An ISO IP protocol suite for the integrated road transport and traffic mobile environment

Kim Laraqui <kim.laraqui@haninge.trab.se>
Magnus Lengdell <magnus.lengdell@haninge.trab.se>
Frank Reichert <reichert@it.kth.se>
Andreas Fasbender <andreas@informatik.rwth-aachen.de>

Abstract

We propose a protocol suite for communication amongst mobile and roadside Intelligent Vehicle Highway Systems (IVHS). The suite seeks inspiration from ISO/OSI addressing and systems management paradigms, as well as from the Internet UDP, and is partly implemented and thus validated on a Telia Research short-range communication testbed.

I. Introduction

This paper originates from research and development activities at Telia Research and the Stockholm Royal Institute of Technology, on communication systems for Intelligent Vehicle Highway Systems (IVHS). The most important IVHS research activities in Europe take place within the DRIVE (*Dedicated Road Infrastructure for Vehicle Safety in Europe*) and PROMETHEUS (*Program for European Traffic with Highest Efficiency and Unprecedented Safety*) programs.

DRIVE [1] quests for the application of advances in information and communications technologies to road transport, which together are known as Road Transport Informatics (RTI), in order to realize an Integrated Road Transport Environment (IRTE) in which there will be significant benefits of traffic efficiency, road safety and reduced environmental pollution. PROMETHEUS [2] is a Eureka project, running from 1986 to 1994, which is undertaken by a consortium of major European car manufacturers. Aims are to enhance future driving through information technology and micro-electronics, including vehicle navigation.

I.A. Communication interfaces

The interfaces (Fig.1) contemplated in this paper are for mobile communication systems in the IRTE (IA of Fig.1), i.e. an application process resides in a vehicle or in an adjacent local road segment application server. The latter may in turn be served by some other application process in the IRTE infrastructure, e.g. through conventional ISO/OSI or Internet protocols.

The two fundamental types of component in protocol architectures are *application process* and *medium/bearer*. The architecture serves a framework where these components may be combined by means of protocol entities. Typical examples of application processes are "Route Guidance", "Toll Payment", and "Emergency Warning" [3][4]. Typical examples of bearers are Short-range Communication systems [4] and SOCRATES (System of Cellular Radio for Traffic Efficiency and Safety) [5].

II. Protocol suites

Work on communication systems in DRIVE and PROMETHEUS [3][6], has illustrated that it is unrealistic to propose a monolithic communication infrastructure based on one bearer. Therefore, the architecture distinguishes between *service provider* which is responsible for the correct operation, maintenance, and management of application processes, and *network operator* which has an equivalent liability to bearers. The above implies that the relationship involving application processes and bearers may be quite complex.

To master this diversity in configurations, we have chosen the following types of layer to serve as combinators between RTI application processes and bearers: *Application Layer*, *Multiplexer Layer*, *Network Layer*, and *Subnetwork Layer*. The selection of common layers reduces work with the interfacing of application protocols on top of bearers of various kinds. This choice is controversial within DRIVE and PROMETHEUS. Some argue that the protocol suite should encompass layers 7, 2 and 1, and that dedicated short-range communication systems should take precedence over other technologies for IA interactions (Fig.1).

Others, including the authors, feel that interactions over IA should be specified generally enough to allow other communication subsystems to provide IA communication services. The mapping of application process onto medium should be guided by quality considerations, and by application semantics (e.g. toll payment application processes naturally do interact over a short-range IA). Our view requires a not too tight coupling between applications and bearers and thus yields the inclusion of intermediate layers.

Protocol entities, however, must be designed cautiously and appreciate the specific communication requirements of mobile environments, where, apart from the problems of mobility management itself, in many cases timeliness and bandwidth requirements are very restricting.

III. Application layer

The purpose of the application layer protocols is to provide normalized interfaces where RTI application processes can access communication services as independently as possible from the type of bearer that is used. The effect of this is to promote application modularity, reusability and portability.

There are two main IA (Fig.1) issues in DRIVE and PROMETHEUS over which IA engineers tend to share a common view [7][8]:

- Roughly speaking, the IRTE/IVHS adheres to a server/client paradigm, with servers provided by the infrastructure and clients in vehicles (human, computers etc.).
- Resources, e.g. in vehicles, should be modelled using object-oriented abstractions.

