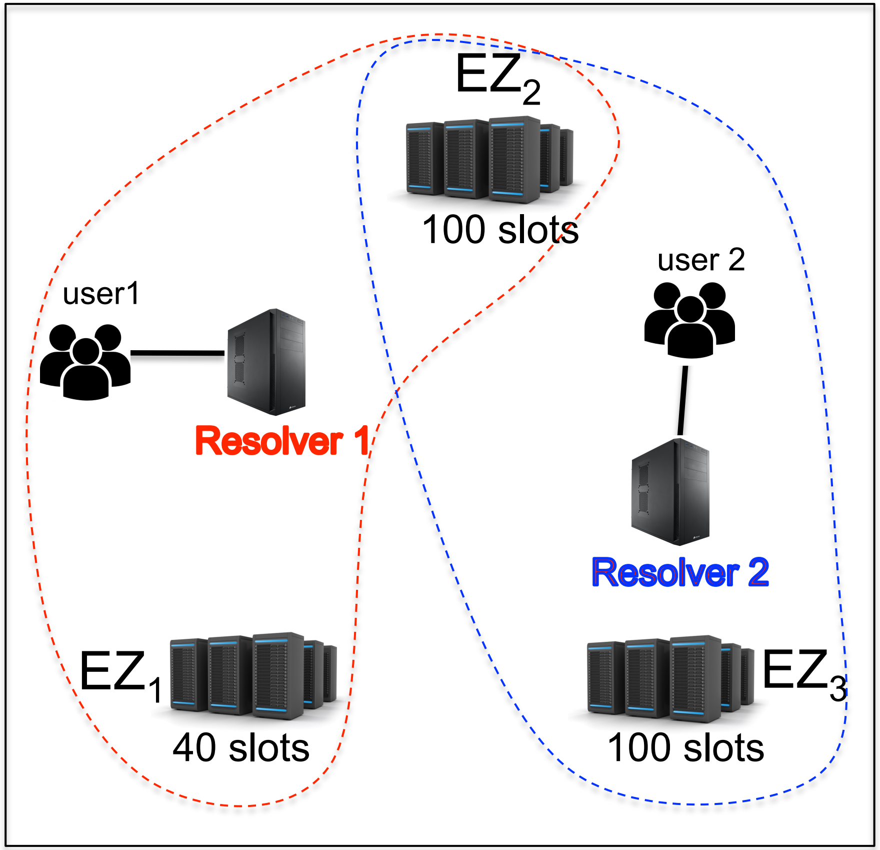


- A (pure) linear program model for utilitarian sever selection.
- Input:
  - All user demands
  - Available session slots of all EZs
  - Budget cost
- Objective: maximizing the total utility.
- Output: fraction of group of users connecting to which EZs.

Pareto Transit Cost vs. Utility



Distributed model



- Each service resolver (SR) has knowledge of:
  - Local user demands
  - Available session slots local EZs and some nearby EZs.
  - A parameter N is used to determine the size of visibility set of EZs.

- Service selection algorithm is independently executed at each resolver to find:
  - Fraction of group users connecting to which EZs (in the visibility set of EZs)

