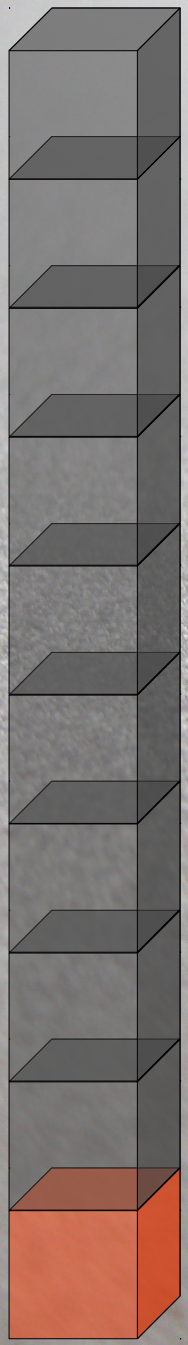


OverSwarm

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

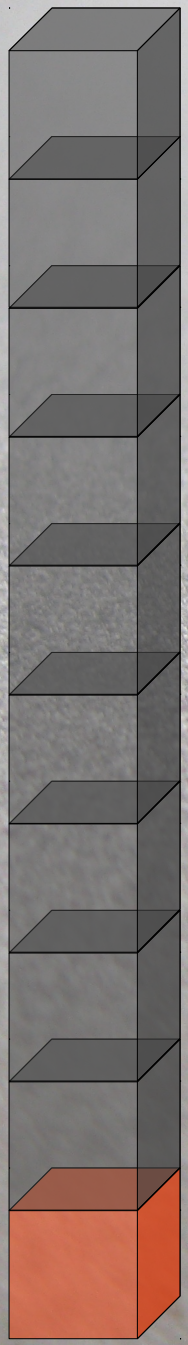
Amos Brocco

17.03.2011

Karlsruhe Institute of Technology

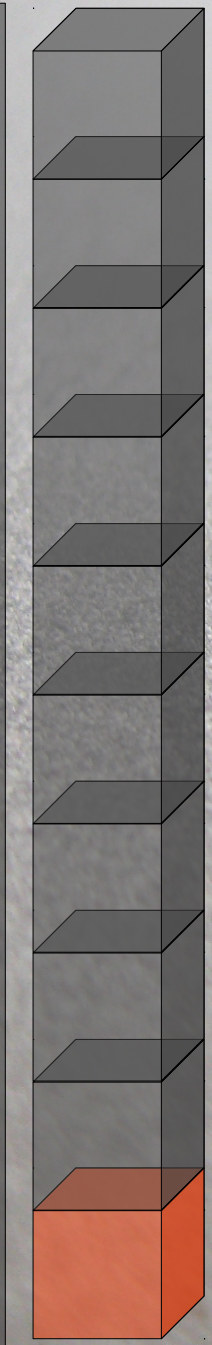
FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION

