Geo-localized mobile systems: a glimpse of theory

Workshop "Internet of Things", Oct 21. LAAS-CNRS

Matthieu Roy (TSF / LAAS-CNRS)

roy@laas.fr

Marc-Olivier Killijian, David Powell (LAAS)

François Bonnet (IRISA)

Leodardo Querzoni, Silvia Bonomi (Univ Roma)

Context

* Two fundamental technological shifts:
  * internet -> ambient systems
    * deployment of user-carried systems
      * wireless communication (short range) + localization devices
    * link between physical and logical (network) world

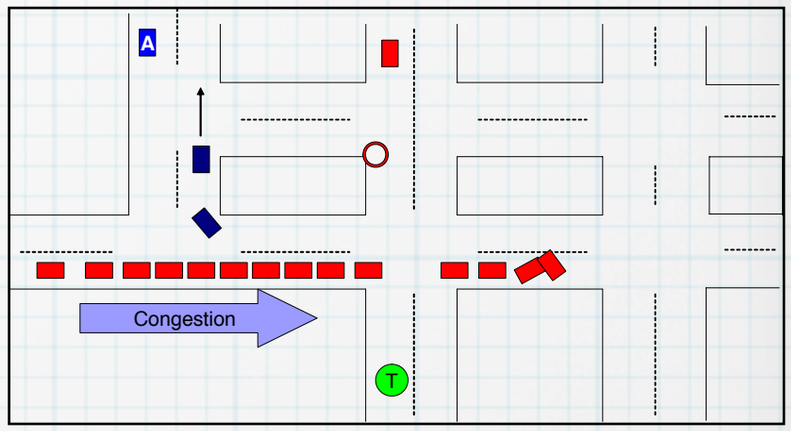
Where do we stand ?

* Extensive research in "closed" systems
  * abstractions, models, algorithms for resilience
* Extensive research on Internet
  * routing, models, structures (overlays)
* Can we get the "best of both world"
  * i.e. provide **localized resilient services**

What's the Challenge ?

* **Formalization** of the system (local)
  * geographic properties
* **Architectural** design (abstractions or building blocks)
  * inspired by (traditional) distributed Σ
* Development of **algorithms**
* (Assessment on a generic experimental platform)

Is there any application to this ?



* Real-life physical examples
* users deploy a white board
* perform better GPS route calculation
* based on users' experience of the traffic
* cooperative backup of critical data
* distributed black box, etc..
* augmented games

System's characteristic parameters

"classical" systems	mobile systems
failure (node, link)	normal behaviour : disconnections, unreliable wireless communication
(small) fixed number of nodes	variable and huge size system
no link between physical world and network	strong coupling with physical environment

So we'll go local only

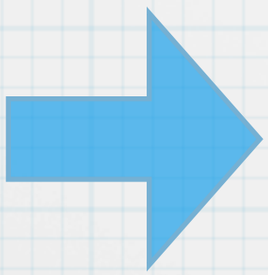
* Local = geo-localized
* Everything must be (re)defined w.r.t. a particular location in space.
* Semantics must be coherent with systems' characteristics:
  * when no user populates a region, it's not possible to keep a state alive

System definition

* Entities $(p_i)_{i>0}$
  * evolve in 2D space with bounded speed
  * equipped with positioning device (infinite precision)
  * communication using wireless device
* Let's concentrate on an area **A**