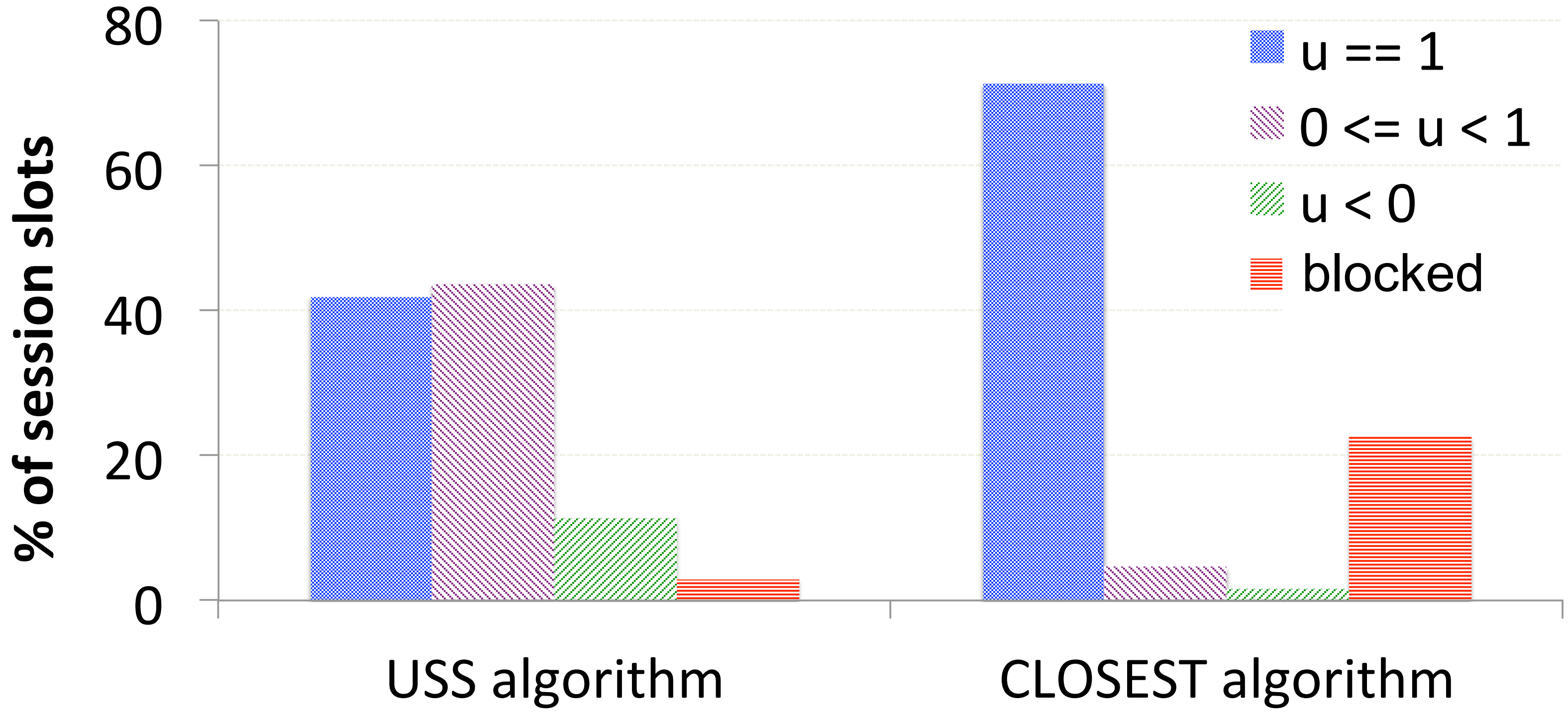
Simulation Setup

- Input data set:

