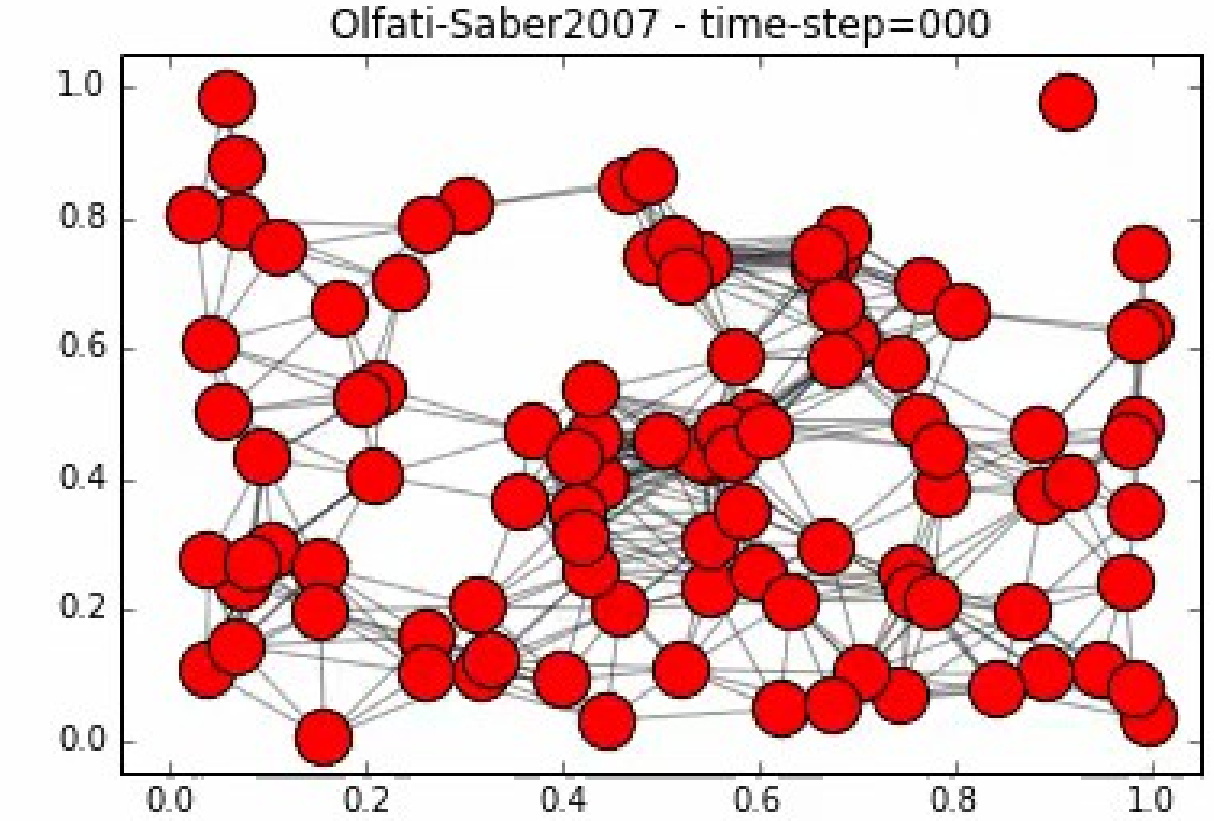
UFMG

Final Project Presentation

Consensus Problem in Multi-Agent Systems

Quicker Consensus with Intelligent Buffer

Alexander Jahn

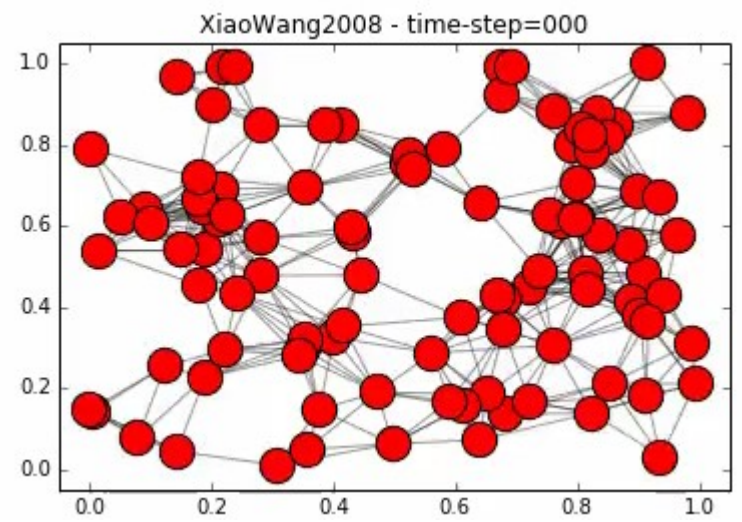


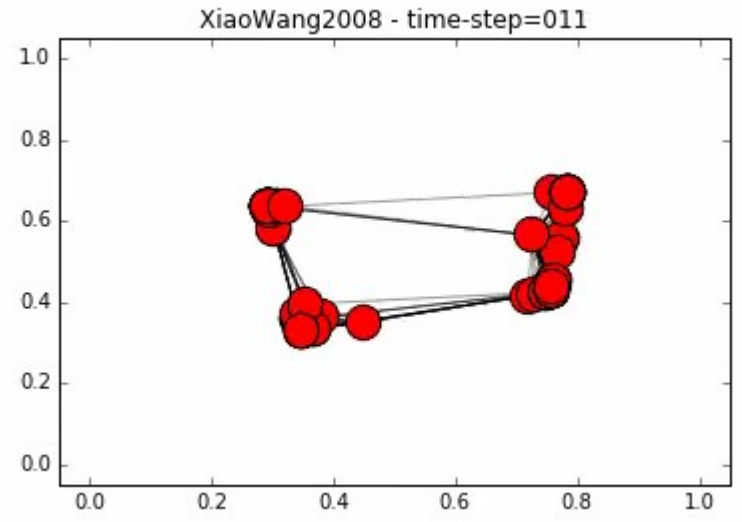
Outline

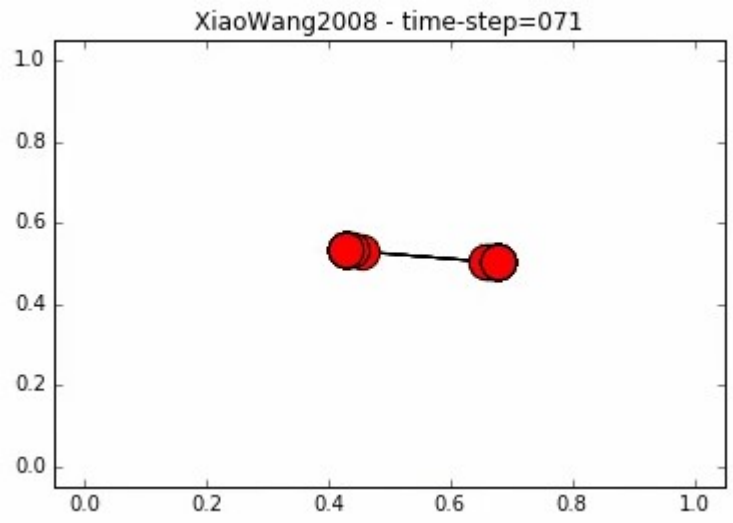
- Multi-Agent Consensus and Problem Formulation
- Agent and Communication Model
- Proposed Intelligent Buffer
- Implementation
- Results
- Future Work

