Exploiting data in dynamic networks

Bertrand Ducourthial

Université de Technologie de Compiègne
UMR CNRS UTC 7253 Heudiasyc

April 2014
1. Context

2. Distributed data collect

3. Distributed data fusion

4. Conclusion

5. Jobs position
1. Context
   - Dynamic networks
   - Distributed data
   - Team
   - Airplug Software Distribution

2. Distributed data collect

3. Distributed data fusion

4. Conclusion

5. Jobs position
Context: dynamic networks

A definition

• **Dynamic network**
  • Short communication link duration
  • Short amount of data exchanged
  • Unstable neighborhood
  • Do not rely on topology
  • Avoid using any remote knowledge

• **Modeling?**
  • Nodes speed ↔ communication protocol
  • p-Dynamic graphs
  • Metric
• **Applications**
  - Large networks are generally dynamic
  - Social networks
  - Peer-to-peer networks
  - Network of laptops Mobile Ad hoc NETworks

• **Examples**
  - Network of pedestrian with personal devices
  - Network of embedded computers
    - Robots networks
    - Vehicular networks (VANET)
Context : dynamic networks

Challenges

- Impact of the dynamic
  - Communication protocols
  - Distributed algorithms
  - Trusty, security
  - ...

- Distributed data?
• Data spread out in a network

• **Decision making**
  - Data on each node → action decided
  - Using physical devices Alarm...
  - Using another (distributed) algorithm

• Require to know about the data

→ **Data collection**
Context: distributed data

Can we trust data?

Techniques for dealing with system faults
  - A fault could damage data
  - Self-stabilization
  - Redundancy
  - ...

Fault: data \neq legitimate data
  - What is a legitimate data?
  - Data are supposed to be precise and certain

Reality: information tainted with imperfection
  - Imprecision
  - Uncertainty
  - Ambiguity

\sim\ Data fusion
Context: team
UTC/CNRS Heudiasyc

- **Université de Technologie de Compiègne**
  ~4500 students, master degree (engineer diploma), PhD
  [http://www.utc.fr](http://www.utc.fr)
  - one of the first French engineering school for computer science
  - close to Paris and Charles de Gaulle airport

- Heudiasyc Lab. from the UTC & CNRS
  Equipex Robotex, Labex MS2T
• Our point of view:
  Dynamic networks are different!

• Our methodology:
  1. Real applications
  2. Designing new algorithms
  3. Proof of concept
     Performance issues
     Analytical proofs
  4. Road tests
  5. Tests or network emulation
  6. Analytical proofs

• Our tools:
  - **Airplug** Software Distribution
  - Communicating embedded disposals

https://www.hds.utc.fr/airplug
## Context: Team

### Research projects

- **Cooperative Mobility for Services of the Future**  
  European Celtic Plus project  2013-2015

- **Inter-vehicles cooperative perception for road safety**  
  National project  2008-2011

- **Distributed system for vehicle dynamic evaluation**  
  Regional project  2008-2011

- **Data gathering from VANET to infrastructure**  
  Industrial project Orange lab  2008-2010

- **Distributed applications for dynamic networks**  
  Regional project  2007-2010

- **SafeSPOT European IP project**  
  2006-2010

- **Network services for com. between mobiles objects**  
  Industrial project Orange lab  2004-2008

- **Road anticipating**  
  Regional project  2004-2007
Data in dynamic networks
B. Ducourthial

Context:
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

Scientific contributions

- Experiments with sensors [WiSARN 2014]
- I2V experiments [IV 2014]
- V2I experiments [IWCMC 2014]
- V2V unicast communication [WCNC 2014]
- Distributed data fusion [SSS 2012]
- Data collect on the road [IV 2012]
- Performances in a convoy of vehicles [VTC 2011]
- V2I architecture [Mobiwac 2010]
- Distributed dynamic group service [SPAA 2010]
- Vehicular networks emulation [ICCCN 2010]
- Simulation of vehicular networks [VTC 2010]
- Experimenting on the road [VTC 2009]
- Messages forwarding [IEEE TVT 2007]
- ...
Context: team

Communication & intelligent vehicles

Data in dynamic networks
B. Ducourthial

Context
Dyn. network
Distributed data
Team
Airplug
Data collect
Introduction
Algorithm
Experiments
Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments
Conclusion
Jobs

