

# OverSwarm

**Toward a comprehensive evaluation of self-organized bio-inspired peer-to-peer solutions**

Dr. **Amos Brocco**

Telematics Institute, Karlsruhe Institute of Technology, Germany

amos.brocco@gmail.com



FONDS NATIONAL SUISSE  
SCHWEIZERISCHER NATIONALFONDS  
FONDO NAZIONALE SVIZZERO  
SWISS NATIONAL SCIENCE FOUNDATION

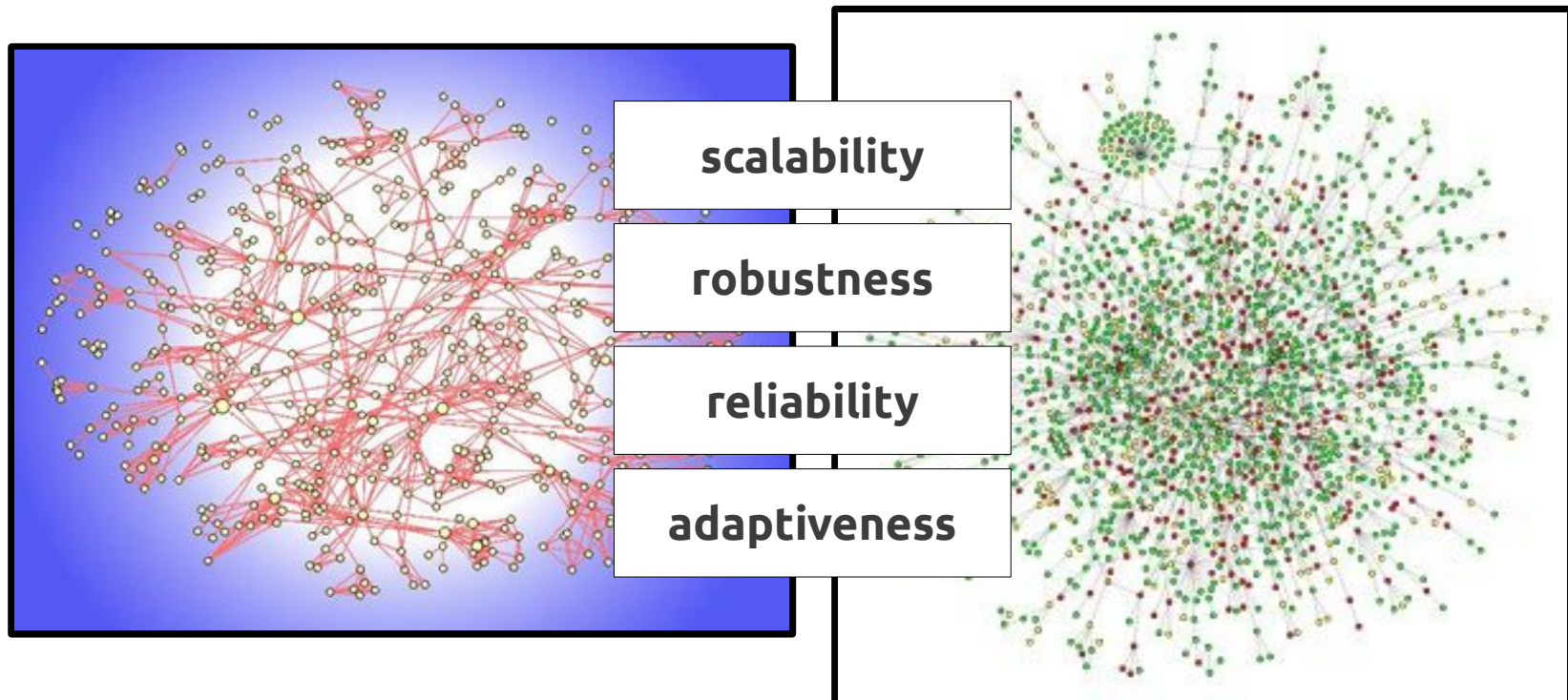
Fellowship  
n°134285

# Outline

Introduction  
Motivation  
OverSwarm  
Conclusion

# Introduction

Scenario: complex P2P distributed systems



# Introduction

## Bio-inspired solutions



**self-organization**

**emergence**

**fault-tolerance**



# Examples

**Messor (load balancing)**

**AntNet (routing)**

**AntHocNet (routing)**

**BlåtAnt (overlay mgmt)**

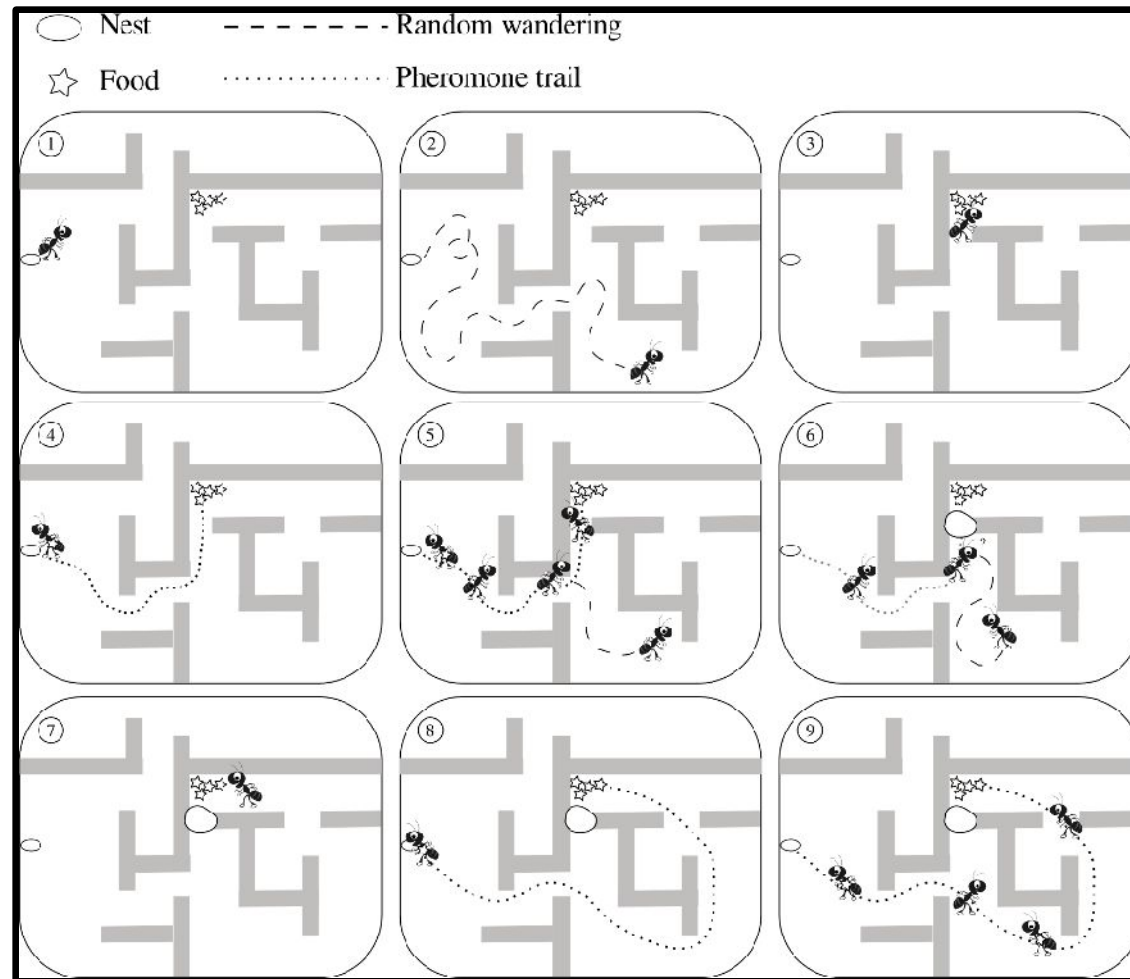
**Self-Chord (DHT)**

**Antares (clustering)**

**SemAnt (res. discovery)**

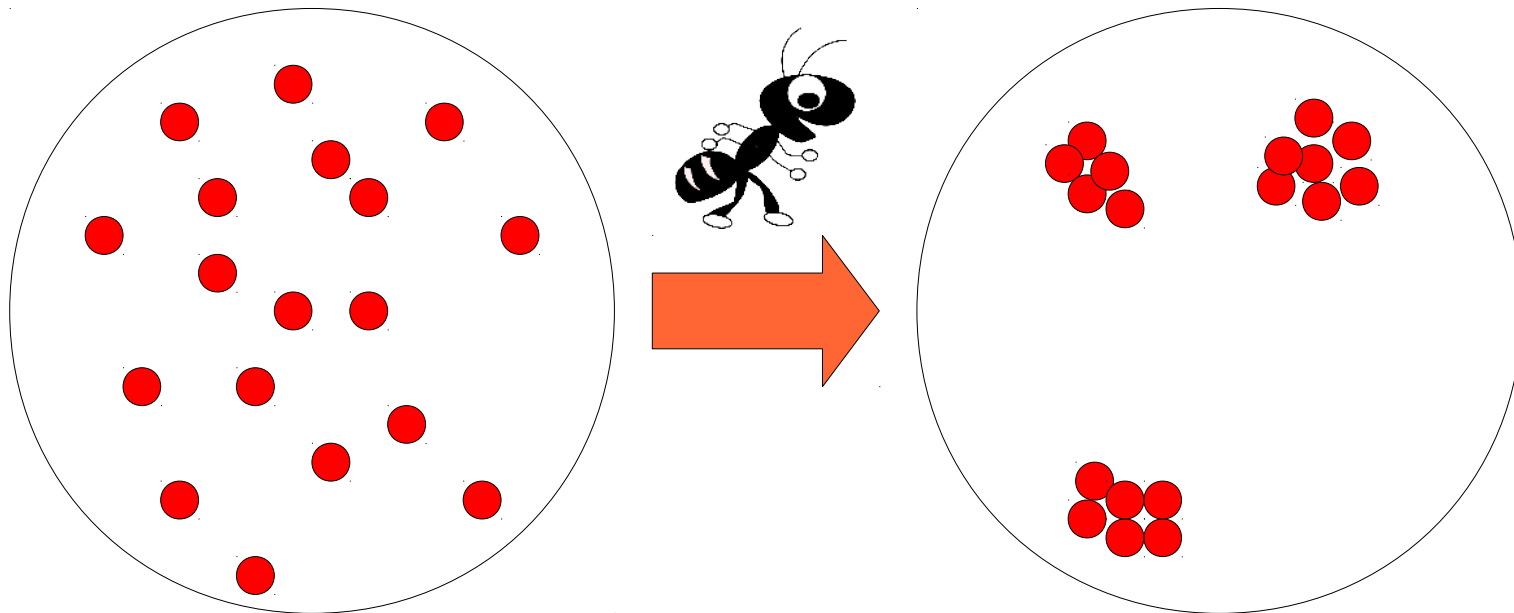
# Example: AntNet

## Ant foraging ~ Routing



# Example: Messor

Real ants (clustering behavior)



# Example: Messor

When an ant is not carrying any object, it wanders about randomly until it encounters an object and picks it up with a probability inversely proportional to the number of objects in the neighborhood.



