

International Journal on

Advances in Telecommunications



2009 vol. 2 nr. 2&3

The *International Journal on Advances in Telecommunications* is Published by IARIA.

ISSN: 1942-2601

journals site: <http://www.ariajournals.org>

contact: petre@aria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Telecommunications, issn 1942-2601
vol. 2, no. 2&3, year 2009, <http://www.ariajournals.org/telecommunications/>

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Telecommunications, issn 1942-2601
vol. 2, no. 2&3, year 2009,<start page>:<end page> , <http://www.ariajournals.org/telecommunications/>

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA

www.aria.org

Copyright © 2009 IARIA

Editor-in-Chief

Tulin Atmaca, IT/Telecom&Management SudParis, France

Editorial Advisory Board

- Michael D. Logothetis, University of Patras, Greece
- Jose Neuman De Souza, Federal University of Ceara, Brazil
- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Reijo Savola, VTT, Finland
- Haibin Liu, Aerospace Engineering Consultation Center-Beijing, China

Advanced Telecommunications

- Tulin Atmaca, IT/Telecom&Management SudParis, France
- Rui L.A. Aguiar, Universidade de Aveiro, Portugal
- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Symeon Chatzinotas, University of Surrey, UK
- Denis Collange, Orange-ftgroup, France
- Todor Cooklev, Indiana-Purdue University - Fort Wayne, USA
- Jose Neuman De Souza, Federal University of Ceara, Brazil
- Sorin Georgescu, Ericsson Research, Canada
- Paul J. Geraci, Technology Survey Group, USA
- Christos Grecos, University of Central Lancashire-Preston, UK
- Manish Jain, Microsoft Research – Redmond
- Michael D. Logothetis, University of Patras, Greece
- Natarajan Meghanathan, Jackson State University, USA
- Masaya Okada, ATR Knowledge Science Laboratories - Kyoto, Japan
- Jacques Palicot, SUPELEC- Rennes, France
- Gerard Parr, University of Ulster in Northern Ireland, UK
- Maciej Piechowiak, Kazimierz Wielki University - Bydgoszcz, Poland
- Dusan Radovic, TES Electronic Solutions - Stuttgart, Germany
- Matthew Roughan, University of Adelaide, Australia
- Sergei Semenov, Nokia Corporation, Finland
- Carlos Becker Westphal, Federal University of Santa Catarina, Brazil
- Rong Zhao, Detecon International GmbH - Bonn, Germany
- Piotr Zwierzykowski, Poznan University of Technology, Poland

Digital Telecommunications

- Bilal Al Momani, Cisco Systems, Ireland
- Tulin Atmaca, IT/Telecom&Management SudParis, France
- Claus Bauer, Dolby Systems, USA
- Claude Chaudet, ENST, France
- Gerard Damm, Alcatel-Lucent, France
- Michael Grottke, Universitat Erlangen-Nurnberg, Germany
- Yuri Ivanov, Movidia Ltd. – Dublin, Ireland
- Ousmane Kone, UPPA - University of Bordeaux, France
- Wen-hsing Lai, National Kaohsiung First University of Science and Technology, Taiwan
- Pascal Lorenz, University of Haute Alsace, France
- Jan Lucenius, Helsinki University of Technology, Finland
- Dario Maggiorini, University of Milano, Italy
- Pubudu Pathirana, Deakin University, Australia
- Mei-Ling Shyu, University of Miami, USA

Communication Theory, QoS and Reliability

- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Piotr Cholda, AGH University of Science and Technology - Krakow, Poland
- Michel Diaz, LAAS, France
- Ivan Gojmerac, Telecommunications Research Center Vienna (FTW), Austria
- Patrick Gratz, University of Luxembourg, Luxembourg
- Axel Kupper, Ludwig Maximilians University Munich, Germany
- Michael Menth, University of Wuerzburg, Germany
- Gianluca Reali, University of Perugia, Italy
- Joel Rodrigues, University of Beira Interior, Portugal
- Zary Segall, University of Maryland, USA

Wireless and Mobile Communications

- Tommi Aihkialo, VTT Technical Research Center of Finland - Oulu, Finland
- Zhiquan Bai, Shandong University - Jinan, P. R. China
- David Boyle, University of Limerick, Ireland
- Bezalel Gavish, Southern Methodist University - Dallas, USA
- Xiang Gui, Massey University-Palmerston North, New Zealand
- David Lozano, Telefonica Investigacion y Desarrollo (R&D), Spain
- D. Manivannan (Mani), University of Kentucky - Lexington, USA
- Himanshukumar Soni, G H Patel College of Engineering & Technology, India
- Radu Stoleru, Texas A&M University, USA
- Jose Villalon, University of Castilla La Mancha, Spain
- Natalija Vlajic, York University, Canada
- Xinbing Wang, Shanghai Jiaotong University, China
- Ossama Younis, Telcordia Technologies, USA

Systems and Network Communications

- Fernando Boronat, Integrated Management Coastal Research Institute, Spain
- Anne-Marie Bosneag, Ericsson Ireland Research Centre, Ireland
- Huaqun Guo, Institute for Infocomm Research, A*STAR, Singapore
- Jong-Hyouk Lee, Sungkyunkwan University, Korea
- Elizabeth I. Leonard, Naval Research Laboratory – Washington DC, USA
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Reijo Savola, VTT, Finland

Multimedia

- Dumitru Dan Burdescu, University of Craiova, Romania
- Noel Crespi, Institut TELECOM SudParis-Evry, France
- Mislav Grgic, University of Zagreb, Croatia
- Christos Grecos, University of Central Lancashire, UK
- Atsushi Koike, KDDI R&D Labs, Japan
- Polychronis Koutsakis, McMaster University, Canada
- Chung-Sheng Li, IBM Thomas J. Watson Research Center, USA
- Artur R. Lugmayr, Tampere University of Technology, Finland
- Parag S. Mogre, Technische Universität Darmstadt, Germany
- Chong Wah Ngo, University of Hong Kong, Hong Kong
- Justin Zhan, Carnegie Mellon University, USA
- Yu Zheng, Microsoft Research Asia - Beijing, China

Space Communications

- Emmanuel Chaput, IRIT-CNRS, France
- Alban Duverdier, CNES (French Space Agency) Paris, France
- Istvan Frigyes, Budapest University of Technology and Economics, Hungary
- Michael Hadjitheodosiou ITT AES & University of Maryland, USA
- Mark A Johnson, The Aerospace Corporation, USA
- Massimiliano Laddomada, Texas A&M University-Texarkana, USA
- Haibin Liu, Aerospace Engineering Consultation Center-Beijing, China
- Elena-Simona Lohan, Tampere University of Technology, Finland
- Gerard Parr, University of Ulster-Coleraine, UK
- Cathryn Peoples, University of Ulster-Coleraine, UK
- Michael Sauer, Corning Incorporated/Corning R&D division, USA

CONTENTS

- Simple vehicle information delivery scheme for ITS networks** **60 - 71**
Katsuhiro Naito, Department of Electrical and Electronic Engineering, Mie University, Japan
Koushiro Sato, Department of Electrical and Electronic Engineering, Mie University, Japan
Kazuo Mori, Department of Electrical and Electronic Engineering, Mie University, Japan
Hideo Kobayashi, Department of Electrical and Electronic Engineering, Mie University, Japan
- Escrow Serializability and Reconciliation in Mobile Computing using Semantic Properties** **72 - 87**
Fritz Laux, Fakultät Informatik, Reutlingen University, Germany
Tim Lessner, School of Computing, University of the West of Scotland, UK
- A Family of Recursive Least-Squares Adaptive Algorithms Suitable for Fixed-Point Implementation** **88 - 97**
Constantin Paleologu, Telecommunications Department, University Politehnica of Bucharest, Romania
Silviu Ciochină, Telecommunications Department, University Politehnica of Bucharest, Romania
Andrei Alexandru Enescu, Telecommunications Department, University Politehnica of Bucharest, Romania
- Adaptive Rate Voice over IP Quality Management Algorithm** **98 - 110**
Eugene S. Myakotnykh, Norwegian University of Science and Technology, Norway
Richard A. Thompson, University of Pittsburgh, USA

Simple vehicle information delivery scheme for ITS networks

Katsuhiro Naito, Koushiro Sato, Kazuo Mori, and Hideo Kobayashi
Department of Electrical and Electronic Engineering, Mie University,
1577 Kurimamachiya, Tsu, 514-8507, Japan
Email: {naito, kmori, koba}@elec.mie-u.ac.jp, k-sato@com.elec.mie-u.ac.jp

Abstract— There has been significant interest and progress in the field of vehicular ad hoc networks (VANETs) in recent years. Intelligent Transport System (ITS) is the major application of VANETs. Vehicle-to-vehicle communication is an important factor for safe driving applications such as blind crossing, prevention of collisions, and control of traffic flows. These applications require exchanges of vehicle information such as vehicle position, cruising speed, direction, and steering angle. Delivery schemes of vehicle information require high delivery ratio, low latency, and high scalability. Additionally, large-size vehicles on actual road environments may interrupt communication between vehicles. Therefore, adequate vehicles should forward vehicle information to their neighbor vehicles in delivery of vehicle information. This paper proposes a new routing protocol for delivery of vehicle information to neighbor vehicles within a specified geographical region. The proposed protocol can deliver new vehicle information with short delay by performing temporal limited flooding before a route construction. Moreover, it can deliver vehicle information effectively with forwarding by adequate vehicles. As a result, our scheme can achieve the high delivery ratio of vehicle information and high scalability. Finally, we assume the different sizes of vehicles in the computer simulations. Then, we evaluate the proposed scheme in the more actual wireless environment. The numerical results show that the proposed protocol can achieve the high delivery ratio with short delay even if the communication between standard-size vehicles is interrupted by the large-size vehicle. Moreover, our protocol has the high scalability in case of increasing of vehicles.

Keywords— VANET, Vehicle-to-vehicle communication, ITS networks, Routing protocol, Vehicle information

I. INTRODUCTION

Vehicular ad hoc networks (VANETs) are new technology to integrate the capabilities of new wireless networks to vehicles. Intelligent Transport System (ITS) is the major application of VANETs [2], [3], [4]. ITS includes several applications such as blind crossing, prevention of collisions, control of traffic flows, traffic monitoring, and nearby information services. These applications can be divided into two major categories. One is called safety application, which improves vehicle safety on the roads. The other is called user application, which provides value-added services such as internet access and entertainment. As for safety applications, their specification requires low latency, high delivery ratio, scalability, etc [5], [6]. VANETs are designed to provide drivers with real-time information through vehicle-to-infrastructure communication or vehicle-to-vehicle communication.

The vehicle-to-infrastructure communication is used for delivering of traffic information, electronic payment of highway tolls, internet accesses, entertainment, etc [7]. Vehicles communicate with many base stations that are equipped along a road. Therefore, vehicles perform handover of base stations one after another. The vehicle-to-infrastructure communication is especially important technology to achieve some user applications in ITS. Meanwhile, vehicles communicate each other in the vehicle-to-vehicle communication. Main service of vehicle-to-vehicle communication is offering vehicle information for safety applications.

The vehicle-to-vehicle communication in VANETs has special attributes that differentiate it from the other types of networks such as mobile ad hoc networks (MANETs). One of the main different features between VANETs and MANETs is related to the behavior of nodes. Vehicles in VANETs are faster than nodes in conventional MANETs. Moreover, the mobility patterns of vehicles in VANETs are more restrictive due to road structures. Therefore, these characteristics are very effective in most of the previous routing protocols [8].

Finding and maintaining routes has many difficulties in the dynamic behavior of vehicles in VANETs. Routing in VANETs has been recently studied and a variety of different protocols were proposed [10]. These protocols can be classified into five categories such as pure ad-hoc routing, position-based routing, cluster-based routing, broadcast routing, and geocast routing.

