A Failure detector for Wireless Networks with Unknown Membership

L.Arantes, P.Sens, V. Simon
Univ. Paris 6 / INRIA/ CNRS

Fabiola Greve
UFBA

[Europar 2011 and Computer Journal 2012]

Outline

- Background :
  - Models of synchronization
  - Consensus problem
  - Unreliable failure detector (FD)
- Current implementations of FDs
- System model for dynamic networks
- Properties to achieve eventually strong FD
- Our proposal algorithm

Background

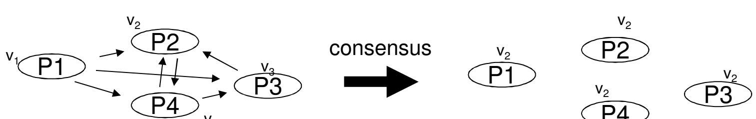
Synchronous model

- A distributed system is <u>synchronous</u> if:
  - there is a <u>known upper bound</u> on the transmission delay of messages
  - there is a <u>known upper bound</u> on processor speed
- A distributed system is <u>asynchronous</u> if:
  - there is <u>no bound</u> on the transmission delay of messages
  - there is <u>no bound</u> on processor speed
- A distributed system is <u>partial synchronous</u> if:
  - There is a global stabilization time (GST)
    - Until GST system is asynchronous
    - After GST system is synchronous
    - GST is not know

Background

Consensus in distributed system

- In the consensus problem, the processes propose values and have to agree on one among these values
  - Solving consensus is key to solving many problems in distributed computing (e.g., total order broadcast, atomic commit, group membership).



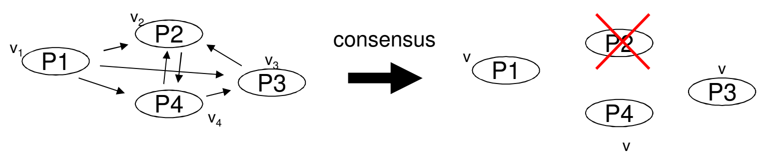
  - Properties :
    - ***Validity***: Any value decided is a value proposed
    - ***Agreement:*** No two correct* processes decide differently
    - ***Termination:*** Every correct* process eventually decides
    - ***Integrity***: No process decides twice

* Correct process: process that never fails

Background

Consensus in asynchronous systems

- ***FLP Impossibility result (Fischer, Lynch, and Paterson 85):*** Consensus cannot be solved deterministically in an asynchronous system subject to even a single process crash.
  - *The idea*: impossible to distinguish faulty hosts from slow ones

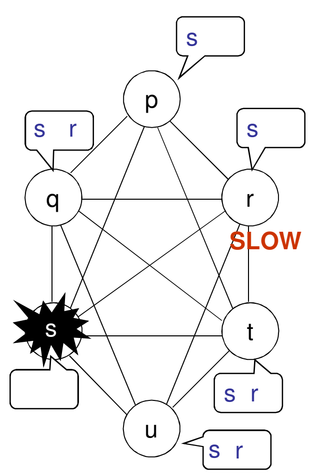


- ***Possibility result (Chandra & Toueg 96):*** Consensus can be solved in an asynchronous system subject to failures with an *unreliable failure detector*
  - The idea: partial synchrony assumptions are encapsulated in the unreliability of failure detectors.