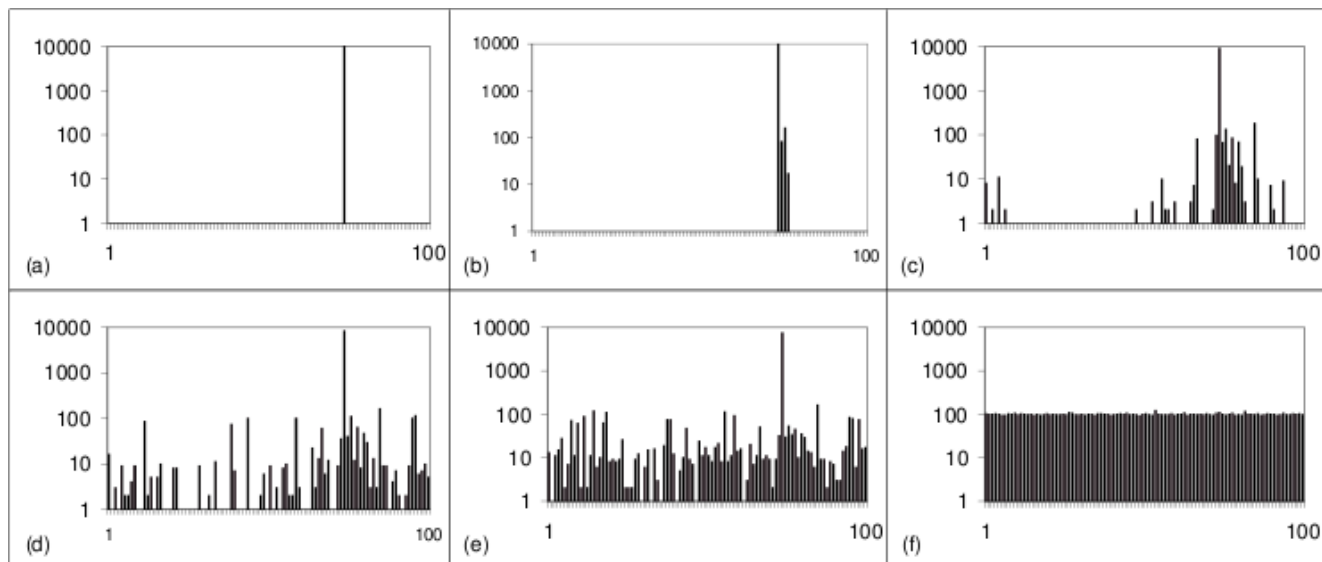
When an ant is carrying an object, the ant drops with a probability proportional to the number of objects in the neighborhood.





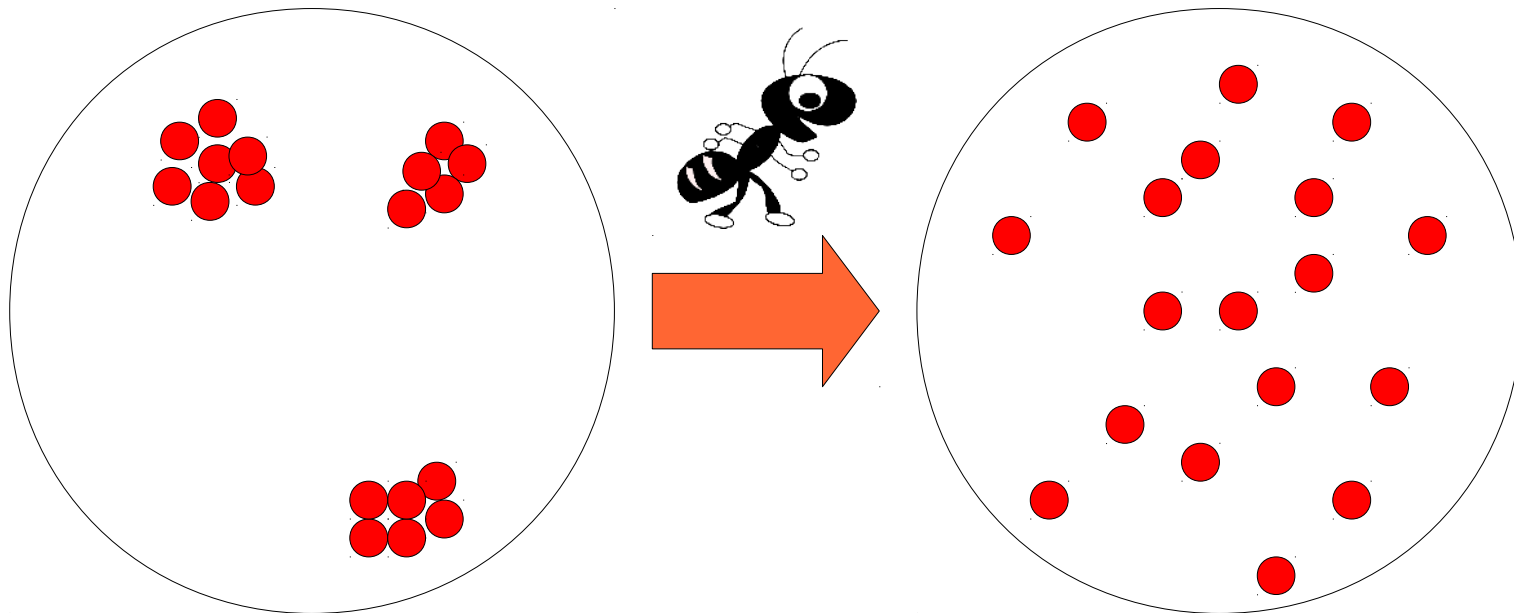
# Example: Messor

## Load balancing in distributed computing systems

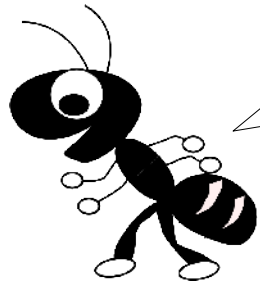


# Example: Messor

Artificial ants (spreading behavior)



# Example: Messor

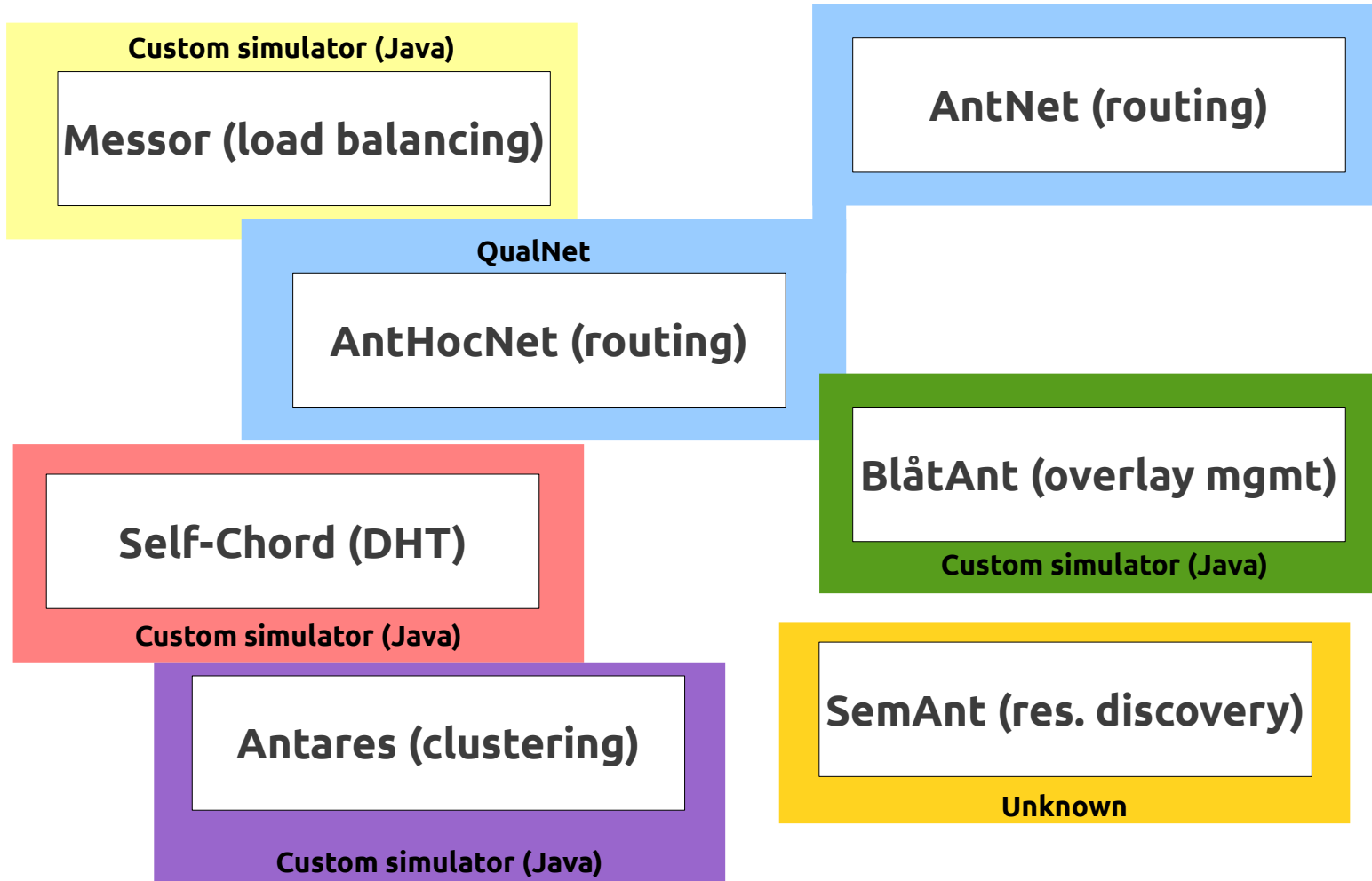


When an ant is not carrying any object, it wanders about randomly until it encounters an object and picks it up;



When an ant is carrying an object, the ant drops it only after having wandered about randomly “for a while” without encountering other objects.