Context: team

Communication & intelligent vehicles
Context: Airplug Software Distribution

Process-based architecture

- POSIX OS
- Core program
  - user-space process
  - networking
- Applications
  - user-space process
  - read on stdin
  - write on stdout
  - API close to IEEE WSMP
- Ensure tasks and OS independence for robustness
- Open to any programming language
**Context** : Airplug Software Distribution

Facilities for developing new protocols

- New protocols developed in user space processes
  - open to new networking solutions
  - cross-layer solutions facilitated

![Diagram of wireless network with Airplug protocols](image-url)

**Data in dynamic networks**

B. Ducourthial

**Context**
- Dyn. network
- Distributed data
- Team
- Airplug

**Data collect**
- Introduction
- Algorithm
- Experiments

**Data fusion**
- Introduction
- Example
- Dist. data fusion
- Neighbor alg.
- Distributed alg.
- Self-stab. alg.
- Experiments

**Conclusion**

**Jobs**
Context: Airplug Software Distribution

• Airplug-term → rapid prototyping
• Airplug-emu → study by emulation
• Airplug-live → real experiments (vehicles, UAV)
• Airplug-ns → adds-on for Network Simulator

remote, notk...
Summary

1. Context

2. Distributed data collect
   Introduction
   Distributed algorithm
   Experiments

3. Distributed data fusion

4. Conclusion

5. Jobs position
Data collect: introduction

Motivations

- Many data produced by vehicles
  - From embedded sensors and calculators
- Could feed intelligent applications
  - infrastructure
  - vehicle-oriented, driver oriented

Problem to solve

- Large amount of data
- Limited network resources
- Dynamic network

Kind of collect

- Data production
  - local, time/geographic aggregation...
- Data sending
  - a single, some, all vehicles...
- Starting
  - Push-based, pull-based...
- Ending?
Data in dynamic networks

B. Ducourthial

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

Fabrication de l'information

collectes passées

passée

instantanée

locale

voisinage

distance k

Collecte de l'information

Envol de l'information

certaines voitures

une voiture

toutes les voitures

une voiture par zone
Data collect : introduction

Related work : a summary

- **Dissemination**
  - Opportunistic, geographic, peer-to-peer, cluster-based... [WU04, LEE06, BON07]
  - Kind of data to be sent?
  - When to send data?

- **Request-based**
  - Propagation of Information with Feedback [SEG83]
    - For fixed networks
  - Wave for MANETs [CHE02]
    - For networks without partitioning
Data collect: introduction

- Fix network
- A single node collects

[Propagation of Information With Feedback, Segall 1983]
smallest value in the network?

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data in dynamic networks

B. Ducourthial

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

Data collect: introduction

Related work: PIF

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

Related work: PIF

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

Related work: PIF

1. Fix network
2. A single node collects

[Propagation of Information With Feedback, Segall 1983]
Data collect: introduction

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect : introduction

Related work : PIF

[Propagation of Information With Feedback, Segall 1983]

• Fix network
• A single node collects
Data collect: introduction

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data in dynamic networks

B. Ducourthial

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

Data collect : introduction

Related work : PIF

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

Related work: PIF

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

Related work: PIF

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

Related work: PIF

- Fix network
- A single node collects

[Propagation of Information With Feedback, Segall 1983]
Data in dynamic networks

B. Ducourthial

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

Data collect : introduction

Related work : PIF

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects

[19]
[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
The smallest value is 2.

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

Related work: PIF

- Fix network
- A single node collects

[Propagation of Information With Feedback, Segall 1983]
Data collect : introduction

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
The smallest value is 2.

[Propogation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects
Data collect: introduction

Related work: PIF

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

---

[Propagation of Information With Feedback, Segall 1983]

- Fix network
- A single node collects

the smallest value is 2
Data collect : introduction

Proposed architecture

- **Start** on some *initiators*
  - Any vehicle
    - Periodically, or on request (local/infrastructure)
  - Service vehicles
  - Road side unit

- **Collect**
  - Data in vehicles up to a given distance
  - Update of dynamic data

- **Termination**
  - Maximal duration
  - Stability of the result

- **Result**
  - Ordered by the distance to the initiator
  - Allow aggregation before exploitation
• **Local view of a node:**
  lists of (node_id, local_data) ordered by the distance to the node
• **Local view of a node:**
  lists of (node_id, local_data) ordered by the distance to the node

\{(a,4)\}, \{(b,8), (c,2)\}, \{(d,5), (f,21)\}, \{(e,12), (h,14), (g,17)\}
**Local view of a node:**

lists of (node_id, local_data) ordered by the distance to the node

\{(a,4), (b,8), (c,2)\}
Local view of a node:
lists of (node_id, local_data) ordered by the distance to the node