Fellowship
n°134285



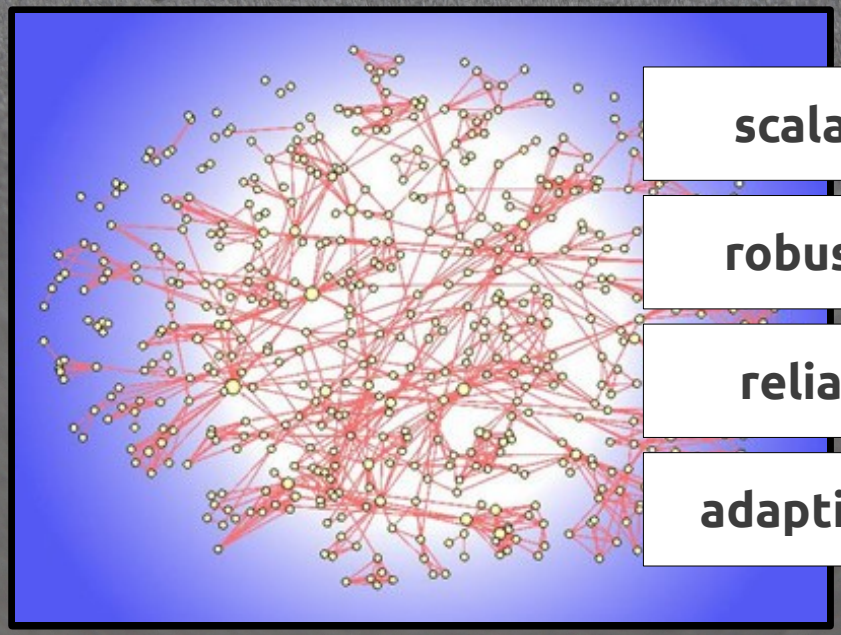
Outline

Introduction
Swarm Intelligence
OverSwarm
Conclusion

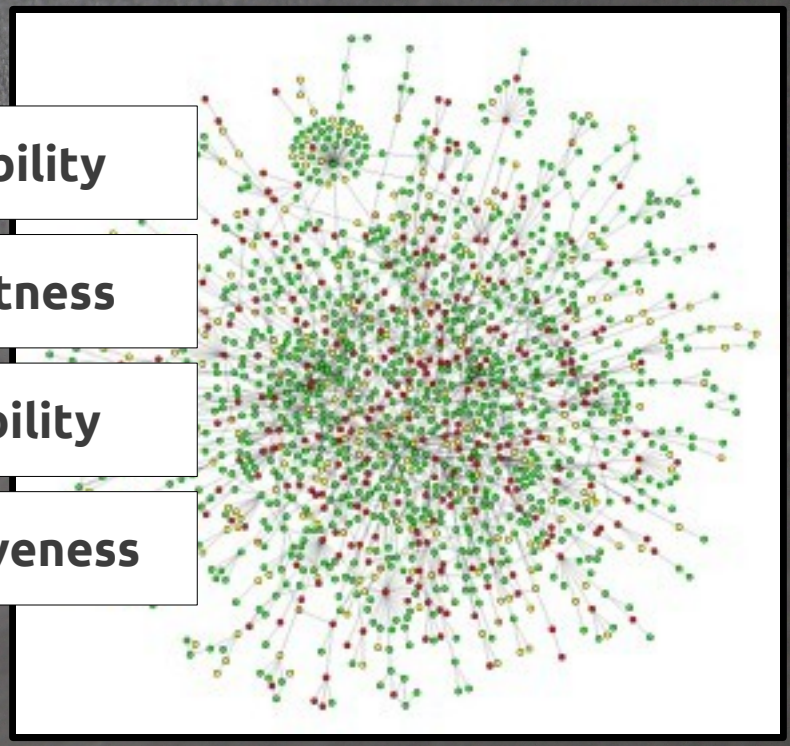


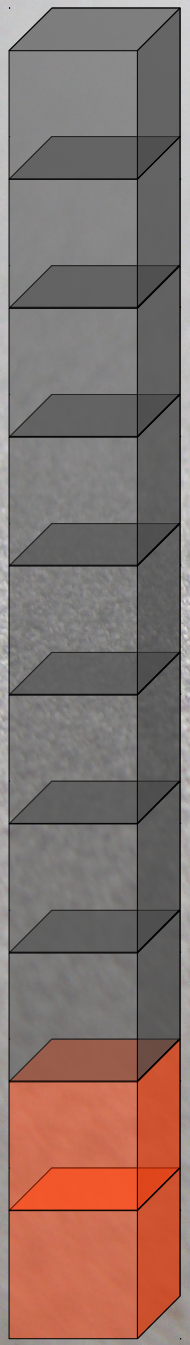
Introduction

Scenario: complex P2P distributed systems



- scalability
- robustness
- reliability
- adaptiveness

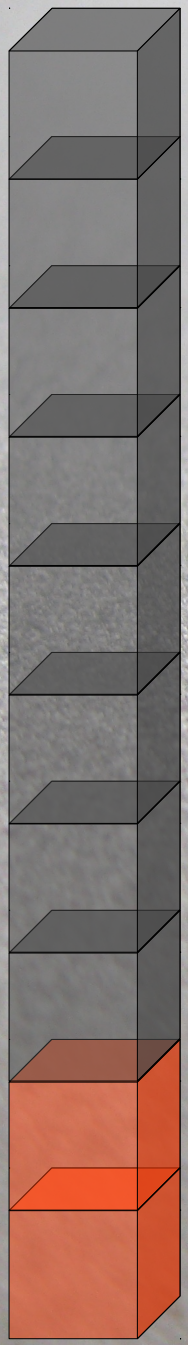




Introduction

Bio-inspired solutions

- self-organization
- emergence
- fault-tolerance



Examples

Messor (load balancing)

AntNet (routing)

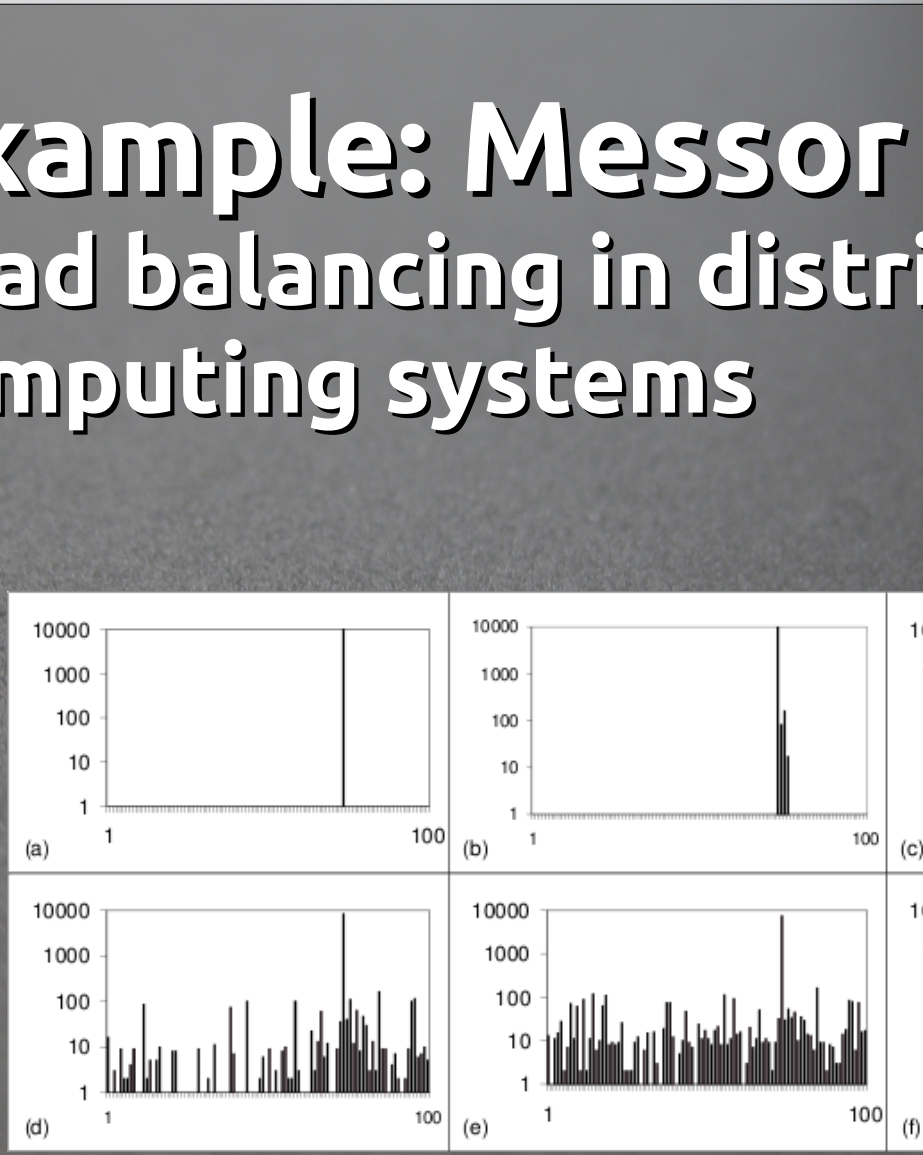
AntHocNet (routing)