Background

Unreliable failure detectors

- Introduced in the beginning of 90's by Chandra and Toueg
- Failure detector = an oracle per node
- Oracles provide a list of hosts *suspected* to have crashed
  - possibly false detections
- Abstractly characterized in terms of two properties: *completeness* and *accuracy*
  - Completeness characterizes the capacity with which failed processes are suspected by correct processes
  - Accuracy characterizes the capacity with which correct processes are not suspected, i.e., restricts the false suspicions that a failure detector can make



Background

Properties of FD

- **Strong completeness**:
  - Eventually every process that crashes is permanently suspected by every correct process
- **Accuracy**:
  - [Eventual] Strong : [There is a time after which] correct processes are not suspected by any correct processes
  - [Eventual] Weak : [There is a time after which] **some** correct processes are not suspected by any correct process

	Accuracy			
	Strong	Weak	Eventual Strong	Eventual weak
Strong completeness	P	S	◊P	◊**S**

◊S the weakest FD to solve consensus

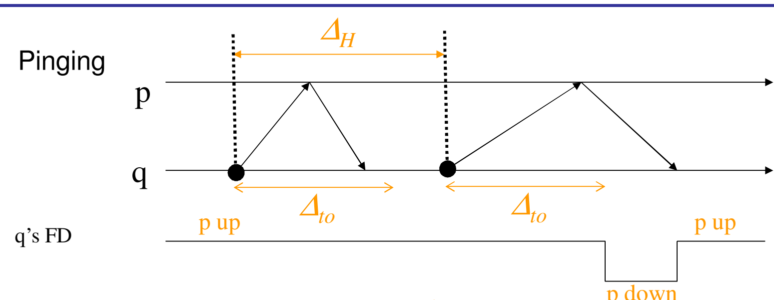
IME -2013

7

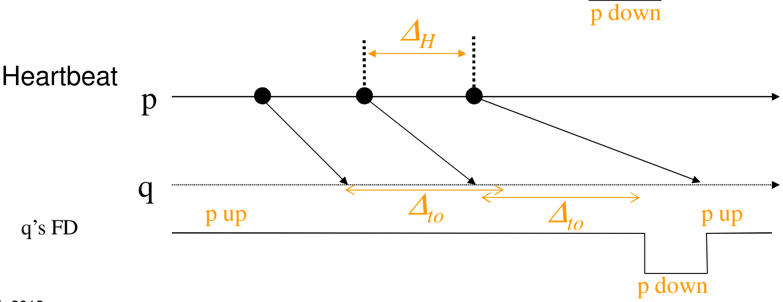
FD Implementation

Timer-based implementation of FD

- Pinging



- Heartbeat



IME -2013

8

FD Implementation

Asynchronous implementations

- Base on query-response mechanism
  [Mostefaoui, Mourgaya, Raynal 03]
- Assumptions :
  - ∏ = {p1,p2, .., pn} known processes
  - Completed graph
  - f = maximum number of crashes
- Principle:

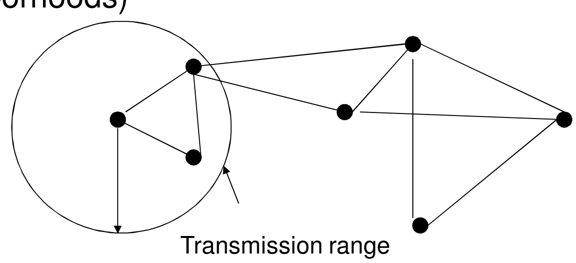
send REQUEST to **n** nodes
wait for **n-**f RESPONSE
suspected = set of nodes that do not response

IME -2013 9

Dynamic System Model

Features of dynamic systems

- Unknown membership
  - set and number of nodes are unknown
- Dynamic graph due to mobility
- Communication via transmission range (broadcast to neighborhoods)



- Complex to fix timeout of transmission delays due to the dynamics of the network

IME -2013 10

Dynamic System Model

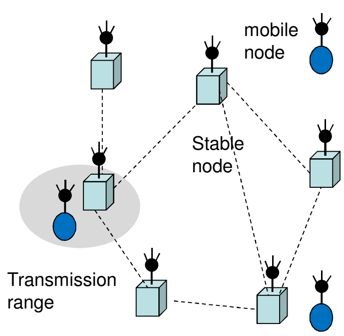
Definitions

- $\prod = \{p_1, p_2, .., p_n\}$
  - $\prod$ and ***n*** *are unknown*
- Processes can crash or leave the system
- The wireless mobile network is represented by a communication graph $G = (V, E)$
  - $V = \prod$
  - E= set of logical links
- R_i be the transmission range of p_i
- N_i^t : set of *1-hop neighbors* (nodes within R_i) at t
- d_i^t: *range density* ($d_i^t = |N_i^t|$)
- f_i = maximum number of failures in the neighborhood of process p_i