Abstractions for mobile systems

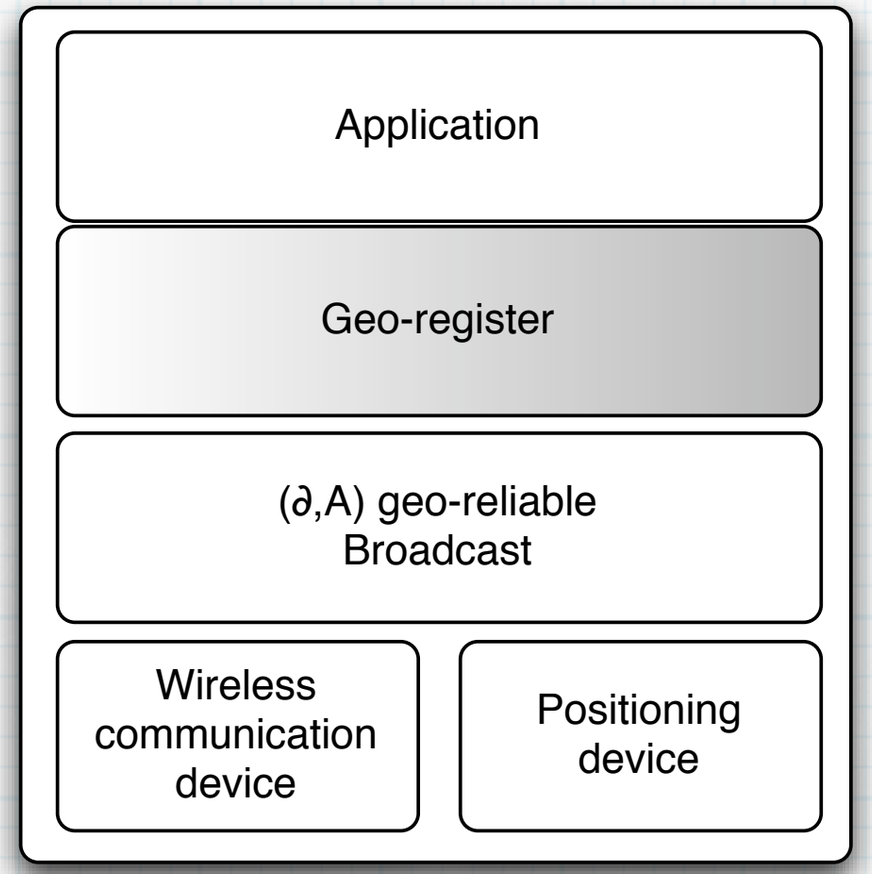
* Traditional distributed abstractions
  * storage: registers, transactional memories
  * agreement: consensus
  * group management: membership



Need to be adapted in geo-aware versions

Building Blocks/ Abstractions

Simple example:

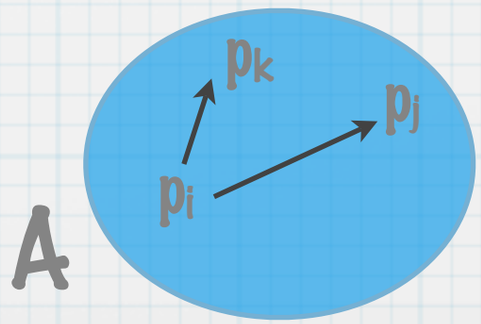


Shared storage/ Register

Geo-reliable broadcast

By hypothesis...

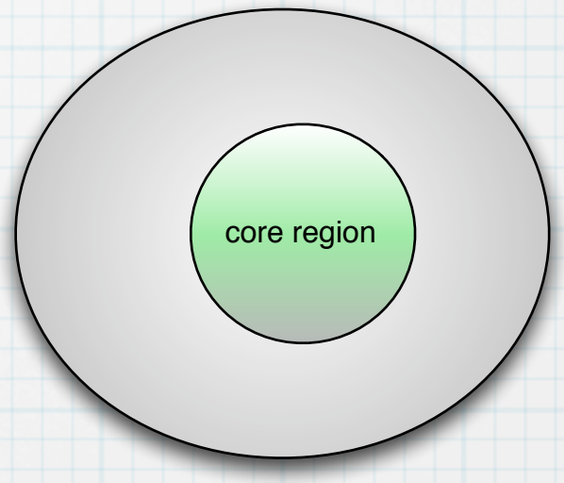
* (∂,A) geo-reliable broadcast:
  * every process in A can issue a broadcast(m)
  * if m is broadcasted at time t by a process that remains in A from t to t+∂ then all processes in A during [t,t+∂] deliver the message



Geo-reliable broadcast

But...

