

Rethinking the Overhead of Geo-localization Information for Vehicular Communications

Jérôme Härrri, Fethi Filali and Christian Bonnet

Institut Eurécom*

Department of Mobile Communications

B.P. 193, 06904 Sophia-Antipolis, France

{Jerome.Haerri,Fethi.Filali,Christian.Bonnet}@eurecom.fr

Abstract—Geo-localization information is a key component for providing location-based services for Intelligent Transportation Systems (ITS). Generally obtained by GPS devices and transmitted by Wireless Vehicular Communications, geo-localization in ITS represents a promising approach to reach objectives such as an increased road safety, transport efficiency, or on-the-road services. Despite its popularity, the issue of the overhead generated by the transmission of geo-localization data has not been addressed by the ITS community.

In this paper, we first discuss the format of the information provided by GPS devices and propose a flexible message structure for exchanging geo-localization data. Then, we illustrate the significant overhead generated by the transmission of such information and propose a compression method achieving up to 70% overhead reduction without loss of precision. We finally test our method on the OLSR routing protocol.

Index Terms—Geo-localization, overhead, compression, format, GPS, VANET.

I. INTRODUCTION

Intelligent Transport Systems (ITS) have been developed in order to improve the safety, security and efficiency of transportation systems. Typically, vehicles establish communications with other vehicles, roadside infrastructures or even pedestrians, and use GPS data as a mean to provide location-based services.

Due to the recent outbreak of this field, no common agreement has been reached neither on the transmission format nor on the content to be transmitted. Moreover, the cost of transmitting GPS data has also been widely ignored by the ITS community.

In this paper, we describe a message format for geo-localization data transmission, and then address the

overhead that will be generated by its use in vehicular communications. We thereafter propose a compression method reaching up to 70% overhead reduction at no precision loss. By using this message format and the proposed geo-localization data compression mechanism, we could improve vehicular communication by enhancing its interoperability and reducing the geo-localization overhead. This solution has also been proposed to the IETF [1] for a possible standardization.

The rest of this paper is organized as follows. Section II describes the various available data formats, and Section III introduces a common packet format for the transmission of GPS data. In Section IV, we illustrate the overhead generated by the use of geo-localization data. Finally, Section V describes an efficient solution to reduce this overhead by compressing the geo-localization information, and Section VI concludes this work.

II. GEO-LOCALIZATION DATA FORMAT

Due to the early simulation stage of location-based research, the community mostly uses Cartesian coordinates to represent a node's location. In deployment, it is envisioned to directly use the coordinates provided by a GPS-like system (and A-GPS for indoor), whose benefits are twofold. First, it provides a standard reference coordinate system, and second, it ensures a global synchronization based on the atomic GPS clock.

A. GPS Data Format

According to the GPS standard, 3D positioning provides the coordinates of a GPS device in a 3-axis referential, whose origin is the gravity center of the GPS satellite constellation. Then, the GPS terminal converts this raw data into exploitable *longitude*, *latitude*, and *elevation* in the World Geodetic System 84 (WGS84) [2], thus providing a worldwide navigational system. The provided data format is as follow:

*Institut Eurécom's research is partially supported by its industrial members: BMW Group Research & Technology - BMW Group Company, Bouygues Télécom, Cisco Systems, France Télécom, Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales.

- **longitude**– describes the location of a place on Earth *east* or *west* of the Greenwich meridian. A longitude is expressed in sexagesimal notation as $23^{\circ} 27' 30'' E$. An alternate representation is a decimal representation of the minutes and degrees 23.45833° , where the East/West suffix is replaced by a negative sign for coordinates west of the Greenwich Meridian. Accordingly, a longitude may be represented by a signed floating point ranging in $[+180^{\circ}, -180^{\circ}]$, usually with a 6 digits precision.
- **latitude**– describes the location of a place on Earth *north* or *south* of the *Equator*. Similarly to the longitude, a latitude is expressed in sexagesimal notation as $13^{\circ} 19' 43'' N$, with an alternate decimal representation 13.32861° , where the North/South suffix is replace by a negative sign for coordinates south of the Equator. Accordingly, a latitude may be represented by a signed floating point ranging in $[+90^{\circ}, -90^{\circ}]$, usually with a 6 digits precision.
- **elevation**– describes the altitude of a place on Earth relative to the WGS-84 ellipsoid. The elevation is therefore expressed by a signed integer ranging from $8000m$ (Mount Everest) to $-11000m$ (Mariana Trench).