# Examples



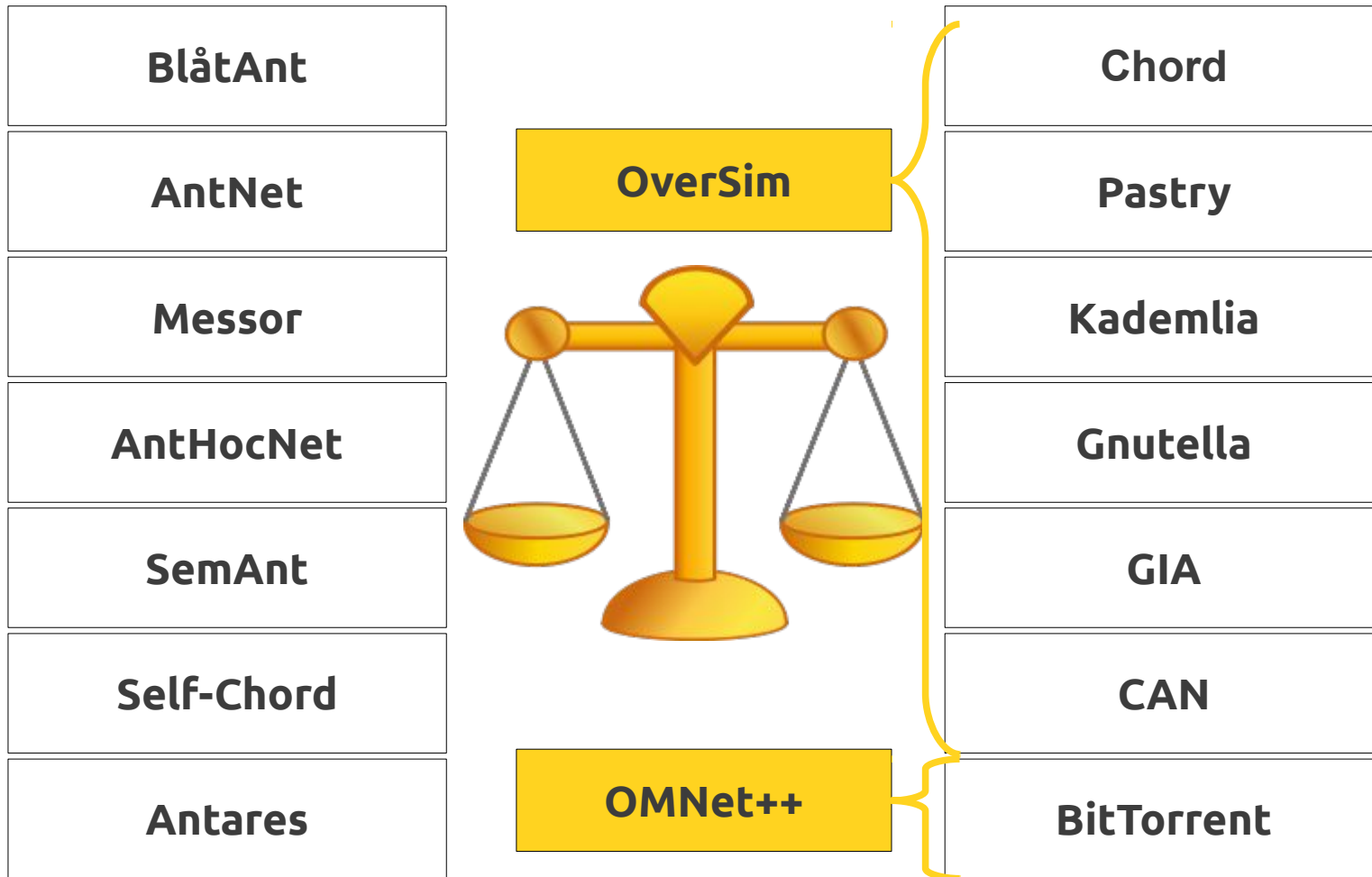
# Goal

<b>BlåtAnt</b>
<b>AntNet</b>
<b>Messor</b>
<b>AntHocNet</b>
<b>SemAnt</b>
<b>Self-Chord</b>
<b>Antares</b>

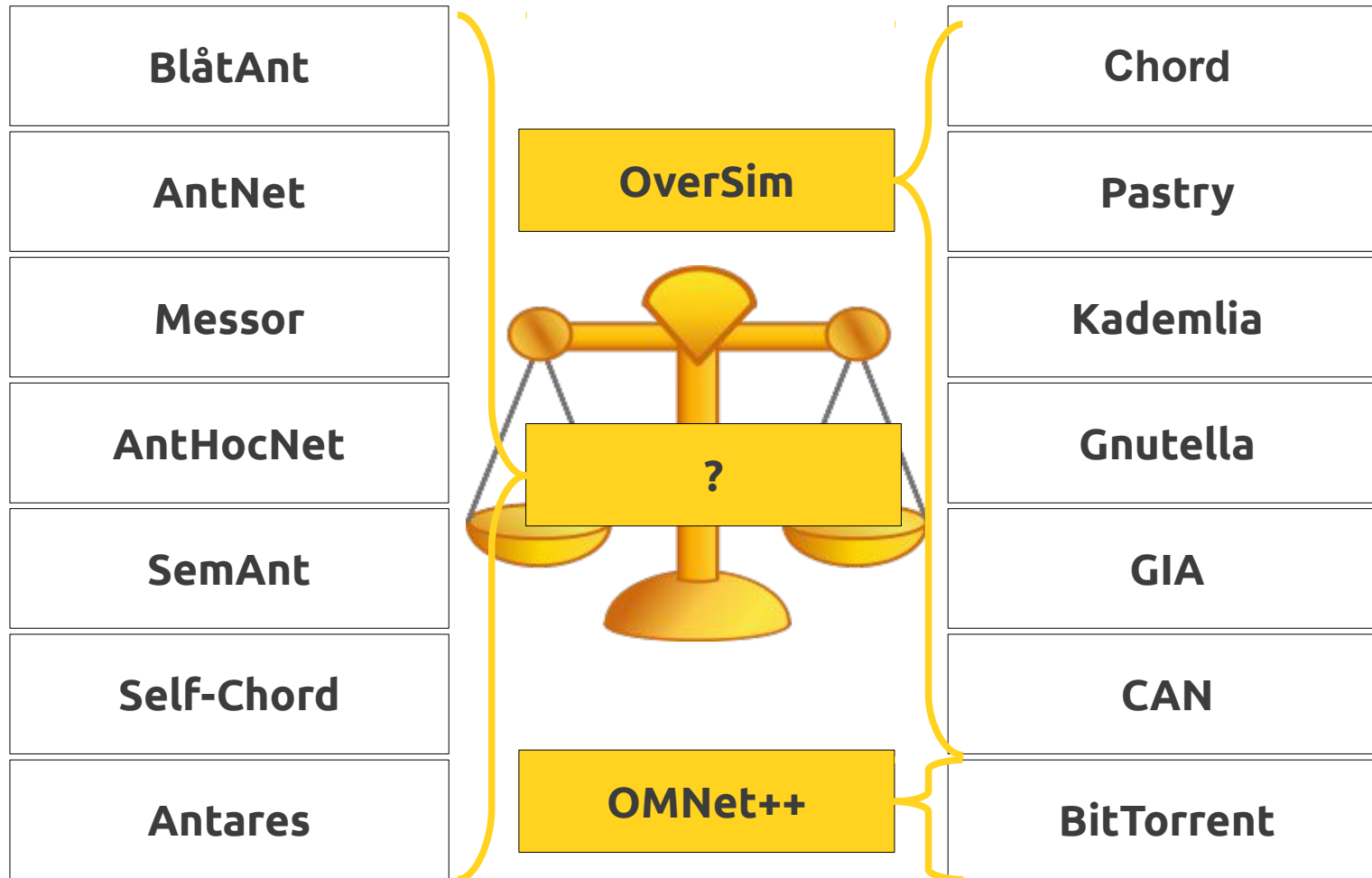


<b>Chord</b>
<b>Pastry</b>
<b>Kademlia</b>
<b>Gnutella</b>
<b>GIA</b>
<b>CAN</b>
<b>BitTorrent</b>

# Goal



# Goal



# OverSwarm

Bio-inspired / swarm framework  
Mobile-agent based protocols

Goals:

- Comprehensive evaluation

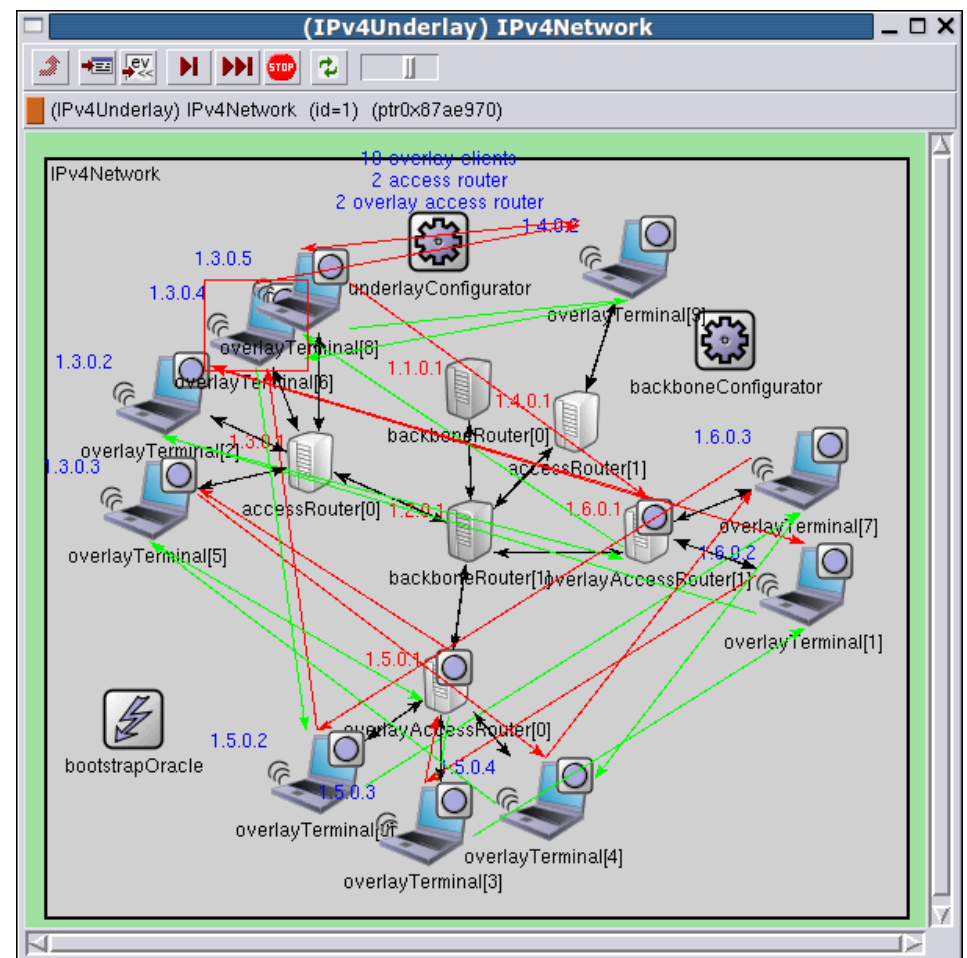
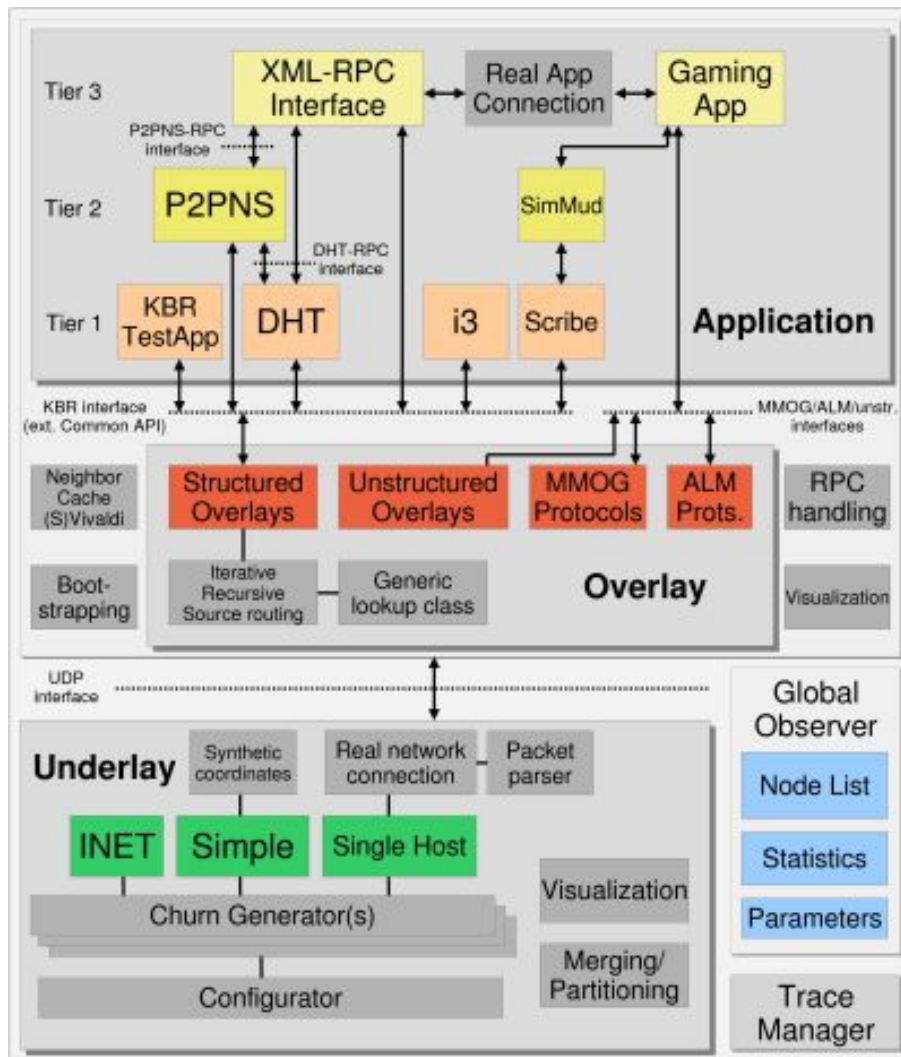
- Comparison

- Experimenting, prototyping

Based on OverSim / OMNET++

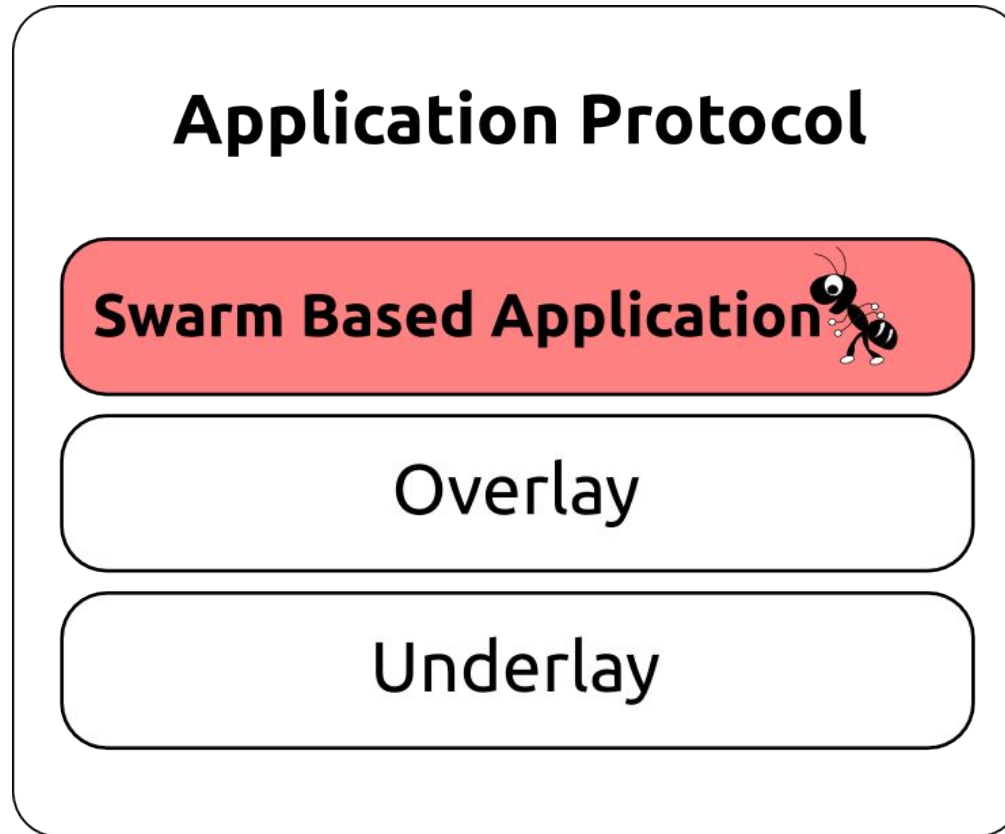


# OverSim

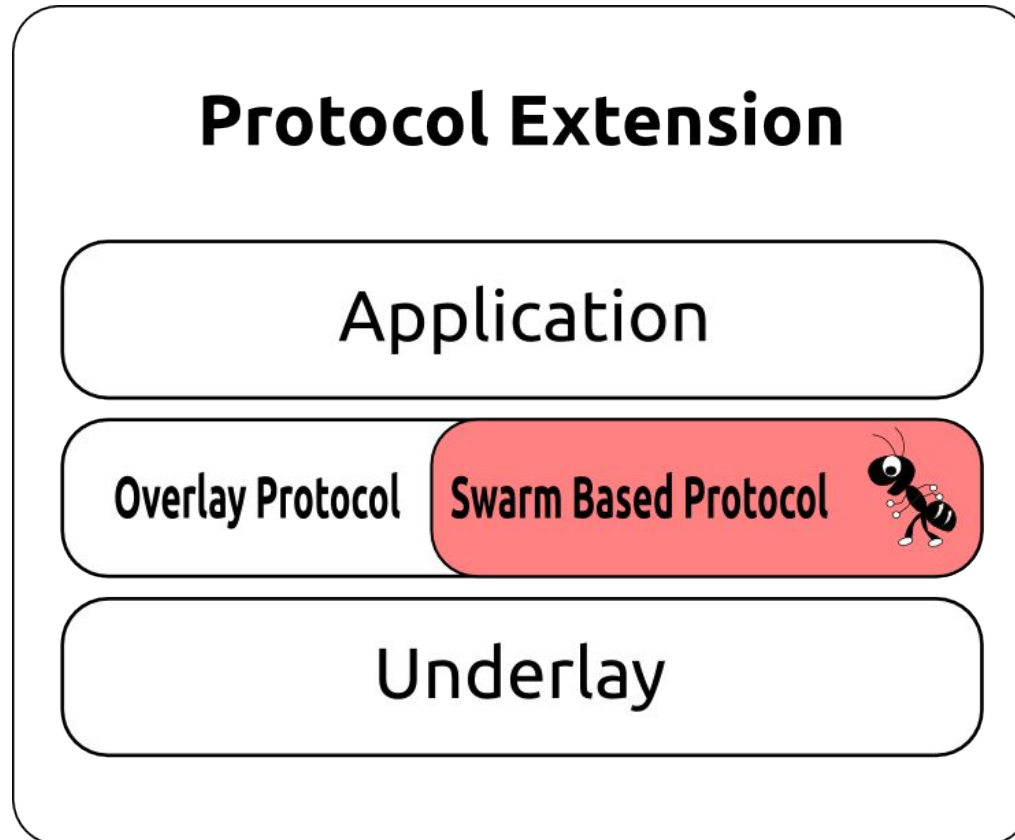


[www.oversim.org](http://www.oversim.org)