BlåtAnt (overlay mgmt)

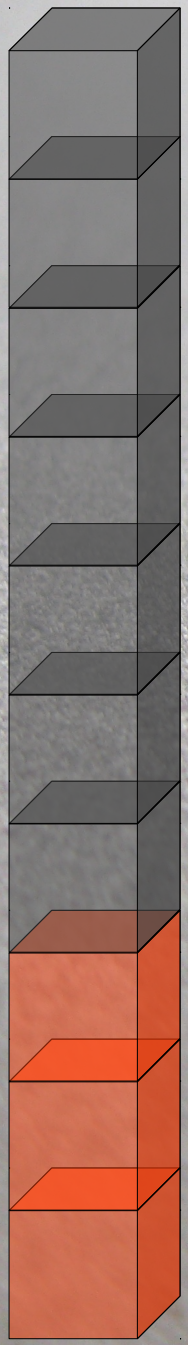
Self-Chord (DHT)

SemAnt (res. discovery)

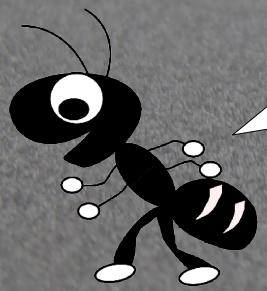
Antares (clustering)



Example: Messor
Load balancing in distributed computing systems



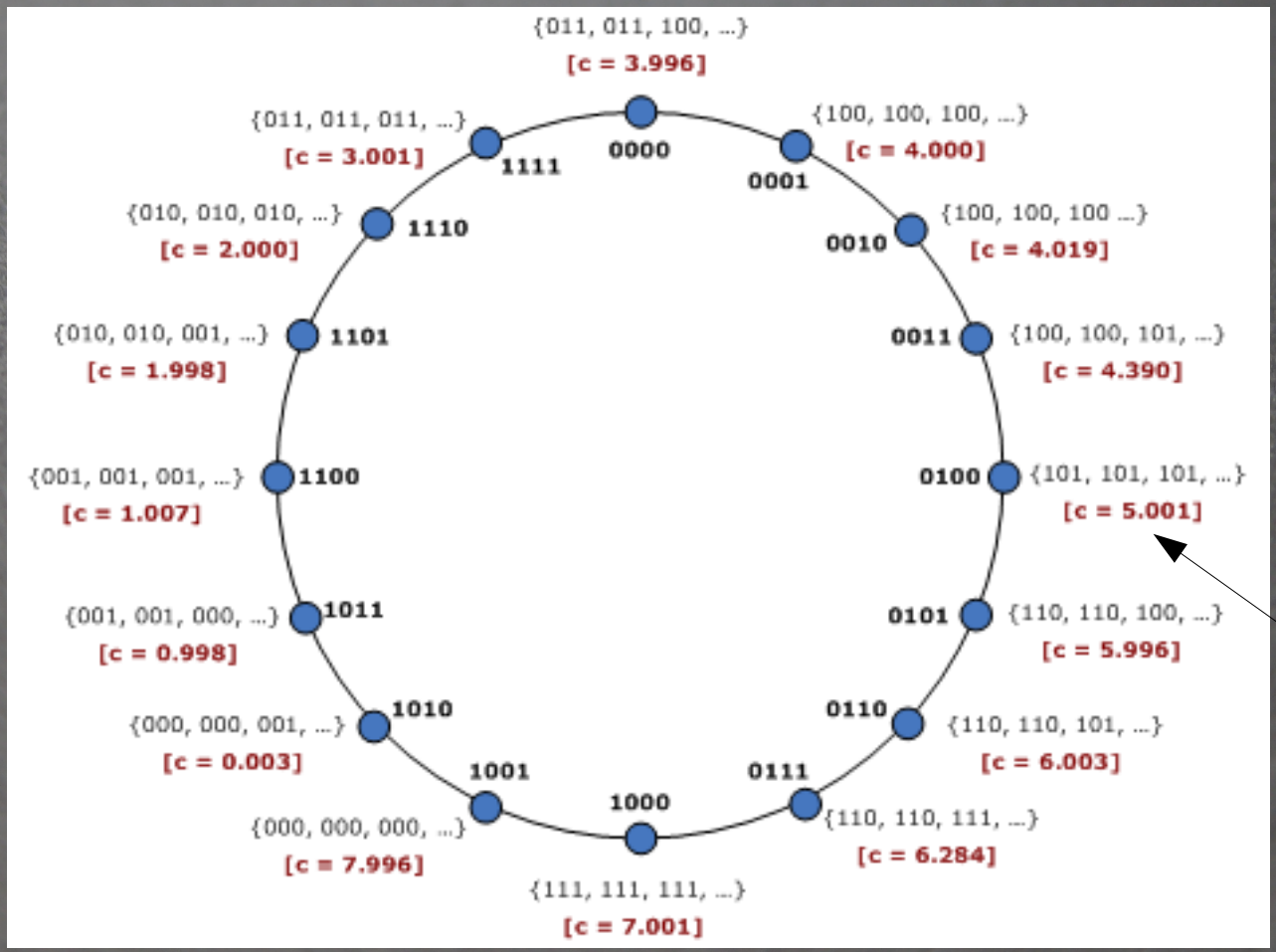
Example: Messor
Load balancing in distributed computing systems