The underlying mechanisms to support the paradigms of above are still debated. There are proposals [7] to model IA interactions as serial line communications on the basis of smart-card standards [9] in which the roadside equipment acts a remote manipulator of vehicle on-board equipment (which is viewed as a kind of smart-card).

III.A. Support from OSI management

We believe the combined server/client and object oriented approach should seek support from OSI management framework (ISO7498-4) since a mapping between the two is straightforward. OSI management, in combination with trading services described below, forms an excellent framework from which to select useful concepts and mechanisms when implementing the distributed IRTE/IVHS:

- Guidelines for the definition of managed objects.
- Remote manipulation based on an adapted version of the OSI Common Management Information Protocol (CMIP, ISO9595).
- Definitions of features such as attributes, notifications, actions, allomorphism for backward compatibility, containment, polymorphism.
- Definitions of functions and objects for logs, access control, accounting, testing, workload monitoring, alarm reporting, etc.
- Simplification of proxy management between roadside and mobile systems, and traffic management centers (e.g. involving both IA as well as IF interface types).

III.B. Application layer structure

In order to avoid the complex OSI mechanisms for context negotiations and management, we choose to explicitly map (application&presentation) con-

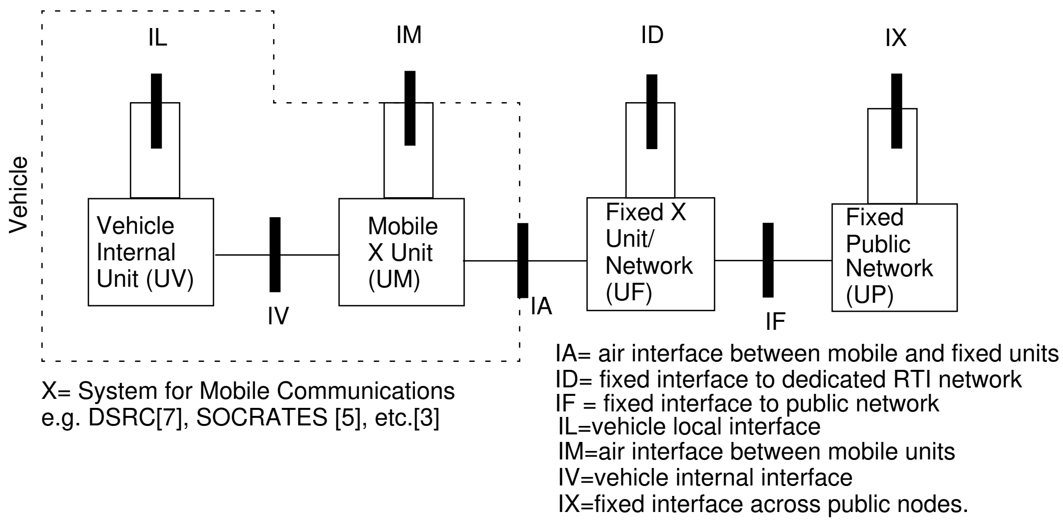


Fig.1

Generic description of interfaces over which RTI interactions may occur.

texts to a set of static "Transport" SAP (Service Access Point) addresses as described in the next section. Further, we plan to use bitmaps to switch contexts within an application association, e.g. between two application protocols over the same roadside-vehicle association. The Application Service Elements (ASEs) we plan to use are the following (they must be streamlined to cater for our specific requirements):

- RTI CMIP, based on the ISO9595 Common Management Information Protocol.
- RTI ACSE, based on the ISO8650 Association Control Service Element Connectionless mode, which will be used for a dynamic setup of associations over non-reserved ports/multiplexer SAPs.
- A dialogue control service element, based on the ISO8327 session service and the ISO9066 Reliable Transfer SE, RTSE.
- RTI CCR, based on the ISO9804 Commitment Concurrency and Recovery ASE.

IV. Multiplexer

The multiplexer (MUX) we have designed on the basis of a combination of ISO8072 Addendum 1 (connectionless mode transport service) and the Internet UDP (User Datagram Protocol, Internet RFC768). Apart from multiplexing, the multiplexer also enables priority scheduling of application data.