VANETs and MANETs share the same principle such as self-organization, low bandwidth, and short radio transmission range. Therefore, most ad-hoc routing protocols are still applicable. Ad-hoc on-demand distance vector (AODV) [11] and dynamic source routing (DSR) [12] are well-known routing protocols for general purpose mobile ad-hoc networks. These protocols can reduce overhead in scenarios with a small number of flows. Meanwhile, VANETs differ from MANETs by their dynamic change of network topology. The conventional studies showed that most ad-hoc routing protocols suffer from highly dynamic nature of vehicle mobility and tend to have low communication throughput due to poor route management performance [13].

Vehicle movement in VANETs is usually restricted in just bidirectional movements constrained along roads and streets [14]. Position-based routing employs routing strategies that use geographical information obtained from navigation system

on-board vehicles. Most position-based routing algorithms are based on forwarding decision upon location information. Some protocols exchange information of location and each vehicle's speed, and select a route with minimum link loss probability [15], [16], [17]. Additionally, greedy perimeter stateless routing (GPSR) [18] is one of the well-known protocols. It works best in a free space scenario. However, direct communication between vehicles may not exist due to buildings and large-sized vehicles. Connectivity-aware routing (CAR) protocol finds paths between a source vehicle and a destination vehicle, considering vehicle traffic and movement of vehicles [19].

In cluster-based routing, each cluster can have a cluster head, which is responsible for intra- and inter-cluster communication [20]. Vehicles in a cluster communicate with neighbor vehicles directly. Inter-cluster communication is performed via the cluster-heads. Many cluster-based routing protocols have been proposed in MANETs. However, the VANETs have different features due to constraints on mobility, high speed movement, and driver behavior. As a result, cluster-based routing protocols can achieve good scalability for large networks. But, vehicles suffer from the long delay and the overhead involved in forming and maintaining clusters in VANETs [21].

Broadcast routing is frequently used for delivering advertisements and announcements in VANETs. The simplest way to implement broadcast mechanisms is flooding, in which each vehicle re-broadcasts packets to all of its neighbors. Flooding performs relatively well for a small number of vehicles. However, it suffers from broadcast storm problems when the number of vehicle in networks increases [22]. Some schemes for the broadcast storm problems have been proposed in ad hoc networks [23], [24], [25]. However, the investigation about the broadcast storm problems is not enough to be considered in VANETs.

Geocast routing is a location-based multicast routing [26]. Therefore, packets are delivered from a source vehicle to all other vehicles with a specified geographical region. The geocast routing is benefit mechanisms in many applications of VANETs. For example, a vehicle can detect some problems in neighbor vehicles to prevent collisions. Most geocast routing schemes are based on directed flooding. In VANETs, each vehicle can obtain its own location by using global positioning system (GPS). Therefore, some researchers have proposed forwarding techniques that reduce redundant transmission by using this location information [27], [28].

However, almost all schemes do not consider intercept of communication by large-size vehicle. In the actual VANETs, sizes of vehicles are also different. Therefore, VANET routing protocols should consider the actual communication environment. Another researcher considers broadcast schemes based on IEEE 802.11 [29], [30]. In these techniques, adequate vehicles for forwarding are selected because vehicle positions

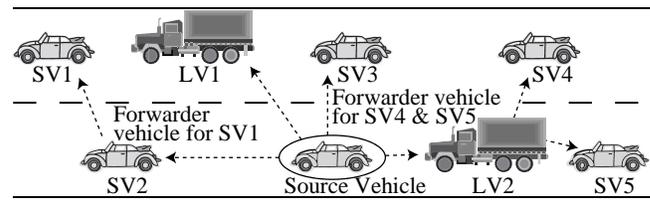


Fig. 1. Vehicle information delivery in ITS.

are exchanged via some control packets. However, actual wireless environments in ITS networks are especially severe from a practical standpoint. For examples, a standard-sized vehicle comes under an influence of blocking by large-size vehicles, and each vehicle suffers from dynamic fluctuation of signal intensity by moving so fast. In these environments, a distance can not be appropriate criteria for selection of forwarder vehicles.

We have proposed a simple delivery scheme for vehicle information [1]. In this paper, we evaluate packet delivery ratio and transmission delay. One characteristic of our scheme is utilizing vehicle information messages (VIMs) themselves for route construction. At first phase, all vehicles forward all vehicle information messages on a temporary basis. Therefore, delivery of new vehicle information can be achieved with short delay. This characteristic will be especially important to achieve blind crossing and prevention of collisions. The reason for this is that almost all routing protocols require several periods to construct routes, and these route construction periods will have big overhead to reduce the delay for recognizing each vehicle. At second phase, each vehicle selects an adequate forwarder vehicle for its vehicle information forwarding. As a result, the number of forwarded vehicle information messages can be reduced to solve broadcast storm problems. This characteristic is an important factor to achieve high scalability with increasing of vehicles. Finally, our scheme utilizes vehicle information instead of hello messages to maintain routes. Consequently, our scheme can check a link status between neighbor vehicles without any control messages, and the number of control messages can be also reduced. We assume the different sizes of vehicles in the computer simulations. Then, we evaluate the proposed scheme in the more actual wireless environment. The numerical results show that the proposed scheme can achieve the high delivery ratio with short delivery delay.

II. SYSTEM MODEL

A purpose of this paper is to achieve a vehicle-to-vehicle communication scheme, which delivers vehicle information within a specified geographical region. Figure 1 is a diagrammatic illustration of vehicle information delivery for safety applications in VANETs. We assume that each vehicle transmits its vehicle information message as a source vehicle

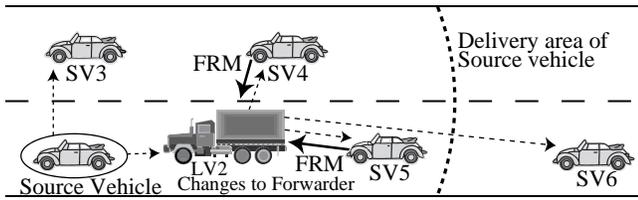


Fig. 2. Example procedure of forwarding request.

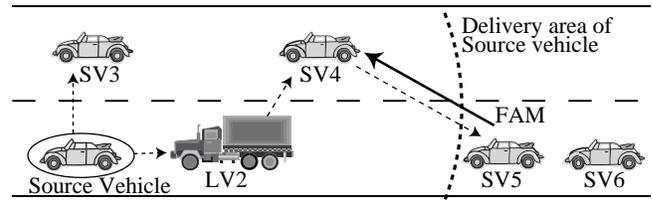


Fig. 4. Example procedure of forwarding abort request.

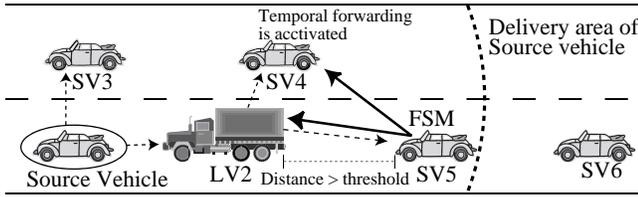


Fig. 3. Example procedure of forwarder search request.

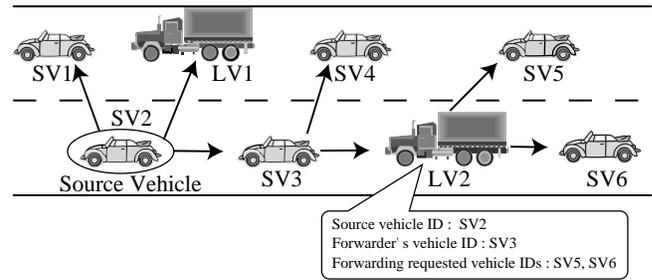


Fig. 5. Example routing information of LV2.

periodically. But, we focus our attention on routes to neighbor vehicles from a source vehicle in Fig. 1. Vehicles *SV2* and *LV2* are forwarder vehicles for their neighbor vehicles. Our protocol can support a mixed environment of standard-sized and large-size vehicles. A vehicle information message is delivered to some vehicles in a limited area. The limited area is defined as the delivery distance, and is determined as a fixed value beforehand. Our scheme can be implemented in a bidirectional road environment by using directional information of vehicles. However, we assume a one-way road in the explanation for simplicity.

In the proposed protocol, three types of control messages are introduced to deliver vehicle information messages; a Forwarding Request Message (FRM), a Forwarder Search Message (FSM), and a Forwarding Abort Message (FAM). The FRM is transmitted when vehicles request neighbor vehicles to activate forwarding function. The FSM is transmitted when vehicles detect link losses. The FAM is transmitted when a distance between a vehicle and its source vehicle is longer than its delivery distance. These example procedures are shown in Figures 2, 3 and 4.

Almost all routing protocols require periodic transmission of control packets because adequate routes may be changed due to moving of vehicles. On the contrary, each vehicle does not transmit the control packets periodically in the proposed protocol. In order to recognize neighbor vehicles, each vehicle uses vehicle information messages as substitutes for special control packets like hello messages. As a result, control messages are only transmitted when vehicles lose links to neighbor vehicles or change links to other neighbor vehicles. Therefore, our protocol can reduce the number of transmitted control messages.

Table I shows the components of the routing table. In the proposed scheme, the routing table in Table I is constructed for each source vehicle. In the assumed ITS networks, vehicle

information is delivered in a limited area near a source vehicle. Therefore, our assumed application is one of multicast application types and the proposed protocol is one of geocast routing protocols. As a result, each source vehicle has a receiver group for vehicle information. In the proposed protocol, the source vehicle ID is used for determining the receiver group for the source vehicle. The source vehicle position is used to detect the delivery area. The final received time of vehicle information is used to remove the routing information if the vehicle does not receive the vehicle information for a long time. The forwarder's vehicle ID is used to maintain its own forwarder vehicle information. The forwarding requested vehicles IDs are an ID list of vehicles, which transmit a Forwarding Request Message to its own vehicle. If this list has some vehicle IDs, the vehicle should forward vehicle information from the source vehicle. The forwarding requested vehicle positions are lists of positions for forwarding requested vehicles. These lists are used to find vehicles that exist outside of delivery area of the source vehicle.

Figure 5 is example routing information of large-size vehicle 2. The LV2 has constructed a route to the SV3 and has been requested to forward vehicle information of the SV2 by the SV5 and the SV6. Therefore, the forwarder's vehicle ID of LV2 is the SV3, and the forwarding requested vehicles IDs are SV5 and SV6.

Figure 6 shows a flow chart of the proposed routing scheme. In this flow chart, a source vehicle *S* transmits a vehicle information message periodically, and a forwarder vehicle *F* forwards the vehicle information message from the vehicle *S*. Finally, a destination vehicle *D* receives it.