IME -2013 11

Dynamic System Model

Processes status in a network with unknown membership



- Wireless Mesh Networks
- Wireless Sensor Networks

- In order to implement FDs with an unknown membership, processes should interact with some others to be known.
  - The actual membership of the system is in fact defined by the KNOWN set. A process is *known* if, after having joined the system, it has been identified by some stable node.
- A *stable process* is a non faulty process that, after had entered the system for some point in time, never departs; otherwise, it is *faulty.*

IME -2013 12

Dynamic System Model

System Membreship

- Let *UP(t)* be the set of processes that are in the system at time t. Let $known_q$ set denotes the partial knowledge of *q*
  - initiallly, $known_q = \{q\}$

STABLE = $\{p : \exists t, \forall t' \geq t, p \in UP(t')\}$

FAULTY = $\{p : \exists t, \forall t', t < t', p \in UP(t) \wedge p \notin UP(t')\}$

KNOWN = $\{p : (p \in STABLE \cup FAULTY) \wedge (p \in known_q, q \in STABLE)\}$

  - The membership of the system is actually defined by the KNOWN set.

IME -2013 13

Dynamic System Model

Communication Model

- *Links reliable:*
  - No message loss neither corruption nor duplication
  - Reliable delivery of broadcast data in transmission range
    - All stable neighbors receive the message
- *Connectivity:*
  - Eventually there is a path between every pair of stable (correct) processes
    - o in spite of changes in the topology of *G, from some point in time t, the set KNOWN* ∩ STABLE forms a ***strongly connected*** *component in G.*

IME -2013 14

Dynamic System Model

◊S^M: Eventually strong FD with unknown membership

- ◊S^M FD is **time-free** and based on a local **query-response** communication mechanism
  - A process p_i *launches the primitive by* sending a query(*m) with a message m to its neighbors within its transmission range.*
  - When a process p_j *delivers this query, it* systematically answers by sending back a response(*m') with a message m' to* p_i.
  - *When* p_i *has received at least* α_i *responses from different* processes, the current query-response *terminates.*
    - $\alpha_i = N^t_i - f_i$

IME -2013 15

Properties

Properties of ◊S^M

- Same properties of ◊S FD, but restricted to known processes
  - *Strong completeness*: every **known and faulty** process is eventually suspected by all known and stable processes.
    $\{\exists t, \forall t' \geq t, \forall p \in STABLE \cap FAULTY \Rightarrow p \in susp_q, \forall q \in KNOWN \cap STABLE\}$
  - *Eventual weak accuracy*: Eventually, at least one **stable and known** process is never suspected by any known and stable processes.
    $\{\exists t, \forall t' \geq t, \exists p \in KNOWN \cap STABLE \Rightarrow p \notin susp_q, \forall q \in KNOWN \cap STABLE\}$

IME -2013 16

Properties

Properties to implement a ◊S^M FD

1) **Stable Termination Property** (SatP): Each QUERY must be received by at least one stable and known node

⇒ *Necessary for the diffusion of the information*

2) **Mobility Property** (MobiP): In its new neighborhood, a moving node should have received a QUERY for at at least one stable neighbor.

