Vehicle Navigation Service Based on Real-Time Traffic Information

A RESTful NetAPI Solution with Long Polling Notification

Giorgio Ghinamo, Francesco Vadalà, Cecilia Corbi
Telecom Italia,
Torino Italy
name.surname@telecomitalia.it

Piergiuseppe Bettassa, Fulvio Risso, Riccardo Sisto
Dept. of Computer and Control Engineering
Politecnico di Torino, Torino, Italy
name.surname@polito.it

***Abstract*—GNSS-assisted vehicle navigation services are nowadays very common in most of the developed countries. However, most of those services are either delivered through proprietary technologies, or fall short in flexibility because of the limited capability to couple road information with real-time traffic information. This paper presents the motivations and a brief summary of a vehicle navigation service based on real-time traffic information, delivered through an open protocol that is currently under standardization in the Open Mobile Alliance forum.**

Keywords: Infomobility, GNSS-Assited Navigation, REST NetAPI.

I. Introduction

Navigation Devices (NDs) represent a common tool for getting driving assistance. However, most of those devices are based on proprietary service providers (e.g., Google, TomTom, Garmin, etc) that cannot guarantee interoperability. Furthermore, those devices are often poorly integrated with real-time dynamic traffic information, which , according to market surveys, are considered an interesting plus for potential customers because of their capability to optimize driving time and fuel consumption, e.g., by avoiding to select a road blocked for an accident. For this reason, the TPEG (Transport Protocol Expert Group) standardization forum defined a technical specification addressing real time traffic information delivery applications. For instance, the ISO TS 18234 series [1] defines the data structures to convey traffic information, in details:

- travelling time and delays structures for each road segments;
- information about road obstructions, due to accidents or works;
- parking lot information.

However the work done in TPEG assumes that a broadcast communication channel (e.g., digital radio broadcasting) is being used by the application; this choice, even if suitable in some cases, has also some cons, and the broadcasting media may represent an obstacle for the deployment of the application for the following reasons:

- Unavailability in many countries of broadcasting bearers offering the bandwidth required by the TPEG application
- Broadcasting technology imposes constraints on the flexibility in data transmission, in particular in the resolution of road segments, in its geographic coverage and time span.

The ISO TS 24530 [2] series issued by TPEG provides an XML encoding schema for info-mobility information entities defined in the TS 18234 series [1], which enables these XML structures to be used to access TPEG information over the (wireless) Internet. However, this results in an approach that mimics the simple download of a file containing traffic information, which implies a rather inefficient access because of the unavailability of filtering mechanisms that enable the selective access to the provided information.

In this technological scenario, delivering traffic information over mobile data network, with efficient bandwidth occupation, gains importance for mobile operators. Using a two-ways communication channel, customized traffic information may be delivered to the ND and filtered according to journey parameters and selected time; optimal routing for the defined journey may as well be proposed and updated along the drive, taking into account real-time and forecast traffic information. Mobile operators have the edge over other actors in delivering traffic and routing information, due to the fact that they can obtain traffic information by exploiting their own assets. Two different strategies may be implemented by mobile operators to calculate driving times and delays for road segments;

- Anonymous tracking of mobile voice calls
- Tracking of GPS satellite navigation (or ND) users

While the former approach is transparent to the user and hence can be deployed by placing the proper HW/SW in the mobile network, the GPS-based navigation device tracking approach offers better accuracy in terms of user position particularly in rural areas, which turns into an higher resolution in road segments definition.

This paper shows how all the above ideas can be put into practice in the definition of an open standard, closely related to the already existing TPEG standards, which is currently under standardization in the Open Mobile Alliance forum. The rest of the paper is organized as follows. Section 2 describes the reasons for a standard protocol for Dynamic Navigation, while Section 3 presents the main scenarios and use cases for Dynamic Navigation service. Section 4 describes the architecture and main functionalities of the proposed solution, while a possible usage scenario with detailed messages flow is presented in Section 5. Finally, some concluding remarks are presented in Section 6, while the main technologies available for mobile operators to estimate road network performance parameters are summarized in Appendix I.