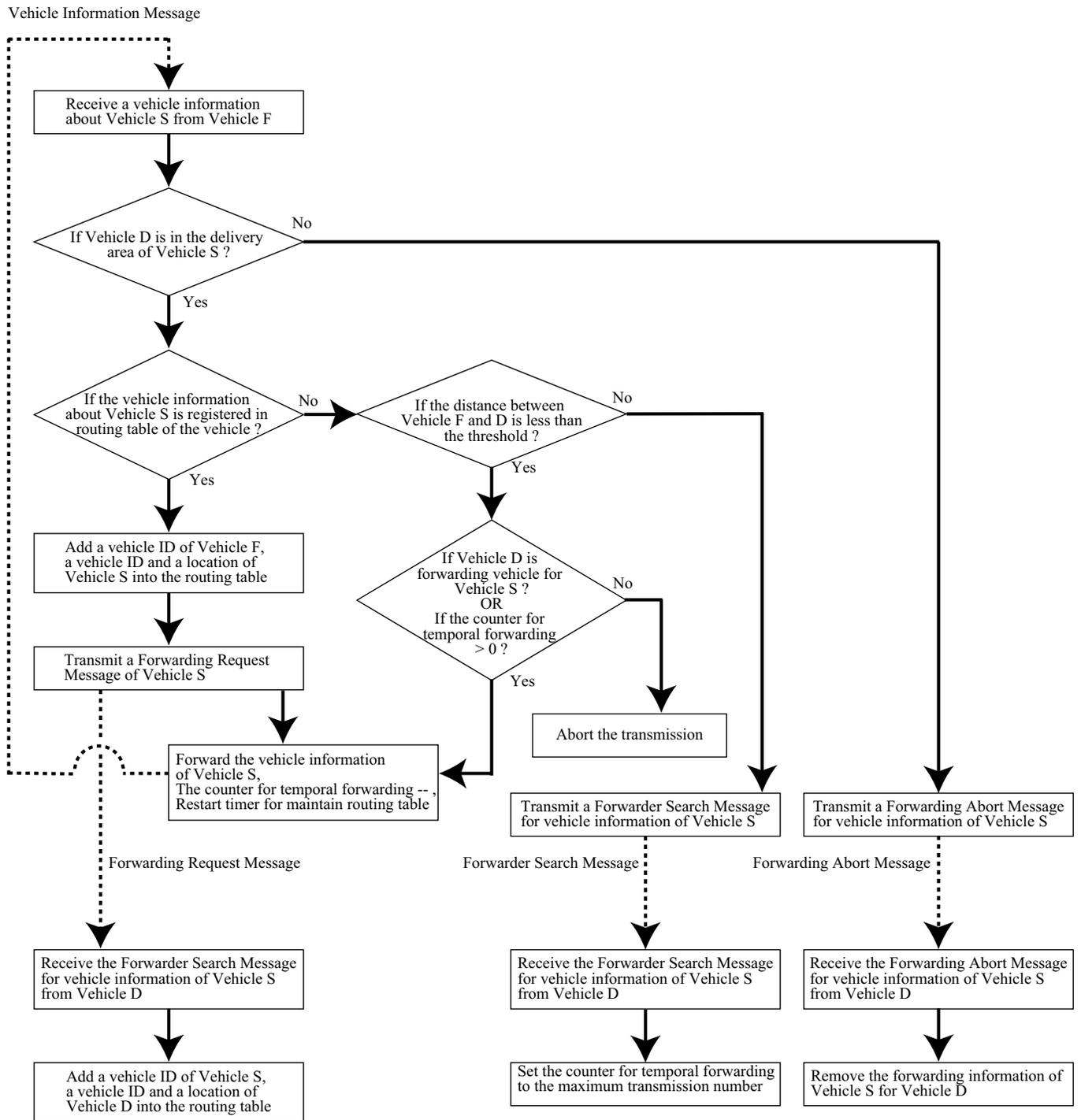


Fig. 6. Flow chart of the proposed routing protocol.

A. Forwarding Procedures

When a vehicle receives new vehicle information messages from neighbor vehicles, two procedures will be performed. The first one is forwarding procedures and the second one is forwarding request procedures. In the forwarding procedures, vehicles forward the received vehicle information message to neighbor vehicles. The procedures are described as follows.

- 1) The vehicle calculates a distance between a previous hop vehicle and itself.
- 2) The vehicle calculates a forwarding delay period according to the distance in order to set priorities of forwarding. The delay period is set to a short time when the distance is long. On the contrary, the delay period is set to a long time when the distance is short. This is because the

TABLE I
COMPONENTS OF ROUTING TABLE.

| |
|--|
| Source vehicle ID |
| Source vehicle position |
| Final received time of vehicle information from source vehicle |
| Forwarder's vehicle ID |
| Forwarding requested vehicle IDs |
| Forwarding requested vehicle positions |

number of hops can be reduced if the distance is long. In the proposed procedures, every vehicle forward vehicle information with prioritized delay on a temporary basis. Therefore, the proposed scheme is tolerant of vehicle movement.

- 3) The vehicle sets a forwarding delay period that is related to the distance.
- 4) The vehicle forwards the received vehicle information message with this forwarding delay period.

B. Forwarding Request Procedures

In the forwarding request procedures, vehicles request to forward vehicle information messages to neighbor vehicles. Procedures are described as follows.

- 1) The vehicle calculates the distance between the source vehicle of the vehicle information message and itself.
- 2) The vehicle checks the routing table to find the source vehicle ID within the vehicle information message when the distance is shorter than the delivery distance.
- 3) The vehicle adds the vehicle ID and a position of the source vehicle into the routing table when the source vehicle ID cannot be found in the routing table.
- 4) The vehicle requests the previous hop vehicle as a forwarder vehicle for itself by transmitting a Forwarding Request Message (FRM).
- 5) The neighbor vehicle that receives the FRM adds a vehicle ID and a vehicle position of the requesting vehicle.
- 6) The neighbor vehicle starts forwarding of vehicle information messages to the requesting vehicle.

Figure 2 is an example procedure when the vehicles $SV4$ and $SV5$ transmit FRMs. In this figure, the vehicle $SV6$ does not transmit a FRM because it exists outside of delivery area of the source vehicle $SV0$. Finally, the vehicle $LV2$ starts forwarding of new vehicle information messages.

C. Forwarder Search Procedures

Following procedures are performed when the distance between a forwarder vehicle and itself becomes longer than a threshold.

- 1) The vehicle tries to find another vehicle as a forwarder vehicle because the current forwarder vehicle is far from itself.

- 2) The vehicle transmits a Forwarder Search Message (FSM) to neighbor vehicles.
- 3) The neighbor vehicles activate each forwarding function of vehicle information messages if each distance between the vehicle transmitting the FSM and themselves is shorter than the threshold.
- 4) The neighbor vehicles start forwarding their vehicle information messages for a while. The maximum retransmission time of vehicle information messages is set to a counter for the temporal forwarding.
- 5) The vehicle transmits a new FRM to an adequate vehicle of its neighbor vehicles when it receives a new vehicle information message from them.

Figure 3 is an example procedure when the vehicle $SV5$ transmits a FSM because the distance between the $LV2$ and the $SV5$ is longer than the threshold. In this figure, the vehicle $SV4$ activates the temporal forwarding procedures for the $SV5$. Hence, the vehicle $SV5$ will transmit a FRM to the vehicle $SV4$.

D. Forwarding Abort Procedures

Following procedures are performed when vehicles move to outside of the delivery area of their source vehicle

- 1) The vehicle transmits a Forwarding Abort Message (FAM) to its forwarder vehicle.
- 2) The forwarder vehicle removes the forwarding information for it from the routing table.

Figure 4 is an example procedure when the vehicle $SV5$ moves to outside of the delivery area, and transmits a FAM to the vehicle $SV4$. The vehicle $SV4$ will inactivate the forwarding procedures for the vehicle $SV5$.

III. EXAMPLE OPERATIONS

In the proposed scheme, each vehicle starts to construct a route by receiving a new vehicle information. In this section, we explain example operations of the proposed scheme with the vehicle layout in Fig. 7. Figure 8 shows an example of packet transmission in this situation. In the example, each vehicle is assumed to deliver vehicle information within radius R .

In Fig. 7, the vehicle V_1 is regarded as a source vehicle. The vehicles $V_2, V_3, V_4,$ and V_5 exist in the area where the vehicle information of the vehicle V_1 can be delivered. The vehicle V_1 transmits the vehicle information messages (VIMs) periodically. The neighbor vehicles V_2 and V_3 register the new vehicle V_1 by checking each routing table. Then, each vehicle calculates a forwarding delay according to relative position to the source vehicle V_1 . The vehicle V_3 sets a shorter delay than the vehicle V_2 because the relative position to V_3 is longer than that of V_2 . This procedure reduces the hop count for vehicle information delivery. Then, vehicles V_4 and V_5 , which receive the vehicle information forwarded by

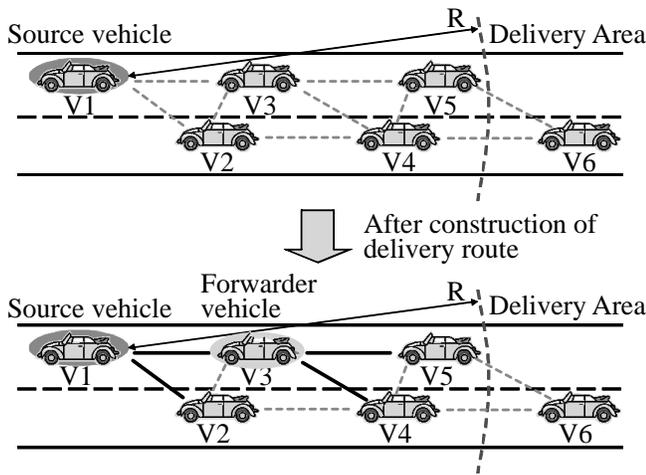


Fig. 7. New vehicle information forwarding and route construction process.

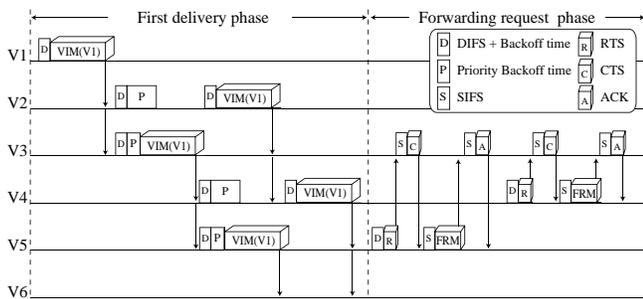


Fig. 8. Time sequence of the new route construction process.

the vehicle V_3 , transmit each Forwarding Request Message to request vehicle information forwarding. The vehicle V_3 registers the forwarding requested vehicle IDs and positions when it receives the Forwarding Request Messages. Finally, the vehicle V_3 constructs a route to the vehicle V_4 and the vehicle V_5 from the vehicle V_1 .

A. Route construction for new vehicles

Each vehicle transmits its own vehicle information periodically. Therefore, vehicles can find that a neighbor vehicle moves into delivery area of their source vehicle. Figure 9 shows an example that the vehicle V_6 moves into the delivery area of the vehicle V_1 . An example for packet transmission is shown in Fig. 10.

In Fig. 9, vehicles V_4 and V_5 can find the position of the vehicle V_6 because these vehicles exchange the vehicle information each other. Moreover, the vehicles V_4 and V_5 can find that the vehicle V_6 moves to the delivery area of the vehicle V_1 because they know the positions of the vehicles V_1 and V_6 .

The vehicles V_4 and V_5 start to forward the vehicle information from the vehicle V_1 when they find that the vehicle V_6 moves to the delivery area of the vehicle V_1 . Consequently, the vehicle V_6 will be able to receive the vehicle information

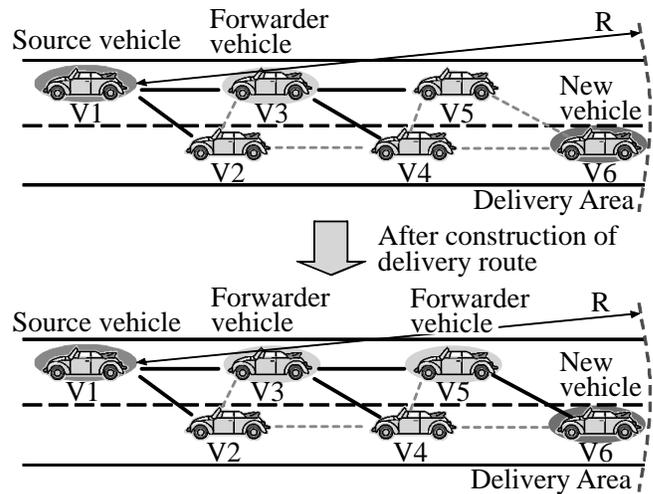


Fig. 9. Route construction for the new vehicle.

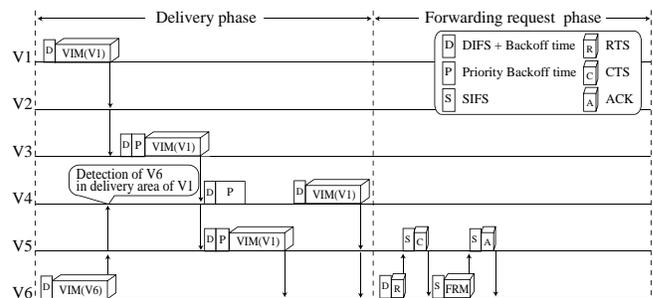


Fig. 10. Time sequence of the new route construction process for the new vehicle.

of the vehicle V_1 through the vehicles V_4 and V_5 . The vehicle V_6 constructs a route by requesting vehicle information forwarding.

