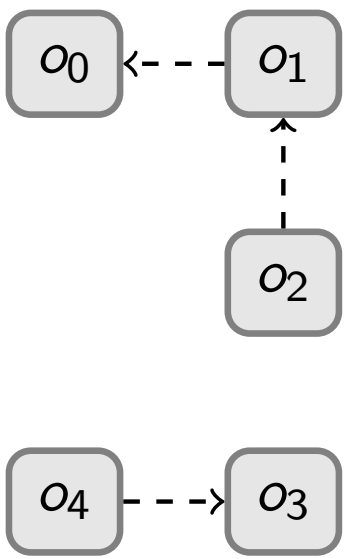
ABS-NET: Fully Decentralized Runtime Adaptation for Distributed Objects

Karl Palmskog
palmskog@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

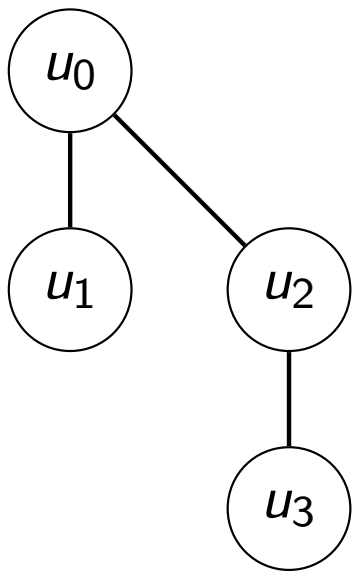
Joint work with Mads Dam, Andreas Lundblad and Ali Jafari

Setting

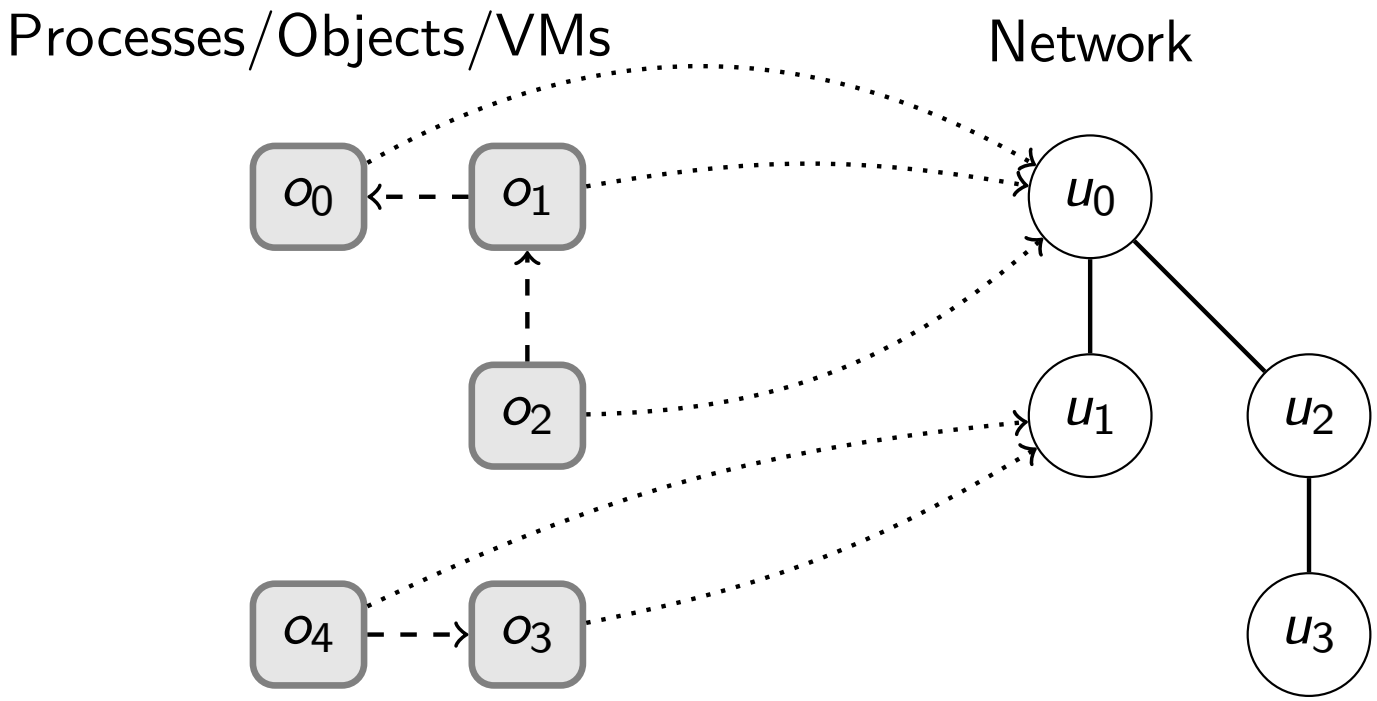
Processes/Objects/VMs



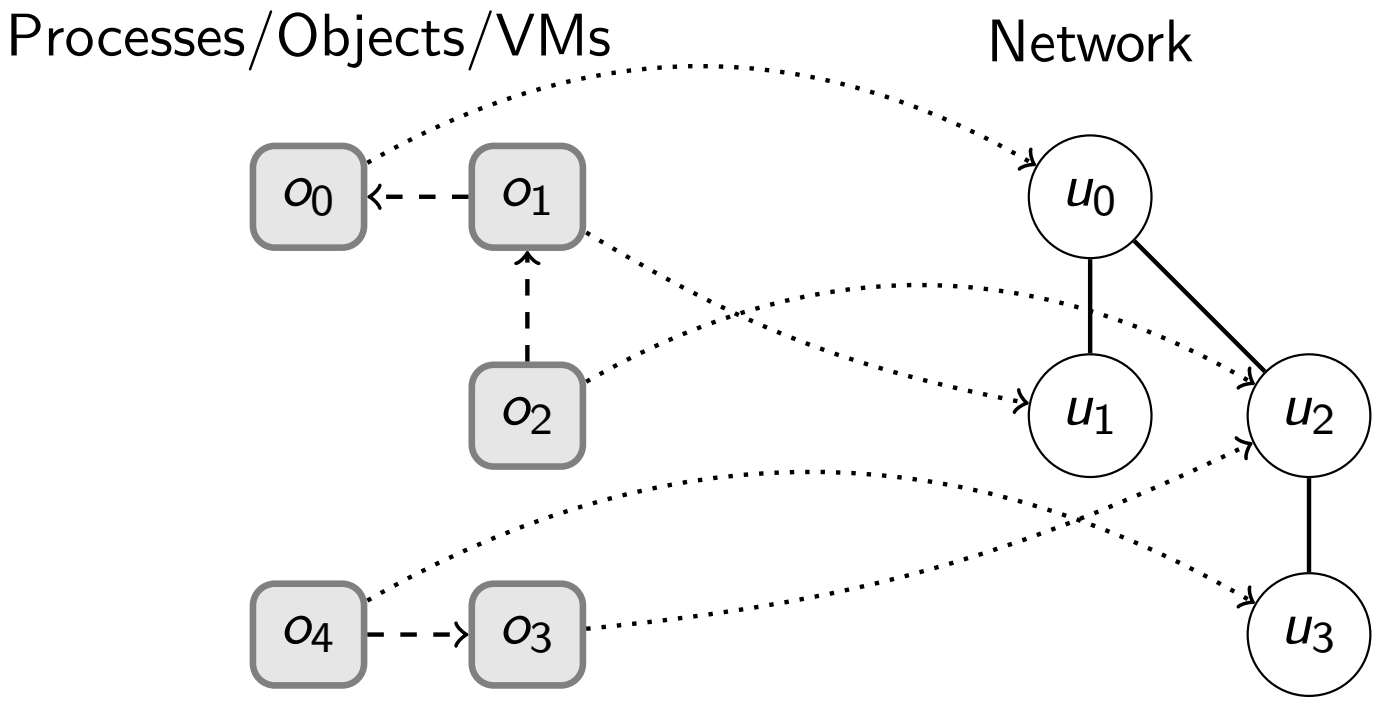
Network



Assignment by Programmer?