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

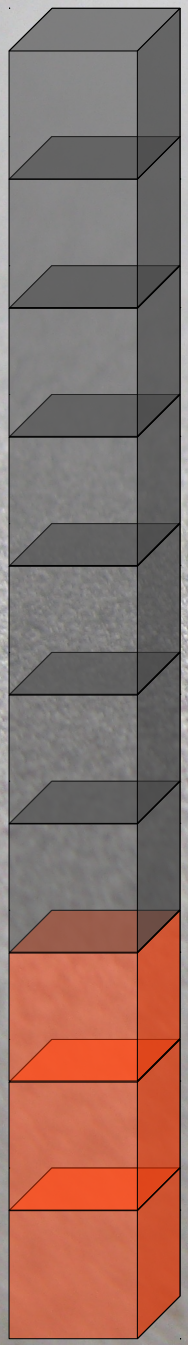
When an ant is carrying an object, the ant drops it only after having wandered about randomly "for a while" without encountering other objects.

Example: Self-Chord
Self-organized Chord DHT

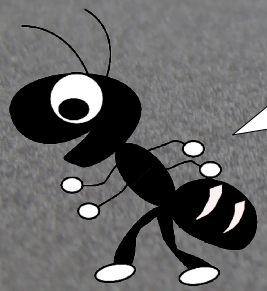


{keys stored in this peer}
[c = centroid of this peer]
peer index

Centroid: value that minimizes the average distance between itself and the keys stored in the local region

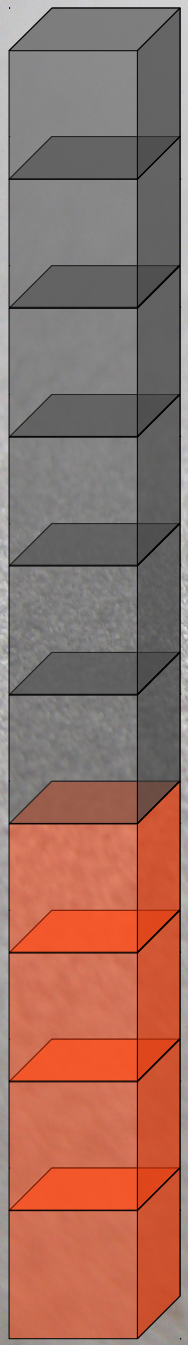


Example: Self-Chord
Self-organized Chord DHT



Mobile agents tend to pick a key from a peer if the key is an “outlier” there, and try to move the key to a peer whose centroid is similar to the key value

The agents try to deposit each key in the peer whose centroid in as close as possible to the key



Evaluation?

Custom simulator (Java)

Messor (load balancing)

AntNet (routing)

QualNet

AntHocNet (routing)

BlåtAnt (overlay mgmt)

Custom simulator (Java)

Self-Chord (DHT)

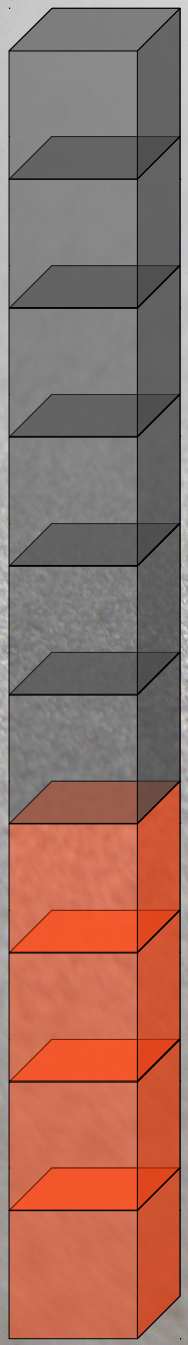
Custom simulator (Java)

SemAnt (res. discovery)

Unknown

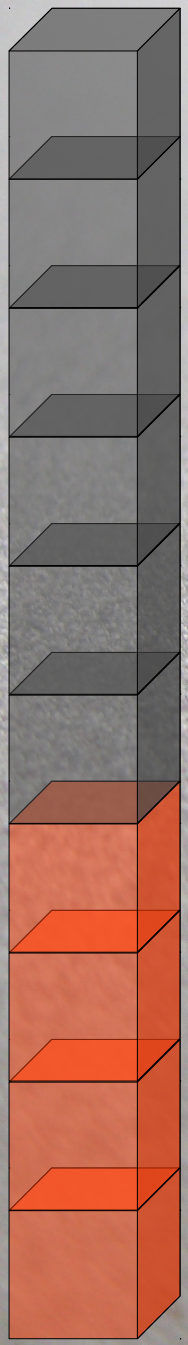
Antares (clustering)

Custom simulator (Java)



Goal

BlåtAnt		Chord
AntNet		Pastry
Messor		Kademlia
AntHocNet		Gnutella
SemAnt		GIA
Self-Chord		CAN
Antares		BitTorrent