The common point of the tree coordinates is they are usually represented by 32 bits each. Accordingly, each geo-localization is usually represented by 96 bits or 12 bytes.

B. GPS Time Representation

In order to precisely determine the position of a GPS device, its internal clock must be synchronized with the satellites atomic clocks. The GPS system therefore provides a global synchronization mean to any application connected to a GPS device.

GPS time is expressed as a number of seconds since the beginning of the GPS epoch on Sunday January 6th 1980 at 0:00 UTC. Initially represented by a 32 bit integer, this value has been increased to a 64 bit long integer at the end of the last century. Accordingly, the transmission of the time in a packet requires 64 bits or 8 bytes.

III. A COMMON GEO-LOCALIZATION MESSAGE FORMAT

This section defines the content and the structure of a mobility message containing a configurable set of geo-localization or mobility information.

All `<mobility>` messages are conformed to the following specification:

`<mobility> = <value-semantic><value>`

`<value-semantic>` is an 8 bit field describing the structure of the `<mobility>` tag.

- *bit 0 (position bit)*: Messages with this bit cleared ('0') do not contain the position of the node. Messages with this bit set ('1') contain position information.
- *bit 1 (velocity bit)*: Messages with this bit cleared ('0') do not contain the velocity of the node. Messages with this bit set ('1') contain the velocity.
- *bit 2 (azimuth bit)*: Messages with this bit cleared ('0') do not contain the azimuth of the node. Messages with this bit set ('1') contain the azimuth.
- *bit 3 (stability bit)*: Messages with this bit cleared ('0') do not contain the stability of the node. Messages with this bit set ('1') contain the stability.
- *bit 4 (Cartesian bit)*: Messages with this bit set ('1') contain Cartesian coordinates instead of GPS's. This bit is used for compatibility between simulation and deployment message formats.
- *bits 5-7 are RESERVED.*

`<value>` is a field containing the mobility parameters. The length of this field may be obtained from the `<value-semantic>` field.

`<value> = <pos><azi><velo>
<stab><time>`

and where

- `<pos>` is a 48 bit field containing the coordinates of a node following the general layout `<pos> = <Longitude><Latitude><Elevation>`, and where each tag is compressed according to Section V-A.
- `<velo>` is an 8 bit field compressed according to Section V-B, and containing the node's velocity in [m/s]. If the Cartesian bit is set ('1'), `<velo>` is instead a 48 bit field containing the Cartesian projection of the velocity following the general layout `<velo> = <dx><dy><dz>`.
- `<azi>` is an 8 bit field compressed according to Section V-B, and containing the node's azimuth in degree
- `<stab>` is an 8 bit field compressed according to Section V-B, and containing the node's stability. It represents the node's eagerness to keep the current mobility parameters.
- `<time>` is a 16 bit field compressed according to Section V-B, and containing the GPS time, in [s], when the mobility parameters have been sampled.

The basic layout of a <mobility> message included in a HELLO packet is illustrated in Fig. 1.

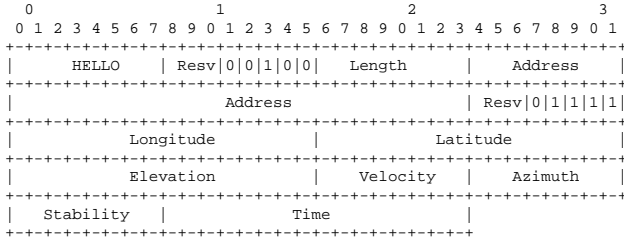


Fig. 1. Hello Packet Containing Geo-localization Information.

IV. THE REAL OVERHEAD OF DIFFUSION OF GEO-LOCALIZATION DATA

In this section, we illustrate the overhead generated by the transmission of geo-localization data in wireless ad hoc networks. We show the non negligible increase of the size of mobility control packets compared to conventional ones.