Assignment by Centralized Scheduler?



Adaptation Requirements

Minimize:

- (global) utilization
- (global) power consumption
- (local) response times
- ...

Context

- ABS language of distributed objects (actors)
- ABS-NET semantics for network execution, object mobility

Simulator

- Developed simulator for ABS-NET execution of ABS programs
- Enabled instrumenting simulator with migration procedures

Investigation of Decentralized Adaptability

1. Created synthetic benchmark scenarios
2. Measured load and communication overhead

ABS

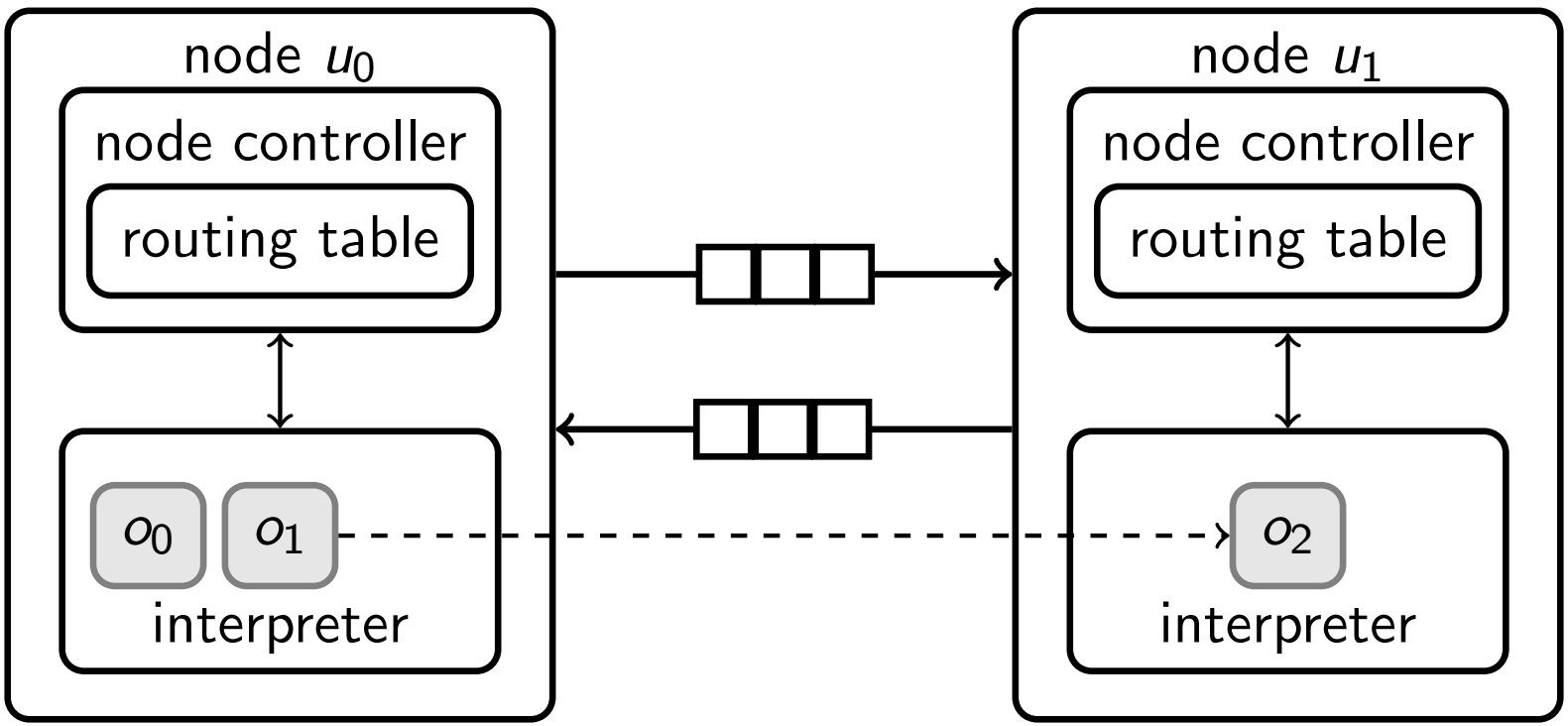
```
interface CastNode {
    Int aggregate();
}

class LeafCastNode(Int val) implements CastNode {
    Int aggregate() { return val; }
}

class BranchCastNode(Int val, CastNode left, CastNode right)
  implements CastNode {
   Int aggregate() {
      Fut<Int> fLeft = left!aggregate();
      Fut<Int> fRight = right!aggregate();
      Int aggregateLeft = fLeft.get;
      Int aggregateRight = fRight.get;
      return val + aggregateLeft + aggregateRight;
    }
}
```

Standard ABS Semantics

ABS-NET Semantics



Object Location Transparency

- Need to transport application-level messages between objects
- Objects mobile across nodes
- Nodes route messages to current location of recipient object

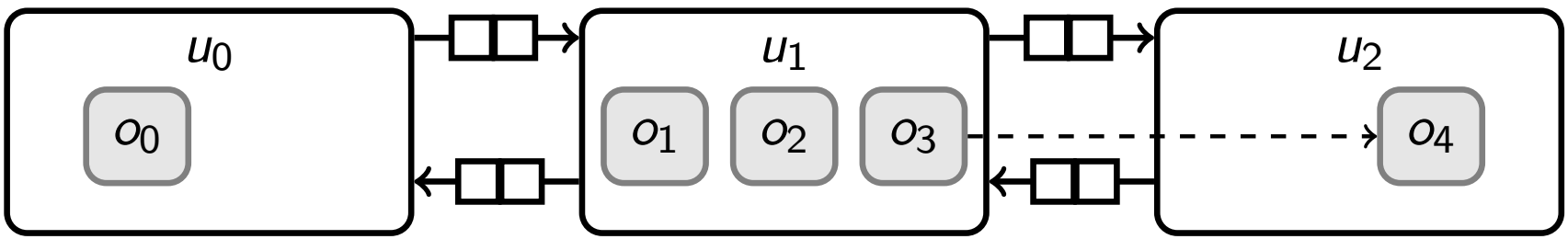
Location-Independent Routing

- Route based on search identifiers (object names)
- Routing information continually exchanged between nodes
- Example node-local routing entries:
  - o_0 found in direction of neighbour u_1, distance 2 hops
  - o_1 found in direction of neighbour u_0, distance 5 hops
- This is a novel approach, explored in ongoing related work!