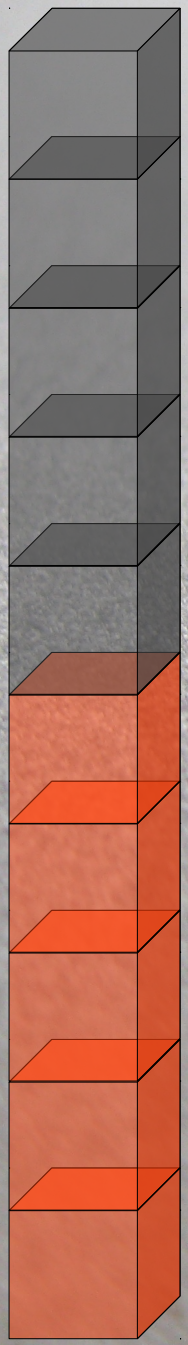
Goal

- BlåtAnt
- AntNet
- Messor
- AntHocNet
- SemAnt
- Self-Chord
- Antares

OverSim

OMNet++

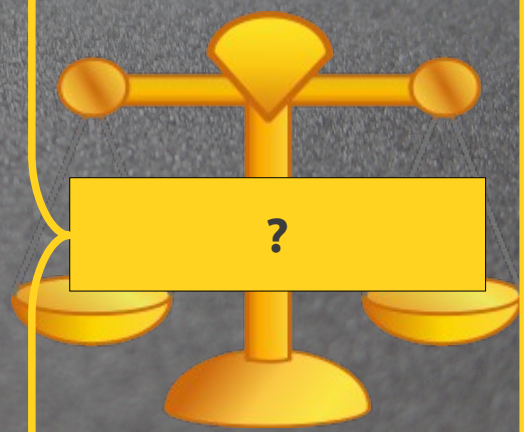
- Chord
- Pastry
- Kademlia
- Gnutella
- GIA
- CAN
- BitTorrent



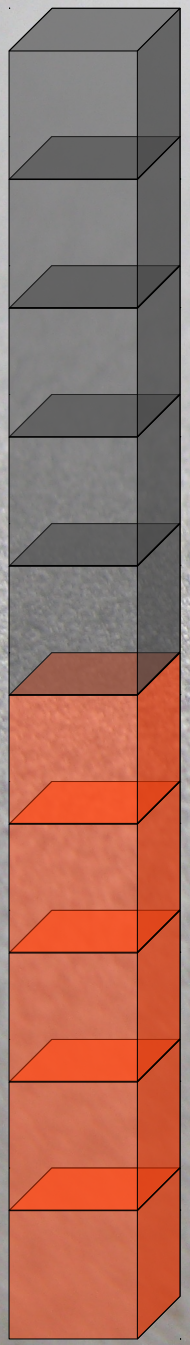
Goal

BlåtAnt
AntNet
Messor
AntHocNet
SemAnt
Self-Chord
Antares

OverSim
?
OMNet++

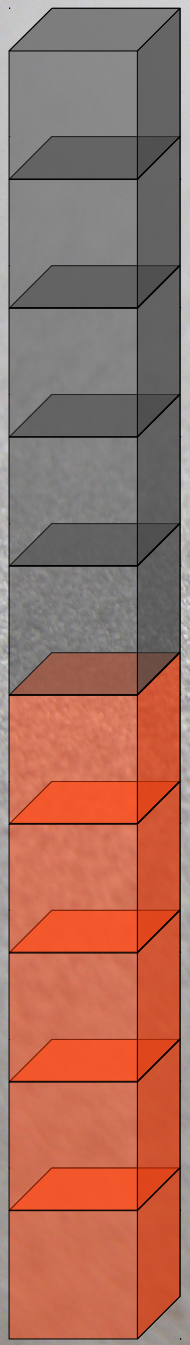


Chord
Pastry
Kademlia
Gnutella
GIA
CAN
BitTorrent



OverSwarm

- Bio-inspired / swarm framework
- Mobile-agent based protocols
- Goals:
  - Comprehensive evaluation
  - Comparison
  - Simplify development
  - Experimenting, prototyping