⇒ *p_i updates its state with recent* information

3) **Stabilized Responsiveness Property** (SRP (p_i)): eventually, the set of responses received by any neighbor of *p_i to their last QUERY always includes a response from p_i*. Moreover, *neighbors of p_i* eventually stop moving outside *p_i's transmission range.*

⇒ *SRP should be hold for at least one stable known node*

⇒ *Necessary for weak accuracy (eventually the "SRP node" will not be suspected)*

IME -2013

17

Algorithm

Time-free ◊S^M FD Algorithm (1)

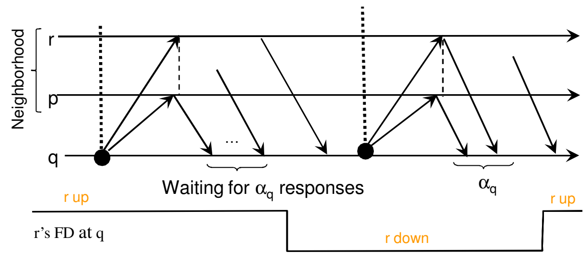
- Principles :
  - Local detection of neighbor's failure based on **query-response** exchange
  - Flooding of failure information (suspected nodes and mistakes)

- Notations :
  - *susp*: set of processes suspected of being faulty
  - *mist.*: set of nodes which were previously suspected of being faulty but such suspicions are currently considered to be a mistake.
    - suspected and mistake information are tagged by a local counter.
  - *rec_from*: set of nodes from which p_i has received responses to its last query message.
  - *known*: denotes the current knowledge of p_i about its neighborhood.

18

IME -2013

Algorithm : Sending of QUERY



Task T1:

Repeat forever

  broadcast QUERY($susp_i$, $mist_i$)

  wait until RESPONSE received from $\geq \alpha_i$ processes

  $rec\_from_i \leftarrow$ all p_j, a RESPONSE is received in line 6

  For **all** $p_j \in known_i \setminus rec\_from_i \mid \langle p_j, - \rangle \notin susp_i$ do

    If $\langle p_j, ct \rangle \in mist_i$

      $Add(susp_i, \langle p_j, ct+1 \rangle)$

      $mist_i = mist_i \setminus \{\langle p_j, - \rangle\}$

    Else

      $Add(susp_i, \langle p_j, 0 \rangle)$

End repeat

Algorithm (2) : Reception of responses

Task T2:

Upon reception of QUERY ($susp_j$, $mist_j$) from p_j do

$known_i \leftarrow known_i \cup \{p_j\}$

For all $\langle p_x, ct_x \rangle \in susp_j$ do

If $\langle p_x, - \rangle \notin susp_i \cup mist_i$ *or* $(\langle p_x, ct \rangle \in susp_i \cup mist_i$ *and* $ct < ct_x)$

  If $p_x = p_i$

    $Add(mist_i, \langle p_i, ct_x + 1 \rangle)$

  Else

    $Add(susp_i, \langle p_x, ct_x \rangle)$

    $mist_i = mist_i \setminus \{\langle p_x, - \rangle\}$

The receiver p_i is suspected : generation of a mistake

Update susp. set with more recent information

For all $\langle p_x, ct_x \rangle \in mist_j$ do

If $\langle p_x, - \rangle \notin susp_i \cup mist_i$ *or* $(\langle p_x, ct \rangle \in susp_i \cup mist_i$ *and* $ct < ct_x)$