We first describe a typical mobility information format. As depicted in Fig. 2, node i transmits its position and velocity to node j . For that matter, it transmits 5 fields: X , Y , V_x , V_y and t^{sample} completely describing its localization, velocity, heading and freshness. In this case, those 5 fields are represented by integers and encoded into 32 bits. The total payload per mobility information is therefore 168 bits (21 bytes).

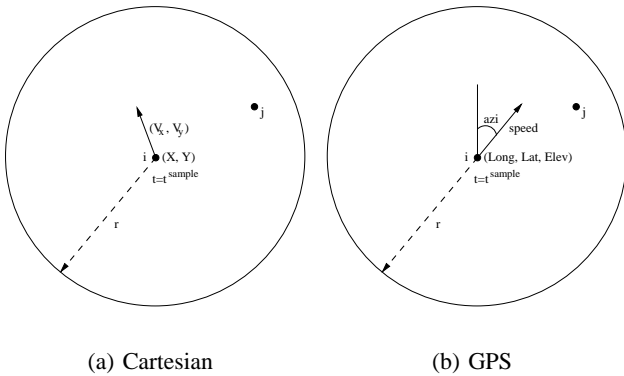


Fig. 2. Neighborhood discovery typical message content.

If the system uses GPS coordinates, the same message consists of the following 5 fields *longitude*, *latitude*, *speed*, *azimuth*, t^{sample} . In that case, the first four fields are represented by 32 bit integers or floating points, while the last field is represented by a 64 bit integer. The total payload per mobility information is therefore 200 bits (25 bytes).

Accordingly, by considering a simple neighborhood discovery heuristic involving a one-hop restricted broadcast of neighborhood information (such approach is used for example in the MPR protocol by OLSR [3]), the total control traffic depends on each node's neighbor degree. Fig. 3 illustrates the drastic overhead increase for the transmission of a single packet as a function of the nodes density. We compare the two previous examples in contrast with the conventional single ID approach. We model a density ranging from 1 neighbor per node to 20 neighbors per node, which is a reasonable assumption for sparse and dense networks.

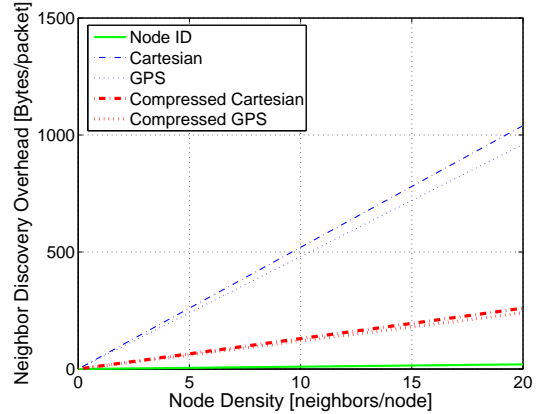


Fig. 3. Illustration of the per packet neighbor discovery overhead with geo-localization data and our proposed compression schema.

We can clearly see the drastic increase in the overhead per packet as a function of the node density. As a typical application usually generates periodic transmission of such packet, we can also extrapolate these results for the overhead created by the use of geo-localization information in mobile ad hoc networks.

V. REDUCING THE GEO-LOCALIZATION OVERHEAD

The motivation of our approach comes from the observation that geo-localization and time data use non-appropriate representation formats. For example, mobility information, if even considered, are usually represented by Cartesian coordinates encoded by a 32 bit integer potentially covering a simulation area of 2^{32} square meters, which is clearly never reached in practice. The representation of longitude or latitude are also done by a 32 bit integer (see [4]), even though the maximum value may only be $180 \cdot 10^6$, including a 6 digit precision. Our approach therefore substitutes the standard representation with a more efficient one based on a *mantissa/exponent* number representation. We aim

at using the minimum number of bits required to cover the full range of applicability of geo-localization data.

A. Compressing GPS Coordinates

As GPS coordinates are represented by a signed integer ranging up to $180 \cdot 10^6$, we use 16 bits. We reserve 1 bit for the sign code c . Of the 15 remaining bits, the most significant 8 bits represent the mantissa a , and the least significant 7 bits represent the exponent b . In the following, K is a constant that is common to all nodes implementing this compression algorithm. As the geo-localization data is represented by an integer, we set $K^{geoloc} = 1$.