The MUX SAPs (see Fig.2) serve two purposes, of which one is static and the other dynamic. The static purpose is one of trading, i.e. there is a set of well-known interfaces with well-known operations that may be invoked by systems that wish to advertize services they offer, or to locate services they wish to use.

The trading facility corresponds to Internet "well-known" UDP ports. We are currently investigating the use of the following RTI adapted Application Service Elements over such well-known MUX SAP addresses: CMIP for 0 and ACSE for 1. The ACSE may then be employed to establish associations dynamically together with simple context specification bitmaps to allocate ASE resources for those associations. Further reserved ports may be RTI application specific, e.g. one for Route Guidance, another for Medium Range Preinformation etc. The use of such ports might not necessarily be required since most interactions could occur through RTI (connectionless) CMIP interactions over MUX SAP 0. Alternatively, RTI application specific ports could be used to imply a node in the object tree from which all object references should be regarded as application specific relative distinguished names (RDNs).

V. Network layer

A Subnetwork Point of Attachment (SNPA) represents a point at which an open system attaches to a subnetwork. The network layer derives SNPA addresses from "globally" unique network addresses by means of explicit mappings. Different service providers on one and the same real system, e.g. beacon, are identified through different NSAP (Network Service Access Point) addresses [10] as depicted in Fig.2.

In our addressing architecture we give a different interpretation to network addresses, as compared to traditional OSI systems, by defining symbolic network addresses. This could be viewed as an alias to an entity that may also be addressed by a "real" network address. Our approach is motivated by the lack, in the OSI Addressing architecture (ISO7498-3), of a real broadcast facility for end systems. Also, it yields small databases to maintain in vehicles.

For vehicles, multiplexer entities may be identified either through a "real" network address or aliases for broadcast purposes. The aliases are also referred to as LEV (Local Environment Vehicle), and *LE* (Local Environment). The LEV names all vehicles in a local environment, and the LE all IA (Fig.2) end system entities in the environment.

Beacons have a LEB (Local Environment Beacon) instead of the LEV that was used for vehicles. The syntax for real NSAP addresses in beacons should somehow reflect the identity of the service provider that is addressed in the road segment. Apart from the broadcast facilities that yield from our approach, we are also able to handle the case of users seeking traffic and transport application services irrespective of service provider (e.g. "Emergency Warning", or "Tourist Information from any - e.g. the cheapest - service provider").

The protocol specified in ISO8473 (the Subnetwork Independent Convergence Protocol, SNICP, i.e. ISO IP) operates over connectionless mode services derived from a wide variety of subnetworks and data links. Subnetworks and data-links that do not provide such a service must make use of a Subnetwork Dependent Convergence Protocol (SNDCP, see ISO8648 on the internal organization of the network layer) or function that translates from subnetwork specific services to the services above. If the underlying service is LLC1 (ISO8802-2) this SNDCP is trivial. For the IRTE, all communication (sub)systems would be expected to design specific SNDCPs; The SNICP would be general and reside atop all system specific SNDCPs.

We use the three profiles of ISO8473: *Segmenting*, *non-segmenting*, and *inactive*. For all of these we have also specified streamlined PDU structures that we are validating in the testbed.

The segmenting and non-segmenting protocols are almost identical to their ISO8473 counterparts. and are primarily intended for subnetworks covering large road segment (e.g. SOCRATES [5]).

The Inactive version does not perform any of the functions of above since it is used to exploit the known characteristics of particular configurations, e.g. toll payment applications on a dedicated Short-range Communication system. Its PDU (there is only one PDU structure) has been defined according to ISO8473, i.e. one octet is added to the application data (and extracted in the other direction) to serve as a protocol identifier. In an implementation, this octet may be omitted by the SNDCP before sending the PDU through the subnetwork, provided peer SNDCP entities agree to this a priori. That would require receiving SNDCP entities to add this octet before conveying the PDU to the SNICP. In this inactive case, the network layer serves the sole purpose of providing a common view on addressing issues, e.g. to distinguish between different service providers available over the same bearer. In a dedicated short-range communication system in which there is only one service provider, the network layer is totally transparent.