  $Add(mist_i, \langle p_x, ct_x \rangle)$

  $susp_i = susp_i \setminus \{\langle p_x, - \rangle\}$

  If $(p_x \neq p_j)$

    $known_i = known_i \setminus \{p_x\}$

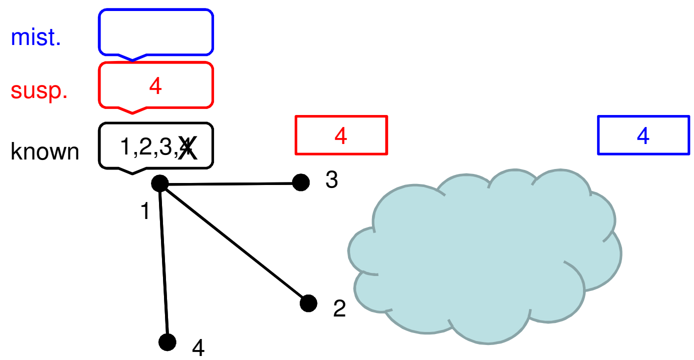
send RESPONSE to p_j

The sender is not the mistake node p_x: suspicion that p_x has moved

Update mist. set with more recent information

Algorithm

Exemple: Mobility of nodes



Performance Evaluation

- **OMNet++**
- **QoS Metrics** (Chen et. al.)
  - ***Detection Time***: time that elapses from *p*'s crash to the time when q starts to suspects *p* permanently;
  - ***Mistake Recurrence Time***: The time between two consecutive mistakes;
  - ***Mistake Duration***: The time it takes for the FD to correct a mistake.

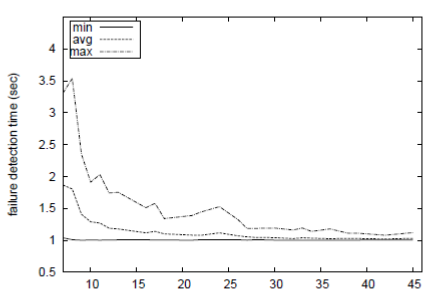
Performance evaluation

- Parameters
  - N=100
  - Two-dimensional regions:
    - 600mx600m
    - 200mx1800m
  - Every node has at least 5 neighbors
  - At most 2 neighbors can crash
    - 10% of total nodes can crash
      - 10 nodes every 70s starting at 10s
  - Delay of 1s between every query was introduced

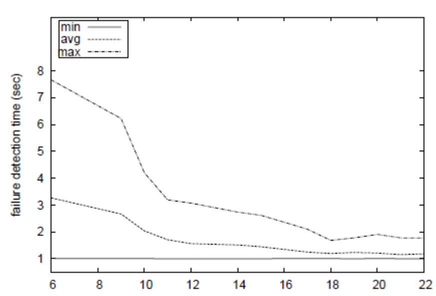
IME -2013 23

Performance Evaluation

- **Impact of the number of neighbors** on the failure detection time.
  - Transmission range r varies from 100m to 380m
    - Variation of the number of neighbors



(a) 600x600 region

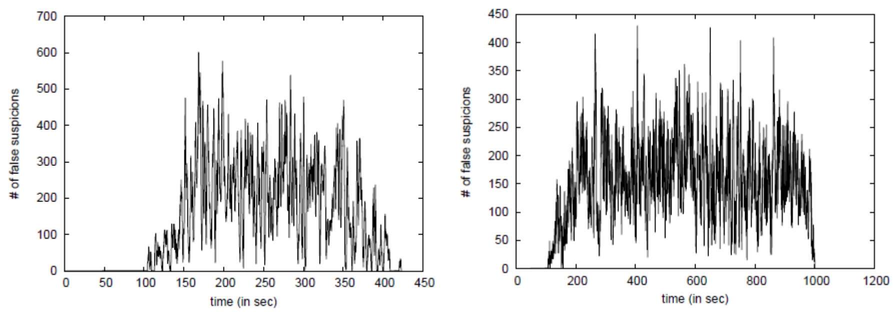


(b) 200x1800 region

IME -2013 24

Performance Evaluation

- **Impact of Mobility**: accuracy property when both ten nodes located at one boundary of the network move at a speed of 2m/s.
  - First one starts moving at 100s and at every 5s a new start moving