B. Route modification for moved vehicles

Vehicles can recognize neighbor vehicles by exchanging vehicle information each other. Therefore, a forwarding requested vehicle can start to find the next neighbor vehicles if it cannot communicate with the forwarder vehicle. Figure 11 shows that the vehicle V_5 moves to the outer area of the vehicle information delivery area of the vehicle V_3 . An example for packet transmission is shown in Fig. 12.

In the Fig. 11, the vehicle V_5 finds that the route through the vehicle V_3 becomes invalid by checking the vehicle information from the vehicle V_3 . Then, it broadcasts the Forwarder Search Message to neighbor vehicles. The vehicle V_4 , which receives the Forwarder Search Message from the vehicle V_5 , starts the temporary forwarding of the vehicle information. Finally, the vehicle V_5 can find the new route through the vehicle V_4 .

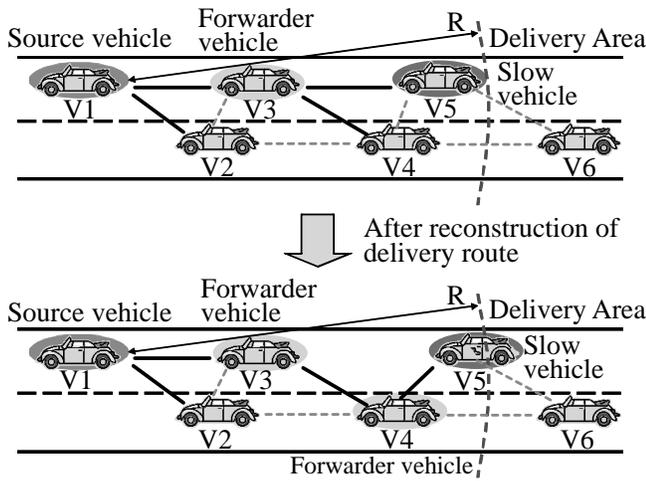


Fig. 11. Route modification process.

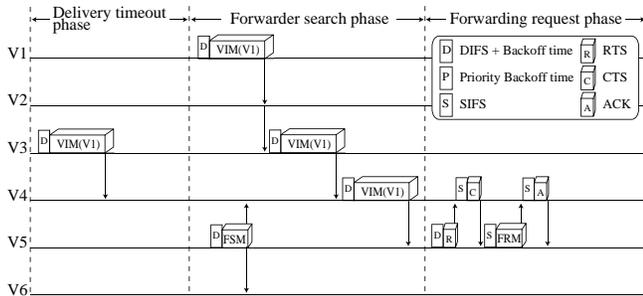


Fig. 12. Time sequence of the route modification process.

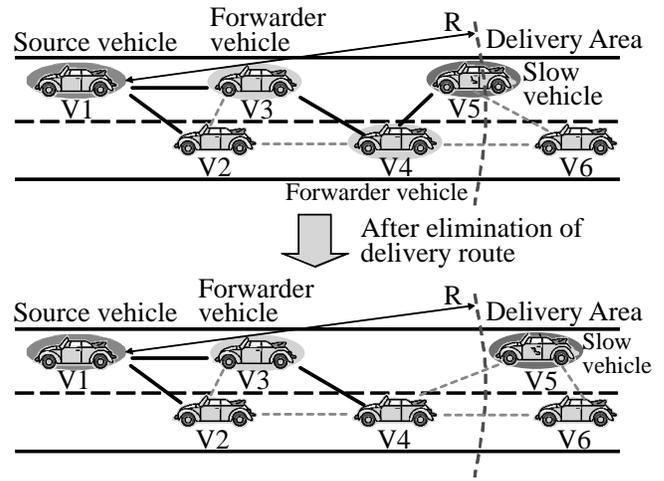


Fig. 13. Route discard process.

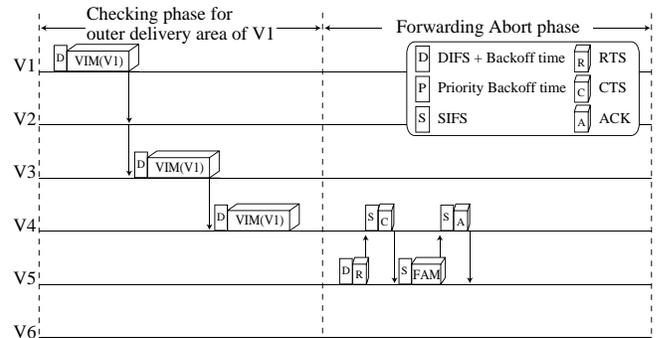


Fig. 14. Time sequence of the route discard process.

C. Route discard for moved vehicles

In the proposed scheme, two procedures for route discard are considered. The first is used in a situation that vehicles in a certain delivery area cannot receive any information of a vehicle, which were in the area, and they cannot recognize it any more. The second is used in a situation that vehicles in a certain delivery area can receive information of a vehicle and can recognize it, but it is moving out to the area. In the situations, they discards the routes in their own routing table. Figure 13 shows an example that the vehicle V_5 moves to the outer area of the vehicle information delivery area of the vehicle V_1 . Figure 14 shows an example for packet transmission in this situation.

The vehicle V_5 uses the route through the vehicle V_4 in Fig. 13. It transmits a Forwarding Abort Message to the vehicle V_4 if it exists in the outer area of delivery area for a given length of time. Finally, the vehicle V_4 stops vehicle information forwarding and removes the route for the vehicle V_5 .

IV. NUMERICAL RESULTS

In order to evaluate the feasibility of the proposed scheme, we performed computer simulations with network simulator QualNet [31]. Qualnet is the well-known wireless network

simulation software that considers the more actual wireless environment. Therefore, packet errors are handled as the packet error ratio according to the received signal-to-interference and noise power ratio (SINR). Each results shows an average of 10 trials of simulation. Our proposed protocol is one of the information delivery schemes by broadcast communication. It is known that broadcast communication suffers from packet collisions when many vehicles exist in a communication area. Therefore, we considered 50 vehicles for small number of vehicles and 200 vehicles for large number of vehicles. We assumed that a road shape is a loop line with a radius equals to 1500 [m] and 2 lanes. Each vehicle is located randomly on the road, selecting the velocity between 90 [km/h] and 110 [km/h] randomly. Therefore, a distribution of vehicle velocity is uniformly between 90 [km/h] and 110 [km/h]. The vehicle runs on the inside lane principally and keeps an inter-vehicular distance as 100 [m]. If there is no vehicle on the outside lane, the vehicle moves to the outside lane from the inside lane to overtake a forward vehicle. After overtaking, the vehicle moves to the inside lane if there is no vehicle on the inside lane. In the simulations, about 50 times of passing occur when

TABLE II
SIMULATION PARAMETERS.

| | |
|-------------------------------------|-------------------------------|
| Simulator | QualNet |
| Simulation time | 150 [s] |
| Simulation trial | 10 [times] |
| Number of vehicles | 50, 200 [vehicles] |
| Vehicle velocity | 90 – 110 [km/h] |
| Size of vehicle information message | 100 [Bytes] |
| Transmission interval | 250 [ms] |
| Communication device | IEEE 802.11b |
| Transmission rates | 11 [Mbps] |
| Transmission power | 15 [dBm] |
| Antenna gain | 0 [dB] |
| Antenna type | Omni directional |
| Antenna height | 1.5 [m] |
| Propagation path loss model | Two ray |
| Wireless environment | AWGN |
| Road shape | Circle with radius = 1500 [m] |
| Number of lanes | 2 [lanes] |

the number of vehicle is 50, and about 150 times of passing occur when the number of vehicle is 200. Finally, the feature of this paper is to consider the effect of large-size vehicles. So, we define the large-size vehicle ratio that means the ratio of the large-size vehicles and the standard-size vehicles. When the large-size vehicle ratio is set to 0, all vehicles are standard-size vehicles.

As the wireless propagation model, we used a two ray propagation model. Moreover, we consider blocking effects due to large-size vehicles. So, we assumed that large-size vehicles are rectangular solids. If a rectangular solid is overlapped with the straight line between two standard-size vehicles, these two vehicles cannot communicate due to blocking.

The final purpose of this study is to fuse vehicle information delivery and communication networks for several network applications. Therefore, we employ IEEE 802.11b for a common communication device. In the simulations, the transmission range is about 500 [m], packet errors are determined due to the received signal-to-interference and noise power ratio (SINR). Our packet error model can consider packet collisions and noises. The size of a vehicle information message is 100 [Byte], and is transmitted with 4 [packets/s]. The delivery area of vehicle information messages is assumed to be 1000 [m].

Our protocol is one of the broadcast communication methods. Therefore, we employ the probabilistic flooding scheme for comparison. The flooding probability is assumed to be 0, 25, 50, 75, 100 [%]. Simulation parameters are shown in detail in Table II.

Figure 15 shows the delivery ratio of vehicle information messages with 50 vehicles. In this study, we define that the delivery ratio is the message received ratio for vehicles in the delivery area. From results, we can find that our proposed protocol can achieve the highest delivery ratio. The delivery ratio of the probabilistic flooding scheme degrades when the

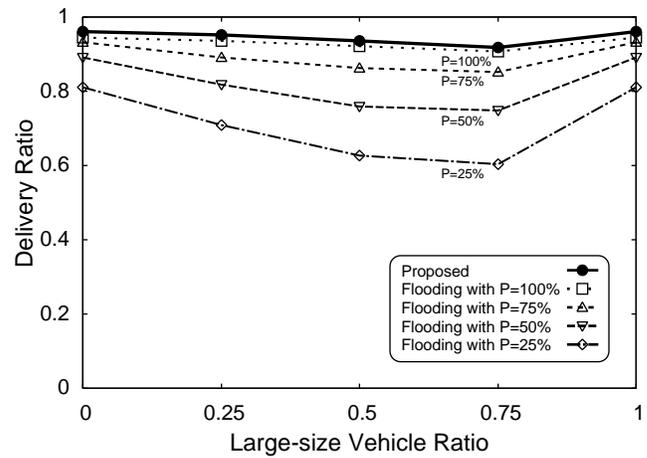


Fig. 15. Delivery ratio of vehicle information (50 vehicles).

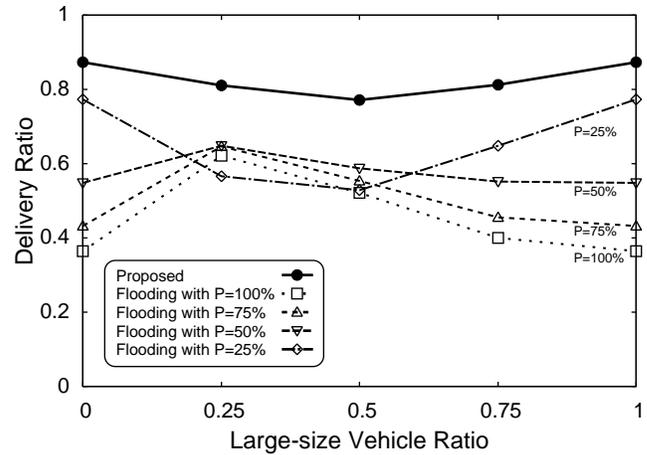


Fig. 16. Delivery ratio of vehicle information (200 vehicles).

flooding probability decreases. This is because several vehicles are required to forward vehicle information messages when there are a small number of vehicles on the road. Moreover, the delivery ratio of all schemes degrades when the large-size vehicle ratio increases. Especially, it degrades much when the value of the flooding probability is set low. The reason for this is that large-size vehicles block communications between standard-size vehicles. So, more vehicles should be required to forward vehicle information messages.

Figure 16 shows the delivery ratio of vehicle information messages with 200 vehicles. From results, our proposed protocol can keep the highest delivery ratio. But, the delivery ratio of the flooding scheme degrades. This is because the flooding schemes suffer from broadcast storm problems. We can find that some flooding schemes can achieve good delivery ratio. However, the optimum flooding probability is also changeable depending on situation change. So, it is difficult to select the optimum flooding probability in the actual system.