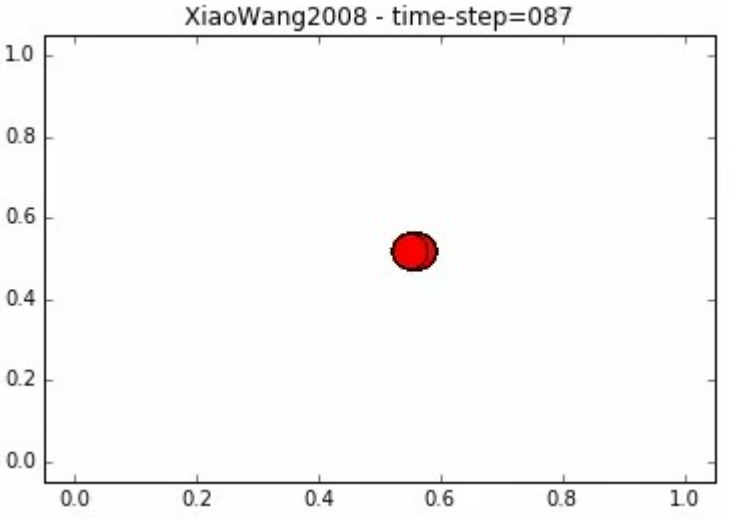
Multi-Agent Consensus

- In a networks of agents (or dynamic systems), consensus means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents.
- A consensus algorithm (or protocol) is an interaction rule that specifies the information exchange between an agent and all of its neighbors on the network.
- **Focus on decentralized Consensus for single integrator model with implementation for rendezvous-problem**