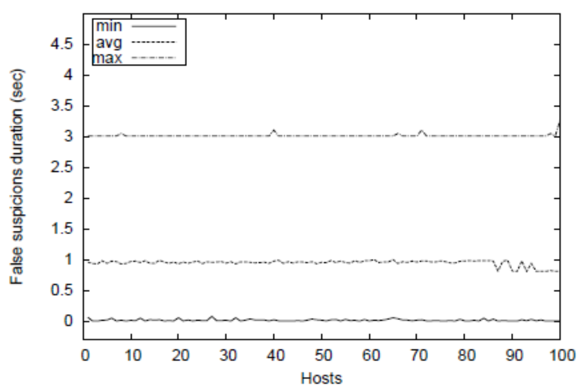


(a) 600x600 region (b) 200x1800 region

IME -2013 25

Performance Evaluation

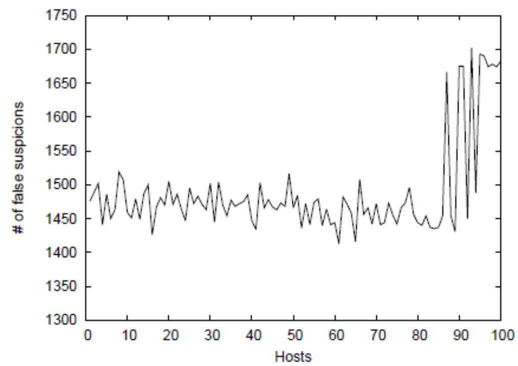
- **Distribution of False Suspicion Duration**
  - 200m x1800m region
  - For all nodes when 10 nodes located at one boundary move



IME -2013 26

Performance Evaluation

- **Distribution of total number of false suspicions**
  - 200m x1800m region
  - for the N=100 nodes when 10 nodes located at one boundary



IME -2013

27

Conclusions

- Implementation of FD for dynamic networks with unknown membership:
  - Timer-free
  - Based on local failure detection and diffusion
- Definition of properties for $\Diamond S^M$
  - Membership
  - Minimum stability of moving nodes
  - Stable Responsiveness to Queries

IME -2013

28

Other Related Work and Perspectives

- A relaxed model with path constructed over time
  - Time Varying Graph (Casteigts et al 2010)
    - Computer Journal 2012
- Byzantine time-free Failure Detector
  - *Strong Byzantine completeness:* eventually, every stable known process suspects permanently every process that has detectably deviated from algorithm $\mathcal{A}$;
  - When $\mathcal{A}$ requires processes to exchange a message *m*, every process *p* waits until the reception of *m* from at least f_p+1 distinct senders.
    - Majority of correct messages: $|N_p| > 2f_p$
    - EDCC 2012 and WRAITS 2011
- Algorithm: Implementation of Ω (eventual leader election)
  - Solve consensus
    - current work

IME -2013 29

Some other recent work

- Distributed Mutual Exclusion
  - Nodes communicate only by message passing.
    - Logically organized: ring, tree, complete graph
  - Shared Resource; The code that access the shared resource is called the *critical section (CS)*.
  - Ensures two properties :
    - ***Safety*** : at most one process can execute the critical section at any given time
    - ***Liveness*** : all critical section requests will be satisfied
  - Divided into two families:
    - *Token-based*
      - A node can access the resource if it holds a token, which is unique in the system.
    - Permission-based
      - A node can access the resource only after having received permission from all nodes

IME -2013 30

Some other recent work

- **Distributed Mutal Exclusion: Dynamic Networks**
  - IME (Alfredo Goldman and Paulo Floriano)
  - Formalization of the Necessary and Sufficient Connectivity Conditions to the Distributed Mutual Exclusion Problem in Dynamic Networks.
    - Framework : Evolving Graphs and Graph Rellabelings
      - NCA 2011 and SBRC 2012

Some other recent work

- **Distributed Mutual Exclusion in Clouds**
  - SLA-oriented mutual exclusion algorithm
    - Service Level Agreement (quality of service) assigned to requests:
      - Priority
      - Response time (deadline)
    - Token-based solutions where nodes are organised on a logical static tree
      - Adaptation of Raymond and Kankar-Chaki algorithms
    - Aim: to reduce the number of SLA violations
    - CCGRID 2012