II. THE NEED FOR AN OMA STANDARD SOLUTION FOR REAL TIME NAVIGATION

So far, application providers targeting dynamic navigation developed mainly proprietary platforms. Nevertheless a standard specification for this service would allow the mobile operators to offer this service to their costumer in an interoperability scenario among different ND vendors and navigation service platforms. This standardization framework aims at allowing large-scale deployments and interoperability of ND Clients and Traffic management Servers in a timely manner. Service providers will have the option to offer dynamic navigation services without having to deploy platforms based on proprietary standards. Such proprietary solutions force users to purchase specific applications provided by incumbent actors. In an open environment, final users will be free to select devices or applications in a fully interoperable scenario.

Functionalities in the navigation domain have already been defined by OMA SUPL framework: a complete set of procedures for tracking and triggering is supported, further more procedures for Assisted GNSS and high accuracy assistance (lane detection) have been included in SUPL [3]. The new enabler represents an additional step toward the full support of navigation applications by the OMA standardization framework.

The OMA DynNav solution presented in this paper reuses all the achievements reached by the TPEG ISO standardization group where data structures for info-mobility have been defined, and it will allow bandwidth optimization with respect to the current TPEG service, filtering delivered information based on routes, geographic areas and time.

The REST NetAPI approach has been adopted by the OMA consortium in order to limit the complexity in the implementations and full adhere to the most recent programming paradigms. The full detail of the OMA DynNav application described in this paper can be found in [4] and [5].

III. MAIN SCENARIOS AND USE CASES FOR NAVIGATION BASED ON REAL TIME TRAFFIC INFORMATION

This section presents some use case scenarios that will be enabled by our DynNav service.

A. *Dynamic vehicle routing: routing information delivery*

This scenario aims at enabling users to connect NDs with a server infrastructure that provides routing information and related traffic information. The ND provides journey information and the server replies with optimal route information for that journey.

The ND requests routing information to the server, sending information about the journey in terms of origin/its current position, final destination, road preferences and vehicle type. The server, based on available real-time and forecast traffic data, estimates the optimal route(s) with related performance parameters and traffic events. From the proposed set of routes, the user deletes the ones he/she is not interested in and subscribes to a notification service in order to receive updated information about *(i)* performance parameters and traffic events for the set of routes, and *(ii)* alternative routes in case of traffic anomalies (e.g. delay or blockage) on current selection, if available.

Optimal routing functionality, based on real time traffic information and current user's position, is not available in current TPEG-compliant solutions.

The DynNav service allows also a different deployment scenario in which a ND proposes a preliminary set of routes to the DynNav server, asking it to provide an estimation of performance parameters and traffic events. As in the previous scenario, the user may subscribe to a notification service to receive updated traffic information and/or alternative routes proposals.

Furthermore, the user can request information about parking lots, public transportation, and Points of Interest (POI) [6] such as restaurants, petrol stations, parking lots, etc., related to the proposed routes.

B. *Traffic information delivery*

This scenario aims at enabling users to connect NDs with a server infrastructure that provides real-time and forecast traffic information. The users request from the server traffic information (real-time and/or forecast) related to a specific area (e.g., in a city neighborhood, or metropolitan area, a region, etc.). In this scenario the route estimation procedure is executed by the ND.

ND communicates to the DynNav server the area of interest (e.g., a city neighborhood, or metropolitan area, a region, etc.) for which it wishes to obtain information. The DynNav Server selects the information related to the specified area and sends it to the ND. Traffic information includes performance parameters for road segments and traffic events. With this information the ND can estimate the optimal route(s) for the user. The user can subscribe to a notification service: at a later stage the server will send updated traffic information to keep the ND aligned with real-time traffic flows and traffic events in the selected area.

C. Market benefits

The user may benefit from this application in terms of travelling time, cost saving and safety; the user may also be informed in real-time about parking lots, public transportation and POIs.

He/she has the freedom to switch to another navigation provider because of the openness of the solution, while currently it is limited to a given service provider unless the entire application is replaced with another one working with the new provider.

IV. TECHNICAL DESCRIPTION OF THE APPLICATION

A. Architecture and main functionalities

The following figure represents the DynNav architecture, showing the DynNav interface and the components.