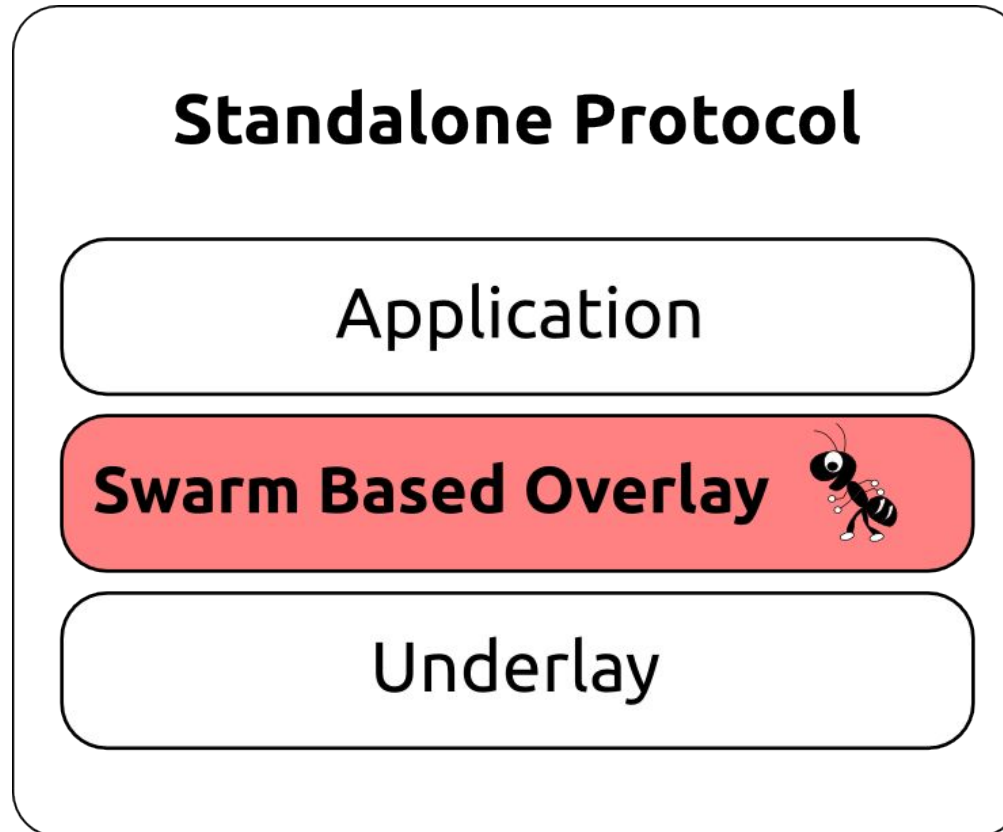
# Application layer



# Overlay protocol extension



# Standalone overlay protocol



# Toolchain

## Programming language (Lisp-like)

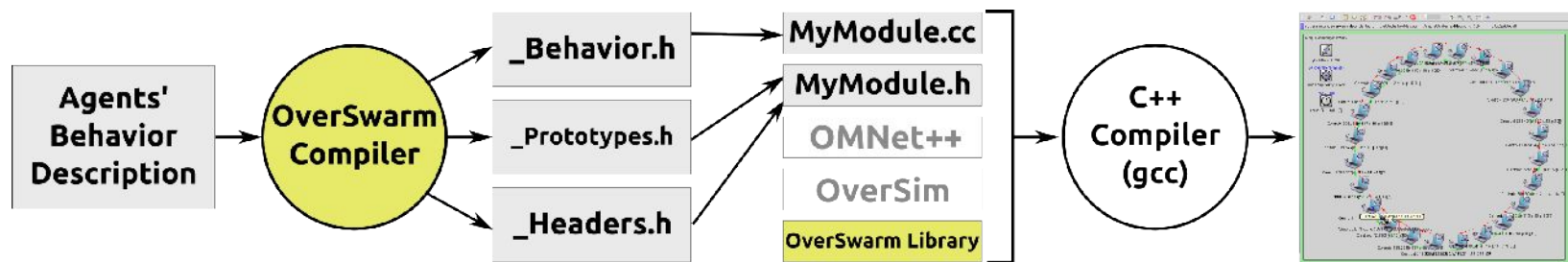
dynamic typing, automatic memory mgmt

strong, transparent migration

pheromone API

topology measurements API

## Compiler (agent language -> C++)



# Programming

## Ant Agent's Behavior

With probability 50% either migrate to the successor, or: migrate to predecessor, doSomething, then migrate back to the previous node and if the result of doSomething was greater than 0 doThis, otherwise doThat.

# Programming

## Ant Agent's Behavior

With probability 50%  
migrate to the successor  
migrate to predecessor  
doSomething, then  
back to the predecessor  
if the result of doSomething  
was greater than  
otherwise doThat.

```
switch(packet->getType()) {  
case 0:  
    if (rand() < 0.5) {  
        packet->setPrevious(this->getAddress());  
        packet->setType(1);  
        sendMessageToUDP(this->predecessor, packet);  
    } else {  
        sendMessageToUDP(this->successor, packet);  
    }  
    break;  
case 1:  
    int result = doSomething();  
    packet->setResult(result);  
    packet->setType(2);  
    sendMessageToUDP(this->getPrevious(), packet);  
    break;  
case 2:  
    if (packet->getResult() > 0) {  
        doThis();  
    } else {  
        doThat();  
    }  
    break;  
default:  
    // Handle unknown message  
}
```

OMNet++ / OverSim

# Programming

## Ant Agent's Behavior

With probability 50%  
migrate to the successor  
migrate to predecessor  
doSomething, then  
back to the previous  
if the result of doSomething  
was greater than  
otherwise doThat.

```
switch(packet->getType()) {  
  case 0:  
    if (rand() < 0.5) {  
      packet->setPrevious(this->getAddress());  
      packet->setType(1);  
      sendMessageToUDP(this->predecessor);  
    } else {  
      sendMessageToUDP(this->successor);  
    }  
    break;  
  case 1:  
    int result = doSomething();  
    packet->setResult(result);  
    packet->setType(2);  
    sendMessageToUDP(this->getPredecessor());  
    break;  
  case 2:  
    if (packet->getResult() > 0) {  
      doThis();  
    } else {  
      doThat();  
    }  
    break;  
  default:  
    // Handle unknown message  
}
```

OMNet++ / OverSwarm

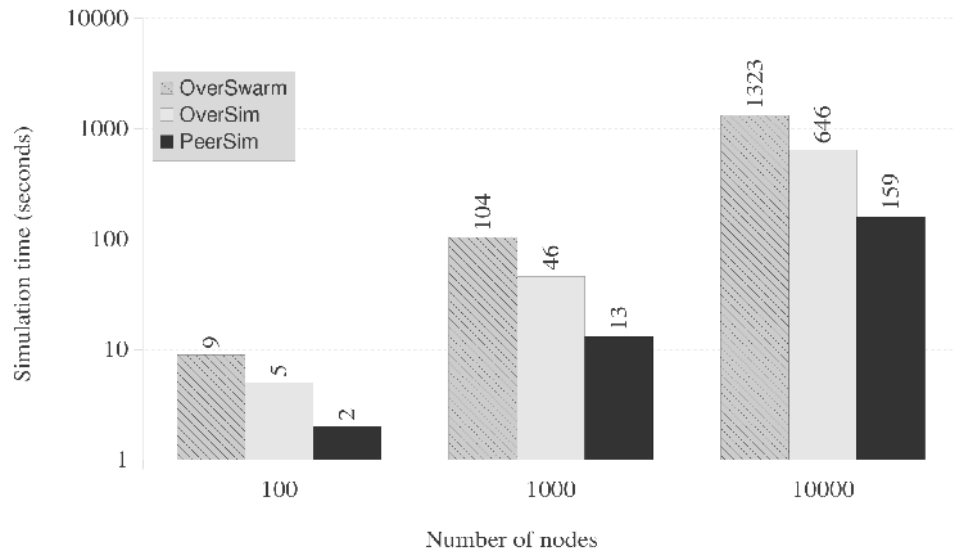
```
(var previous nil)  
(if (< (rand) 0.5) (begin  
  (set! previous (getThisNode))  
  (migrate (getPredecessor))  
  (var result (doSomething))  
  (migrate previous)  
  (if (> result 0)  
    (doThis)  
  else  
    (doThat)))  
else  
  (migrate (getSuccessor)))
```

OverSwarm

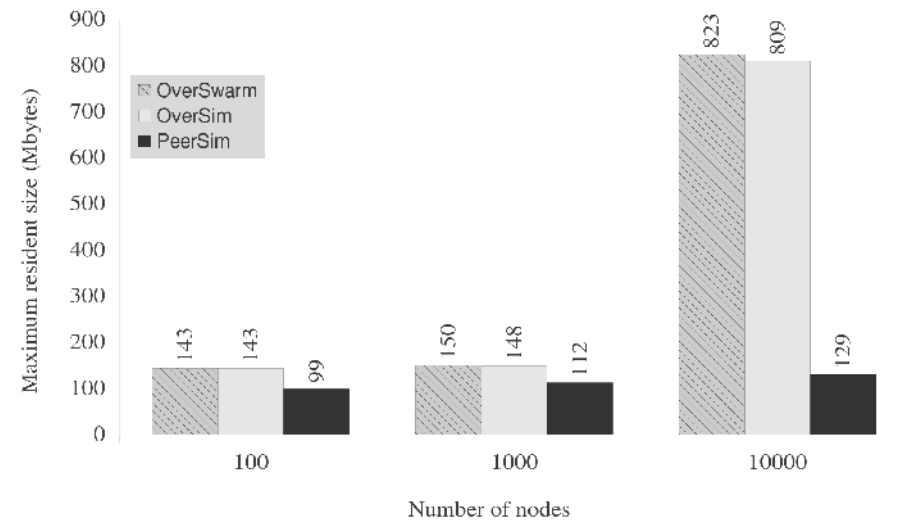


# Benchmark

## Simulation Time



## Memory usage



# Example: BlåtAnt

Overlay management protocol

Self-structured overlay

- bounds distances between each pair of peers
- limits small cycles
- pure peer-to-peer (no superpeers, hubs)