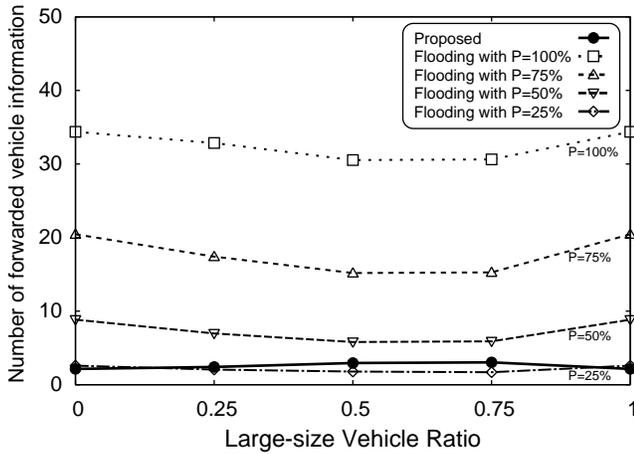


Fig. 17. Number of forwarded vehicle information (50 vehicles).

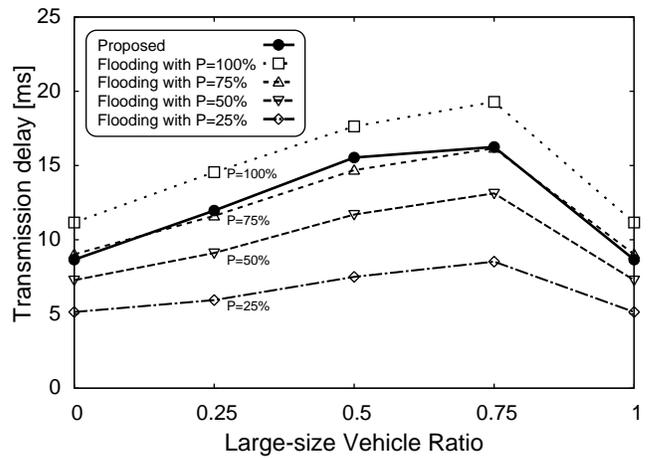


Fig. 19. Delay performance (50 vehicles).

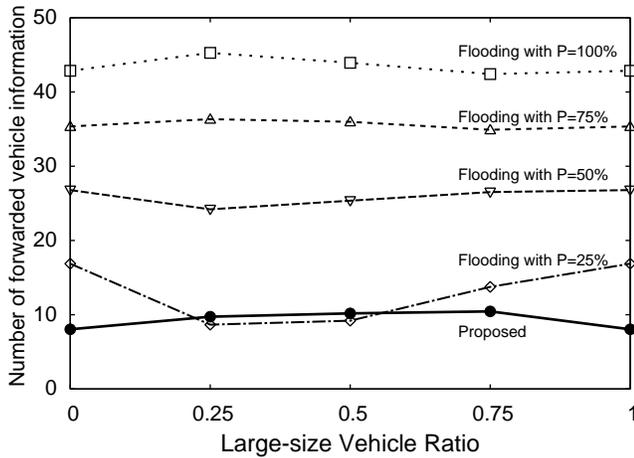


Fig. 18. Number of forwarded vehicle information (200 vehicles).

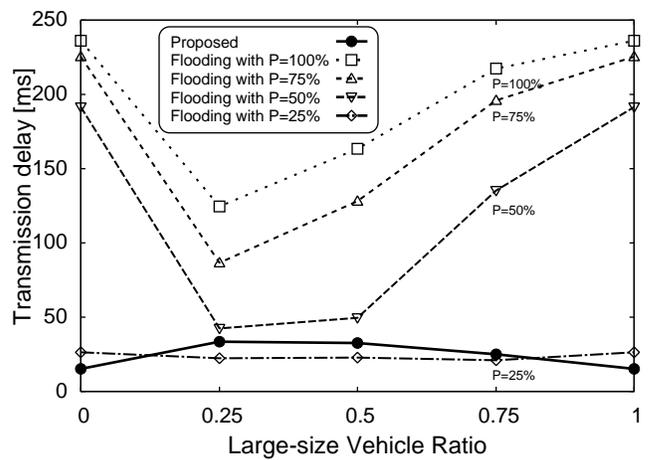


Fig. 20. Delay performance (200 vehicles).

Incidentally, the delivery ratio of the proposed protocol can achieve high performance even if the large-size vehicle ratio is changed because each vehicle selects an optimum vehicle as its forwarder vehicle in the proposed protocol. In the conventional research, the objective packet delivery ratio is assumed to be 90 [%]. In the broadcast communication, packets may be corrupted due to hidden terminal problems. Therefore, it is difficult to achieve high delivery ratio when special media access control (MAC) method is not employed. In our delivery ratio, we evaluate packet delivery ratios at all receiver vehicles. Therefore, we think that our protocol can be used for the actual environments by employing the Forward Error Correction (FEC).

Figure 17 shows the number of forwarded vehicle information in the delivery area with 50 vehicles. From results, we can find that the flooding schemes require several times of forwarding. Therefore, the flooding schemes can achieve high delivery performance. However, these excess forwarding are

unreasonable from the viewpoint of the wireless resource. The probabilistic flooding can decrease the number of forwarding. But, the delivery ratio is also degraded. On the contrary, our proposed protocol requires small number of forwarding like the probabilistic flooding with 25 [%]. However, the proposed protocol can achieve the high delivery ratio like the full flooding scheme. Therefore, our protocol is a reasonable scheme from the viewpoint of wireless resources.

Figure 18 shows the number of forwarded vehicle information in the delivery area with 200 vehicles. From results, the performance of the probabilistic flooding with 25 [%] keeps small number of forwarded vehicle information messages. However, the delivery ratio of the probabilistic flooding with 25 [%] degrades due to blocking by large-size vehicles. This is because it is difficult to forward vehicle information message appropriately when the flooding probability decreases. Meanwhile, the proposed protocol can keep the smallest number of forwarded vehicle information messages. Moreover, the

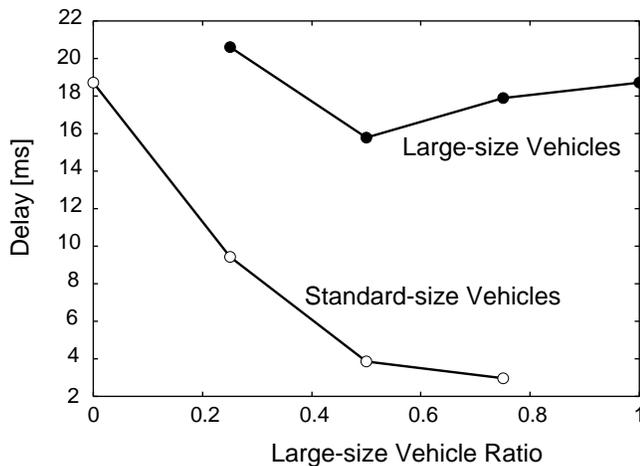


Fig. 21. Delay performance of standard and large-size vehicles.

performance of the proposed protocol achieves a stable delivery ratio and a stable forwarding performance because each vehicle selects its forwarder vehicle by considering blocking due to large-size vehicles.

Figure 19 shows the delay performance with 50 vehicles. The delay period starts when a source vehicle transmits a vehicle information message, and ends when the vehicle information message is received at all vehicles in the delivery area. Therefore, the accurate delay of each vehicle is different due to the positions of the vehicles. So, the delay performance averages delays of all vehicles in the delivery area. From results, the delay performance of the proposed protocol is a little shorter than that of the full flooding scheme. The delay performance of all schemes increases when the large-size vehicle ratio increases. The reason for this is that blocking by the large-size vehicles causes degradation of the actual transmission range. Therefore, more forwarder vehicles are required to transmit the vehicle information messages.

Figure 20 shows the delay performance with 200 vehicles. From results, the delay performance of the proposed protocol can keep short values when the large-size vehicle ratio changes. On the contrary, the flooding schemes have especially long delay when the large-size vehicle ratio equals to 0 or 100 [%]. The actual transmission range becomes long when there is no effect of blocking due to the large-size vehicles. Therefore, broadcast storm problems occur.

Figure 21 shows the delay performance of standard-size and large-size vehicles in the proposed protocol. This kind of delay is required to transmit vehicle information in MAC layer. From results, we can find that delays of large-size vehicles decrease according to increasing of the large-size vehicle ratio because large-size vehicles block communications between standard-size vehicles and the number of vehicles in a certain communication area also decreases. Therefore, each vehicle can obtain more opportunities to transmit vehicle information. On the

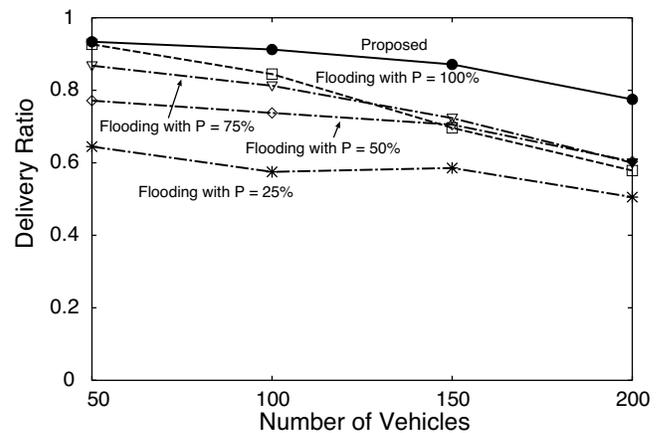


Fig. 22. Delivery ratio of vehicle information (large-size vehicle:40[%]).

contrary, the delay performance of large-size vehicles is also constant. This is because large-size vehicles can communicate with standard-size vehicles and large-size vehicles. Moreover, these communication are not blocked. Then, the number of vehicles sharing the same communication is also increasing. As a result, it is difficult for large-size vehicles to obtain opportunities to transmit vehicle information.

Figure 22 shows the delivery ratio of vehicle information with a large-size vehicle ratio equals to 40 [%] and 200 vehicles. From results, the performance of the all flooding mechanisms degraded according to increasing in the number of vehicles. The reason for this is that it is difficult to select adequate forwarding vehicles in the situation the large-size vehicle ratio equals to 40 [%]. Meanwhile, the proposed protocol has good scalability performance. The scalability is one of the most important factor in ITS. This is because the proposed protocol is especially simple and only a few control messages are exchanged when a vehicle joins certain networks, it changes its forwarding vehicle and drops out the networks. Moreover, the proposed protocol can select an adequate forwarding vehicle, and improve effectiveness of channel resource.

Figure 23 shows the number of forwarded vehicle information with a large-size vehicle ratio equals to 40 [%] and 200 vehicles. From results, we can find that the proposed protocol can keep a small number of forwarded vehicle information. However, the performance of the proposed protocol is little larger than that of the probabilistic flooding with $P = 25$ [%] because the proposed protocol can select the forwarding vehicles. Therefore, more vehicles are selected as a forwarding vehicle when large-size vehicles blocks communication between standard-size vehicles.

Figure 24 shows the delay performance with a large-size vehicle ratio equals to 40 [%] and 200 vehicles. From results, the delay performance degrades according to increasing in the

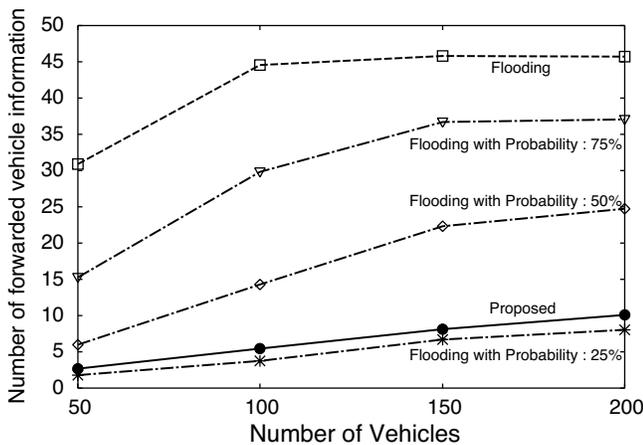


Fig. 23. Number of forwarded vehicle information (large-size vehicle:40[%]).