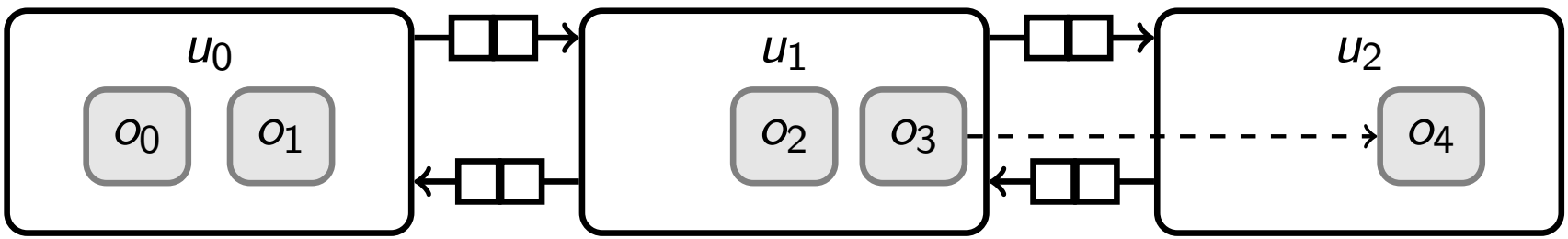
Quality of Service Objectives

- Node load (# active tasks)
- Arc load (# inter-node application messages sent over time)
- Message latency (# message hops)

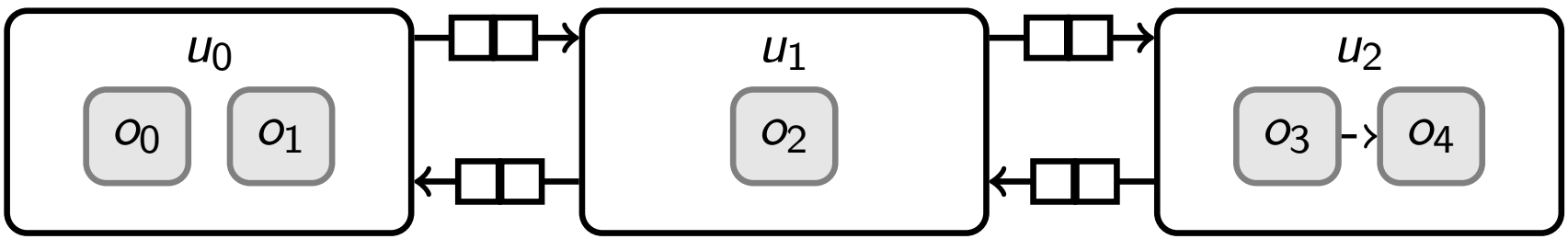
Runtime Adaptability in Practice



Runtime Adaptability in Practice, continued



Runtime Adaptability in Practice, continued



Adaptability Assumptions

- Migration decisions taken locally
- Analysis only during runtime
- Static network

Simulator

- Written in Java
- Implements ABS-NET backend for ABS frontend
- Node controllers are Java threads, arcs are TCP sockets
- Network topology and migration configurable through CLI

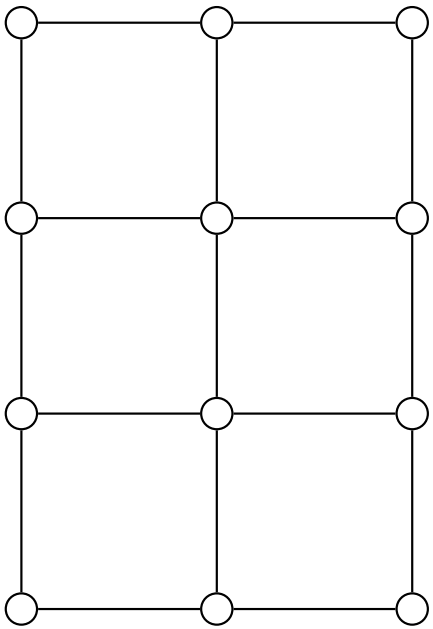
Migration Procedures

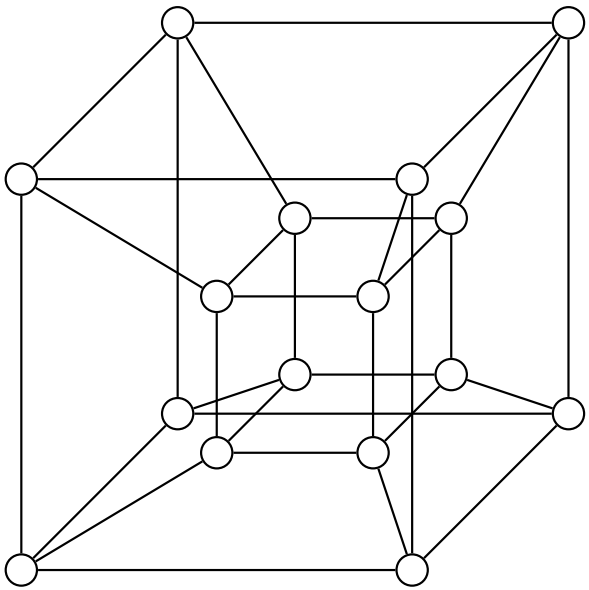
for each active object o **do**
  u' is a neighbour chosen uniformly at random
  l is the current load
  l' is the last known load of u'
  if $l > l' + 1$ **then**
    send o to u' with probability $1 - l'/l$

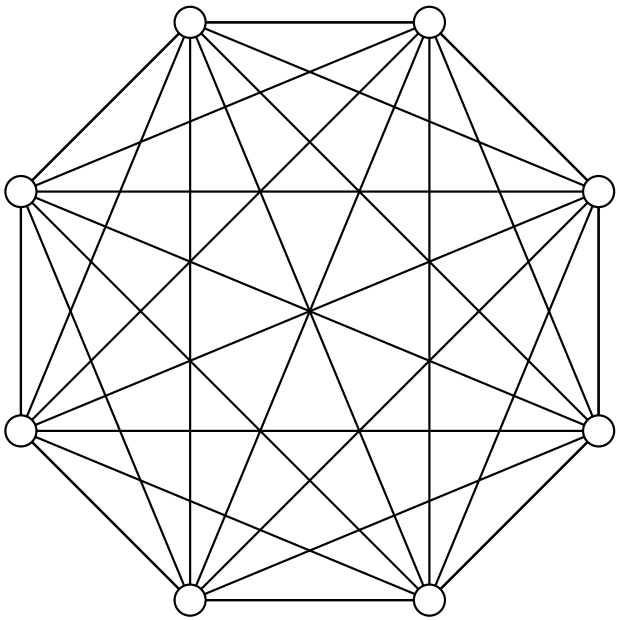
Migration Procedures, continued

for each active object o **do**
  u' is a neighbour chosen uniformly at random
  l is the current load
  l' is the last known load of u'
  if $l > l'$ **then**
    send o to u' with probability $1 - l'/l$