{(a,4)}, {(b,8), (c,2)}, {(d,5), (f,21)}
Local view of a node:
lists of (node_id, local_data) ordered by the distance to the node

\{(a,4), (b,8), (c,2), (d,5), (f,21), (e,12), (h,14), (g,17)\}
Data in dynamic networks

B. Ducourthial

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

---

Data collect: distributed algorithm

Local view: operator ant

• Views \( \mathcal{V}_1 \) and \( \mathcal{V}_2 \):

\[
\mathcal{V}_1 = \{(a, 4)\}, \{(b, 8)\}, \{(d, 5), (f, 21)\}
\]

\[
\mathcal{V}_2 = \{(c, 2)\}, \{(d, 5)\}, \{(b, 8), (e, 12)\}
\]

• Shifting

\[
\mathcal{V}_1 = \{(a, 4)\}, \{(b, 8)\}, \{(d, 5), (f, 21)\}
\]

\[
r(\mathcal{V}_2) = \{\}, \{(c, 2)\}, \{(d, 5)\}, \{(b, 8), (e, 12)\}
\]

• Merging

\[
\mathcal{V}_1 \oplus r(\mathcal{V}_2) = \{(a, 4)\}, \{(b, 8), (c, 2)\}, \{(d, 5), (f, 21)\}, \{(b, 8), (e, 12)\}
\]

\[
\mathcal{V}_1 \oplus r(\mathcal{V}_2) = \{(a, 4)\}, \{(b, 8), (c, 2)\}, \{(d, 5), (f, 21)\}, \{(e, 12)\}
\]

• \( r \)-operator ant:

\[
ant(\mathcal{V}_1, \mathcal{V}_2) = \mathcal{V}_1 \oplus r(\mathcal{V}_2)
\]

\( \leadsto \) self-stabilizing distributed algorithm
Data collect : distributed algorithm

Local view : example

\[ \mathcal{V}_1 \oplus r(\mathcal{V}_2) = \{(a, 4)\}, \{(b, 8), (c, 2)\}, \{(d, 5), (f, 21)\}, \{(e, 12)\} \]
$\mathcal{V}_1 \oplus (\mathcal{V}_2) = \{(a, 4), (b, 8), (c, 2), (d, 5), (f, 21), (e, 12)\}$
\[ V_1 \oplus r(V_2) = \{(a, 4)\}, \{(b, 8), (c, 2)\}, \{(d, 5), (f, 21)\}, \{(e, 12)\} \]
Data collect: distributed algorithm

\[ \mathcal{V}_1 \oplus r(\mathcal{V}_2) = \{(a, 4), (b, 8), (c, 2), (d, 5), (f, 21), (e, 12)\} \]

\{(a, 4), (b, 8), (c, 2), (d, 5), (f, 21), (e, 12)\}

\{(c, 2), (d, 5), (b, 8), (e, 12)\}
Algorithm for message reception

```java
receive(parameters, view)

if no current collect then
    Reset variables; store the parameters
    Set the lifetime of the sender to maxloss
    Store the view of the sender
    Set the timer
else if message for current collect
    Set the lifetime of the sender to maxloss
    Store the view of the sender
else
    Drop the message
end if
```
Algorithm for timer expiration
Decrement the lifetime of each known neighbor
Reset any data of neighbors with lifetime=0
Update local_view with local data
for each view previously stored do
    local_view ← ant(local_view, view)
end for
Truncate local_view to maxdst first elements
if local termination is false then
    set the timer
    send(parameters, local_view)
end if
Algorithm for Local termination detection

\[
\text{if initiator } \notin \text{ local\_view then return true}
\]

\[
\text{count\_dur } \leftarrow \text{ count\_dur } + 1
\]

\[
\text{if count\_dur } == \text{ maxdur then return true}
\]

\[
\text{if old\_local\_view } \equiv \text{ local\_view then}
\]

\[
\text{count\_stb } \leftarrow \text{ count\_stb } + 1
\]

\[
\text{else}
\]

\[
\text{count\_stb } \leftarrow 0
\]

\[
\text{end if}
\]

\[
\text{if count\_stb } == \text{ maxstb then return true}
\]

\[
\text{return false}
\]
Data collect : experiments

Using the Airplug Software Distribution

- **Proof of concept on the road**
  - 5 vehicles with Dell mini-9, WiFi devices and roof antenna
  - Ubuntu 8.04, Airplug, GPS and COL programs embedded
  - [see movie on-line http://www.hds.utc.fr/airplug]