VI. Management of network and transport SAPs

Management is related to activities which control or monitor the use of resources providing data storage, data processing and interconnecting capabilities. As a first step towards management of RTI systems, we use the ISO9542 connectionless protocol which permits end systems to exchange configuration and routing information. It is designed to operate in close conjunction with ISO8473 and its addenda. Among others, ISO9542 aims at solving the following practical problem: How do end systems discover the existence and reachability of other end systems on the same subnetwork? In the IRTE, we are primarily interested in configuration management aspects of ISO9542, and not so much in routing facilities.

Vehicles may receive RTMP (RTI Transfer Management Protocol, where "Transfer" refers to Transport+Network) messages whenever they are in the transmission range of an appropriately equipped beacon. The RTMP PDUs contain all relevant information about service providers and available applications in the local environment covered by the beacon issuing the message (note that such messages could also be provided by specialized RTI management entities through other communication systems).

In the RTMP PDU structure, we use an identifier field to uniquely identify a service provider (i.e. an NSAP address [10]). The identifier is followed by a MUX SAP field, of which the bit pattern indicates available (well-known) "ports" for the service provider in the local beacon environment. Availability of

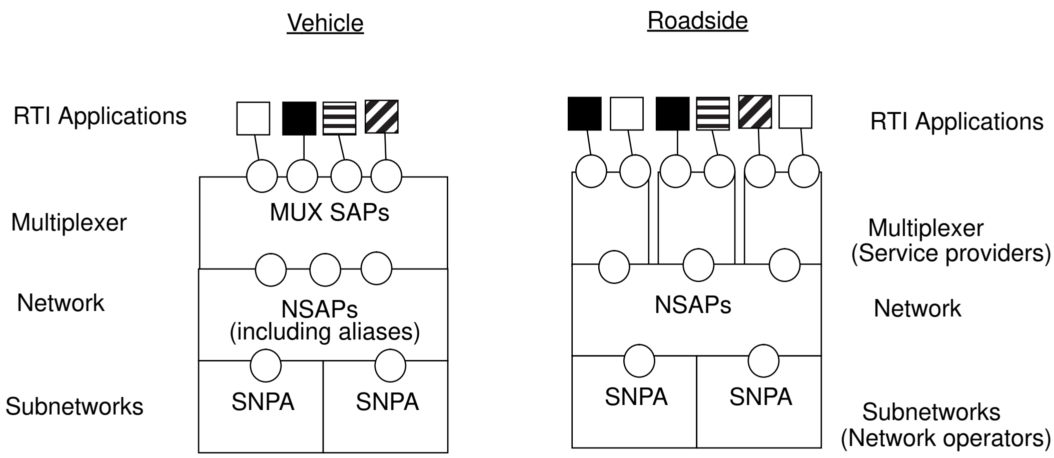


Fig.2

Relationships involving Service Access Points and Addresses.

a SAP is represented with the SAP's position bit set to '1'. Extension bits in the PDU structure allows for future inclusions of providers and/or RTI MUX SAP addresses.

VII. The testbed

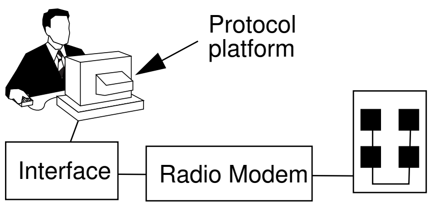
The protocols running on our Short-range Communication (SRC) hardware support an application layer interface, a multiplexer similar to ISO8072 Ad.1, ISO8473, ISO9542, the Media Access Control (MAC, ISO8802-3), and the physical layer including interfaces to the hardware.

The platform consists of several units, each of which includes a PC (PC/AT486, 25 MHz), which serves as a host system to application and communication processors. The physical channel interface includes the antennas, the radio modem and the data recovery unit. A unit can be configured to operate as a fixed (roadside) unit, or as a mobile (vehicle) unit, depending on the scenario under test.

VII.A. Layer implementation issues

Each layer offers one service access point for sending (SSAP) and one for reception (RSAP) of service data units (SDUs). If desired, each SAP is buffered via an input queue for storage of multiple SDUs. However, because of real-time requirements, we have designed the system with the ambition not to buffer data in between layer entities.