Algorithm 1 Signed Integer Compression

Require: A signed integer i and $K > 0$
Ensure: A compressed unsigned integer

```

1: a=b=0
2: c = SIGN(i) {Returns 1 if c < 0; Returns 0 if c ≥ 0}
3: j = ABS(i) {To use it as unsigned}
4: while  $\frac{j}{K} \geq 2^b$  do
5:   b++
6: end while
7: b--
8: if b < 0 then
9:   a = 0
10:  b = 0
11: else if b > 127 then
12:   a = 255
13:   b = 127
14: else
15:   a = 256 ·  $\frac{j}{(K \cdot 2^b)} - 1$ 
16: end if
17: if c > 0 then
18:   return (a · 128 + b) | 0x8000
19: else
20:   return (a · 128 + b) & 0x7FFF
21: end if

```

Algorithm 2 Signed Integer Decompression

Require: Compressed unsigned integer i and $K > 0$.
Ensure: A signed integer

```

1: c = (i >> 15) & 0x01
2: j = i & 0x7FFF {To remove the sign bits}
3: a = j >> 7
4: b = j - (a · 128)
5: return  $1 + \frac{a}{256} \cdot 2^b \cdot K \cdot (-1)^c$ 

```

Using this method, the minimal value representable is $(-1)^c \cdot K$ and the maximum value is $(-1)^c \cdot 3.39 \cdot 10^{38} \cdot K$. We finally add the sign bit representation to obtain the 16 bit representation of a 32 bit signed integer.

This method may therefore be used to represent the geo-localization information in GPS or Cartesian coordinates with a 50% reduction of the number of bits without loss of precision.

B. Compressing GPS time

As GPS time is represented by an unsigned integer with a maximal value of 2^{64} , we also use 16 bits. The most significant 8 bits represent the mantissa a , and the least significant 8 bits represent the exponent b . Using this method, the minimal value representable is K and the maximum value is $1.15 \cdot 10^{77} \cdot K$. Similarly to the geo-localization case, as GPS time is represented by an integer, we set $K^{time} = 1$. This method may therefore

Algorithm 3 Unsigned Integer Compression

Require: An unsigned integer i and a constant $K > 0$
Ensure: A compressed unsigned integer

```

1: a=b=0
2: while  $\frac{i}{K} \geq 2^b$  do
3:   b++
4: end while
5: b--
6: if b < 0 then
7:   a = 0
8:   b = 0
9: else if b > 255 then
10:  a = 255
11:  b = 255
12: else
13:   a = 256 ·  $\frac{i}{(K \cdot 2^b)} - 1$ 
14: end if
15: return (a · 255 + b)

```

Algorithm 4 Unsigned Integer Decompression

Require: Compressed unsigned integer i and a constant $K > 0$.
Ensure: An unsigned integer

```

1: a = i >> 8
2: b = i - (a · 256)
3: return  $1 + \frac{a}{256} \cdot 2^b \cdot K$ 

```

be used to represent the GPS time representation using a 75% reduction of the number of bits without loss of precision.

The same approach may be used for the speed, azimuth or stability, representing a and b with 4 bits each. Using this method, the minimal value representable is 0 and the maximum value is $63488 \cdot K$. By fixing a specific K to reach a target precision, we are therefore able to reduce the size of the transmission of this data by 75%.

As the stability is represented by a strictly positive integer, we propose to fix $K^{stab} = 1$.

The speed representation should consider the application requirements. It is represented in m/s . Therefore, by considering a 2 digits precision, we reach a range between $0m/s$ and $634m/s$, which provides a sufficient range for its representation. Accordingly, we fix $C^{speed} = 0.01$.