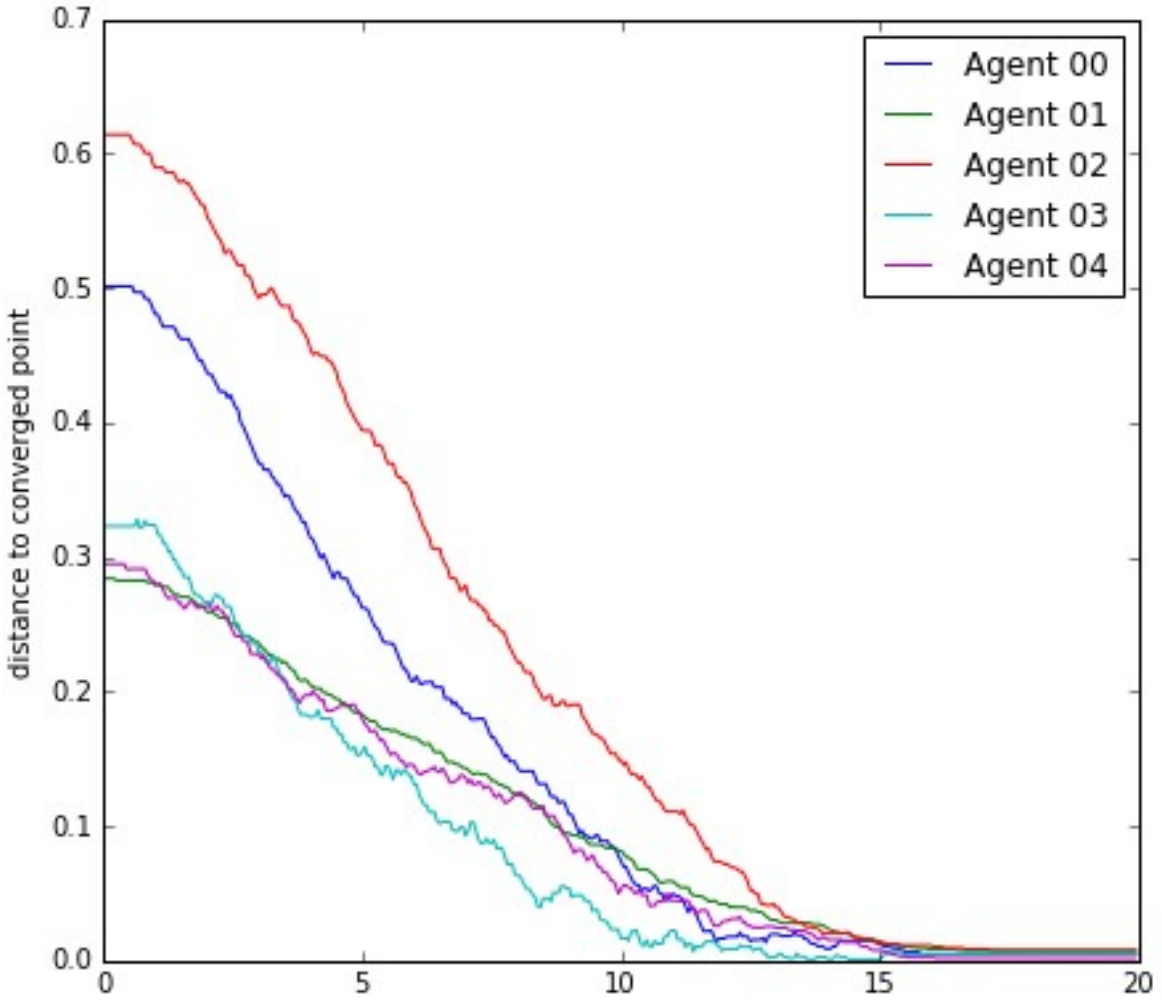

XiaoWang2008 - time-step=000
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


XiaoWang2008 - time-step=011
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


XiaoWang2008 - time-step=071
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


XiaoWang2008 - time-step=087
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Rendezvous-Problem