Ant-like mobile agents

# Example: BlåtAnt

For a user defined parameter  $D$

Create and remove logical links:

## **Connection Rule**

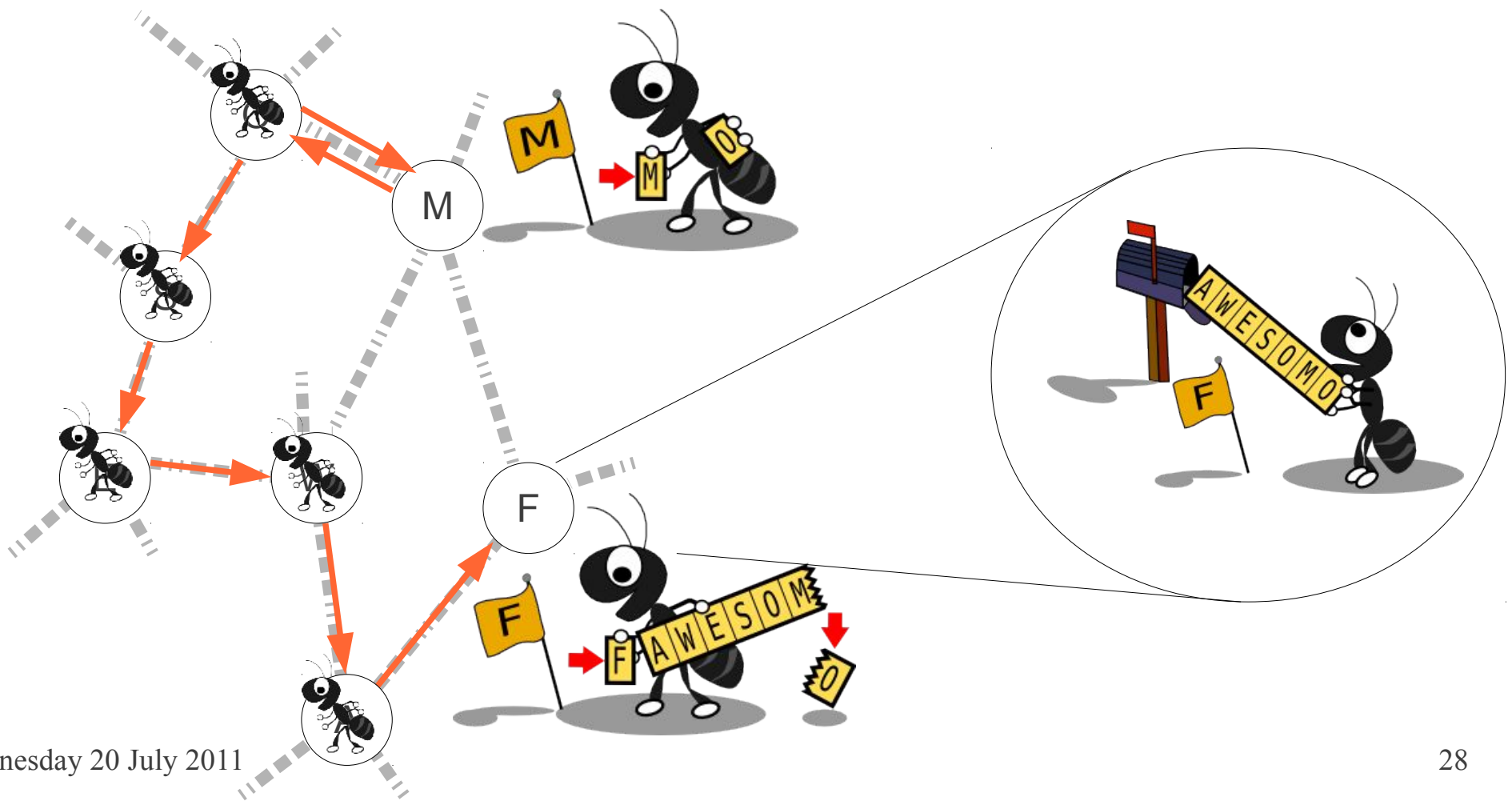
connect two nodes if distance (hops)  $\geq 2D - 1$

## **Disconnection Rule**

break cycle if length  $< 2D$

# Example: BlåtAnt (discovery)

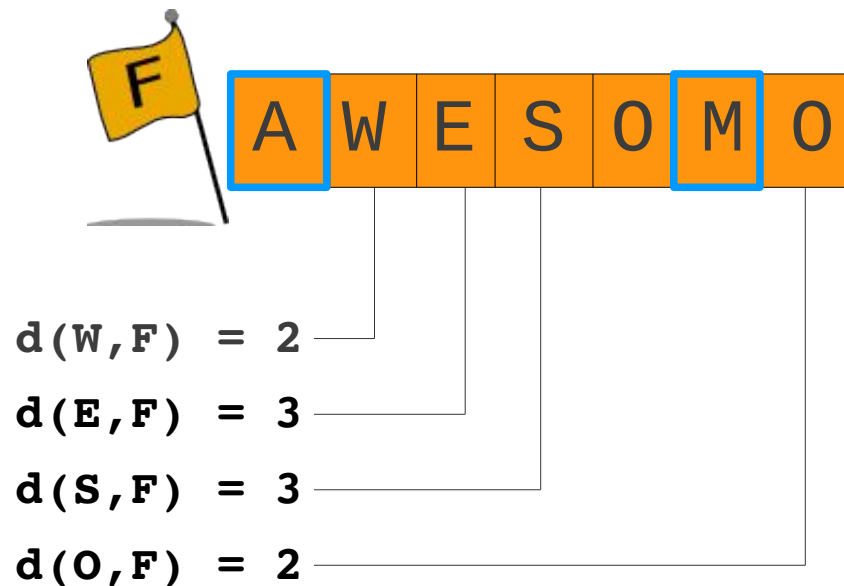
Discovery ants collect and disseminate information across nodes...



# Example: BlåtAnt (inform)

Infer minimal distances to (non-neighbor) nodes

- **Connection Rule:** if distance  $\geq 2D - 1$ , create a new connection

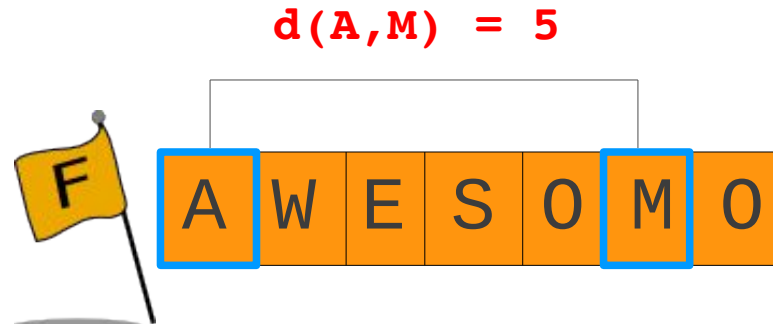


 = neighbor of F

# Example: BlåtAnt (inform)

Infer minimal distances between (neighbor) nodes in the overlay (not crossing the node itself)

- **Disconnection Rule:** if distance  $< 2D-2$ , disconnect one of the neighbors



 = neighbor of F

# Example: BlåtAnt (discovery,inform)

```
(while 1 (begin
  (if (<= steps 0) (break))

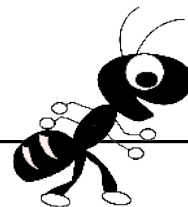
  ;; Pass the information to the node
  (inform vector)

  ;; Address of the current node
  (var currentNode (getThisNode))

  (push vector currentNode)

  ;; Trim vector (if necessary)
  (if (> (len vector) vectorlength) (erase vector 0))

  ;; Now, choose next step
  (var nextStep nil)
  (var candidates (getNeighbors))
  ;; Remove previously visited nodes
  (foreach v in vector (begin
    (remove candidates v)))
  ;; No good destination? Forget all information, and start over
  (if (= (len candidates) 0) (begin
    (set! candidates (getNeighbors))
    (set! vector [])))
  ;; Count this step
  (set! steps (- steps 1))
  ;; Exploration or exploitation?
  (if (< (random) kappa)
    (migrate (getLowestGammaTrail candidates))
  else
    (migrate (choose candidates))))))
```