Network Topologies



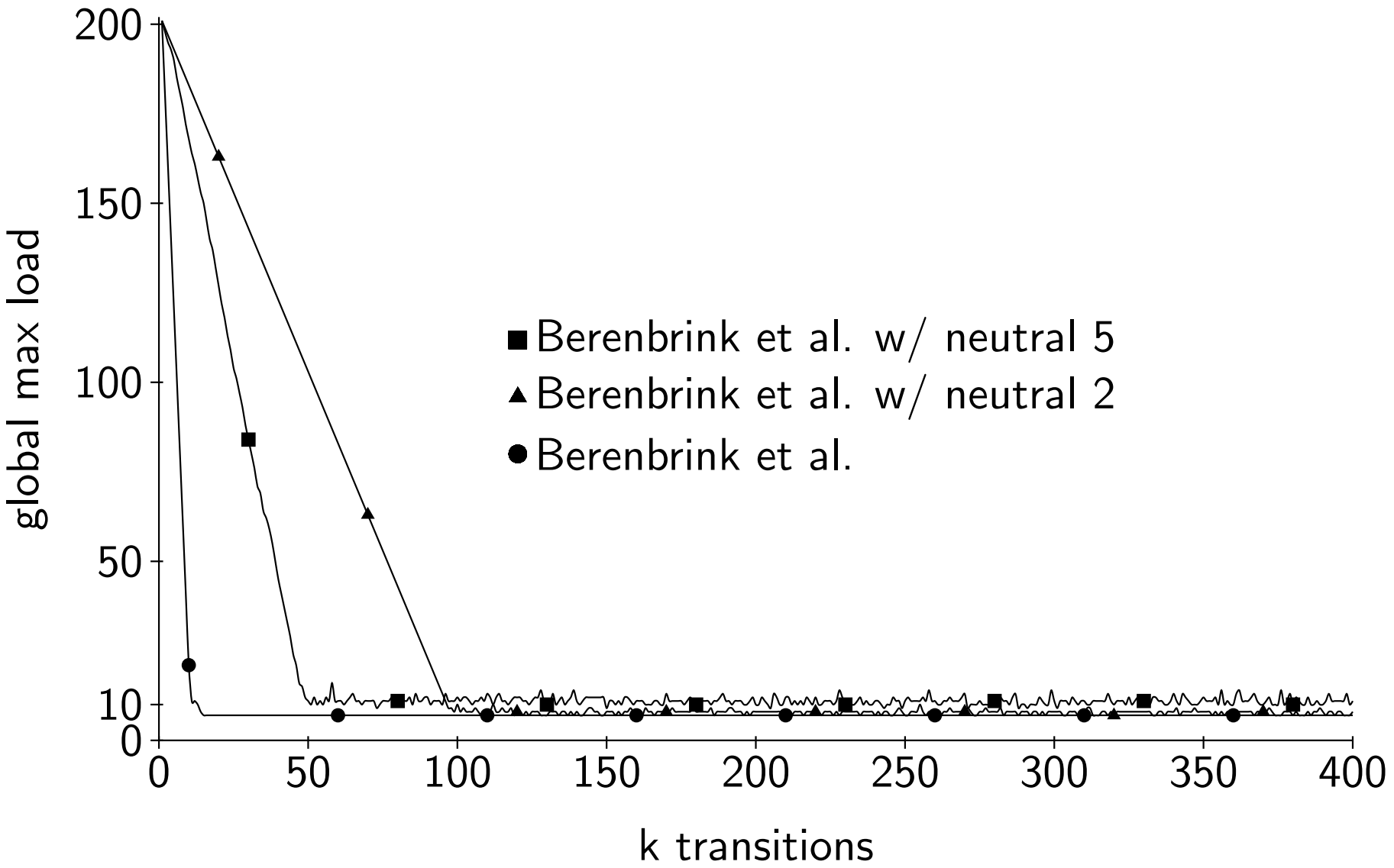




IndependentTasks.abs

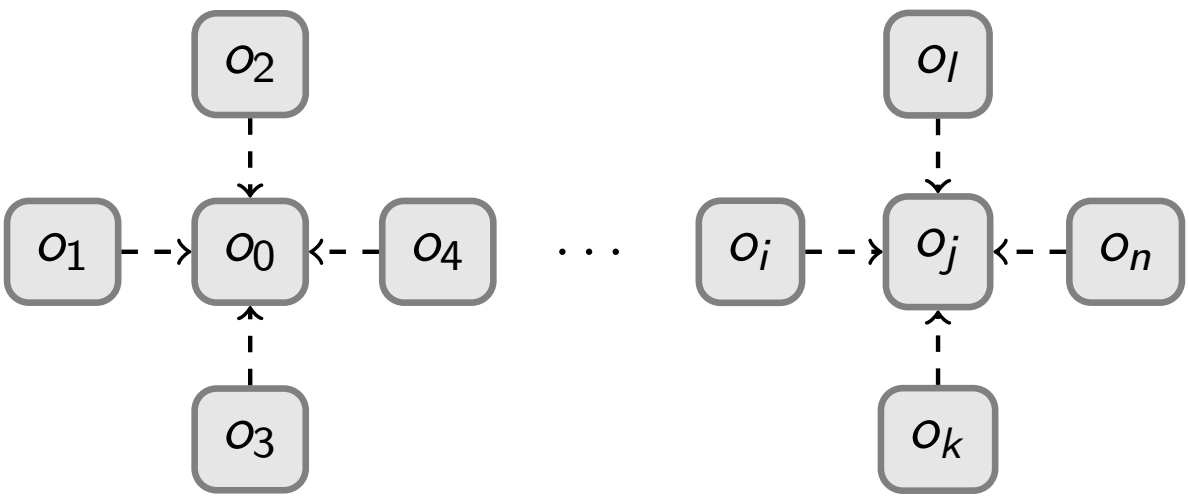
- Creates objects that run tasks without communication
- Used to evaluate procedures w.r.t. pure load balancing