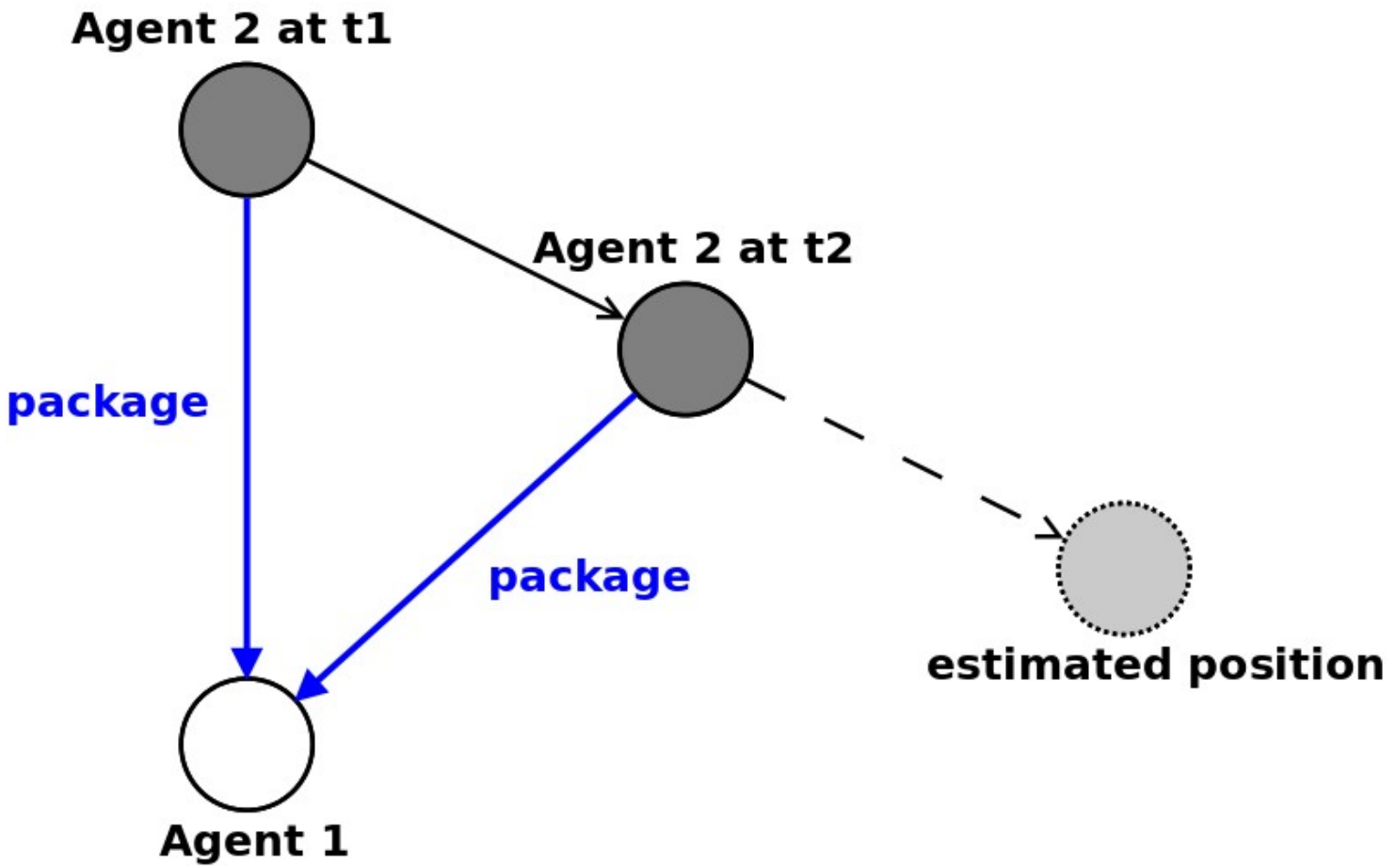
Agent and Communication Model

- Different models are possible
- Most basic one is the single integrator
- The communication is implemented with synchronous and asynchronous behavior
  - Asynchronous with lower and upper bounded uniform probability distribution
- The time of each agent is uniformly distributed with lower and upper bound

Proposed Intelligent Buffer

- Two approaches implemented and tested
  - Extrapolate better data from multiple packages
  - Keep the newest data in memory
- Default procedure is to discard old information and only use a single data set of each sender