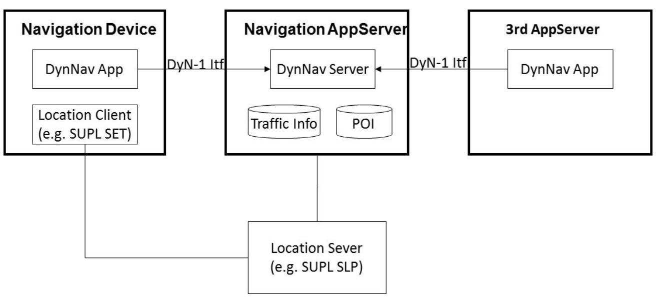

Figure 1: DynNav Enabler Architectural Diagram

The DynNav Server performs the following main functions:

1. Analysis of journey parameters defined by the ND and proposal of a set of routes, based on real-time and forecast traffic data;

2. Provision of real-time and forecast traffic information related to a set of routes proposed by the DynNav server components or by the ND;

3. Provision of real-time and forecast traffic information related to one or more areas defined by the ND;

4. New route proposal in case one of the following conditions holds:

    - Performance of current route become unacceptable;

    - User diverts and deviates from the current route;

5. Provisioning of complementary information (e.g., POIs) related to a route or an area;

6. Notification service of traffic information updates related to the set of routes and or areas previously defined;

Current position of the user may be uploaded on the DynNav server through an external application (i.e. OMA SUPL [3]) or by updating the parameters of the journey.

The DynNav Application, requesting Navigation Information from the server, may reside in a Navigation Device (for turn-by-turn navigation) or in a 3rd party application server. A scenario where the DynNav application resides on an 3rd party application server is represented by journey planning tools offered through the web: the end user will access to real-time traffic information and routing information through an Web interface exposed by the 3rd party application server, the 3rd party application server will access the requested route and traffic information through DyN-1 interface.

B. REST approach with COMET long polling

DynNav follows the OMA RESTful Network NetAPIs guidelines [12] to specify the communication protocol between DynNav server and clients. This section introduces the REST approach and then it discusses how this has been integrated in DynNav.

The REST (REpresentational State Transfer) architecture was defined in 2000 by R. Fielding [10]. The main idea behind REST is that clients and servers interact via (stateless) request and response HTTP messages [11], which refer to a resource identified and addressed by a Uniform Resource Locator (URL). REST does not specify how resources are stored into the server, but only how they are exchanged between server and clients. For instance, on the server side resources are usually kept into a database (or in any other data structure chosen by the implementation) and are then translated into a more neutral representation such as XML or JSON, that will be transported by an HTTP message with the proper encoding. As a consequence, the REST API can be seen as a sort of “implementation detail”, as the most interesting part of the solution is determined by the structure of the data on the server, which is specified in the next section.

Being HTTP-based, REST communications can exploit all the HTTP features in terms of authentication, caching and content negotiation, enabling the creation of rather simple APIs based on existing Web standards [11]. With respect to HTTP methods, REST messages make use of the following commands:

- GET, to retrieve a resource
- POST, to create a resource on the server
- PUT, to update the state of a resource
- DELETE, to remove or delete a resource

Responses must explicitly define themselves as cacheable, or not, to prevent clients reusing old or inappropriate data in response to further requests.

The inherent advantages of the REST approach are to simplify the component implementation and maintenance, due to the clear separation of concerns between client and server, and the possibility to reach high level of architectural scalability, because of the stateless nature.. Please note, that "stateless" does not mean the complete absence of a state, but rather that the server is not required to retain session information about each client for the duration of multiple requests. The result is that in a REST service state is maintained as part of the content transferred from client to server back to client.

DynNav is a REST-compliant service, so it takes advantage of all these features. Moreover, since the stateless approach transfers the entire state at every request, a session can be picked up where it was left off by merely accessing the URI at a later time, regardless of client or server is changed. This is particularly useful in a mobile navigation scenario, where disconnections may happen, and then it is not necessary to introduce overhead to manage reconnections.

Nevertheless, one of the limitations of the REST approach is that transfers have to be started by the client. This is incompatible with some applications in which the servers asynchronously have to initiate the communication toward the client. For instance, the DynNav server has to notify the clients that turned on the subscription for a specific event (e.g., accidents) when such those events happen in the area of interest. This style of communication is called "server push" technology, in contrast to the classic "client pull", where the initial request for data originates from the client.