* If a process leaves **A** during the sending interval... no guarantee
* Core region (geographic definition)
  * a subset **A'** of **A** s.t. every message sent by a process in **A'** will be delivered by all correct processes that were in **A'** when the message was sent

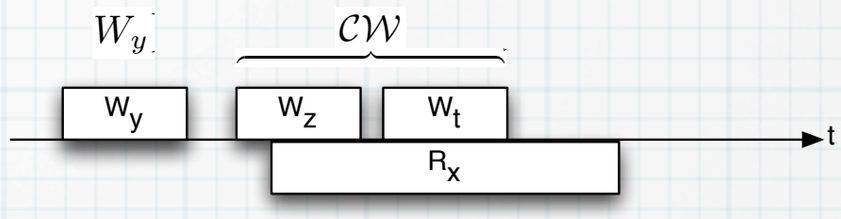


Geo-registers

* Simple case : Non concurrent writes
  * write is allowed in the core region A'
  * read is allowed in A (roughly)
    * a read operation tries to return the last written value

Non concurrent write semantics

What is the "last written value" ?



* V={last written value, concurrently written values} (here $V=\{y,z,t\}$)
* If, since the last completed write operation,
  * 1) core region was never empty, then $v \in V$ must be returned
  * 2) else it returns $v \in V$ or $\perp$

Geo-registers

Geographically controlled thread:

when p enters A:
  $R_p \leftarrow void$;
  wait for
    $\square$ ($W(x)$ is received) : $R_p \leftarrow x$; exit;
    $\square$ (2δ time delay elapsed)
  RB_send(REQ)
  wait for
    $\square$ ($REP(v)$ is received) : $R_p \leftarrow v$;
    $\square$ ($W(x)$ is received) : $R_p \leftarrow x$;
    $\square$ (2δ time delay elapsed) : $R_p \leftarrow \perp$;
when p leaves A:
  free(R_p);

Communication controlled thread:

upon reception of (REQ) : **if** ($R_p \neq void$) **then RB_send**($REP(R_p)$)
upon reception of ($W(x)$) : $R_p \leftarrow x$

Read and Write operations:

When p is in A:
`read()` : **wait until** ($R_p \neq void$) **then return**(R_p);

When p is in A':
`write`(x) : **RB_send**($W(x)$);

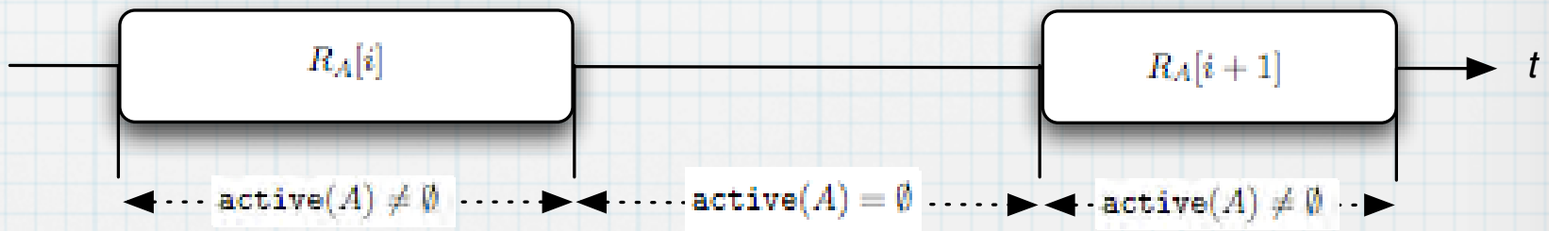
Structure induced by the model:

1 geographic thread

1 comm. thread

Properties...

* Region/core region interest:
  * abstracts away physical parameters (network parameters, speed)
* simple implementation of shared storage
* Semantics:
  * applications that need to store information only when users populate an area

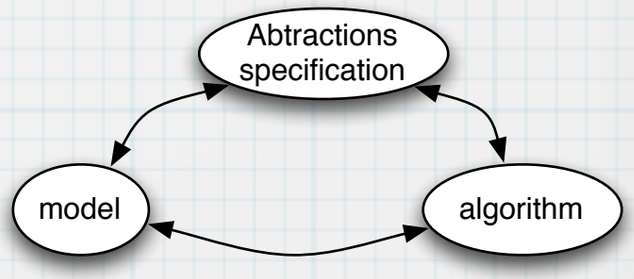


Properties...

* Event-based programming
  * events:
    * application-driven
    * communication/interactions between users
    * movements: interactions with physical world

Extensions

* Current status
  * implementation for one-hop communication model
  * concurrent writers case =? behaviour in presence of failures
* Future work
  * new abstractions/building blocks



Conclusion

* develop building blocks for spatial-based distributed computing
* simple programming
* more resilient applications
* proven building blocks
* new applications ?