The SDUs are passed across SAPs of different layers by means of reference pointers. This avoids time consuming copying of SDUs which would otherwise occur at each SAP. Each layer adds its information in the memory allocated for the packet and hands a new updated pointer to the next layer. A real-time kernel offers parallel processing of tasks and mechanisms to synchronize and exchange interface control information. Each layer represents one task for transmission and reception. A single task avoids the efforts of solving intralayer communications and reduces the number of tasks in the system, and this scheme has therefore been chosen in our testbed.



Technical specifications:

Transmission range - 350 m.
Dynamic range - 70 dB (receiver)
Bitrate - 2Mbps
Packet length - 480 bits (adjustable)
Modulation - FSK
Bandwidth - 20 MHz (adjustable)

Fig.4

Example protocol suites for RTI communication systems.

A SDU is only read from an input SAP when it can be computed completely. A computation consists of reading the SDU from an input queue, computing the embedded PDUs and Protocol Control Information (PCI), generating a new SDU and writing it to an output queue. The queues which interconnect adjacent layers hold SDUs which flow from one communication layer to the next. In this flow, communication layers have equal priority and the layer transmission path and reception path are treated equally.

A layer's task needs only be active when a SDU must be processed. Therefore the queue handling has been combined with the layer task control and a layer has not direct access to the queue. Instead, two functions, `READ` and `WRITE`, hide the queue's internal structure. These functions update a layer semaphore which keeps track of SDUs waiting to be processed by that layer's entity. The semaphore is used by the real-time kernel to reactivate a task that awaits SDUs.

A `READ` to a layer's input queue checks whether a SDU is available. If it is, the corresponding output queue is checked for a free memory space to put it in. If both conditions are fulfilled `READ` decrements the layer semaphore and returns the SDU. A call to `WRITE` is always successful. It increases the semaphore of the corresponding layer which processes the SDU. After a `WRITE`, a layer will automatically suspend itself in order to give other layers time to process a SDU. After completion of processing a SDU, the layer's task hands control over to the real-time kernel.

As all layer tasks, with the exception of the application layer, have equal priority another layer may now compute a SDU. This implies that the entity which received the SDU, from the last active layer task, has the opportunity to compute it and forward it to the next layer and so forth. The reason to allow higher priorities to application entities is to guarantee reliable scheduling of hard real-time control processes.

From a radio transmissional point of view, the system will serve a tool in our efforts to achieve an optimized microwave channel, in particular with respect to coding, methods of modulation, and spectral efficiency. We use our own microstrip patch antennas (+/- 15 degrees), although they may easily be replaced by others. We have chosen to use a conventional radio modem based upon well known technology. After the down conversion of thew 5.8 GHz signal, the clock and data is recovered at 170 MHz.

The receiver has three outputs: 1) The actual data output, which provides the host with the received data (PDUs); 2) The RSSI (Received Signal Strength Indicator), which is used by the host to check that the power of the received signal is sufficient or to find an empty time slot for sending; and 3)The data eye of the demodulated data.

The data output is made available to the interface hardware where the packet detection takes place. The data is then handed on to the host by means of a 16 bit parallel port.

VII.B. Application layer

Currently, we have only implemented specific RTI application protocols and not the generic entities described earlier. The RTI application, which represents user application processes in vehicles or beacons, uses Vehicle Status Application and Medium Range Preinformation Application protocols [1] to compute a local overview of the traffic. All application protocols conform to the recommendations of ISO8509 on OSI service conventions.

If the station emulates a beacon, messages can be generated and bypassing traffic monitored. If the station emulates a vehicle, messages from other vehicles, as well as from beacons, are collected through Medium Range Preinformation. The Vehicle Status Application generates packets at regular intervals (1/10 sec.) for broadcast of vehicle status message to all vehicles in the environment. Parameters are the vehicle's position, speed and current manoeuvre. For field trials, a Text Terminal Application offers the possibility to exchange text messages among stations to coordinate manoeuvres.

The multiplexer takes messages from the three application processes and multiplexes them into one stream for the network layer. It generates a `N_UNITDATA.Request` for interaction with the network layer at the SNICP_RSAP. The inactive protocol is used as much as possible to save bandwidth and minimize computational complexity.