Since the solution of setting up an HTTP server on mobile devices, to receive pushed notifications, is unfeasible due to limited hardware resources or network limitations (e.g. NATted networks, firewalls), alternative that emulates the pushing are currently used.

Limiting the area of interest to that of the mobile navigation devices, mainly two alternatives to emulate server push are used: Comet long polling [13] and OMA Push [14].

DynNav service, notifications are handled by an additional server, called Notification Server. This server may implement one or both of above solutions, DynNav does not specifies constraints about the choice.

OMA Push solution is SMS-based: an SMS containing URL of updated resources are send in an text message, then client can retrieve updated resource on the server.

The rest of the section shows the long polling technique and its implementation known as Comet. From a research point of view, long polling is a more stable solution, since it does not require a real mobile device and can also be used easily inside emulators. In a real business scenario, the choice between OMA Push and long polling (or both) solutions has to be evaluated on an ad hoc basis.

Long polling is a variation of the traditional polling technique and emulates the push of an information from a server to a client. With long polling, the client requests information from the server in a way similar to a normal poll. However, if the server does not have any information available right away, the server holds the request and waits for some information to be available instead of sending an empty response. Once the information becomes available (or after a predefined timeout), a complete response is sent back to the client. In case of an empty response, the client will prepare itself for the next iteration by issuing a new request immediately, enabling fresh and promptly responses from the server.

Figure 2 depicts how long polling can be used for this purpose (initial phase where trip and routes is omitted).

If a client wants to receive notifications about routes and traffic information, it shall contact DynNav server to create subscriptions for a given class of events. Then, in order to be notified with updates related to those subscriptions, client has to contact the Notification Server (which is known both to the client and the DynNav server) an initialize a Comet session.

When one or more event, which modifies client's subscribed resources, pops up in the DynNav server, a message is sent (in a standard HTTP way) to inform notification server, which has to forward the notification to the client over the Comet session. Then, client can get updated resources from DynNav server an send a new request to notification server, waiting for future notifications.

