Vehicular network emulation

A. Buisset, B. Ducourthial, F. El Ali, S. Khalfallah

Bertrand.Ducourthial AT utc.fr

Laboratoire Heudiasyc (UMR UTC-CNRS 6599)
Université de Technologie de Compiègne
France

19th International Conference on Computer Communications and Networks (ICCCCN 2010)
2-5 August 2010, Zurich, Switzerland
1. Context

2. Airplug software suit

3. Airplug-emu

4. Performances

5. Conclusion
1 **Context**

Intelligent Transport Systems

Scientific issues

Contribution

Team

2 **Airplug software suit**

3 **Airplug-emu**

4 **Performances**

5 **Conclusion**
Intelligent Transport Systems

• ITS motivations
  • Improving transportation in terms of safety, mobility, productivity, environmental impact...
  • main goals: reduce road fatalities, improve infrastructure management, offer new on-board services

• ITS applications
  • Infrastructure oriented applications for optimizing the infrastructure management (transit, freeway, freight, emergency organization...)
  • Vehicle oriented applications for increasing the road safety (crash prevention, alerts, visibility distance...)
  • Driver oriented applications for improving the road usage (traffic jam, road work information, payment...)
  • Passenger oriented applications for offering new services on board (Internet access, distributed games, tourist info...)
Scientific issues
Highly dynamic ad hoc networks

• Next step in networking and distributed alg.

- Impact of the dynamic
  - impact on network layers
    link (2), network (3), transport (4)
  - impact on distributed algorithms
    fault tolerance, data sharing...
  - impact on trusty and security
    who believe? what information is reliable?
  - algorithms necessary embedded
    context-aware optimization, adaption...

~> Strong problems, new solutions expected
How to validate new ideas

- **By proofs:**
  - for distributed algorithms
  - require communication and synchronization models
  - + exact result; - models far from reality

- **By simulations:**
  - for networking protocols
  - require propagation and MAC model, packets traffic and node mobility model
  - + scalable; - models far from reality

- **By experiments:**
  - for proof of concept and performance measuring in situ
  - require equipments and logistic
  - + exact results; - not scalable, not reproducible

↔ Need for a complementary approach
Contributions

- **Emulation**: powerful tool for vehicular networks
  - between theory and practice, road experiments and simulation
  - parts are real: applications, protocols, mobility, traffic
  - parts are artificially reproduced: layers 1 and 2

- **How to efficiently reproduce layers 1 and 2?**
  - **Airplug-emu**
  - can do hybrid emulation (including real links)

- **The Airplug software suite**
  a complete environment for studying dynamic networks
Team

- 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
  Automatic, Computer Science, Networking, Knowledge...

- Vehicular networks team
- Intelligent vehicles team
  several equipped cars
Some of the team projects

- **Road anticipating**  
  Regional grant DIVA, Heudiasyc - CREA  
  2004-2007

- **Network services for com. between mobiles objects**  
  Industrial grant FTR&D  
  2004-2008

- **Co-operative Systems for Road Safety**  
  "Smart Vehicles on Smart Roads"  
  IP SafeSPOT, 6th PCRD / IST / eSafety  
  2006-2010

- **Distributed applications for dynamic networks**  
  Regional grant Heudiasyc - LaRIA  
  2007-2010

- **Data gathering from VANET to infrastructure**  
  Industrial grant FTR&D  
  2008-2010

- **Distributed system for vehicle dynamic evaluation**  
  Regional grant Heudiasyc - MIS  
  2008-2011

- **Inter-vehicles cooperative perception for road safety**  
  National project ANR JC, (Heudiasyc)  
  2008-2011
Some of the team contributions

- Distributed dynamic group service [SPAA 2010]
- V2I architecture [MobiWac 2010]
- Simulation of vehicular networks [VTC 2010]
- Road experiments [VTC 2009]
- Messages forwarding [IEEE TVT 2007]
  conditional transmissions
- IEEE 802.11 fairness [MedHocNet 2006]
- Capacity of vehicular networks [VTC 2005]
Summary

1. Context
2. Airplug software suit
   Protocols design
   Software suit
   Platform
   API
3. Airplug-emu
4. Performances
5. Conclusion
Airplug architecture

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
Airplug architecture
Facilities for developing new protocols

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

![Airplug architecture diagram]

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

TCP/UDP
IP
AIRPLUG
TCP/IP
over
802.11
UDP/IP
over
802.11
VANET
protocol
over
802.11
VANET
protocol
over
802.15
GPS TST HOP
802.15
802.11
802.11
wireless network
Airplug software suite