`IndependentTasks.abs` on Hypergraph

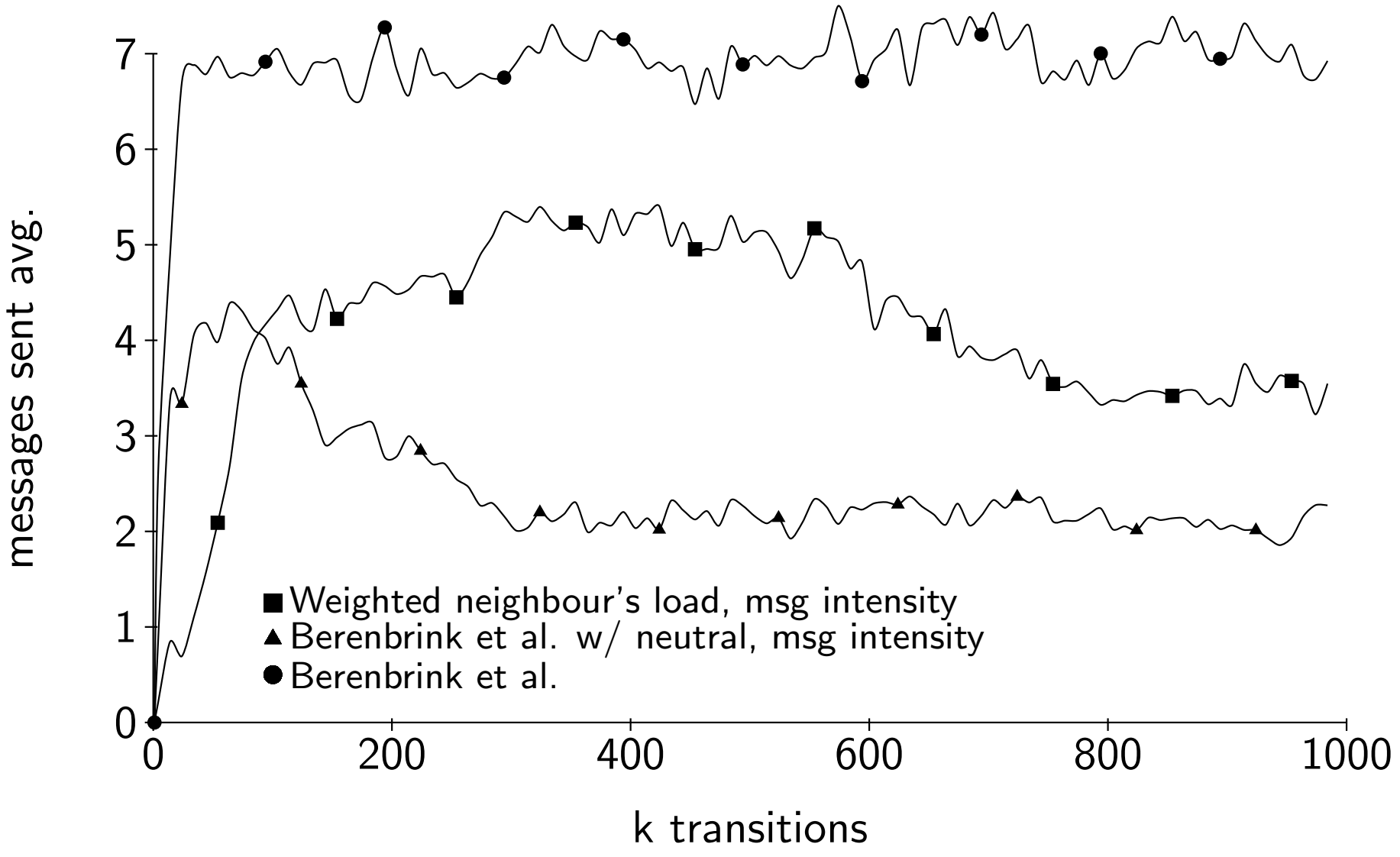


Star.abs

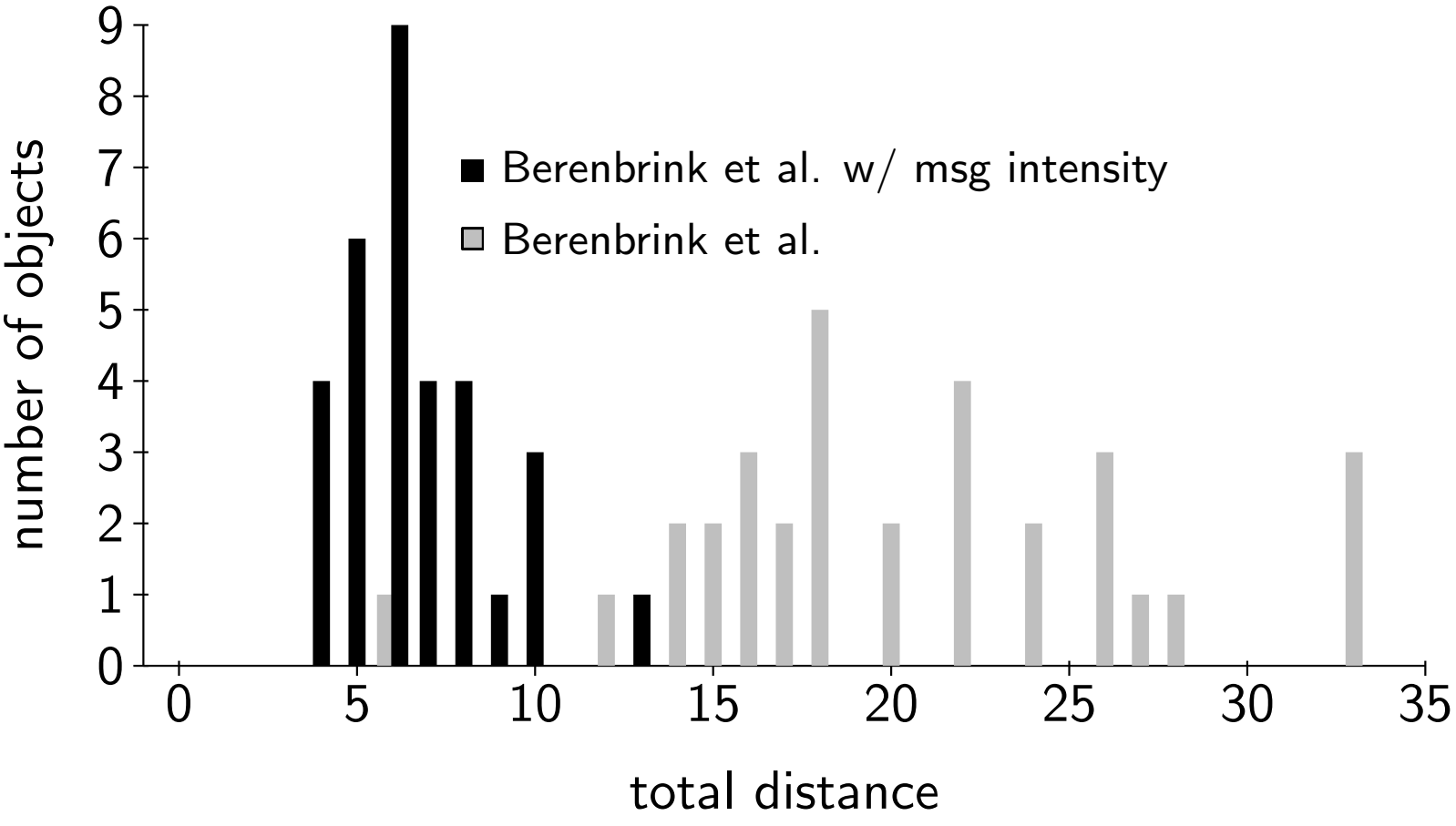
- Communication only between fringe and center objects
- Fringes and their centers need to be on nodes at close distance