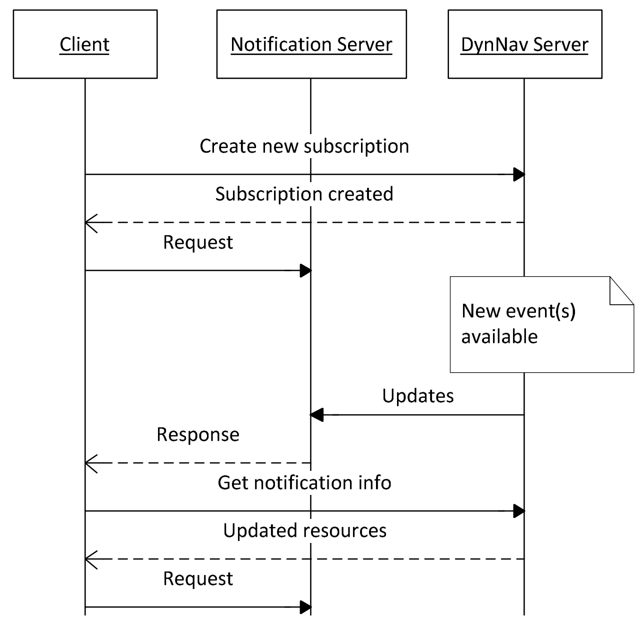

Figure 2 Long polling technique

C. *Resource Tree and data structure*

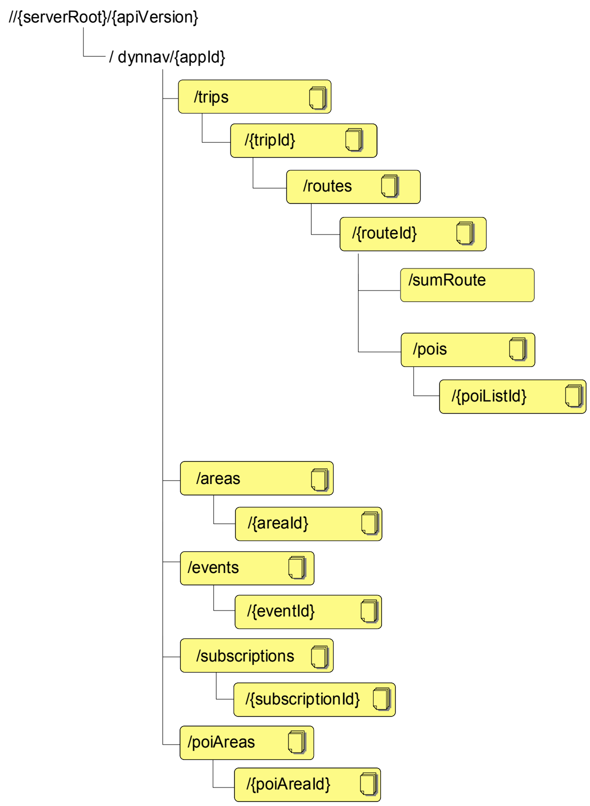

Figure 3 Resource structure

Figure 3 describes the resource tree of the DynNav service. The *trip* resource contains the information about the journey, which is used by the server to propose a set of routes based on real time and forecast traffic information. In fact, a trip can contain the links to a set of *routes* resources. Upon reading the route resource, the application can access routing information and related traffic information (traffic events and road network performance parameters). In details, the route resource contains the sequence of the proposed road segments that allow to reach the final destination; furthermore performance parameters are provided (regular travelling time, delay, speed) for each segment. In the route resources, links to related traffic events are also available. In case of a long and complex route, the route itself can be preliminary encoded in summarized format and the server will provide only the most important segments of the route.

By reading the DynNav *route* resource, the ND can have access to traffic information, as defined by TPEG, in an more efficient way, requesting from the server customized information, related to the expected route and for the selected time interval. This flexibility, originally not possible in old TPEG implementations because of the unidirectional communication channel, is enabled in our implementation because of the bidirectional interaction allowed between ND and servers; as a consequence, performance parameters can be estimated with higher resolution than it was in previous TPEG implementations (broadcast media, XML file access).

Another resource of DynNav is the *area*: it is used to get access to information (traffic events, POI, etc.) based on a geographical area; applications can select the geographical area, the time interval and the categories of interest, as defined by TPEG.

The *subscription* resource is used by the user to subscribe to the notification service, which is available for the following resources:

- Trip: in order to get notified about:
  - new traffic events related to the set of routes defined for the trip
  - updated performance parameters for each route defined for the trip
  - new alternative route proposals in case of congestion
- Area: in order to be notified of new traffic events and performance parameters updates

The notification mechanism enables advanced feature in the DynNav application such as:

- rerouting in case of obstruction on the proposed route
- delivery of new traffic events and performance parameters with throughput optimization with respect to broadcasting and XML approaches.

The last resource is the list of Point of Interest [6] available in a selected area; POI categories can be selected by the user according to classification defined in [7] . As the POI list may be related to a route, a child resource of route has been defined for this purpose.

D. *Data Structures*

This application is designed to reuse as much as possible data structures already defined by standardization fora. In details, the application refers to the structure defined in rtmML document [2] for traffic events and network performance, in locML [2] for location entity description (both documents have been issued by TPEG group in ISO TC 204), in the IETF RFC 4776 [8] for Civic Address data format, and finally to a W3C specification [6] for POI information.

As no public standards are available for encoding route information, a novel encoding schema has been proposed: a route is constituted by a sequence of segments, each segment includes information on origin and destination, link name, measured or forecast performance parameters (travelling time in regular conditions, delays, expected speed), segment shape (sequence of points for graphical representation on the map).

V. Detailed description of a possible information flow

This section describes a typical scenario of a DynNav application where a lightweight ND requests route and traffic information from the DynNav server. The main functionalities defined for this scenario are: (1) the delivery of route information, (2) the subscription to notification services for new traffic events, updated road network performance parameters, (3) current position reporting by the application, (4) the re-routing in case of congestion along the proposed route and in case of deviation and diversion from the route in use. With this set of functionalities, the user receives all along the journey real-time optimal routing information, based on measured/forecast traffic parameters and reported traffic events, encoded according to ISO TPEG schema. In the proposed scenario, route estimation is provided by a centralized entity (DynNav server) exploiting an overall perspective on road network performances; this feature is not possible in current TPEG technology, where routing functionality always pertains to the ND.