Applications for infrastructure, vehicles, drivers, passengers...

http://www.hds.utc.fr/~ducourth/airplug
Complete research platform

- **On the road**: airplug-road
  - in Compiègne, France
  - in Michelin circuit, France
  - test-bed with 6 cars with France Telecom R&D
  - test-bed with 7 cars with France Telecom R&D

[see movies on-line](http://www.hds.utc.fr/~ducourth/airplug)

Complete research platform

- **On the road**: airplug-road
  - in Compiègne, France
  - in Michelin circuit, France
  - test-bed with 6 cars with France Telecom R&D
  - test-bed with 7 cars with France Telecom R&D

*see movies on-line*

http://www.hds.utc.fr/~ducourth/airplug
Complete research platform

- **On the road**: airplug-road
  - **In the laboratory**: airplug-lab
    - GPS position replaying
    - new trajectories derived \(\leadsto\) convoys
    - out of range messages filtered (soon)

Context
- ITS
- Scientific issues
- Contribution
- Team

Airplug
- Protocols design
- Software suit
- Platform
- API

Airplug-emu
- Communications
- Node
- Mobility
- Scenarios
- Core program

Performances
- Inputs
- Reproducing road testbeds

Conclusion
Complete research platform

- On the road: airplug-road [VTC 2009]
- In the laboratory: airplug-lab [ICCCN 2010]
  - In a computer: airplug-emu
    - using shell facilities
    - emulation of vehicular networks

**Team**
- Airplug
- Protocols design
- Software suit
- Platform
- API

**Airplug-emu**
- Communications
- Node
- Mobility
- Scenarios
- Core program

**Performances**
- Inputs
- Reproducing road testbeds

**Conclusion**
Complete research platform

- On the road: **airplug-road**  [VTC 2009]
- In the laboratory: **airplug-lab**
- In a computer: **airplug-emu**  [ICCCN 2010]
- Remotely: **airplug-rmt**
  - a specific application controls remote access from external applications
  - portability of the applications
  - transparent usage stand-alone / remotely / locally
  - heterogeneous vehicular networks emulation
Complete research platform

- On the road: airplug-road [VTC 2009]
- In the laboratory: airplug-lab
- In a computer: airplug-emu [ICCCN 2010]
- Remotely: airplug-rmt
  - In Network Simulator: airplug-ns [VTC 2010]
Complete research platform

- **On the road**: airplug-road [VTC 2009]
- **In the laboratory**: airplug-lab
- **In a computer**: airplug-emu [ICCCN 2010]
- **Remotely**: airplug-rmt
- **In Network Simulator**: airplug-ns [VTC 2010]

- In all these usages, the same codes are used

```
  VIS  ALT  WTR  CTD  MSG  JEUX
TST  FTP  IMG  CNV  DIP  PTH  MVS
  GPS  NBH  IO   HOP  NTL  TNL
  AIRPLUG  CTL
```

```
  VIS  ALT  WTR  CTD  MSG  JEUX
TST  FTP  IMG  CNV  DIP  PTH  MVS
  GPS  NBH  IO   HOP  NTL  TNL
  AIRPLUG  CTL
```

- In all these usages, the same codes are used.
Airplug architecture
Application Programming Interface

- **Addressing for dynamic networks**  [WINITS 2007]
  - **area** : LCH, AIR, ALL
  - **applications** :
    - a given application
    - all those that subscribed to the sender app
  - **Note** : similarities with
    - IEEE WAVE Short Messages Protocol
    - messages-oriented frameworks (eg. JMS)

- **Three types of communication**
  - to simplify the development
    - **what**, **whatwho**, **whatwhowhere**
  - automatic guessing or safe mode

- **Libraries for easy developments**
  - eg. message formatting

- **Hierarchical makefile for easy installation**
1. Context

2. Airplug software suit

3. Airplug-emu
   - Communications
   - Node
   - Mobility
   - Scenarios
   - Core program

4. Performances

5. Conclusion
Communications emulation

- Airplug applications rely on standard IO for communication \(\rightarrow\) can be managed by the shell

- Examples
  - Unidirectional communication
    \[
    ./gps \mid ./pro \mid ./pro
    \]
  - Bidirectional link between two applications
    \[
    mkfifo link1 link2 \\
    ./pro < link1 > link2 \\
    ./pro < link2 > link1
    \]
  - Several neighbors
    \[
    tee link2 link3 < link1
    \]
  - Moving nodes:
    \[
    \text{kill} \quad \text{-STOP} \quad \text{pid} \\
    \text{create new connections} \\
    \text{kill} \quad \text{-CONT} \quad \text{pid} \\
    \text{kill} \quad \text{-KILL} \quad \text{old_connections}
    \]