`Star.abs` on Grid

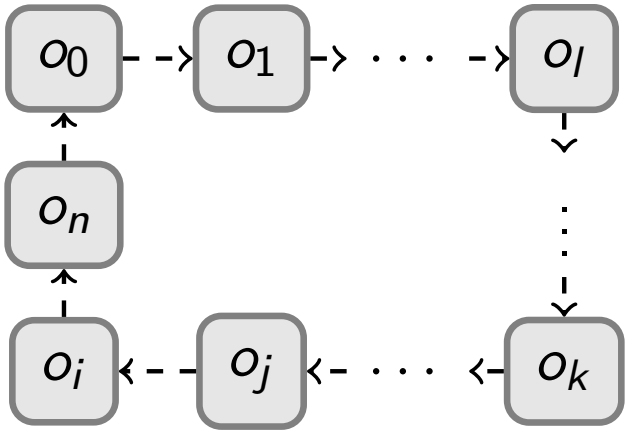


`Star.abs` Object Distance Distribution

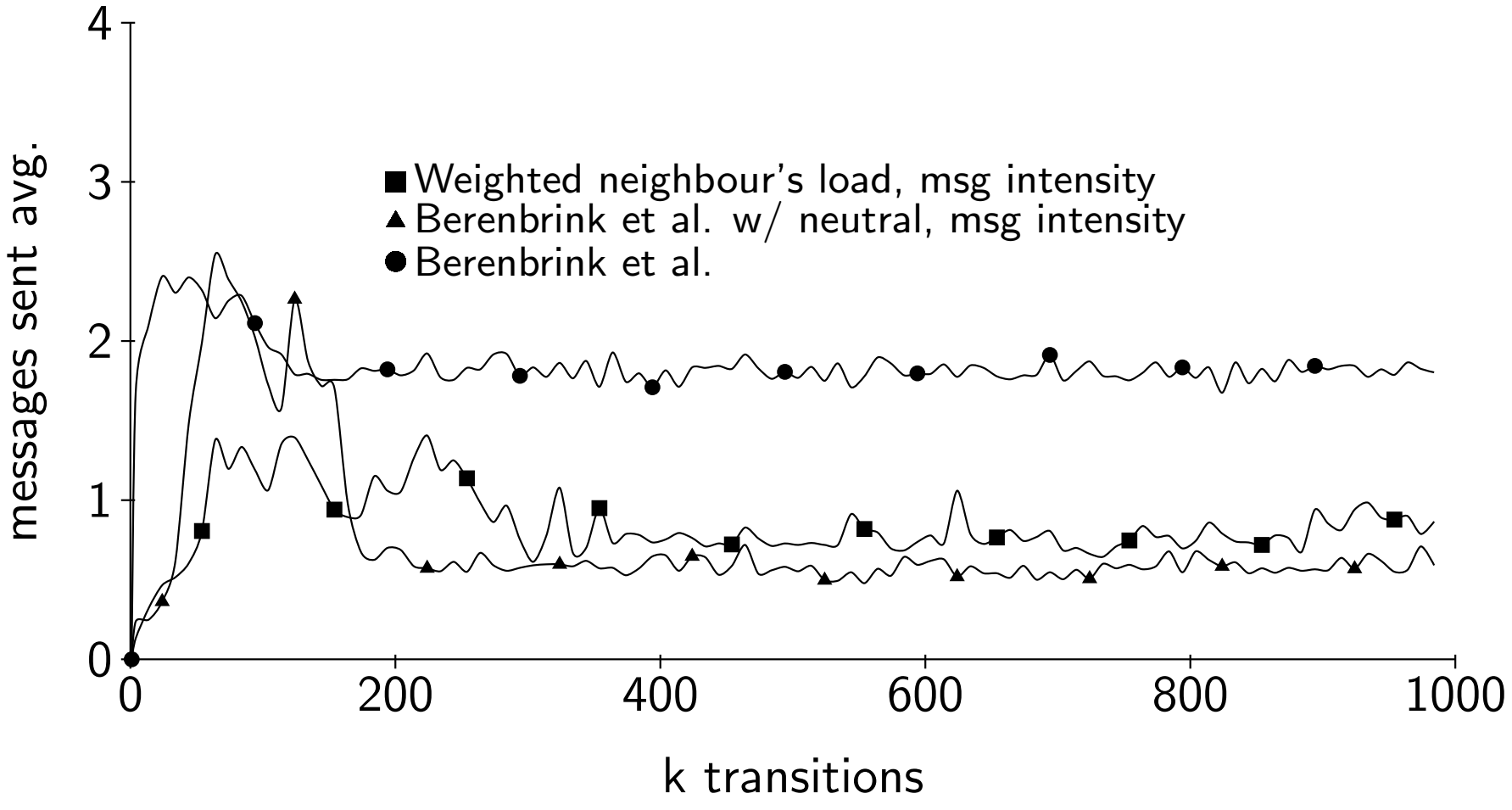


Ring.abs

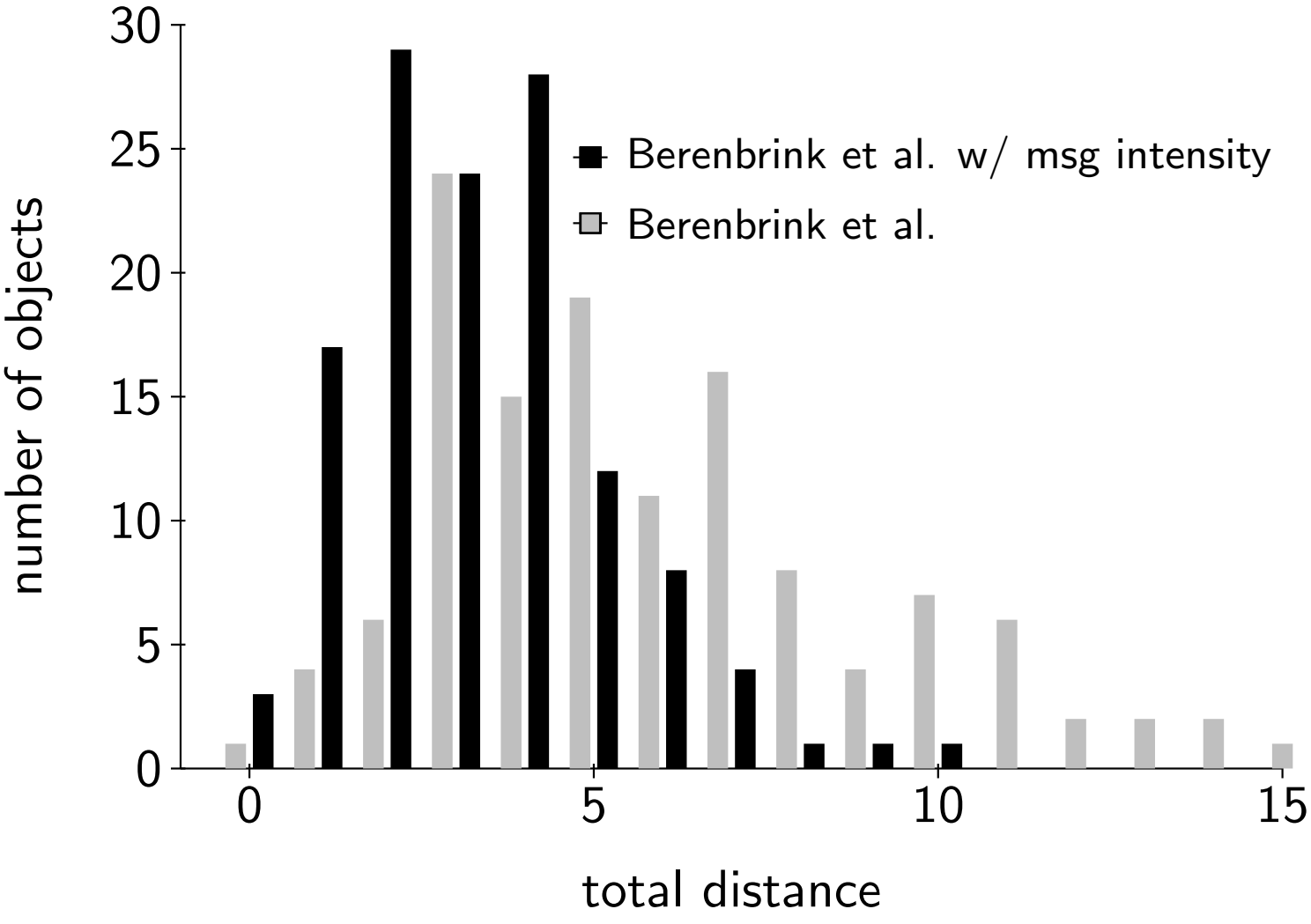
- Creates objects with references that form a ring
- All calls go through whole ring and back