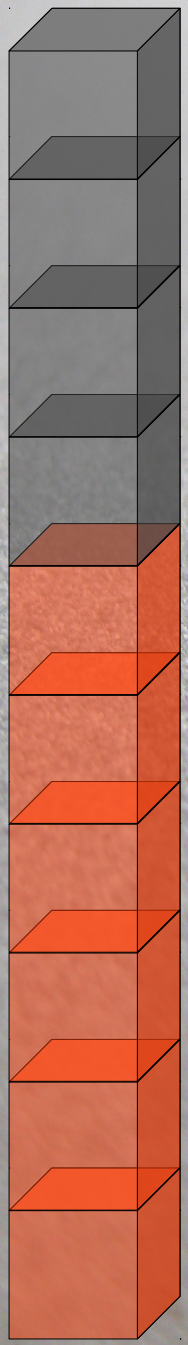
Application layer

Application Protocol

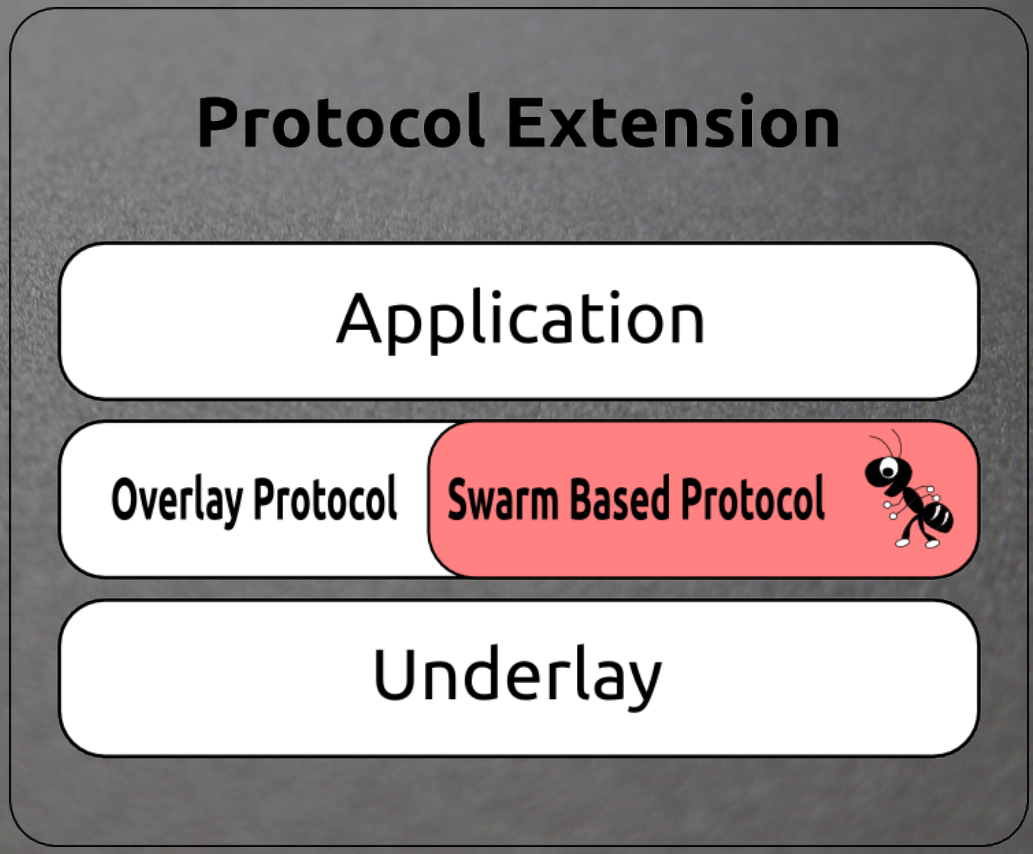
Swarm Based Application

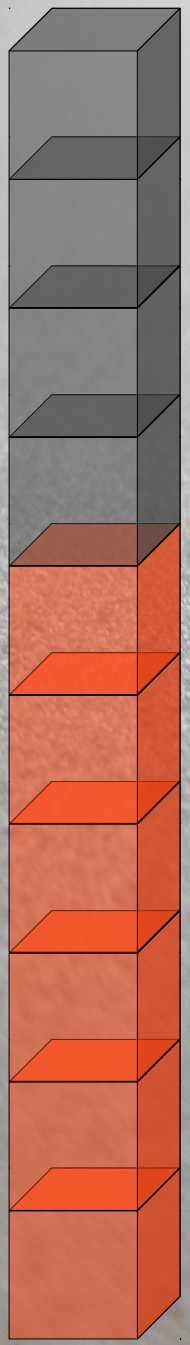
Overlay

Underlay



Overlay extension





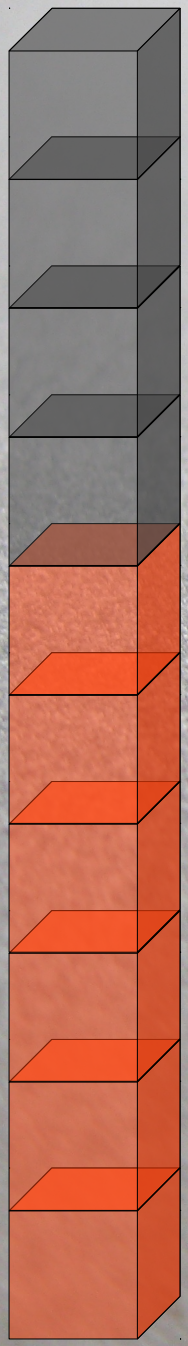
Stand-alone overlay

Standalone Protocol

Application

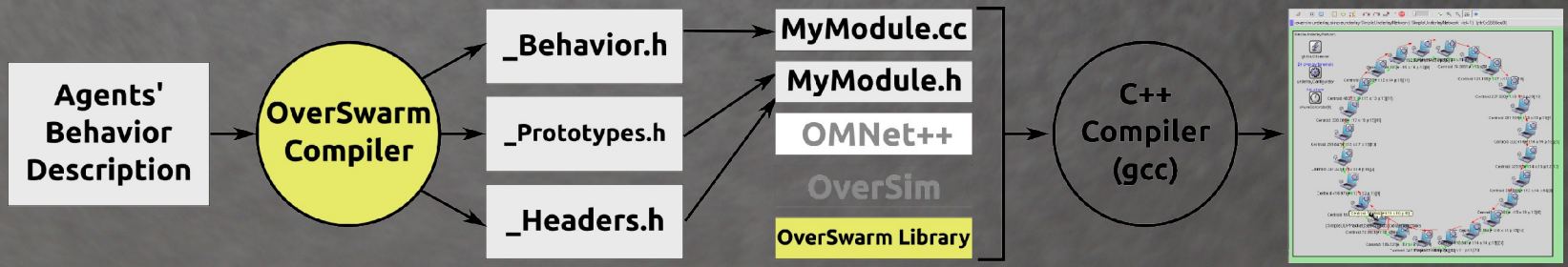
Swarm Based Overlay

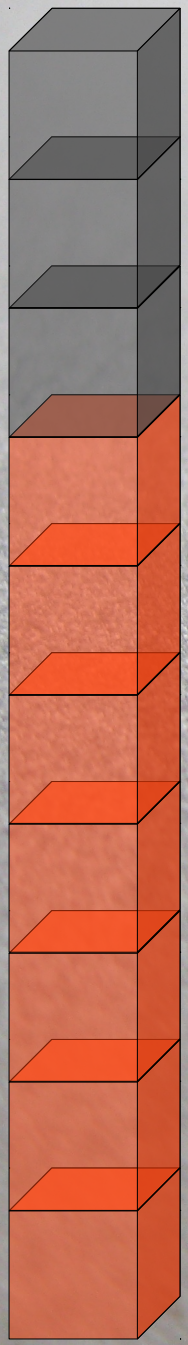
Underlay



Toolchain

- Programming language (Lisp-like)
  - dynamic typing
  - automatic memory management
  - strong, transparent migration
- Compiler

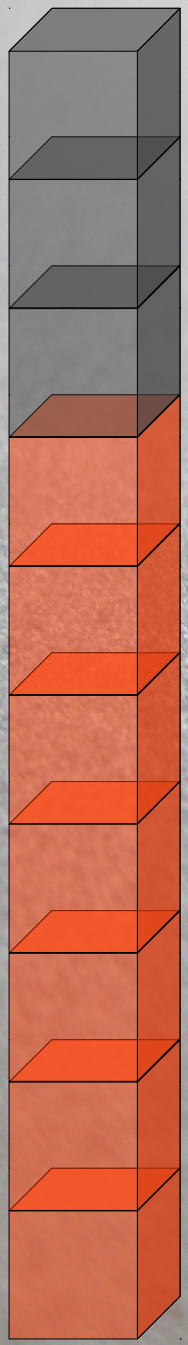




Programming

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.



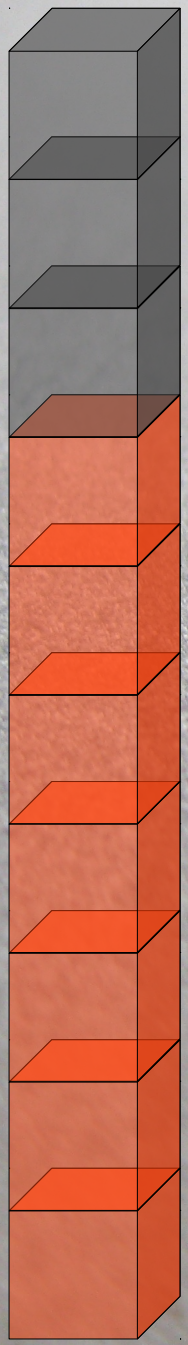
Programming

Ant Agent's Behavior