In the proposed implementation example, a user of the DynNav application defines the journey in terms of starting point, destination and other preferences, which are sent to the DynNav server by the ND. The DynNav server will reply with a set of routes that match with journey parameters, taking into account real time and forecast traffic information. For bandwidth optimization, the routes are available in the DynNav server in two different formats, summarized and full. In a preliminary stage, the application accesses to the proposed routes in summarized format; with this information the user can select one or more preferred routes: requesting for them the full format (the not selected roots are removed by the application). In case of limited complexity and length of the journey, the proposed routes may be accessed right from the beginning in full format.

The application subscribes to notification services for receiving traffic information updates for the current routes (performance parameters and traffic events for selected categories) and alternative route proposals in case of congestion along the proposed ones.

The application may optionally update its current position on the DynNav server after the vehicle drives a certain distance. With this information, the server will delete segments already travelled from the reference route(s) and remove routes that are no longer compatible with the current position.

Afterwards, the user deviates and diverts from the current route, the application uploads its updated current position, and the DynNav server recognizes that the current position is not compatible with current route(s) and estimates a new one that is notified to the application. To minimize the interaction with the user (for safety reasons), the notification service will be automatically extended to the new proposed route(s) for receiving related updates (events and performance parameters).

Later, due to a traffic jam on the selected route, the DynNav server notifies the application of updated traffic information for the selected route and a proposal of an alternative route, estimated on road network performances measurements. The application accesses to new traffic parameters for the current route and to the proposed alternative route. The DynNav server will automatically provide notification service for the new proposed route if not explicitly removed by the user.

The following Figure 4 provides the message flow for the described scenario.

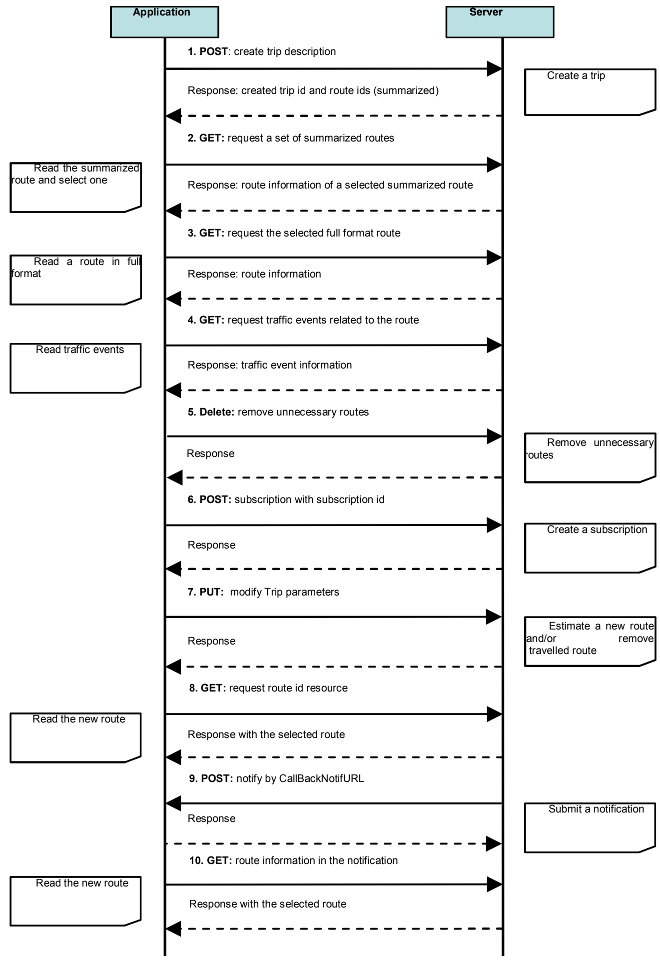

Figure 4 Message flow of possible implementation

VI. Conclusions

This article describes an application for vehicle routing based on real time traffic information on the road network. This service has been endorsed as standard solution for Dynamic Navigation by Open Mobile Alliance OMA standardization Forum.