`Ring.abs` on Grid

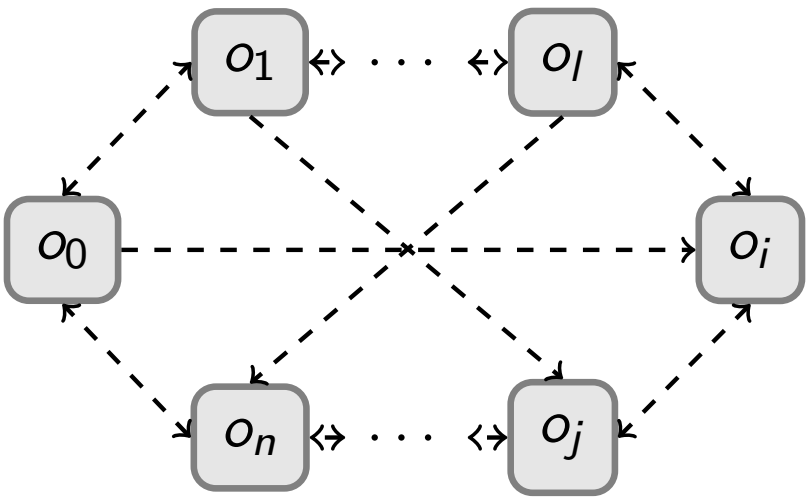


`Ring.abs` Object Distance Distribution

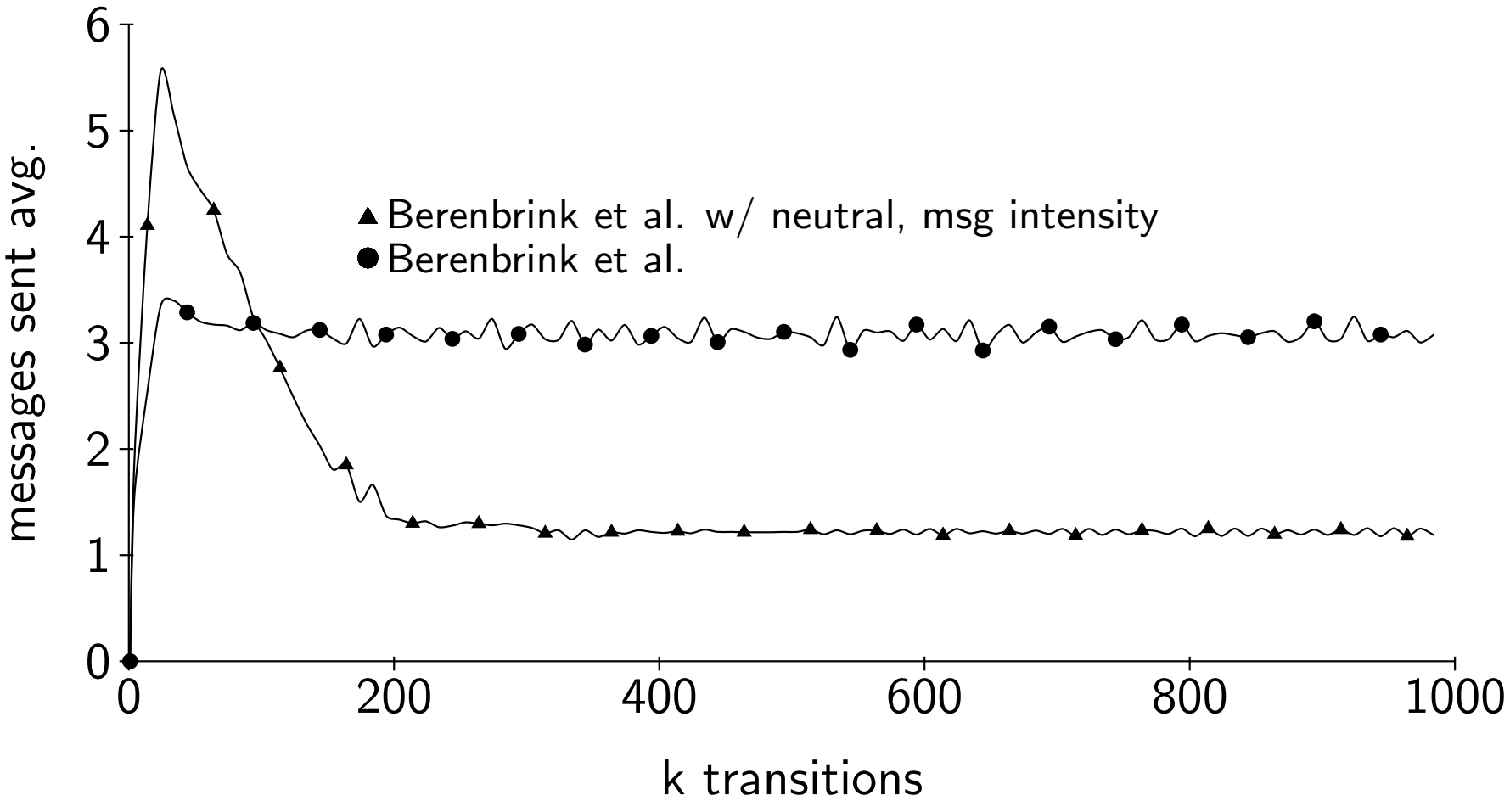


ChordDHT.abs

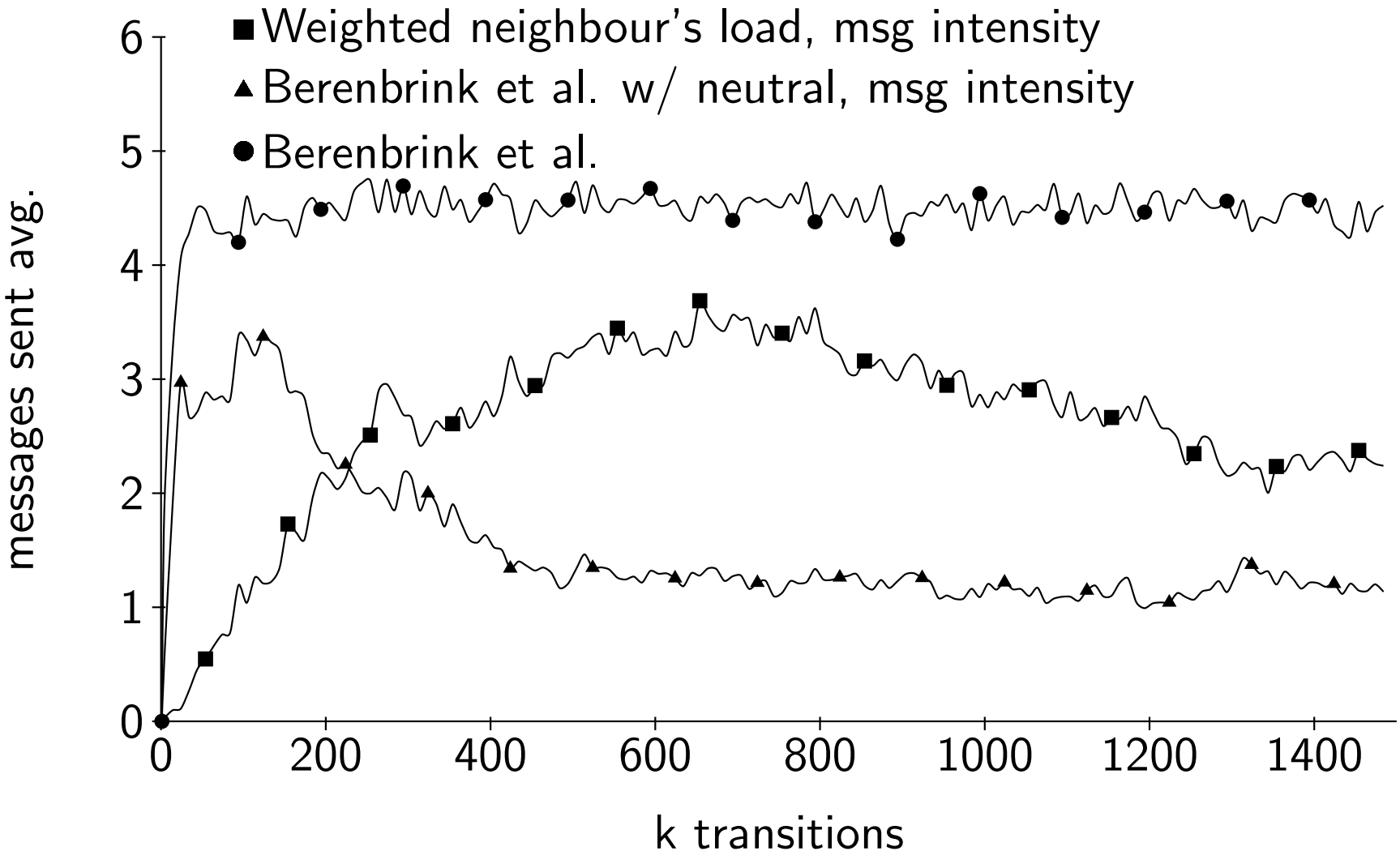
- Creates distributed hash table node objects
- Producer objects add key-value pairs
- Consumer objects request values for pseudorandom keys



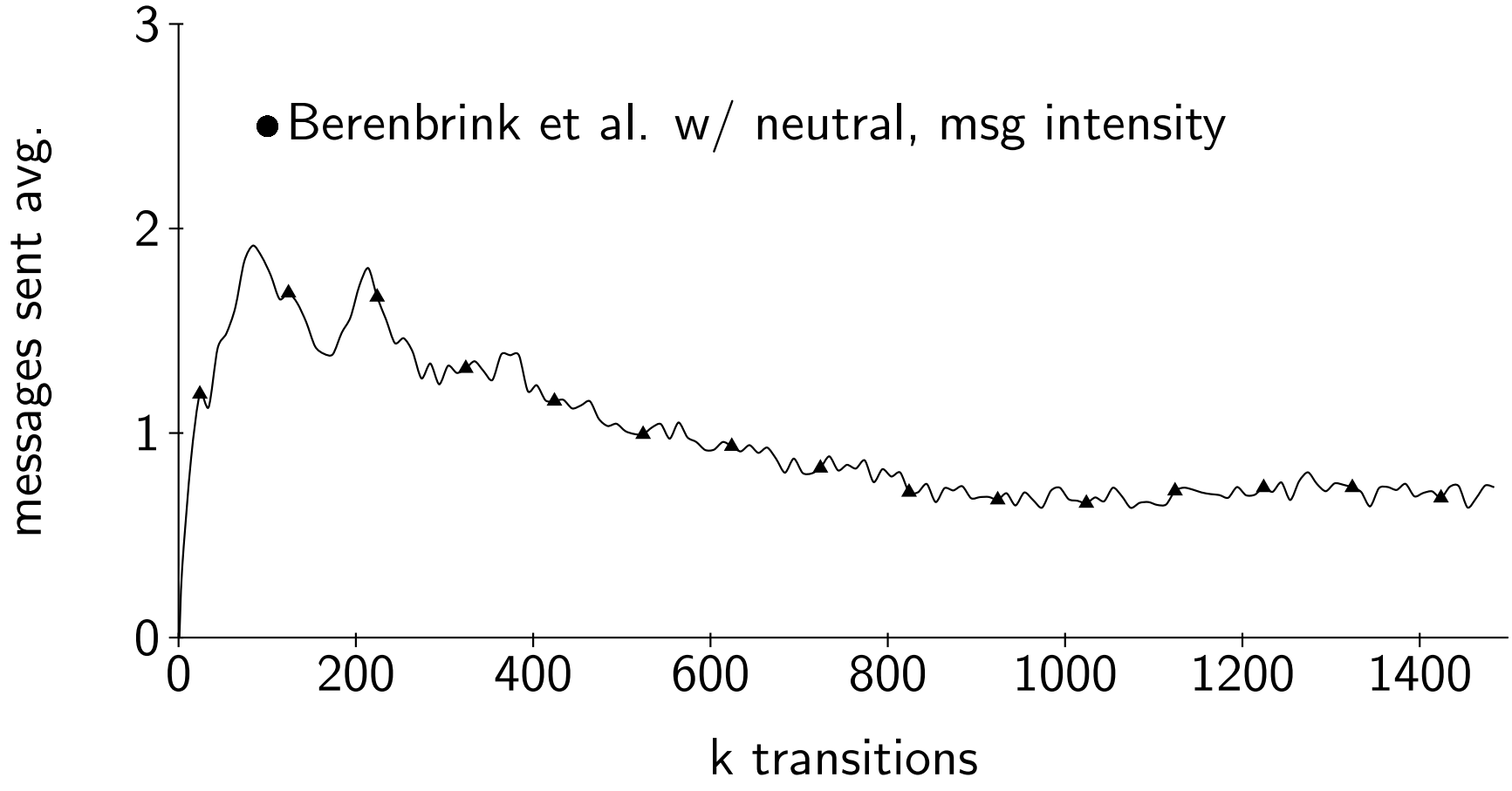
`ChordDHT.abs` on Grid



`Star64.abs` on Grid: 450 objects, 64 nodes



`Star128.abs` on Grid: 650 objects, 128 nodes



Conclusions

- Feasible to both:
  - do decentralized load balancing
  - consistently reduce communication overhead
- Validated applicability of ABS-NET model to decentralized runtime adaptation

Future Work

- Enhance simulator for large networks, rate limitation, latency
- Extend analysis to benignly dynamic networks
- Theoretical and experimental exploration of semantics/model