- **Performance evaluation by emulations**
  - 13 vehicles, series of 50 experiments
  - Variations of the timer duration, the links robustness and the life duration of a neighbor

- **Demonstration**
Data collect: experiments

Road experiment replay

Click on the image for loading the video (in the web browser)

[https://www.hds.utc.fr/airplug]
Data collect: experiments
Percentage of received data versus Link reliability
Data collect: experiments

Percentage of received data versus timer duration

![Graph showing the percentage of collected data (%) versus duration (ms) for different reliability values.](image.png)
• **Qualitative result**
  • Success of the proof of concept
  • Support the network dynamic
    Including network partitioning

• **Quantitative results**
  • $\uparrow$ link reliability $\Rightarrow$ % collected data $\uparrow$
  • Small influence of the timer duration

• **Auto-adaptation of maxd$st$**
  Maximal distance

• **In highly dynamic network**:
  • Increasing maxloss Neighbor lifetime
    To the detriment of up-to-date local view
  • Decreasing the timer duration
    To the detriment of bandwidth

• Towards self-adaptation
1. Context

2. Distributed data collect

3. Distributed data fusion
   - Introduction
   - Data fusion example
   - Distributed data fusion
   - Neighborhood confidence algorithm
   - Distributed confidence algorithm
   - Self-stab. alg.
   - Experiments

4. Conclusion

5. Jobs position
Distributed data fusion

Introduction: data?

• How to deal with imprecise and uncertain data?
  • Imprecision:
    Set Membership Approach uncertainty?
  • Aleatory uncertainty:
    Probability theory imprecision?
  • Theory of Belief Function: generalizes both Transferable Belief Model Dempster-Shafer Theory of Evidence

• Belief Function Framework
• Information modeling + combination rules

[Dempster 1968, Shafer 1976, Smets 1990s]
Data in dynamic networks

B. Ducourthial

Introduction: data representation

• Data $X$ with value in $\Omega$

• Item of information about $X$
  • (value, confidence)
  • value: subset of $\Omega$
  • confidence: indication on the reliability of the item of information

• Interest:
  • Imprecision of $X \rightsquigarrow$ value
  • Uncertainty of $X \rightsquigarrow$ confidence

[Dubois, Prade 1988]

<table>
<thead>
<tr>
<th>Value</th>
<th>certain</th>
<th>uncertain</th>
</tr>
</thead>
<tbody>
<tr>
<td>precise</td>
<td>20</td>
<td>probably 20</td>
</tr>
<tr>
<td>imprecise</td>
<td>between 15 and 25</td>
<td>probably between 15 and 25</td>
</tr>
</tbody>
</table>
• Frame of discernment $\Omega$ finite or infinite

• Basic belief assignment (bba)
  • **Mass function**
    • $m^\Omega : \mathcal{P}(\Omega) \to [0, 1]$
    • $\sum_{X \subseteq \Omega} m^\Omega(X) = 1$
  • Other representations: commonalities, weights

• **Conjunctive operator**
  • Combines two mass functions by emphasizing the agreement, providing they are reliable and independent [Smets 1990, Shafer 1976]
  • $m_1 \otimes_2 (A) = \sum_{B \cap C = A} m_1(B) \cdot m_2(C)$
  • Conflict is the mass obtained on $\emptyset \subseteq \Omega$

• **Dempster operator**
  • Conflict ignored
  • Spread over other sets

• Other operators: disjunctive, cautious...
Distributed data fusion

Data fusion example: weather forecast (1)

- Pressure measurement

- Weather forecast
  - Compare current measure with the last one
Distributed data fusion
Data fusion example: weather forecast (2)

- Barometer?

- Measure:
  - Pressure measurement: interval \( I \subset \mathbb{R}^+ \)
  - Pressure gradient: interval \( \Delta I \subset \mathbb{R} \)
  - Simple mass function:
    - Only two subsets: \( \Delta I \) and \( \mathbb{R} \)
    - \( \mathbb{R} \): lack of knowledge
    - \( m_{\mathbb{R}}(\Delta I) = 1 - \alpha \)
    - \( m_{\mathbb{R}}(\mathbb{R}) = \alpha \)
    - \( \alpha \): reliability of the barometer
Distributed data fusion

Data fusion example: weather forecast (3)