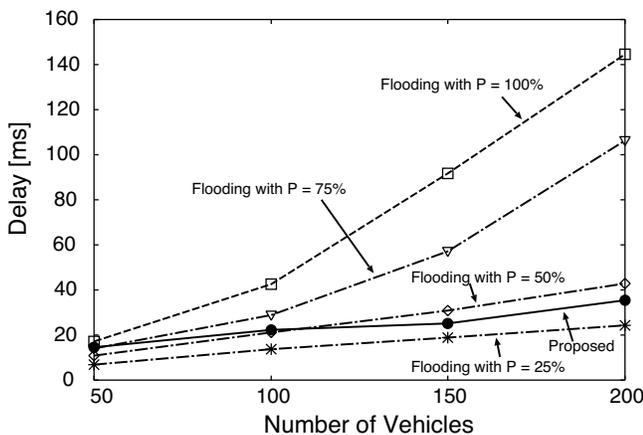


Fig. 24. Delay performance (Big size vehicle:40[%]).

flooding probability. This is because broadcast storms occur and it is difficult for almost all vehicles to transmit vehicle information. On the contrary, the proposed protocol can keep the short delay even if the number of vehicle increases.

Figure 25 shows the continuous drop ratio of vehicle information with a large-size vehicle ratio equals to 40 [%] and 200 vehicles. The continuous drop ratio means the ratio that the vehicle cannot receive the vehicle information continuously. The continuous drops of vehicle information are not suited characteristics for ITS communication because these drops cause temporal interruption of communication between neighbor vehicles. From results, we can find that the proposed protocol has good tolerance to burst packet losses.

V. CONCLUSION

In this paper, we have proposed a new routing protocol for delivery of vehicle information to neighbor vehicles in a specific area. The proposed protocol can deliver new vehicle information with short delay by performing temporal limited flooding before a construction of routes. Moreover, it can

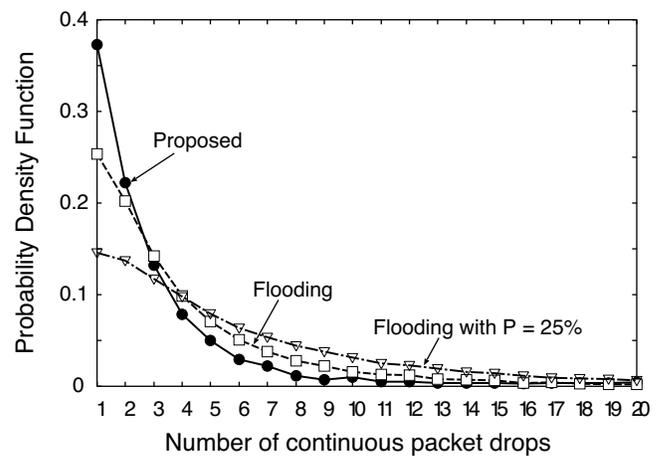


Fig. 25. Continuous drop ratio of vehicle information.

deliver vehicle information effectively with forwarding by an adequate vehicle. The feature of the protocol is utilizing a vehicle information message itself to detect each vehicle status. Moreover, our protocol can be extended for a bidirectional road by using directional information. As a result, our protocol does not require periodic transmission of control messages. In addition, we have evaluated an environment with the mixed factor of standard-size and large-size vehicles. In the actual environment, it is important to support this mixed factor for real safe driving systems. Finally, we can find that our protocol can achieve the high delivery ratio with short delay even if large-size vehicles influence the communication. Moreover, we can provide required quality in communications if we employ the forward error correction (FEC) to recover the packet loss. Considering all these results mentioned above, the proposed method could be one of the fundamental schemes for achieving ITS.

VI. FUTURE WORK

In this paper, we evaluated the performance with two sizes of vehicles in additive white gaussian noise (AWGN) environment. Therefore, our evaluation can assume more actual vehicle conditions and wireless communication environment. However, multi-path fading is also significant degradation factor in city environment. Then, it is important to handle the dynamic fluctuation of wireless channel. Moreover, the proposed scheme was evaluated with the IEEE 802.11b system. Therefore, it is not difficult to implement on embedded system with IEEE 802.11 device. Authors has a schedule to implement the proposed scheme on a Linux router board with a mini-PCI IEEE 802.11device.

ACKNOWLEDGMENT

This work was supported by Grant-in-Aid for Young Scientists (B)(20700059), Japan Society for the Promotion of Science (JSPS).

REFERENCES

- [1] K. Naito, K. Sato, K. Mori, and H. Kobayashi, "Proposal of Distribution Scheme for Vehicle Information in ITS Networks," IARIA The Eighth International Conference on Networks (ICN 2009) Mar. 2009.
- [2] O. Andrisano, R. Verdone, and M. Nakagawa, "Intelligent transportation systems: the role of third generation mobile radio networks," IEEE Communications Magazine, vol. 38, no. 9, pp. 144–151, Sep. 2000.
- [3] M. Rudack, M. Meincke, K. Jobmann, and M. Lott, "On traffic dynamical aspects inter vehicle communication(IVC)," IEEE Vehicular Technology Conference (VTC03 Spring), Apr. 2003.
- [4] C. Dermawan and A. Sugiura, "Simulation of Bluetooth Wireless Communication for ITS," IEICE Transactions on Communications, vol. E86-B, no. 1, pp. 66–67, Jan. 2003.
- [5] S. Y. Wang, "On the effectiveness of distributing information among vehicles using inter-vehicle communication," IEEE Intelligent Transportation Systems 2003, vol. 2, no. 12–15, pp. 1521–1526, Oct. 2003.
- [6] F. Gil-Castineira, F.J. Gonzalez-Castano, and L. Franck, "Extending Vehicular CAN Fieldbuses With Delay-Tolerant Networks," IEEE Transactions on Industrial Electronics, Vol. 55, No. 9, pp. 3307–3314, Sep. 2008.
- [7] C. Sommer and F. Dressler, "The DYMO Routing Protocol in VANET Scenarios," IEEE Vehicular Technology Conference, 2007 (VTC-2007 Fall), pp. 16–20, Oct. 2007.
- [8] Y. Toor, P. Muhlethaler, A. Laouiti, and A. de La Fortelle, "Vehicle Ad Hoc networks: applications and related technical issues," IEEE Communications Surveys and Tutorials, Quarter 2008, vol. 10, no 3, p. 74–88, 2008.
- [9] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC 2002), pp. 194–205, Jun. 2002.
- [10] J. Bernsen and D. Manivannan, "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification," ACM Pervasive and Mobile Computing, Vol. 5 , No. 1, pp. 1–18, Feb. 2009.
- [11] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF Request for Comments 3561, Jul. 2003.
- [12] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," IETF Request for Comments 4728, Feb. 2007.
- [13] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," IEEE Vehicular Technology Magazine, vol. 2, no. 2, pp. 12–22, Jun. 2007.
- [14] R.A. Santos, A. Edwards, R.M. Edwards, and N.L. Seed, "Performance evaluation of routing protocols in vehicular ad-hoc networks," International Journal of Ad Hoc and Ubiquitous Computing, vol. 1, no. 1-2, pp. 80–91, 2005.
- [15] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer, "Position-aware ad hoc wireless networks for inter-vehicle communications: the Fleetnet project," ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC 2001), pp. 259–262, Oct. 2001.
- [16] Z. Mo, H. Zhu, K. Makki, and N. Pissinou, "MURU: A Multi-Hop Routing Protocol for Urban Vehicular Ad Hoc Networks," International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS 2006), Jul. 2006.
- [17] F. Granelli, G. Boato, and D. Kliazovich, "MORA: a Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks," IEEE Workshop on Automotive Networking and Applications (AutoNet 2006), Dec. 2006.
- [18] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," ACM International Conference on Mobile Computing and Networking (MobiCom), pp. 243–254, 2000.
- [19] V. Naumov and T.R. Gross, "Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks," IEEE International Conference on Computer Communications (INFOCOM 2007), pp. 1919–1927, May 2007.
- [20] J. Wu, "Dominating-set-based routing in ad hoc wireless networks," Wiley Series On Parallel And Distributed Computing, pp. 425–450, 2002.
- [21] T.D.C. Little and A. Agarwal, "An information propagation scheme for VANETs," IEEE Intelligent Transportation Systems (ITSC 2005), pp. 155–160, Sep. 2005.
- [22] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," Wireless Networks, vol. 8, pp. 153–167, 2002.
- [23] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," IEEE Transactions on Mobile Computing, vol. 1, no. 2, pp. 111–122, 2002.
- [24] I. Stojmenovic, M. Seddigh, J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 1 pp. 14–25, 2002.
- [25] Y.-C. Tseng, S.-Y. Ni, E.-Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," IEEE Transactions on Computers, vol. 52, no. 5, pp. 545–557, May 2003.
- [26] C. Maihofer, "Survey of Geocast Routing Protocols," IEEE Communications Surveys & Tutorials, vol. 6, no. 2, pp. 32–42, Q2 2004.
- [27] M. T. Sun, W. C. Feng, T. H. Lai, K. Yamada, H. Okada, and K. Fujimura, "GPS-based message broadcasting for inter-vehicle communication," International Conference on Parallel Processing 2000, pp. 279–286, 2000.
- [28] C. H. Chou, K. F. Ssu, and H. C. Jiau, "Geographic Forwarding With Dead-End Reduction in Mobile Ad Hoc Networks,"
- [29] G. Korkmaz, E. Ekick, F. Ozguner, and U. Ozguner, "Urban Multi-Hop Broadcast Protocol for Inter-Vehicle Communication Systems," ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004), Oct. 2004.
- [30] E. Fasolo, A. Zanella, and M. Zorzi, "An Effective Broadcast Scheme for Alert Message Propagation in Vehicular Ad hoc Networks," IEEE International Conference on Communications (ICC '06), pp. 3960–3965, Jun. 2006.
- [31] QualNet, URL:<http://www.scalable-networks.com>

Escrow Serializability and Reconciliation in Mobile Computing using Semantic Properties

Fritz Laux
Fakultät Informatik
Reutlingen University
D-72762 Reutlingen, Germany
fritz.laux@reutlingen-university.de

Tim Lessner
School of Computing
University of the West of Scotland
Paisley PA1 2BE, UK
timlessner@lesshome.net

Abstract

Transaction processing is of growing importance for mobile computing. Booking tickets, flight reservation, banking, ePayment, and booking holiday arrangements are just a few examples for mobile transactions. Due to temporarily disconnected situations the synchronisation and consistent transaction processing are key issues. Serializability is a too strong criteria for correctness when the semantics of a transaction is known. We introduce a transaction model that allows higher concurrency for a certain class of transactions defined by its semantic. The transaction results are "escrow serializable" and the synchronisation mechanism is non-blocking. The model copes with many mobile scenarios and is able to improve existing synchronization approaches through an automatic replay approach, whereas transaction migration or transactional composition in mobile interaction is not considered. Rather we provide an optimistic transaction model residing at middleware layer. Experimental implementation showed higher concurrency, transaction throughput, and less resources used than common locking or optimistic protocols.

1. Introduction

Mobile applications enable users to execute business transactions while being on the move. It is essential that online transaction processing will not be hindered by the limited processing capabilities of mobile devices and the low speed communication. In addition, transactions should not be blocked by temporarily disconnected situations. Traditional transaction systems in LANs rely on high speed communication and trained personnel so that data locking has proved to be an efficient mechanism to achieve serializability.

In the case of mobile computing neither connection quality or speed is guaranteed nor professional users may be assumed. A reliable end-to-end protocol (ISO/OSI level 4) like TCP is not sufficient as a user transaction (ISO/OSI level 7) may span multiple sessions. The communication delay due to retransmissions occupies resources e.g. blocks

data elements. This means that a transaction will hold its resources for a longer time, causing other conflicting transactions to wait longer for these data. If a component fails, it is possible that the transaction blocks (is left in a state where neither a rollback nor a completion is possible).