"alpha->inform" as "inform",

```
osvPtr antFunction(Alpha::inform, osvPtr vector)
{
  OvsWList v = unwrapOvsWList(vector);
  return wrap(inform(v));
} C++
```

```
bool Alpha::inform(OvsWList& info)
{
  int distanceFromNeighbor = 2;

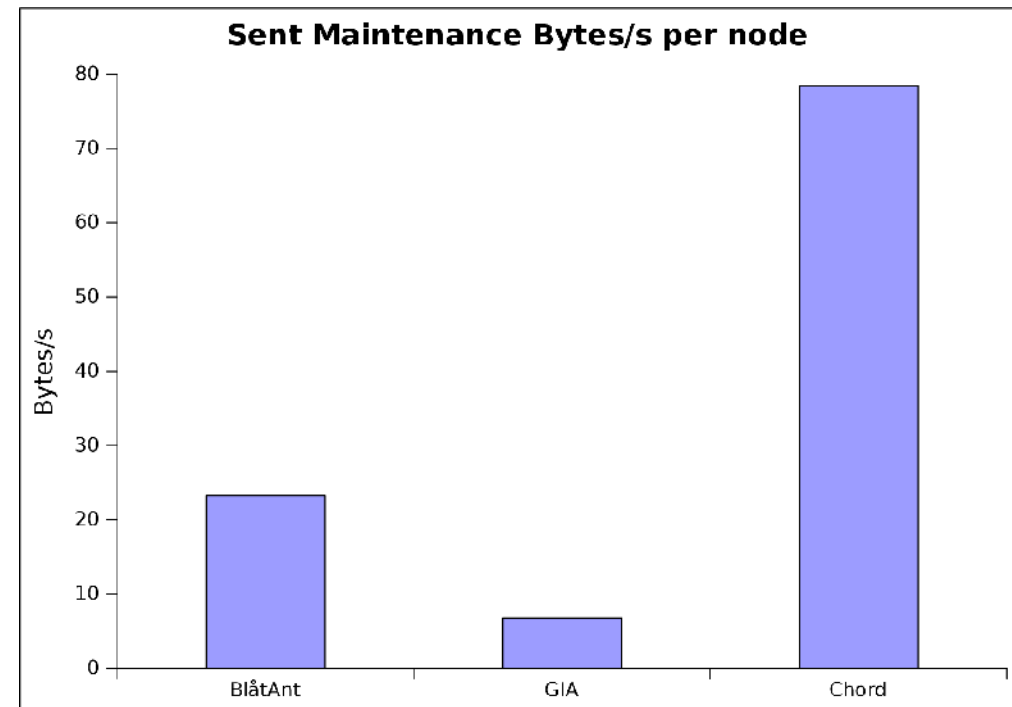
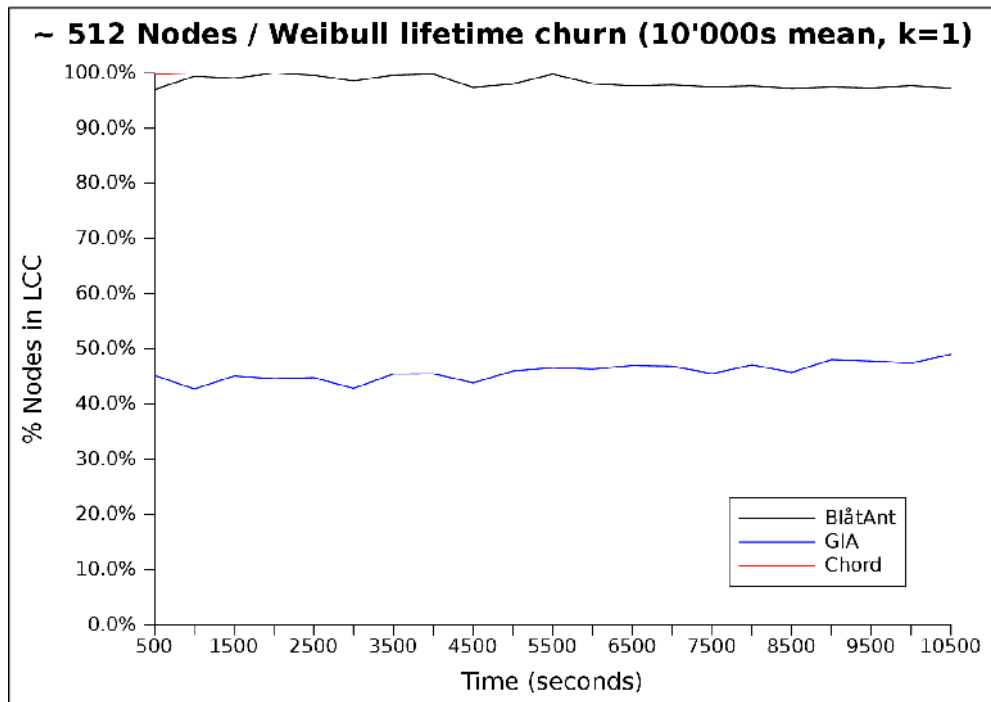
  if (info.empty()) return false;

  int count = (info.size() - 2);

  for (int i = count; i >= 0; i--) {
    TransportAddress el = unblobTransportAddress(info.at(i));
    if (el.isUnspecified()) {
      continue;
    }
    if (isNeighbor(el)) {
      distanceFromNeighbor = 2;
    } else {
      Entry& e = table->getOrNew(el);
      e.distance = distanceFromNeighbor;
      e.entryAge = getTimestamp();
      distanceFromNeighbor++;
    }
  }

  distanceFromNeighbor = info.size();
} C++
```

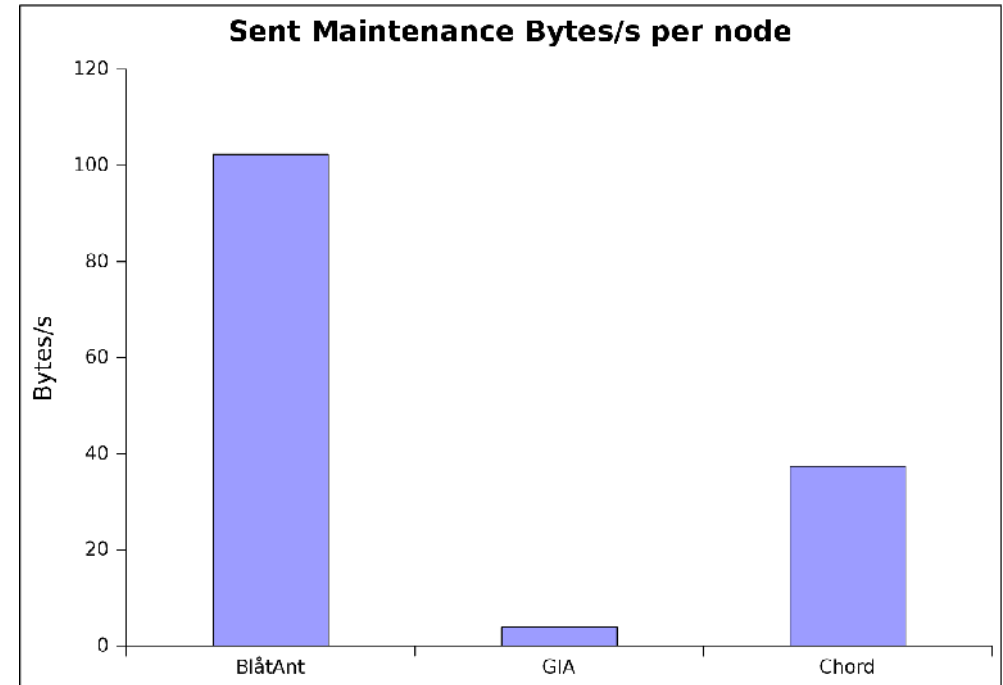
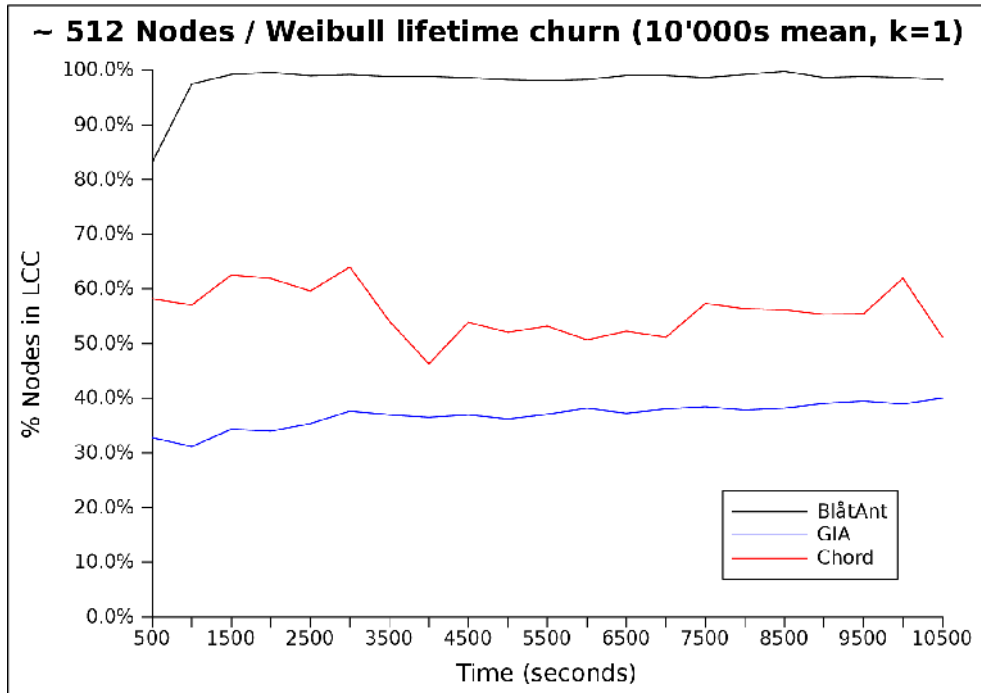
# Example: Evaluation



+ / - ~20 peers every 500 seconds  
BlåtAnt: max 8 neighbors  
GIA: min 3, max 20 neighbors