Fault tolerant k-mutual exclusion

- K-mutex: At any time k units of the resources can be used
  - Fault tolerant solution for Hypercube topologies
    - Hi-ADSD diagnostic system
      - Spanning tree
  - Universidade Federal do Parana
    - WTF 2012 and ISPDC 2013

IME -2013

33

Some other recent works

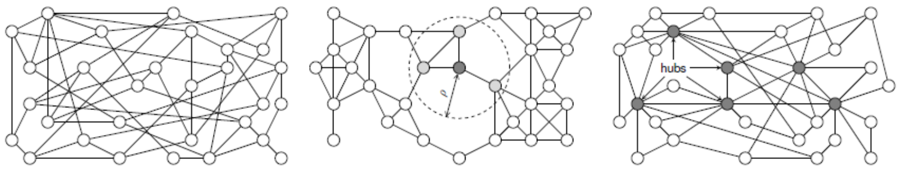
- **Gossip algorithms on large-scale networks**
  - Information dissemination to all other sites
  - Topology is a not complete graph.
  - Flooding algorithm presents poor performance
    - Upon the first reception of a message, every site of the network relays it once to its neighbors
      - » High number of redundant messages
  - Gossip protocols
    - Upon the first reception of a message, every site of relays it once to its neighbors based on some probability
      - » reduces the number of redundant message
      - » Should guarantee high reliability: percentage of nodes that receive all broadcast messages.

IME -2013

34

Some other recent works

- Performance evaluation comparison of gossip algorithms over different topologies, message complexity, dissemination of information, number of infected nodes, latency, etc.
  - SRDS 2012
- A new gossip protocol for scale-free topologies that exploits the dissemination power of hubs of scale-free networks
  - Submitted to ICPP 2013



(a) Bernoulli Graph (b) Random Geometric Graph (c) Scale-Free Graph

- STIC-Amsud (Yahoo Research (Chile), UFSCar, San Louis (Argentina), LIP6) : search of information in disaster area

Laboratoire d'Informatique de Paris 6 LIP6

- Attached to the University Pierre et Marie Curie (Paris 6) and CNRS
  - 177 researchers and 200 PhD students
  - 5 Departments :
    - ***Réseaux et Systèmes Répartis***
      - ***Regal***
      - Move,
      - ARP
      - NPA,
      - Phare
      - Complex Network
    - Calcul Scientifique
    - Decision, Systèmes Intelligents, Recherche Opérationelle
    - Données et Apprentissage Artificiel
    - Systèmes Embarques sur Puce

Regal

- Joint project-team with INRIA Research Team
  - ✓ 8 faculty members, 3 researchers, 2 engineers, 12 PhD students

Scientific context

Distributed system for large and dynamic networks

Focus on information sharing, distributed algorithms

Features
- Large number of resources
- Heterogeneity
- Asynchronous networks
- Dynamicity (Failure, disconnection)
- Security

no global state

Challenges
- Fault Tolerance
- Scalability
- Data Storage, availability
- Deployment/efficient accesses to remote services
- Dynamic adaptation

Regal approach : end-to-end (from algorithms to experimentations)

→ Algorithm →Prototype → Experimentation → Evaluation

IME -2013

37

Regal

- Some Projects:
  - **Distributed Algorithms or large scale systems, mobile systems, dynamic systems, Grids, and Clouds**
    - Fault tolerance, failure detection, dynamicity, auto-stabilisation, elasticity, MapReduce
  - **P2P Data Storage**
    - Pastis (File System) , Publish-subscriber Systems, Indexing/caching, replication
  - **Replication and Consistency for large scale systems**
    - Optimistic approach, weaker models, *Actions-Constraints Formalism* formal model
  - **Dynamic configuration of OS**
    - VVM : programming and an execution environment allowing to adapt the java virtual machine on the fly.
    - Dynamic detection of bugs

IME -2013

38