The azimuth needs further considerations. In literature, the azimuth is represented in degree with a 6-digits precision, ranging between 0° to 360° . As such type of format cannot be represented by the 8-bits compression, we could increase the size of a and b . However, similarly to the velocity, we should also analysis the applications using this information. The azimuth is directly used as projection in order to obtain the direction of movement. By using the 8-bits compression, we obtain a 2-digits precision, which generates a 0.015% error in cosine projection and 0.006% in sine projection. Accordingly, we are convinced that the error generated by the loss of 4 precision digits are negligible and therefore assume a loss-less azimuth compression. We therefore set $K^{azimuth} = 0.01$

Scenarios	Size			Value K	Compr.
	a [bit]	b [bit]	c [bit]		
GPS	8	7	1	1	50%
Cartesian	8	7	1	1	50%
Time	8	8	0	1	75%
Azimuth	4	4	0	0.01	75%
Speed	4	4	0	0.01	75%
Stability	4	4	0	1	75%

TABLE I

SUMMARY OF AND VALUES OF PARAMETERS FOR DIFFERENT MOBILITY DATA.

Figure 3 illustrates the compression efficiency. We show the overhead reduction as a function of the node density for a 2D geo-localization including velocity, and we can see the significant benefits achieved by our proposed solution. Then, in Fig. 4, we describe the compression efficiency depending on the format of the geo-localization data transmitted for an average density of $10nbrs/node$. We can also clearly see that, already reaching a 66% reduction for a simple 2D geo-localization, the efficiency reaches 71% when we transmit 2D geo-localization, velocity, azimuth, time and stability.

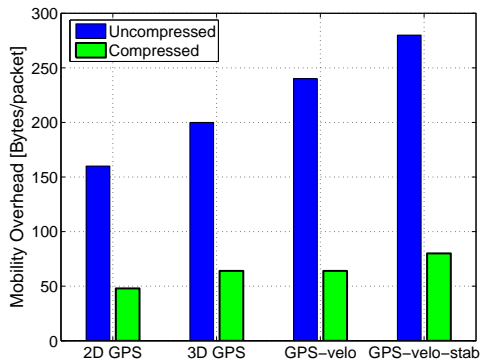


Fig. 4. Compression Efficiency depending on the data format.

Finally, in Fig. 5, we illustrate the Routing Overhead Ratio of the OLSR [3] routing protocol as a function of the node density. We used CBR traffic at a rate of 50kb/s with 10 sources and control traffic of 1 *HELLO* packets per 2 seconds. We can clearly see that as the density increases, so does the cost of carrying geo-localization data. Yet, the proposed compression is able to significantly reduce this drawback.

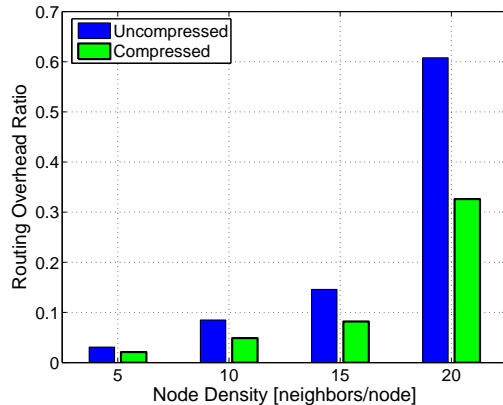


Fig. 5. Illustration of the routing overhead ratio of OLSR with geo-localization information.

VI. CONCLUSION

In this paper, we discussed the overhead generated by the transmission of geo-localization information. We proposed a compression algorithm managing to reduce this overhead up to 71%. We also introduced a Wireless Vehicular Communication message format defining a structure for the transmission of mobility information. We finally illustrated how our solution could improve the routing overhead of the OLSR protocol up to 46%. We therefore provided a framework for optimized and configurable transmission of geo-localization information and believe our approach could ease interoperability and improve the performance of location-based solutions in ITS. Our proposition has also been proposed for a possible standardization within the IETF [1].

REFERENCES

- [1] J. H. *et al.*, "Manet position and mobility signaling format," June 2007, internet Draft, <http://tools.ietf.org/id/draft-haerri-manet-position-signaling-01.txt> (work in progress).
- [2] Wikipedia - World Geodetic System, <http://en.wikipedia.org/wiki/WGS84>.
- [3] P. J. *et al.*, "Optimized link state routing protocol (olsr)," request for Comments, <http://www.ietf.org/rfc/rfc3626.txt>.
- [4] U.S. Census Bureau - Topologically Integrated Geographic Encoding and Referencing (TIGER) system, <http://www.census.gov/geo/www/tiger>.