The usual way to avoid blocking of transactions is to use optimistic concurrency protocols.

In situations of high transaction volume the risk of aborted transaction rises and the restarted transaction add further load to the database system. Also this vulnerability could be exploited for denial of service attacks.

In order to make mobile transaction processing reliable and efficient a transaction management is needed that does not only avoid the drawbacks outlined above but also fits well into established or emerging technologies like EJB, ADO, SDO. Such technologies enable weakly coupled or disconnected computing promoting Service Oriented Architectures (SOA).

These data access technologies basically provide abstract data structures (objects, data sets, data graphs) that encapsulate and decouple from the database and adapt to the programming models. We propose a transaction mechanism that should be implemented in the middle tier between database and (mobile) client application. This enables to move some application logic from the client to the application server (middle tier) in order to relief the client from processing and storage needs. Validation, eventual transaction rewrites, reconciliations or compensations are implemented in the middle tier as shown in Figure 1. A client transaction T_1 executes entirely locally after loading the read set $RSet^1$ into the client. On commit the middleware has to check $RSet^1$ for possible changes which happened in the mean time due to other transactions e.g. T_2 using the write set $WSet^2$. In case of serialization conflicts the transaction manager has to resolve the situation. If there are legacy applications not running through this middleware the consolidation must take into account the current database state S^s as well.

The present paper is an extended version of [1], and it provides more detailed information about the server phase, the requirements for transaction splitting, and other

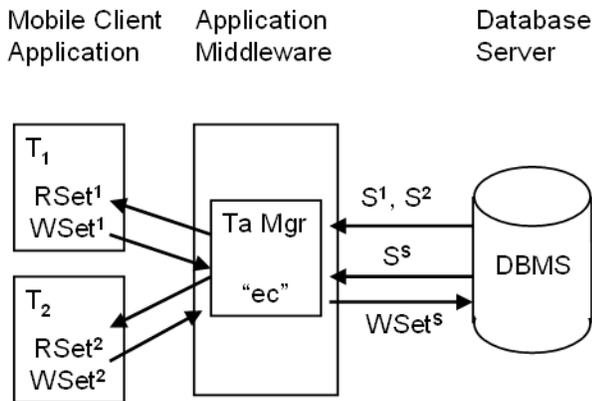


Figure 1. Three tier architecture for mobile transaction processing

implementation issues.

1.1. Motivation

The main differences between mobile computing and stationary computing are temporary loss of communication and low communication bandwidth. However increased local autonomy is required at the same time. Data hoarding and local processing capability are the usual answers to achieve local autonomy. The next challenge is then the synchronisation or reintegration of data after processing [2], [3], [4]. As pointed out above, blocking of host data is not an option.

The challenge is to find a non-blocking concurrency mechanism that works well in disconnected situations and that is not leading to unnecessary transaction cancellations.

We need a mechanism to reconcile conflicting changes on the host database such that the result is still considered correct. This is possible if the transaction semantic is known to the transaction management. In this paper we propose to automatically replay the transactions in case of a conflict.

We illustrate the idea by an example and defer the formal definition to the next Section. Assume that we have transactions T_1 and T_2 that withdraw €100 and €200 respectively from account a . If both transactions start reading the same value for a (say €1000) and then attempt to write back $a := €900$ for T_1 and $a := €800$ for T_2 then a serialization conflict arises for the second transaction because the final result would lead to a lost update of the first transaction.

However, if in this case the transaction manager aborts the transaction, re-reads a (= €900 now) and does the update on the basis of this new value then the result (= €700) would be considered as correct. In fact, it resulted in a serial execution from the host's view. Clearly this transaction replay is only allowed if it is known that the second transaction's subtract value does not depend on the account value (balance). This

precondition holds within certain limits for an important class of transactions: Booking tickets, reserving seats in a flight, bank transfers, stock management.

There are often additional constraints to obey: A bank account balance must not exceed the credit limit, the quantity on stock cannot be negative, etc.

We will introduce a transaction model based on this idea that allows higher concurrency for a certain class of transactions defined by its semantic. The transaction results are "escrow serializable" and the synchronisation mechanism is non-blocking.

The next section sketches out the related work and addresses some drawbacks of existing approaches. Section 3 and 4 introduce our model and provide the required definitions for escrow serializability as well as the transactions' semantics. Section 5 describes theoretically the client and server phase in detail, whereas section 6 provides information about an implementation based on Service Data Objects (SDO). Section 7 focuses on an alternative conflict detection using Row Version Verification (RVV) and the performance of the escrow model is presented in section 8. The paper's conclusions are presented in section 9.

2. Related Work

For making transaction aborts as rare as possible essentially three approaches have been proposed:

- Use the semantic knowledge about a transaction to classify transactions that are compatible to interleave.
- Divide a transaction into subtransactions.
- Reconcile the database by rewriting the transaction in case of a conflict.

Semantic knowledge of a transaction allows non serializable schedules that produce consistent results. Garcia-Molina [5] classifies transactions into different types. Each transaction type is divided into atomic steps with compatibility sets according to its semantic. Transaction types that are not in the compatibility set are considered incompatible and are not allowed to interleave at all. Farrag and Özsu [6] refine this method allowing certain interleaving for incompatible types and assuming fewer restrictions for compatibility. The burden with this concept is to find the compatibility sets for each transaction step which is a $O(n^2)$ problem. Our proposed model is a $O(n)$ problem, because for each operation of a transaction it has to be decided if the operation is reconcilable or not, and it is not required to define the compatibility with every concurrent transaction.

Dividing transactions into subtransactions that are delimited by breakpoints does not reduce the number of conflicts for the same schedule but a partial rollback (rollback to a subtransaction) may be sufficient to resolve the conflict. Huang and Huang [7] use semantic based subtransactions and a compatibility matrix to achieve better concurrency

Table 1. Comparison of high concurrency mechanisms

| Mechanism | Bibliography | Drawbacks |
|--|-----------------------|--|
| uses Ta semantics to build compatibility set | [5], [6] | semantic classification complexity is $\mathcal{O}(n^2)$ |
| uses subtransactions to build compatibility matrix | [7], [8], [9] [10] | manual division into sub-Ta, $\mathcal{O}(n^2)$ |
| uses multiversions and conflict resolution function | [11], [3] | not performant in case of hot spots |
| uses semantic to reconcile Ta (escrow-serializability) | [13], [1] | semantic dependency function required |

for mobile database environments. Local autonomy of the clients may subvert the global serializability. The solutions proposed by Georgakopoulos et al. [8] and Mehrotra et al [9] came for the prize of low concurrency and low performance. Huang, Kwan, and Li [10] achieved better concurrency by using a mixture of locking to ensure global ordering and a refined compatibility matrix based on semantic subtransactions. Their transaction mechanism still needs to be implemented in a prototype to investigate its feasibility. The reconciliation mechanism proposed in this paper attempts to replay the conflicting transactions and produce a serializable result. This method has been investigated in the context of multiversion databases. Graham and Barker [11] analysed the transactions that produced conflicting versions. Phatak and Nath [3] use a multiversion reconciliation algorithm based on snapshots and a conflict resolution function. The main idea is to compute a snapshot for each concurrent client transaction which is consistent in terms of isolation and leads to a least cost reconciliation. The standard conflict resolution function integrates transactions only if the read set $RSet$ of the transaction is a subset of the snapshot version $S(in)$ into which the result needs to be integrated. In the case of write-write conflicts this is not the case, as $RSet \not\subseteq S(in)$.

We illustrate this by an example using the read-write model with Herbrand semantics (see [12]). Assume we have two transaction: $T_1 = (r_1(a), w_1(a), r_1(b), w_1(b))$ transfers €100 from account a to account b and $T_2 = (r_2(b), w_2(b))$ withdraws €100 from account b . If both transactions are executed in serial, the balance for account b will end up with its starting value. Now assume, that snapshot version $V(0) = \{a^0, b^0\}$ is used and both transactions start with the same value b^0 . Assume the schedule $S = (r_1(a^0), r_2(b^0), r_1(b^0), w_1(a^1), w_2(b^1), c_2, w_1(b^1), c_1)$. S is not serializable and no other schedule either if both transactions use the same version of b . The last transaction attempting to write account b will produce a lost update and should abort.

The multiversion snapshot based reconciliation algorithm of Phatak and Nath [3] will not be able to reconcile T_1 as $RSet(T_1) = V(0) = \{a^0, b^0\} \not\subseteq V(1) = \{a^0, b^1\}$. $V(1)$ is the result of transaction T_2 . If no snapshot would

have been taken and making sure that the update (read-write sequence) of b is not interrupted (interleaved) the result would have been the serializable schedule $R = (r_1(a), w_1(a), r_2(b), w_2(b), c_2, r_1(b), w_1(b), c_1)$. This shows the limitations of snapshot isolation compared to locking in terms of transaction rollbacks. On the other hand the schedule R leads to low performance because no interleaving operations for the read-write sequence are allowed. Table 1 gives an overview on transaction mechanisms used to reduce or resolve concurrency conflicts.

Many mobile replication and synchronization models are introduced in literature. Some of these models could be improved by an application of the *ec* - model. The Isolation Only (IO) [14] for example, enables disconnected operations in a private workspace and distinguishes between 1^{st} and 2^{nd} class transactions, whereas only the 1^{st} class is serializable with all committed transactions (SR). The 2^{nd} class is only local serializable with other 2^{nd} class transactions (LSR). SR is granted if a local transaction was successfully reintegrated (LSR \rightarrow SR). IO defines global serializability (GSR) to be the next level of serializability, and the difference between LSR, SR, and GSR is that GSR is not testable during a transaction's execution. If a test for GSR fails, the IO model also proposes to re-execute a transaction with the current DB's state. However, the IO model doesn't define the types of conflicts where re-execution is applicable. The *ec*-model is capable to extend the IO model since our model provides a semantics based classification for conflicts and a mechanism to automatically replay conflicting transactions. Furthermore, the IO model is based on Kung's OCC model to ensure SR (see [15]), and in section 5 we present how to extend an OCC with an additional reconcile phase. So the *ec* - model is able to improve GSR and SR in the IO model. The idea to replay a transaction on a stationary DBMS is referred to as transaction oriented synchronization and is described in the Two - Tier - Replication model ([16]). Reconciliation is based on the transaction's semantics. Our model focuses on the server or middleware and the transaction's semantics has to be made available for the server only, i.e. the TM. If local transactions have to be aware of the semantics, because the *ec*-model is applied to local DBMS, a "Combat" mechanism as introduced in the Pro-Motion transaction model ([17], [18]) is a proper solution.

Our approach is to abort a conflicting transaction and automatically replay the operation sequentially. The isolation level should be read committed to avoid cascading rollbacks or compensation transactions. Our model relies on the optimistic snapshot validation without critical section [19] algorithm or row version verifying (RVV) [20], and to ease the reconciliation processing we classify transaction in terms of its semantics.

3. Transaction Model

A database D may be viewed as a finite set of entities or elements a, b, \dots, x, y, z (see [21]). If there exists more than one version of an entity, we denote it with the version number, e.g. x^2 . These entities will be read (read set $RSet$) and modified (write set $WSet$) by a set of transactions $\mathbf{T} = \{T_1, T_2, \dots, T_n\}$. The database D at any given time exists in a particular state D^S . A snapshot of D is a subset of a database state D^S (see [22]).

Our mobile computing system consists of a database server, an application middleware with mobile transaction management, and a mobile client with storage and computing capabilities as sketched out in Figure 1. A mobile transaction is a distributed application that guarantees transactional properties. We assume that the data communication is handled transparently by a communication protocol that can detect and recover failures on the network level. Mobile client and server have some local autonomy so that in case of network disconnection both sites can continue their work to some extent.

The data base consists of a central data store and snapshot data (at least the $RSet$) on the mobile client for each active transaction. From a transactional concept's view the transactions on the client are executed under local autonomy. The local commit is "escrowed" along with the changes to the server. The transaction manager tries to integrate all escrowed transactions into the central data store. In case of serialization conflicts reconciliation can be achieved if the semantic of the transaction is known and all database constraints are obeyed.