- Advantages: simple, powerful, robust
- **RCP**: receives messages from neighbor nodes, can delay or loss some of them
- **TST and HOP**: two Airplug-compatible applications on this node
- **DIR**: analyzes message header to forward messages either locally or to neighbor nodes
- **GTW**: frozen when the connections are updated
- **Airplug GPS application**:  
  - can decode NMEA frames sent by the GPS device *(on the road)*  
  - can forward positions to applications willing them  
  - can store and replay positions  
  - can modify them to create new realistic ones

- **Airplug-emu** can use:  
  - GPS logs  
  - ns-2 traces  
  - fixed positions  
  - other mobility generators outputs *(customizable)*
- Scenarios are described using an XML configuration file:

```xml
<map width="2500" height="2500">
  <node id="vehicle_1">
    <app name="TST" zone="LCH" exe="/tst.tk" />
    <app name="HOP" zone="AIR" exe="/hop.tk" />
    <move type="gpsfile" path="../data/log.gps" delay="0"/>
  </node>
  <node id="vehicle_2">
    <app name="TST" zone="LCH" exe="/tst.tk" />
    <app name="HOP" zone="AIR" exe="/hop.tk" />
    <move type="gpsfile" path="../data/log.gps" delay="10"/>
  </node>
  <node id="vehicle_3">
    <app name="TST" zone="LCH" exe="/tst.tk" />
    <app name="HOP" zone="AIR" exe="/hop.tk" />
    <move type="gpsfile" path="../data/log.gps" delay="20"/>
  </node>
</map>
```
Airplug-emu core program

- **Init**: read the XML file, launch the processes applications and protocols run in independent processes as on the road

- **Loop**: with a user-defined frequency,
  - read new positions
  - compute the links
    - using range and hazard to avoid perfect disk
    - positions remain in lists ordered by $x$ and $y$
    - complexity generally less than $O(n^2)$
  - update the shell links

- **Messages**: forwarded by the shell can be lost or delayed, allows to mimic the road

- **Hybrid emulation**: remote nodes connected by WiFi

- **Output**:
  - real-time animation
  - shell-script to reproduce all the emulation using only the shell
1. Context

2. Airplug software suit

3. Airplug-emu

4. Performances
   - Inputs vs. results
   - Reproducing road testbeds

5. Conclusion
• Delays and loss rates measured on the road → Airplug-emu → results

• Accuracy depends on inter-packet gap (IPG)
  Left : exact values for each test. Right : mean values
Reproducing road testbeds

- Left: convoy of 7 stopped vehicles.
  Right: moving convoy of 5 vehicles

- Average inputs only for Airplug-emu and ns-2
Summary

B. Ducourthial

Context

ITS
Scientific issues
Contribution
Team

Airplug
Protocols design
Software suit
Platform
API

Airplug-emu
Communications
Node
Mobility
Scenarios
Core program

Performances
Inputs
Reproducing
road testbeds

Conclusion
Conclusion 1: emulation of vehicular networks

- Strong motivations for ITS
  - Intelligent Transport Systems

- Lot of applications imagined and studied
  - interesting scientific challenges
    - link / network / transport layers
    - distributed algorithms
    - security

- How to validate?
  - proofs...
  - simulations...
  - real testbeds...

- Place for network emulation!
  - Airplug-emu: simple and robust architecture
  - Good performances
Conclusion 2: Airplug-emu advantages

- **Designing new protocols**
  - facilities using interface and on-line parameters range, link robustness, network dynamic...
  - easy scenarios creation
    - mobility from GPS, ns-2 or others

- **Protocols study**
  - easy to reproduce experiments with new sets of parameters
  - tuning, performance evaluation

- **Accuracy of the results**
  - depends on inputs from the testbed
  - mean measures only \( \sim \) very good approximation for IPG larger than 100 ms

- **Protocol deployment**
  - in network simulator ns-2
    - with very few transformation if written in Tcl/Tk
  - on the road
Conclusion 3: complete platform

- **The Airplug Software Suite**
  a platform for studying dynamic networks
  - on the road Airplug-road
  - on the lab Airplug-lab
  - on the emulator Airplug-emu
  - on Network Simulator Airplug-ns
  - + remote access

- Used by research teams and for teaching
- Large set of applications developed and tested
- Easy to design new protocol or application
- Available on demand

[http://www.hds.utc.fr/~ducourth/airplug]