The service is based on information structure defined by ISO TC 204 (Travel and Transport Information) and represents a step ahead with respect to existing services. State of the art IT Technologies have be exploited: REST NetAPI for its advantage for SW developers, Long Polling technology for Push functionalities of real time updated information.

ACKNOWLEDGMENT

We thanks Hongbeom Ahn (LG Electronics) for his commitment in the development of the OMA DynNav Application.

BIBLIOGRAPHY

[1] Traffic and Travel Information (TTI)" ISO/TS 18234, URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=37265

[2] Traffic and Travel Information (TTI)" ISO/TS 24530, URL:http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=54706

[3] "Secure User Plane Location", Version 2.0, Open Mobile Alliance™, OMA-ERP-SUPL-V2_0, URL:http://www.openmobilealliance.org/

[4] "OMA Dynamic Navigation Enabler", Open Mobile Alliance™, OMA-ER-DynNav-V1_0, URL:http://www.openmobilealliance.org/

[5] "OMA Dynamic Navigation Enabler", Open Mobile Alliance™, OMA-TS-REST-NeAPI-DynNav-V1_0, URL:http://www.openmobilealliance.org/

[6] W3C Points of Interest Core, URL:http://www.w3.org/TR/2011/WD-poi-core-20110512/

[7] Forte et Al; IETF draft Labels for Common Location-Based Services, URL:http://tools.ietf.org/html/draft-forte-ecrit-service-classification-03

[8] IETF RFC 4776, Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information

[9] M. Colonna, P. Lovisolo, et al. "Real-Time Urban Monitoring Using Cell Phones: A Case Study in Rome", IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS

[10] "Architectural Styles and the Design of Network-based Software Architectures", Roy Fielding, 2000, URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[11] "Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Internet Engineering Task Force RFC2616, June 1999.

[12] "OMA Guidelines for RESTful Network APIs V1.0", Open Mobile Alliance™, 2012, URL: http://www.openmobilealliance.org/Technical/release_program/GuidelinesREST.aspx

[13] "RESTful Web Services and Comet", C. McDonald and R. Palkovic, 2008, URL: http://www.oracle.com/technetwork/systems/articles/cometslideshow-139170.html

[14] "OMA Push V2.2", Open Mobile Alliance™, URL: http://www.openmobilealliance.org/technical/release_program/push_v2_2.aspx

Appendix I: Approaches for Traffic Performance Parameters Estimation

DynNav application is based on the availability of traffic information in terms of road network performance parameters and traffic events. Performance parameters may be estimated with different approaches.

From Mobile Operator perspective, anonymously locating and tracking procedures of mobile terminals in conversation status in the networks may be used to estimate road traffic information at least in areas with dense radio cells density. With this approach [9], traffic maps may be generated in near real time, estimating average speed of vehicles on the road.

Mobile terminal position can be periodically estimated, accessing to the results of radio channel measurements made by the Mobile Terminals and Base Stations available in the mobile network.

A light positioning algorithm, based on theoretical path-loss model, allow to process a large amount of power measurements reports tracking all the active phones in the network. From experimental data, implementing a light positioning algorithm based on power measurements, the accuracy of location procedure can reach in urban area with dense cell distribution a precision of 180m in the 67% of samples and 180m and 300m in 95% [9]. These levels of accuracy allow to generate reliable speed information on main road segments in urban area.

Accuracy enhancements, up to 50m in 67%, may be achieved with more sophisticated location algorithms, based on measured or simulated radio coverage maps, with high computational cost as drawback.

Filtering techniques on position estimation may further improve the positioning accuracy increasing the reliability of estimated road performance parameters.

However this approach based on anonymous tracking of calls is applicable only in urban areas with high cell density, it cannot be used in rural area where the level of accuracy is limited to one third of cell radius and cannot be used for user speed estimation.

In rural areas, other sources of information should be used to estimate performance parameters (i.e., travelling time and traffic delays) in the road network to be used in DynNav application. Alternative source for traffic information are:

- GNSS receiver tracking or
- traffic authorities performance measurements.