With probability 50%
migrate to the succ
migrate to pr
doSomething, then
back to the previous
if the result of do
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

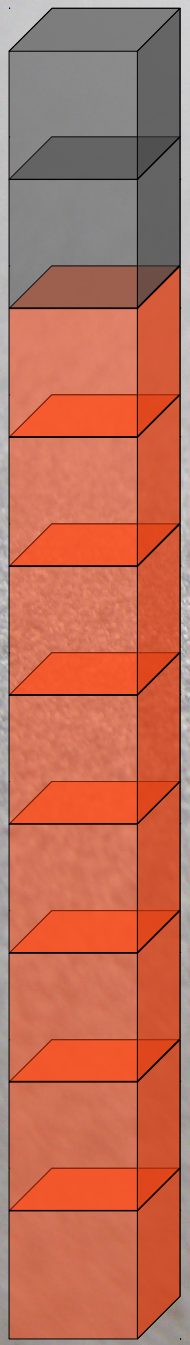
*With probability 50%
migrate to the succ
migrate to pr
doSomething, then
back to the previous
if the result of do
was greater than
otherwise doThat.*

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

OMNet++ / Ov

```
(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

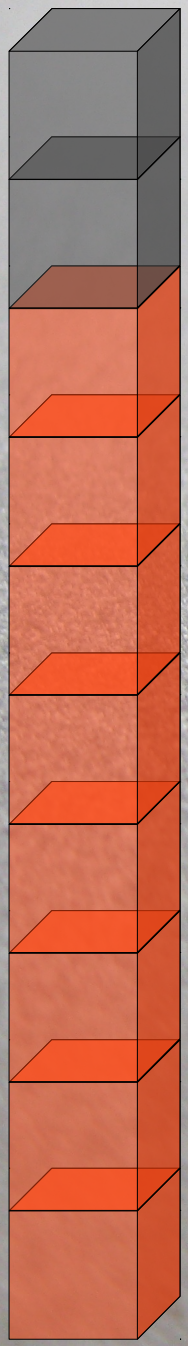


Example: Self-Chord

Self-organized Chord DHT

Reuses / Extends OverSim's Chord implementation

Adds ant-like agents to the existing protocol

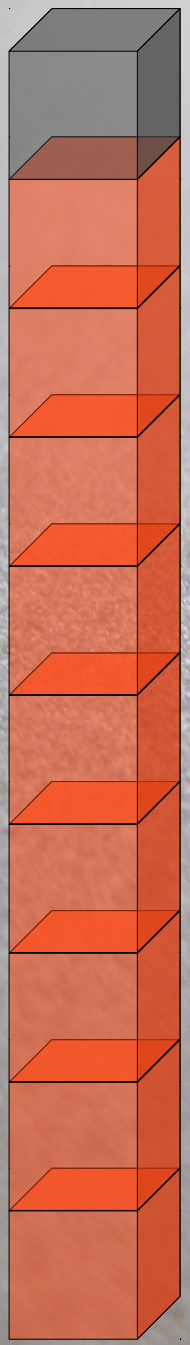


Example: Self-Chord
Self-organized Chord DHT