- 658 cities
- 2507 data centers
- 1834 group of users

Source: https://github.com/richardclegg/multiuservideostream

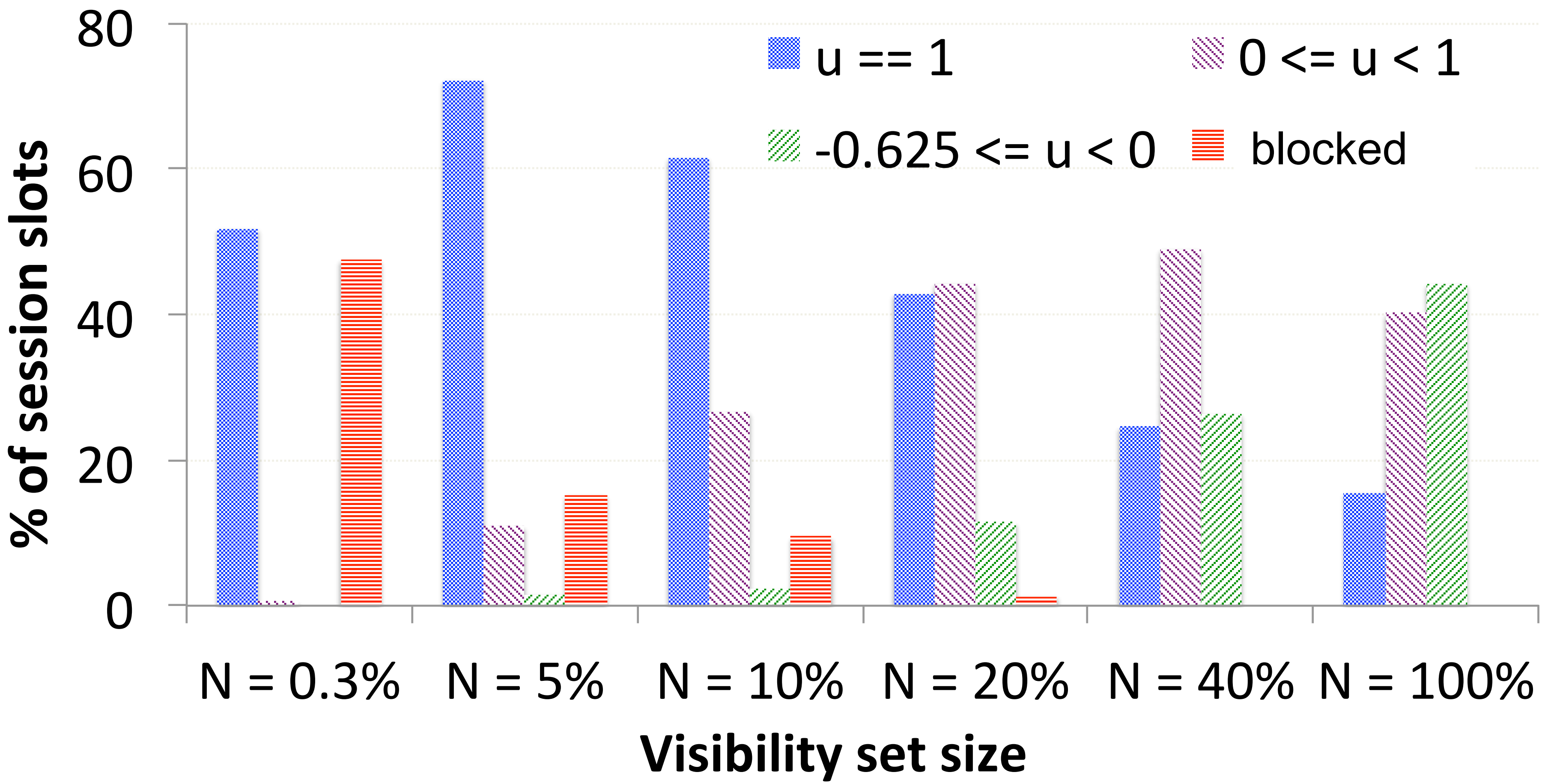
Utility vs. CLOSEST Server Selection



Utility for voice service

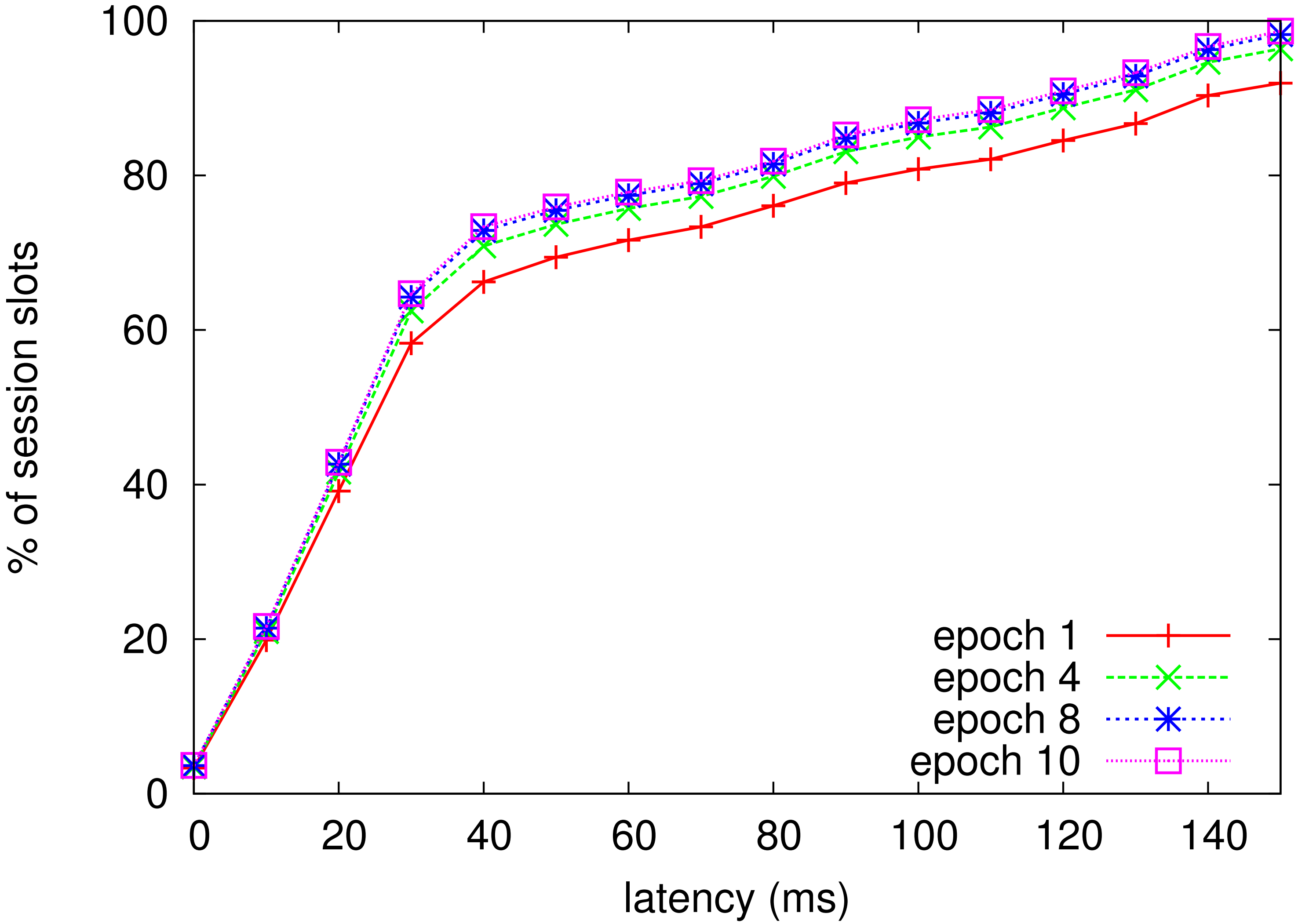
(R_{min} = 20 ms, R_{med} = 100 ms, R_{max} = 150 ms)

Different visibility sets



Distributed algorithm

Convergence of distributed algorithm



CDF latency, visibility set N = 20%

Conclusions

- Server selection tailored to service requirements
- Takes into account network and server performance
- Allows for a wide implementation of traffic policies
- Low overhead