- Coarsening:
  - Finite frame of discernment instead of intervals of $\mathbb{R}$
  - $\Omega = \{\text{wet, cloud, sun}\}$
  - Mass function:

\[
egin{array}{cccccc}
\Delta I << 0 & \Delta I < 0 & \Delta I \approx 0 & \Delta I > 0 & \Delta I >> 0 \\
\{\text{wet}\} & \{\text{wet, cloud}\} & \{\text{wet, cloud, sun}\} & \{\text{cloud, sun}\} & \{\text{sun}\}
\end{array}
\]

- Several independent measures can be combined using the Dempster rule.
- Decision: from mass to *pignistic* probability

\[
P(A) = \sum_{\emptyset \neq B \subset \Omega} m(B) \frac{|A \cap B|}{|B|}
\]
Motivation

- Problem to solve
  - **Direct confidence** (regularly) produced locally
  - Using an external mean
  - **Node’s confidence** computed using other values
- Avoiding data collection and centralized approach
- Locality
  - One result per node
  - Depend on the position of the node in the network

\[ \text{Distributed approach for data fusion} \]
Distributed data fusion

Related work

- Centralized approach
  - Geographic distance between sources of information
  - Age of information
- Distributed approach
  - Spanning tree
  - Vehicular networks
    - Spanning tree?
    - Loops \( \sim \) data incest
    - Idempotent combination rule
    - \( \sim \) Cautious operator
      - Defined on weights functions
- Network always supposed to be reliable

Centralized approach

- Geographic distance between sources of information
- Age of information

Distributed approach

- Spanning tree
- Vehicular networks
  - Spanning tree?
  - Loops \( \sim \) data incest
  - Idempotent combination rule
  - \( \sim \) Cautious operator
    - Defined on weights functions

Network always supposed to be reliable
**Distributed data fusion**

**Contribution**

- **Problem to solve:**
  - **Direct confidence** (regularly) produced locally
    - Using an external mean
  - **Node’s confidence** computed using other values

- **Neighborhood confidence algorithm**
  - Combine the direct confidences in the neighborhood

- **Distributed confidence algorithm**
  - Combine all direct confidences of the system
  - Discount information regarding the distance
    - Confidence decreases at each hop

- **Properties**
  - Finite data set
    - Discretization + adapted operators
  - Asynchronous and anonymous system
  - Unreliable message passing system
  - Intermittent faults on memories/messages
  - Crash faults on nodes
• Result on any node $v$ depends on its neighbors only.
• Result on any node \( v \) depends on its neighbors only.
- Result on any node \( v \) depends on its neighbors only.
Upon (local) timer expiration

\[ \text{PRIV}_v \leftarrow \text{current direct confidence} \]

\[ \text{OUT}_v \leftarrow \text{PRIV}_v \]

for each entry \( u \) in \( \text{IN}_v \) do

\[ \text{OUT}_v \leftarrow \text{OUT}_v \uplus \text{IN}_v[u] \]

end for

\textbf{push( PRIV}_v \text{ )}

Reset \( \text{IN}_v \)

Restart the timer
• **Legitimate configuration**
  - combination of the direct confidence with those of the neighbors
  - ⨂ : discretization of Dempster operator
  - Commutative

\[
\forall v \in \mathcal{S}, \quad \text{OUT}_v(c) = \bigoplus_{u \in \Gamma_0 v} \text{PRIV}_u(c)
\]

• **Result**

*Convergence in finite time to a legitimate configuration after the last occurrence of a transient fault and the last modification of either the topology or the direct confidences (inputs).*
**Distributed data fusion**

**Definition**

- Result on any node \( v \) now depends on all other nodes, not only its neighbors.
Upon (local) timer expiration

\[
\text{PRIV}_v \leftarrow \text{current direct confidence} \\
\text{OUT}_v \leftarrow \text{PRIV}_v \\
\text{for each} \ v \ \text{entry} \ u \ \text{in} \ \text{IN}_v \ \text{do} \\
\quad \text{OUT}_v \leftarrow \text{OUT}_v \land \text{IN}_v[u] \\
\text{end for} \\
\text{push( OUT}_v ) \\
\text{Reset IN}_v \\
\text{Restart the timer}
\]
Distributed data fusion

Result

- **Legitimate configuration**
  - combination of the direct confidence with those of all the nodes in the network
  - $\bigoplus$: cautious operator defined on weights
    Associative, commutative, idempotent

\[
\forall v \in S, \quad \text{OUT}_v(c) = \bigoplus_{u \in \Gamma_v} \text{PRIV}_u(c)
\]

- **Result**

  *Stabilization in a fixed topology starting from an initial configuration where all memories have been reset, assuming the direct confidences (inputs) stabilizes.*
Distributed data fusion
Self-stabilizing algorithm : motivation

- **The previous algorithm**
  - Give one result per distributed component
  - Does not support erroneous messages
  - See demonstration

- **What happens in large network?**
  - Convergence time?
  - Interpretation of the result?