```
;; Pick function
(define (doPick) (synchronized
    (var c (getCentroid))
    (foreach r in (getResources) (begin
        (if (and
                (shouldPickA c r direction)
                (shouldPickB c r)) (begin
            (set! resource (pick (key r)))
            r
            (break)))))))

;; Drop function
(define (doDrop) (synchronized
    (if (shouldDrop (getCentroid) resource) (begin
        (drop resource)
        (set! resource nil)
        (end)))))

;; Body of the behavior
(while 1 (begin
    (if resource (doDrop) else (doPick))
    (if (= step 0)
        (if (not resource)(end))
    else
        (set! step (- step 1)))
    (if (and LOGARITHMIC_HOPPING resource) (begin
        (migrate (getNextHop (key resource))))
    else (begin
        (if (= direction LEFT)
            (migrate (getPredecessor))
        else
            (migrate (getSuccessor)))))))
```

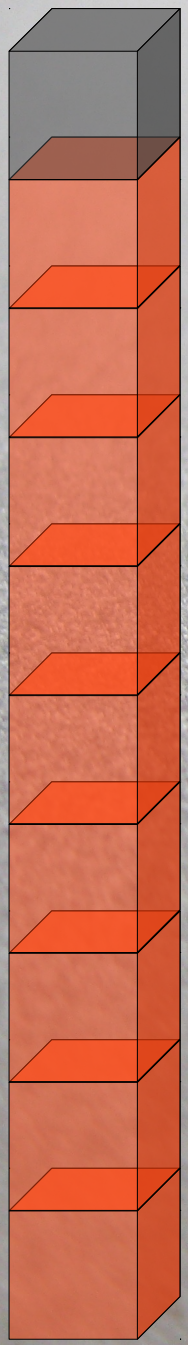
```
import for "oversim" {
    "dataTable->lock"            as      "lockDataTable"
    "dataTable->unlock"          as      "unlockDataTable"
    "dataTable->pick"            as      "pick",
    "dataTable->drop"            as      "drop",
    "dataTable->getCentroid"     as      "getCentroid",
    "dataTable->getRangeOfKeys"  as      "getRangeOfKeys",
    "dataTable->getResources"    as      "getResources",
    "getPredecessor"             as      "getPredecessor",
    "getSuccessor"               as      "getSuccessor",
    "getThisNode"                as      "getThisNode",
    "dataTable->setPredecessorKeys"  as      "setPredecessorKeys",
    "dataTable->setSuccessorKeys"    as      "setSuccessorKeys",
    "getLogarithmicNextHop"      as      "getNextHop"
}
```

```
OvSwValue::Ptr SelfChordDataTable::pick(void* owner, OvSwStack* st, OvSwValue::Ptr token)
{
    int key = unwrapInt(token);
    for (unsigned int i=0; i < data.size(); i++) {
        int hvalue = data[i].first;
        if (key == hvalue) {
            TableEntry t = data[i];
            OvSwMap* m = new OvSwMap();
            (*m)["key"] = wrap(key);
            (*m)["value"] = wrap(t.second.first);
            (*m)["owner"] = wrap(t.second.second);
            data.erase(data.begin() + i);
            updateCentroid();
            return wrap(m);
        }
    }
    return OVSWFALSE;
}

OvSwValue::Ptr SelfChordDataTable::drop(void* owner, OvSwStack* st, OvSwValue::Ptr token)
{
    OvSwMap m = unwrapOvSwMap(token);
    /* Unpack data */
    int key = unwrapInt(m["key"]);
    string value =  unwrapString(m["value"]);
    TransportAddress ta = unwrapTransportAddress(m["owner"]);
    /* Store data in the table */
    DataValue d = DataValue(value, ta);
    TableEntry te = TableEntry(key, d);
    addTableEntry(te);
    updateCentroid();
    return OVSWFALSE;
}
```



Current status

- Compiler + Tools working
- Several protocols implemented
  - Self-Chord (DHT)
  - BlåtAnt-S (overlay mgmt)
  - Newscast (gossiping)
  - Ozmos (load balancing)
  - Messor (load balancing)

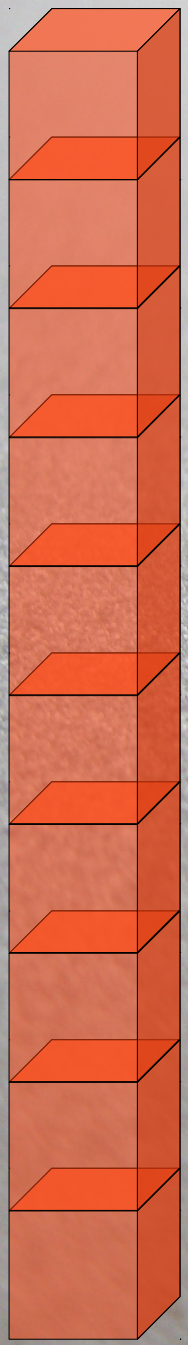


Present & Future work

- Additional bio-inspired protocols
- Documentation

- Bio-inspired function library:
  - Pheromone mgmt.
  - Epidemic protocols

- Unstructured overlays:
  - Topology measurements (diameter, av. path length, degree distribution,...)



Conclusion

- OverSwarm' goal:
  - comprehensive evaluation of bio-inspired P2P systems
  - comparison with existing approaches

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

References

- 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.