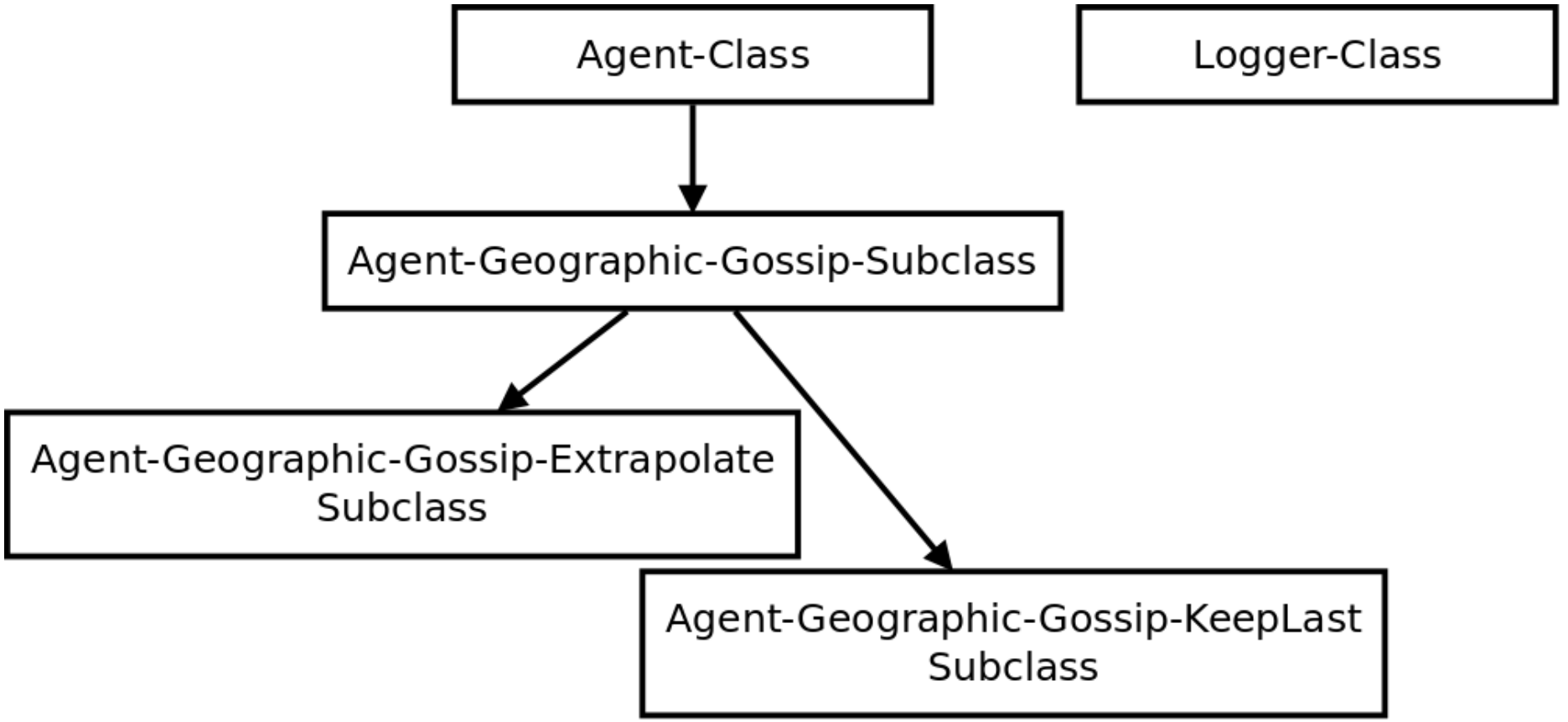
Extrapolate Better Data



Implementation

- IPython framework (Jupyter)
- Packages utilized:
  - Networkx (for graph related methods)
  - Simpy (Discrete Event Simulation, DES)
- Uploaded to my Github:
  - https://goo.gl/OH7S56
  - Video compilation requires ffmpeg (not working on Github online)
  - Loads slowly (1.8Mb)