- **What happens in case of erroneous message?**
  - Introduced accidentally
  - Introduced intentionally
  - Due to the dynamic
Upon (local) timer expiration

\[ \text{PRIV}_v \leftarrow \text{current direct confidence} \]
\[ \text{OUT}_v \leftarrow \text{PRIV}_v \]

\textbf{for each} entry \( u \) in \( \text{IN}_v \) \textbf{do}

\[ \text{OUT}_v \leftarrow \text{OUT}_v \odot r(\text{IN}_v[u]) \]

\textbf{end for}

\textbf{push( OUT}_v )

Reset \( \text{IN}_v \)

Restart the timer
Distributed data fusion

Self-stabilizing algorithm: discounting

\[
\text{OUT}_v \leftarrow \text{OUT}_v \ominus r(\text{IN}_v[u])
\]

- \textbf{r} is called a discounting
- It decreases the information in a given bba
  Basic belief assignment
- It is application-dependent

- **Condition 1**: endomorphism
  \[ r(w_1 \ominus w_2) = r(w_1) \ominus r(w_2) \]

- **Condition 2**: expansion
  \[ w \prec \ominus r(w) \]
The cautious operator along with the discounting $r$ defines an r-operator which ensures the self-stabilization of the algorithm.

[SSS 2005, SSS 2007]

- $\ominus$: cautious operator defined on weights
- $r$: discounting function
- $\ominus$: r-operator defined by $x \ominus y = x \ominus r(y)$
• **Legitimate configuration**

  • combination of the direct confidence with those of all the nodes in the network, discounted as many time as their distance.
  • $\bigodot : r$-operator defined using $\bigotimes$ and $r$

\[
\forall v \in S, \quad \text{OUT}_v(c) = \bigodot_{u \in \Gamma_v} \text{PRIV}_u(c) \otimes_{u \in \Gamma_v} r^{\text{dist}(u,v)}(\text{PRIV}_u(c))
\]

• **Result**

    *Stabilizes in finite time to a legitimate configuration after the last occurrence of a transient fault and the last modification of either the topology or the direct confidences (inputs).*
• **Stabilization time**  supposing a synchronous system
  • $O(k + D)$
  • $k$ defined by $r^k$ (smallest value) = largest value
  • $D$ : diameter of the stabilized topology
  • Previous example : $k = 10$

• **Message size**
  • Depends on coarsening $\sim |\Omega|$
    Previous example : $|\Omega| = 3$
  • and the discretization
    Example : $(0, 1]$ discretized up to the thousandth
  • In the example : 60 bits per message
**Testbed**

- 3 RSU, 6 sensors + vehicles

---

**Proof of concept**
on-line demonstration using Airplug-emu
Context

1. Context

2. Distributed data collect

3. Distributed data fusion

4. Conclusion

5. Jobs position
Data in dynamic networks

B. Ducourthial

Context
Dyn. network
Distributed data
Team
Airplug

Data collect
Introduction
Algorithm
Experiments

Data fusion
Introduction
Example
Dist. data fusion
Neighbor alg.
Distributed alg.
Self-stab. alg.
Experiments

Conclusion

Jobs

How to exploit data in dynamic networks?

- Distributed data collect
  Self-stabilizing algorithm
- Distributed data fusion
  Self-stabilizing algorithm

Next steps

- Modeling
- Proofs of usability in dynamic networks
- Adaptivity
- Large testbed

Vehicle equipped with a device not connected to its CAN bus
Vehicle equipped with a device connected to its CAN bus
Road-Side Unit
Data on an infrastructure server

Internet
1. Context

2. Distributed data collect

3. Distributed data fusion

4. Conclusion

5. Jobs position
• **Open postdoc position**
  • Distributed algorithms and protocols
  • One year

• **Open engineer position**
  • Development and experiments
  • European project
  • Two years

• **Heudiasyc Lab. UMR CNRS UTC 7253**
  Equipe Ex Robotex
  Labex MS2T
  [https://www.hds.utc.fr](https://www.hds.utc.fr)