VII.C. Network layer

The SNICP entity sends packets to the SNICP_SSAP and receives packets from the SNDCP_RSAP. When a packet is received (originating from the multiplexer) the `SNICP_Send` function is called. This function controls if the source network or NSAP address is reachable according to the routing table. If the address exists in the routing table and is active, the packet is sent to the destination entity. If the address cannot be mapped onto a SNPA address in the routing table, the packet is discarded.

The `SNICP_Receive` function reads a SDU (in reality an IDU, Interface Data Unit) from the SNDCP_RSAP. Depending on the type of PDU, different functions are executed.The header indicates either a Segmenting PDU, a Non-segmenting PDU, or an Inactive PDU.

VII.D. Network (systems) management

A subset of all functions of ISO9542 is implemented in the testbed. Every time a configuration timer expires, an ESH packet is computed and sent to the SNDCP interface described previously. If an ESH packet is received at the SNDCP, the information is analyzed and the routing table is updated. The routing table is initialized by the function `Init_routing_Table`. If new network or NSAP addresses or new subnetworks are made public to the local IRTE segment, the routing table is updated.

When an update timer expires, the routing table is manipulated by a control function. This function flushes old NSAPs. If the information is too old (i.e. (GetTime - Last_UpdateTime > Holding_Timer) the NSAP is set to NULL meaning it is not available.

VII.E. Data-link layer

The MAC Sublayer has two subfunctions. The first function selects a suitable channel (here: time slot) and monitors it for interference from other stations [11]. The second function transforms the requested logical channel/slot number into a time when the packet has to be transmitted over the hardware. In addition, it is used to harmonize the timeliness of frames with other stations to minimize loss of bandwidth due to overlapping packets.

The current MAC implementation is based on a Fixed Time Division Multiple Access (FTDMA) protocol. The variable number of transmitted frames/second is between one and 128. One frame consist of a number of slots between 16 to 1024.

The benefits of a static frame structure is that the implementation is easier as compared to a dynamic one. The problem with a static implementation is when one station starts sending in a slot already used by another system, in which case a collision is inevitable. With a dynamic frame structure collisions occur less frequently.

If the number of stations in the environment is increased, the Dynamic Time Division Multiple Access (DTDMA) protocol is preferable. The DTDMA protocol is able to detect and avoid collisions, and will be considered in future versions of our testbed.

VII.F. Physical layer

The Physical layer adds a physical frame structure to all packets. The frame consists of a lead-in sequence, a header, a CRC-code and a trailer code. It provides the MAC layer with a hardware independent interface for the exchange of packets.

To isolate the system from hardware specific details, the "hardware layer" (which in ISO/OSI terminology could be viewed as a combination of Physical layer and local hardware environment management entity) takes a complete physical frame and hands it over to the desired hardware. This layer was necessary to provide the SRC testbed with the required flexibility.

The hardware supported by this testbed are a parallel radio modem interface (2Mbps), RS-232, and a local software loopback null modem. Further data on hardware are depicted in Fig.3.

VIII. Conclusions

Practical experience with Internet protocols and theoretical features of the ISO/OSI reference model can be used as important sources of inspiration when designing RTI communication systems.

We exemplified this by proposing a communication systems' architecture based on trading and OSI management on top of connectionless transport, network, and subnetwork entities.

Hands-on experience with protocols and channel characteristics is the most efficient way to gain detailed understanding of real technology and standardization barriers and facilitators. Initial such experience led us along an interesting and realistic specification and implementation path. Nevertheless, further projections onto technology are necessary before we can present more exhaustive conclusions with respect to a complete communication system for the IRTE/IVHS.

In the meantime, we welcome comments to our specification and implementation approaches.

References

(Note: For OSI standards of this paper, we refer to the ISO [12].)

[1] DRIVE, Telecommunications, Information Industries and Innovation, Directorate F, DRIVE 91, 200, Rue de la Loi, B-1049 Brussels, Belgium

[2] PROMETHEUS office c/o Daimler-Benz AG, Postfach 60 02 02, D-7000 STUTTGART 60, Germany.