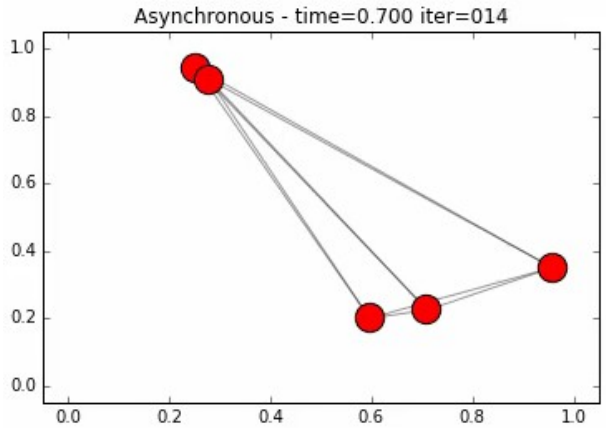
Implemented Class Hierarchy

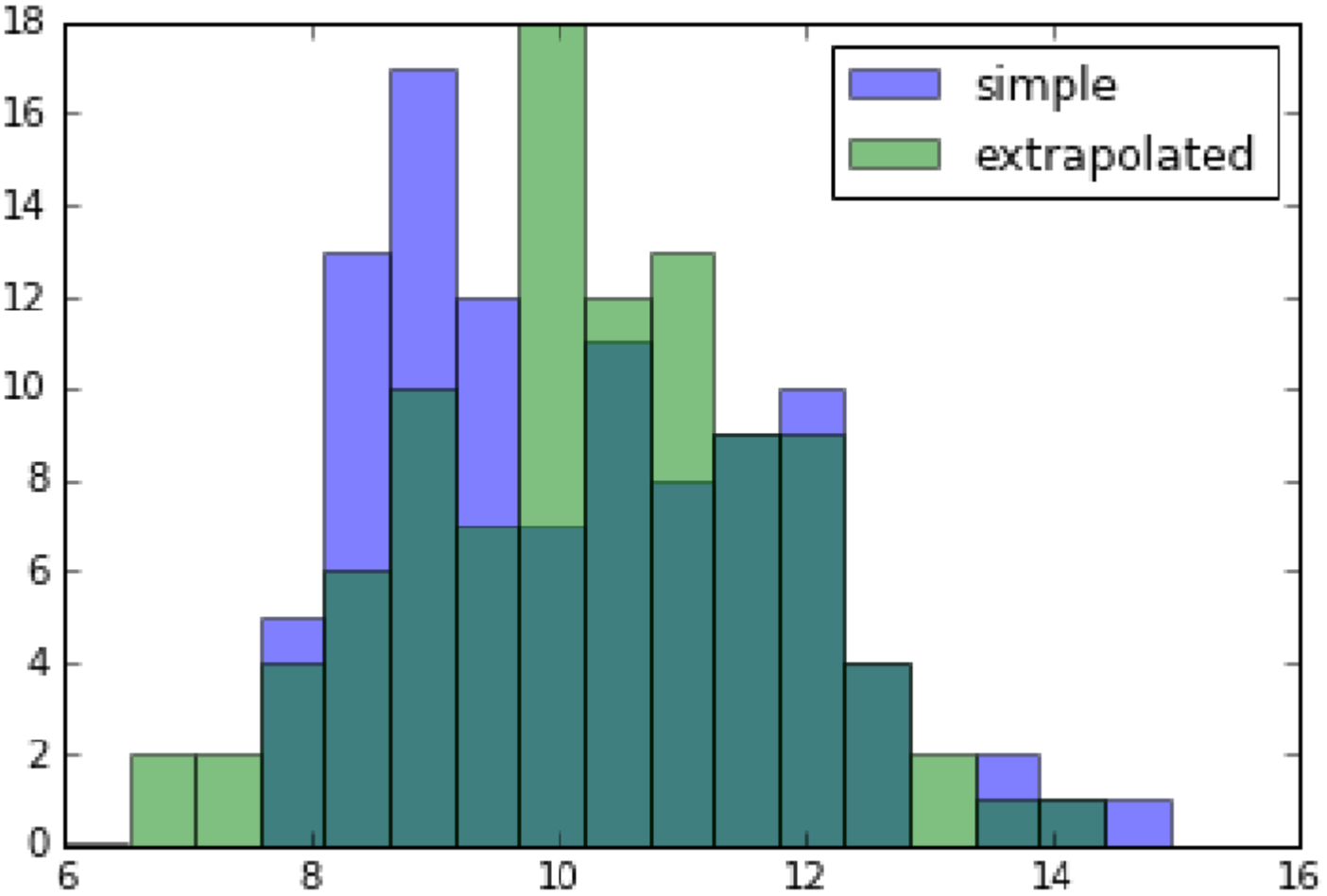


Results: Extrapolation vs Simple

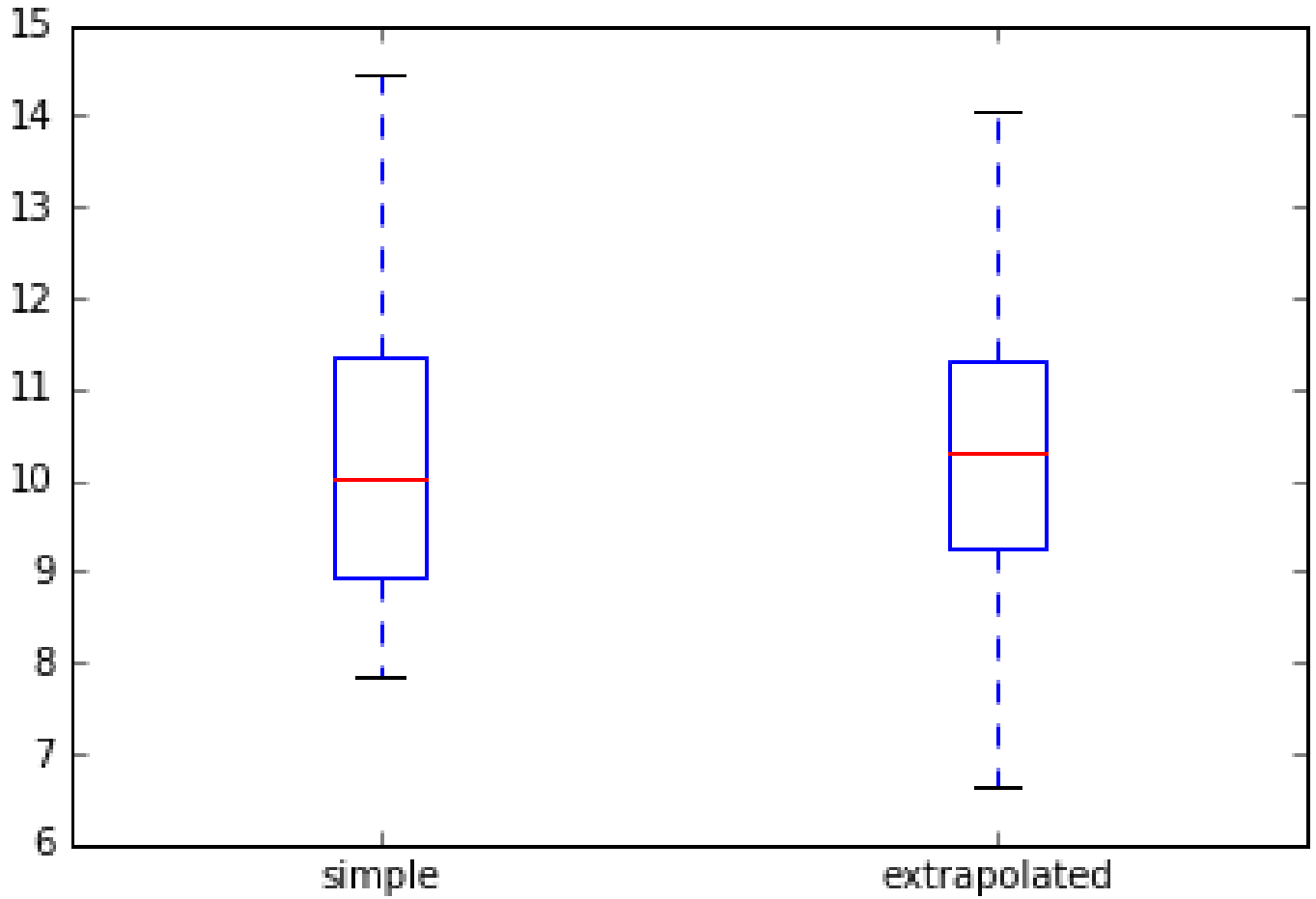
- Extrapolation is not that efficient in practice

```
Results for 100 samples:
        simple  extrapolated
avr     10.19   10.28
std     1.54    1.52
dR avr  15.12%
dR std  6.56%
with
{'Tbufmax': 1.0,
 'Tbufmin': 0.1,
 't_delay_max': 1.0,
 't_delay_min': 0.1 }
```