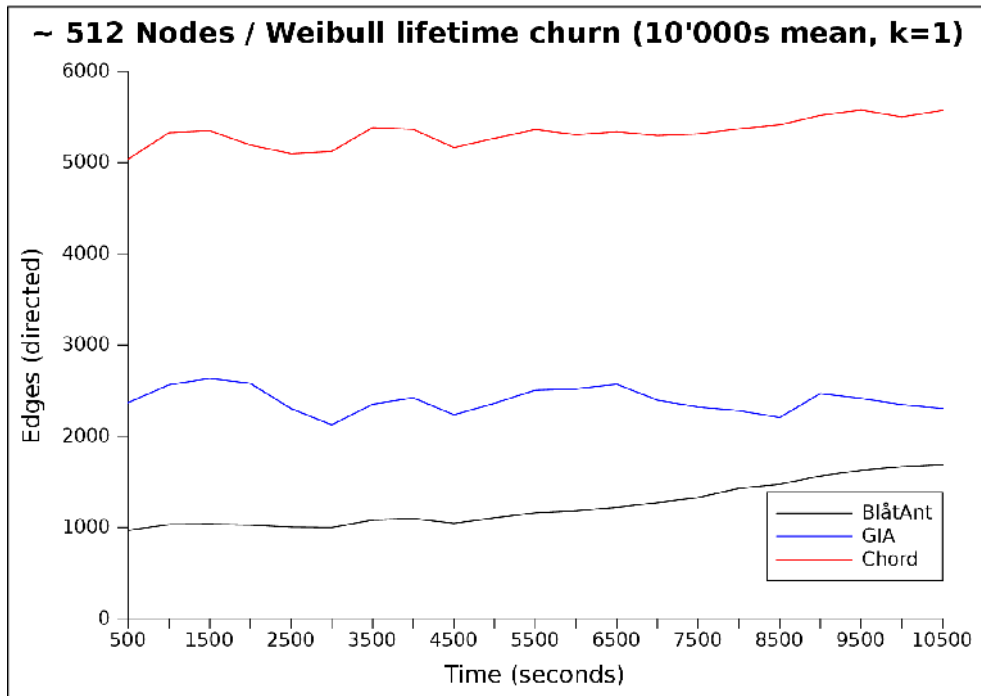


# Example: Evaluation

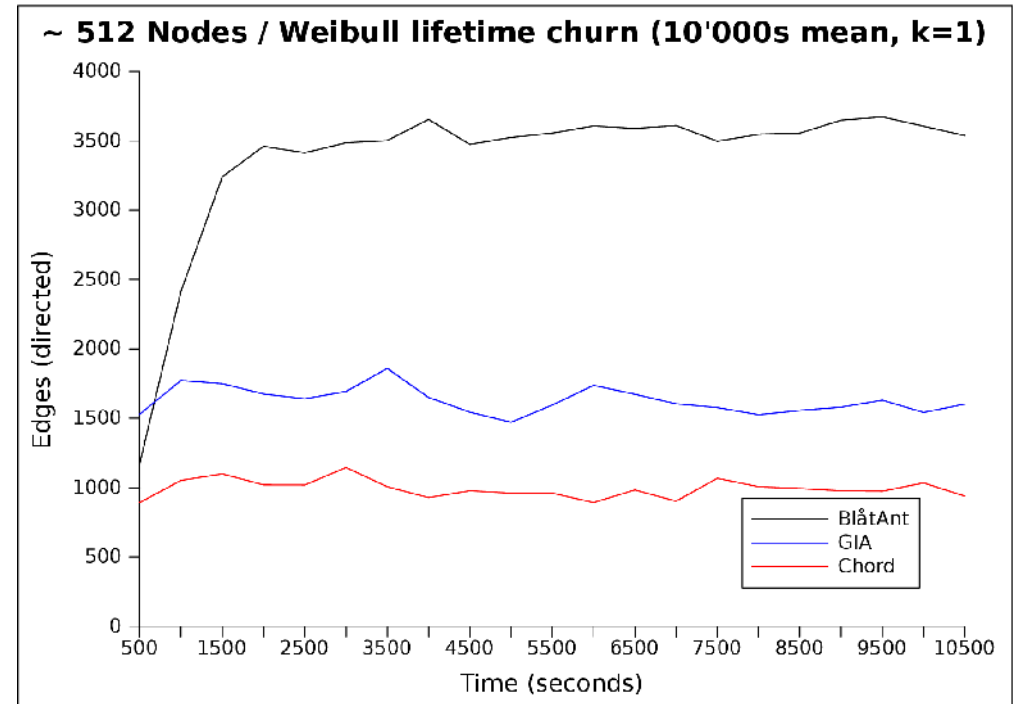


+ / - ~20 peers every 500 seconds  
10% packets lost  
BlåtAnt: max 8 neighbors  
GIA: min 3, max 20 neighbors

# Example: Evaluation



No packet loss



10% packets lost

# Conclusions

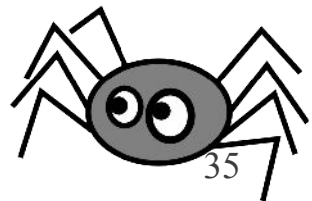
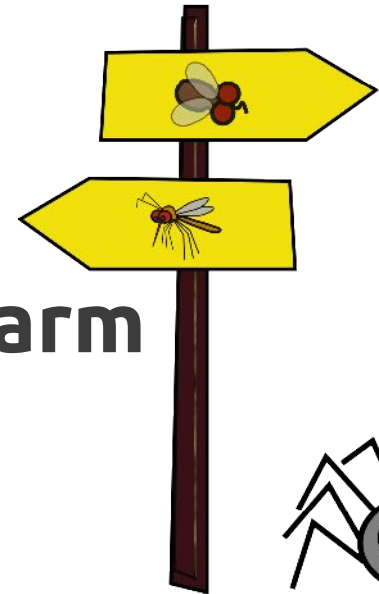
## OverSwarm' goals

Comprehensive evaluation of bio-inspired P2P systems

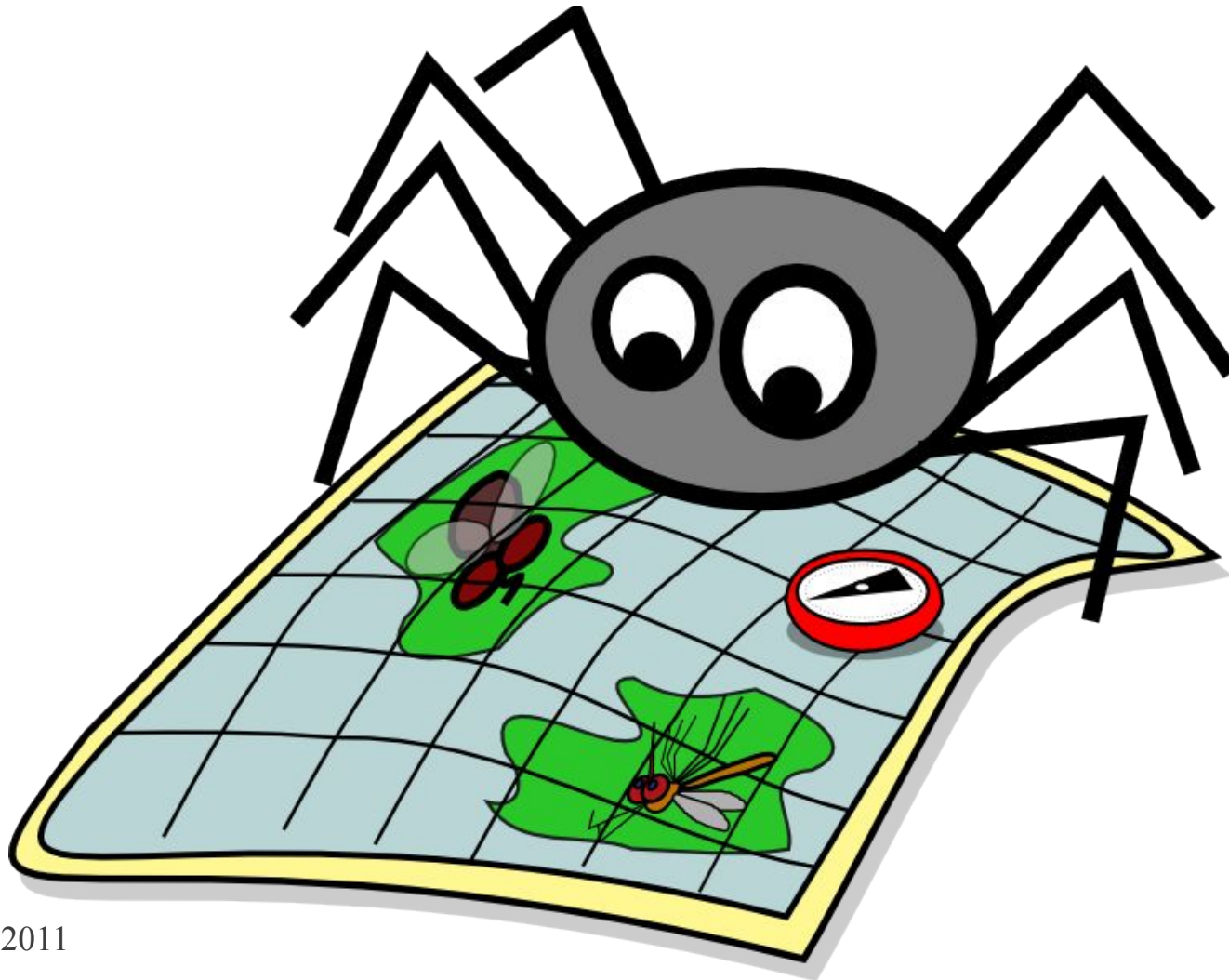
Comparison between bio-inspired and “traditional”

Easy to use development framework

<http://syscall.org/doku.php/overswarm>



# Thank you for your attention... questions?



- Messor: Load-Balancing through a Swarm of Autonomous Agents by: Alberto Montresor, Hein Meling, Özalp Babaoğlu. In Proceedings of 1st Workshop on Agent and Peer-to-Peer Systems (2002), pp. 125-137. Key: citeulike:4971427
- A. Forestiero, E. Leonardi, C. Mastroianni, M. Meo, Self-Chord: a Bio-Inspired P2P Framework for Self-Organizing Distributed Systems. IEEE/ACM Transactions on Networking, vol.18, n.5, October 2010, pp. 1651-1664. PDF.
- Amos Brocco, Apostolos Malatras, and Béat Hirsbrunner, "Proactive Information Caching for Efficient Resource Discovery in a Self-Structured Grid", in: Workshop on Bio-Inspired Algorithms for Distributed Systems, ACM, ICAC 2009, Barcelona, Spain, June, 2009.
- Elke Michlmayr. 2006. Self-organization for search in peer-to-peer networks: the exploitation-exploration dilemma. In Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems (BIONETICS '06). ACM, New York, NY, USA
- Di Caro G., Dorigo M., "AntNet: Distributed Stigmergetic Control for Communications Networks", Journal of Artificial Intelligence Research (JAIR), Vol. 9, Pag. 317-365, 1998.
- Ducatelle, F., Adaptive Routing in Ad Hoc Wireless Multi-hop Networks, PhD thesis, Università della Svizzera Italiana, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, 2007.
- Forestiero, A.; Mastroianni, C.; Spezzano, G.; , "Antares: an ant-inspired P2P information system for a self-structured grid," Bio-Inspired Models of Network, Information and Computing Systems, 2007. Bionetics 2007. 2nd , vol., no., pp.151-158, 10-12 Dec. 2007
- Márk Jelasity, Wojtek Kowalczyk, and Maarten van Steen. Newscast computing. Technical Report IR-CS-006, Vrije Universiteit Amsterdam, Department of Computer Science, Amsterdam, The Netherlands, November 2003.
- Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. 2003. Making gnutella-like P2P systems scalable. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03). ACM, New York, NY, USA, 407-418.
- Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. SIGCOMM Comput. Commun. Rev. 31, 4 (August 2001), 149-160.