[3] L. Egnell, T. Lazraq, "Methods for an individual-oriented road traffic information and communication network", The Royal Institute of Technology, Department of Applied Electronics, S-100 44 Stockholm, Sweden.

[4] J. Walker (editor), "Mobile communication systems", Artech House, 685 Canton Street, Norwood, MA 0262, USA, ISBN 0-879006-340-0.

[5] I. Catling, F. Op De Beek, C. Casimir, R. Mannings, F. Zijderhand, W. Zechnall, J. Hellåker, "SOCRATES - System of cellular radio for traffic efficiency and safety", Advanced Telematics in Road Transport, Proceedings of the DRIVE Conference, Brussels, Volume I, February 4-6, 1991.

[6] G. Freij, "An Integration Framework for RTI Systems: Agreed Service Requirements and Architecture Aspects", DRIVE I SECFO (V1057), WP4.8/3, August 1991 (refer to [1]).

[7] H.Haddering, S.Hoff (Editors), "DSRC application layer structure and definition of services", Working document, Version 0.5, CEN TC278/WG9/SG L7, March 1994.

[8] K. Laraqui, A. Nazari, F. Reichert, "RTI systems interconnection - Reference model description", Proceedings of the 25th ISATA (International Symposium on Automotive Technology and Automation) Dedicated Conference on RTI/IVHS, Florence, Italy, 1st-5th June, 1992, ISBN 0 947719 52 0, Published by: Automotive Automation Limited.

[9] ISO 7816-4, "Draft, Identification cards - Integrated circuit cards with contacts - Inter-industry standards for interchange" (see ref. [12]).

[10] ISOTR10730, "Tutorial on naming and addressing" (see ref. [12]).

[11] D. Hübner, K. Jakobs, F. Reichert, "Taking advantage of the disadvantage for improved decentral radio channel access", Proc. of the 41st IEEE VTC, St. Louis May 1991.

[12] ISO, International Organization of Standardization, 1, Rue de Varembe, Case Postale 56, CH-1211 Geneva 20, Switzerland, Tel: +41 22 734 1240, Fax: +41 22 733 3430.

Authors

Mr. Andreas Fasbender received his M.Sc. degree in Computer Science from the Aachen University of Technology, Germany, in 1991. His research interests cover modelling and analysis of mobile data networks, and efficient implementation of communication architectures. Address: Andreas Fasbender, Institute of Computer Science IV, Aachen University of Technology, Ahornstr. 55, D-52704, AACHEN, GERMANY, Tel:+49.241.80 214 10, Fax: +49.241.80 214 29.

Mr. Kim Laraqui has a M.Sc. degree in Computer Science and Engineering from the Royal Institute of Technology, Stockholm, where he is a part-time Ph.D. student at the Department of Teleinformatics (Prof. Björn Pehrson). He is a member of the High-Performance Networking group of Telia Research. Currently, his research focuses on protocols and management systems for data communications over high-speed networks. He is also involved in European standardization activities for the design of mobile protocols for Intelligent Vehicle Highway Systems. Address: Kim Laraqui, Telia Research, S-13680 Haninge, Sweden, Tel: +46.8.7075568, Fax: +46.8.7075480.

Mr. Magnus Lengdell has a M.Sc. degree in Computer Science and Engineering from the University of Luleå, Sweden. For the past years, his work has focused on mobile protocol design and implementation for short-range communication systems, and in the development of object-oriented simulators for such systems. Currently, his main activities are with SS7 and Intelligent Network (IN) functions to support mobile communication systems' infrastructures. Address: Magnus Lengdell, Telia Research, S 136 80 Haninge, Tel: +46.8.7075362, Fax: +46.8.7074684.

Dr. Frank Reichert received his Dipl.-Ing. degree in 1986 and his Ph.D. degree in 1991, from the Aachen University of Technology, Germany. His research areas were decentral control of packet radio networks, in particular those applied to future IVHS communications. His current research interests cover the field of future mobile multi-media communication where he focuses on the network architecture and protocols. Address: Frank Reichert, Royal Institute of Technology, TSLab, Electrum/204, S-164 40 KISTA, SWEDEN, Tel: +46.8.752 1434, Fax: +46.8.751 1793.