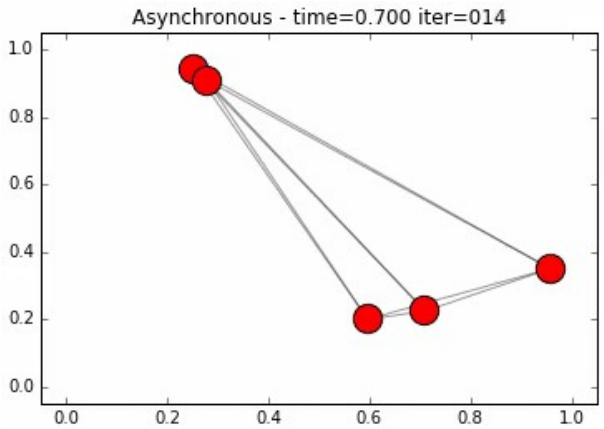


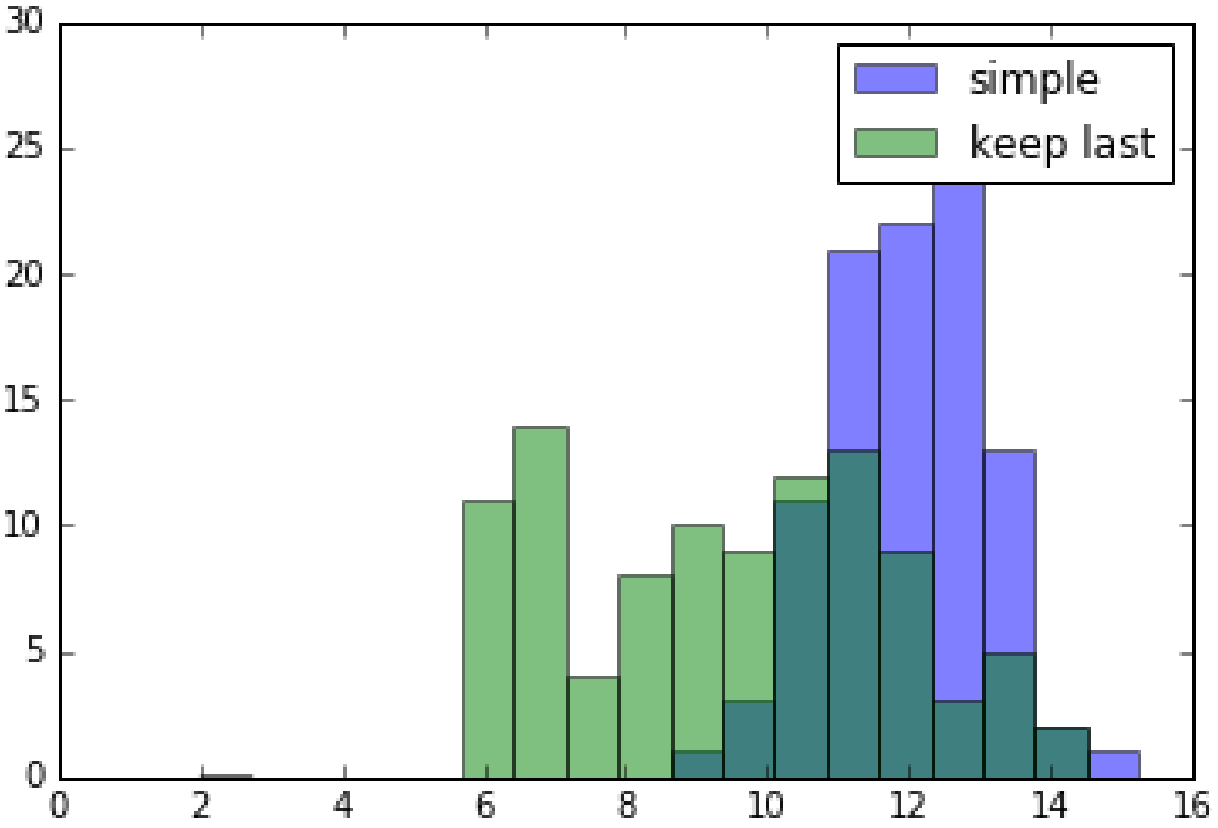


Results: Extrapolation vs Simple

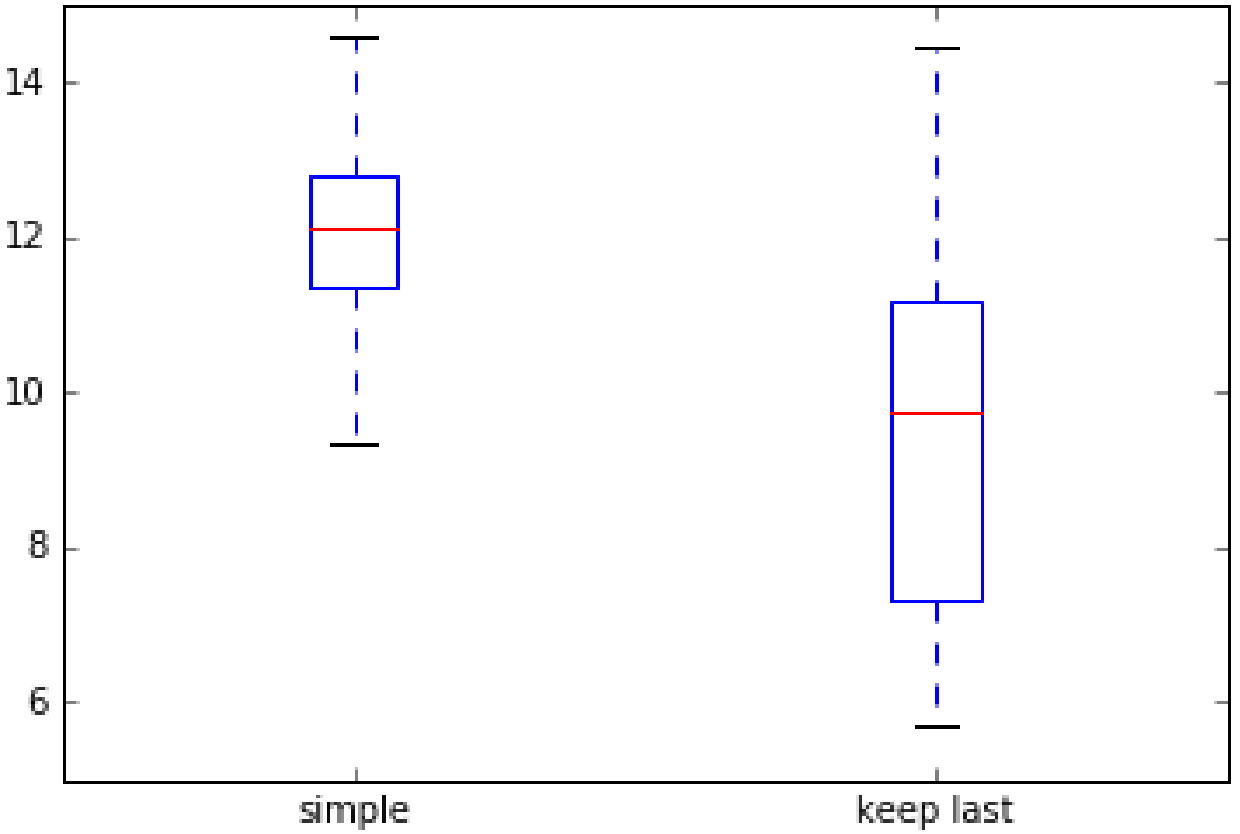


- Works well


Asynchronous - time=0.700 iter=014
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


simple
keep last
30
25
20
15
10
5
0
0
2
4
6
8
10
12
14
16

Results: Keep Last vs Simple



Future Work

- More complex models for the agent and/or communication (closer to reality, the results should imply practical use)
- Implementation of time-invariant network topology