3.1. Escrow Serializable

For the sake of availability we want to avoid locked transaction as far as possible. One approach is to use optimistic concurrency, the other way is to relax serializability. Optimistic concurrency suffers from transaction aborts when a serialization conflicts arises [22]. The multiversion based view maintenance could minimize that risk but it requires a reliable communication at all times [3].

Much research was invested to optimize the validation algorithms [23], [24], [25], [26] for serialization. We prefer to allow non-serializable schedules that produce consistent results for certain types of transactions similar to [27].

A transaction T transforms a consistent database state into another consistent state. This may be formalized by considering a transaction as a function operating on a subset of consistent database states \mathbf{D} , i.e. $D^2 = T(D^1)$ with suitable $D^1, D^2 \in \mathbf{D}$, where $RSet \subseteq D^1$ and $WSet \subseteq D^2$. If we want to make the user input u explicit we write $D^2 = T(D^1, u)$.

Definition 1: (escrow serializable)

Let Q be a history of a set of (client) transactions $\mathbf{T} = \{T_1, T_2, \dots, T_n\}$ that are executed concurrently on a database D with initial state D^0 . For each transaction T_i the user input is denoted by u_i . The history Q is called *escrow serializable* (ec) if

- 1) there exists a serial history S for \mathbf{T} with committed database states $\mathbf{D}^S = (D^1, D^2, \dots, D^n)$, where
- 2) $\exists r \in \{1, 2, \dots, n\}$ with $D^1 = T_r(D^0, u_r)$ and
- 3) $\exists s \in \{1, 2, \dots, n\}$ with $D^k = T_s(D^{k-1}, u_s)$ for each $k = (2, 3, \dots, n)$

Please note that this kind of serializability is descriptive as it is not based on the operations but on the outcome (semantic) of the transactions. Escrow serializability means that the outcome is the same as with a serial execution using the same user input.

The name *escrow serializable* stems from the idea that a mobile client "escrows" its transaction to the server. On the server site the transaction manager reconciles the transaction if all database constraints are fulfilled. This can be achieved by analysing the conflicting transactions and producing the same result as a serial execution would have done. We demonstrate this with the following example:

Example 1: (withdraw)

Let T_1 and T_2 be two withdraw transactions that takes € 100 resp. € 200 from account x . We denote by c_i (resp. a_i, ec_i) the commit (resp. abort, escrow commit) command. The history

$$S^c = r_1(x)r_2(x)w_2(x' := x - 200)ec_2w_1(x'' := x - 100)ec_1$$

normally produces a lost update, but it is escrow serializable. The transaction manager on the server will detect the conflicting transaction. T_1 is aborted and automatically replayed with the previous input data. The resulting history on the server will be

$$S^s = r_1(x)r_2(x)w_2(x' := x - 200)c_2w_1(x - 100)a_1r_1(x')w_1(x' - 100)c_1.$$

Schedule S^s is equivalent to the serial execution (T_2, T_1) .

If there exists a constraint, say $x > 0$ any violating transaction has to abort. Assume that $x = 300$ and take the same operation sequence as in schedule S^c then transaction T_1 has to abort because $x' - 100 \not\geq 0$.

3.2. Escrow Reconciliation Algorithm

Escrow serialization relies on reconciling transaction in such a way that the outcome is serializable. This is only possible if the semantic of the transactions including the user input are known. The idea is to read all data necessary for a

```

ensure: set of transactions  $\mathbf{T} = \{T_1, T_2, \dots, T_n\}$ 
ensure: actual database state  $D^s$ , set of constraints  $C(D)$ 
ensure: only committed data in read set  $RSet(i)$  of  $T_i$ 
ensure:  $T_i = (op_{ik}, i = 1, 2, \dots, k_i)$ 
for  $\forall ec_i \in \{ec_1, ec_2, \dots, ec_n\}$  received do
  // test if  $T_i$  conflicts with  $D^s$ 
  if  $RSet(i) \subseteq D^s$  and  $\forall c \in C(D): (c = true)$  then
    commit  $T_i$ 
  else // abort and replay transaction
    abort  $T_i$ 
    ensure: serial execution
    for each  $op_{ik} \in op(T_i)$  do  $op_{ik}$ 
    if  $\exists c \in C(D)$  with  $(c = false)$  then // c violated
      abort  $T_i$ 
    else
      commit  $T_i$ 
    end if
  end for
end for

```

Figure 2. Reconciliation algorithm for ec serializability

transaction and defer any write operation until commit time. If a serialization conflict arises at commit time this means that a concurrent transaction has already committed. In this case the transaction is aborted and automatically replayed with the same input data.

Our transaction model is divided into two phases:

- **client phase**

During the processing on the client site, data may only be retrieved from the server. It is important that the read requests are served in an optimistic way. Technically a read set of data, a data graph or any other snapshot could be delivered to the mobile client. The client transaction terminates with an escrow commit (ec) or an abort (a).

- **server phase**

When the server receives the ec along with the write set and no serialization conflict exists the transaction is committed. In case of a conflict the transaction is aborted. The replay is done automatically with pessimistic concurrency control or serial execution. This prevents nested transactions conflicts or the starvation [22] of a transaction. If no constraints are violated the replayed transaction is committed.

A possible reconciliation algorithm using the abort-replay mechanism is presented in Figure 2. Section 5 describes the server phase in detail and provides an extended version of this algorithm.

Care has to be taken with transactions not using the abort-replay mechanism. In this case the database should work in isolation level "serializable".

If the abort-replay mechanism is always used to integrate the transactions on the server there is no need for a certain isolation level as the read sets only contain consistent results.

Any competing transactions will not alter the database until the server integrates the result. As the transaction results are integrated one-by-one, no read phenomena may occur and serial results are ensured.

So far we have illustrated the model with transactions that produce a constant change for a data item (see Example 1). The model is valid for any transaction with a known semantic (see Theorem 1). For instance the transaction $T_3 = (r_3(x), w_3(x := 1.1x), ec_3)$ increases the prize x of a product by 10%. If the first read of x and the reread differ ($r'_3(x) \neq r_3(x)$), then the replay will produce a 10% increase based on the actual value.

For an automatic replay it is essential to know which transactions are "immune" or depend in a predicted manner from the read set. These are the candidates for escrow serializability.

There is a technical issue for the banking example. Here we do not really need the actual withdraw amount of the transaction to replay it. It is sufficient to know three database states since the new value can be calculated by $a := a^1 + a^c - a^0$ where a^1 is the actual balance, a^c is the new balance calculated by the client transaction, and a^0 is the basis on which the value a^c was computed. This observation gives reason to find classes of transactions that are ec serializable without knowing the actual user input.

An implementation using SDO technology is described later in Section 6.

4. Semantic Classification of Transactions

To facilitate the task for the reconciliation algorithm we shall classify the client transaction according to their semantic, in particular the dependency of the input from the read set.

Definition 2: (dependency function)

Let T be a transaction with $RSet = \{x_1, x_2, \dots, x_n\}$ and $WSet = \{y_1, y_2, \dots, y_m\}$ on a database D . The function $f_i : \vec{x} \rightarrow y_i$ with $\vec{x} = (x_1, x_2, \dots, x_n)$ and $y_i \in WSet$ is called *dependency function* of y_i .

Let $x_k \in RSet$ and $y_i \in WSet$ be numeric data types for all k . If f_i is a linear function then y_i is called *linear dependent* and we can write

$$y_i = f_i(\vec{x}) = \vec{a}_i^T \vec{x} + c_i \quad (i = 1, 2, \dots, m) \quad (1)$$

with \vec{a}_i^T being the transposed vector $a_i = (a_{i1}, a_{i2}, \dots, a_{in})$.

If all functions f_i are linear dependent, then

$$\vec{y} = A\vec{x} + \vec{c} \quad (2)$$

with $m \times n$ -Matrix $A = (a_{ik})$ and m -dimensional vector \vec{c} . We call the corresponding transaction T *linear dependent*.

If $f_i(\vec{x}) = \vec{1}^T \vec{x} + c_i$ then f_i is called *linear dependent with gradient 1* ($\vec{1}$ is the vector with magnitude 1).

If $f_i(\vec{x}) = \vec{b}^T \vec{x} + c_i$ then f_i is called *linear dependent with gradient \vec{b}* .

In our banking example the accounts are linearly dependent with gradient 1. The 10% price increase is an example for a transaction that is linearly dependent with gradient $b = 1.1$.

If the values of the *WSet* however depend in a non-formalized user dependent manner from the *RSet* then there is no way to reconcile the transaction automatically. The escrow serializable execution of a transaction depends on the fact that the outcome does change in a known functional manner.

Theorem 1: (escrow serializable)

Let \mathbf{T} be a set of transactions where each transaction T has known dependency functions f_i ($i = 1, 2, \dots, m$). Then the concurrent execution of \mathbf{T} is escrow serializable using the abort-replay algorithm of Figure 2.

Proof 1: (escrow serializable)

Let D^0 be a consistent state of a database with transactions $\mathbf{T} = \{T_1, T_2, \dots, T_n\}$. Let H be a history of \mathbf{T} and let w.l.o.g. the commit order be the same as the transaction index. We construct a serial transaction order that matches the definitions of escrow serializability using the abort-replay algorithm. Any write operations of the transactions T_i are postponed until commit time. The read set of T_1 is a subset of database state D^0 . Then we have $D^1 = D^0 \cup S^1 := T_1(D^0)$ after the first commit c_1 . When a subsequent transaction T_k tries to commit and $RSet_k \cap (\cup_{\kappa \leq k-1} S^\kappa) = \emptyset$ then there is no serialization conflict and the commit succeeds. In case of a conflict, the transaction is aborted and replayed with the same user data. During the replay the algorithm ensures serial execution, so further commits are queued. Finally we have $D^k = D^{k-1} \cup T_s(D^{k-1}, u_s)$ for $k = 1, 2, \dots, n$. QED

Let $\{r_1, r_2, \dots, r_n\} \subseteq D^c$ be the read set values of a client transaction and let $\{s_1, s_2, \dots, s_n\} \subseteq D^s$ be the read set values on the server when the transaction tries to commit. Then the abort-replay mechanism produces $WSet(T) = T(D^s, \vec{u}) = A\vec{s} + \vec{u}$. The value of any numerical data item $x \in WSet$ for a linear dependent transaction is computed

as

$$\begin{aligned} x^T &= \Pi_x T(D^s, \vec{u}) = \vec{a}^T \vec{s} + u \\ &= \vec{a}^T \vec{s} + (\vec{a}^T \vec{r} + u) - \vec{a}^T \vec{r} \\ &= \vec{a}^T (\vec{s} - \vec{r}) + \Pi_x D^c \\ &= \Pi_x A(\vec{s} - \vec{r}) + \Pi_x D^c \end{aligned} \quad (3)$$

From the above equation we see that the reconciliation for transactions with a linear dependent write set may be simplified. For the transaction manager it is sufficient to know the client state D^c , the read set D^s at commit time and the state produced by $T(D^c, u)$.

Corollary: A linear dependent transaction can be reconciled (replayed) in a generic way, if client state D^c at begin of transaction, the read set D^s at commit time and the state produced by $T(D^c, u)$ are known.

The corollary statement is similar to the reconciliation proposed by Holliday, Agrawal, and El Abbadi [4].

4.1. Quota Transaction

In many cases the semantic of a transaction has well known restrictions. We can guaranty the successful execution of certain transactions if the user input remains within a certain value range.

Assume a reservation transaction. If the transaction is given a quota of q reservations then the success can be guaranteed for reservations within these limits. It is the responsibility of the transaction manager to ensure that the quota does not violate the consistency constraints. For example if there are 10 tickets left and the quota is set for 2 tickets, then only 5 concurrent transactions are allowed. As soon as a transaction terminates with less than two reservations the transaction manager may allow another transaction to start with a quota that ensures no overbooking.

Quota transactions in this sense are similar to increment or decrement of counter transactions with escrow locking (see [12]).

Definition 3: (quota transaction)

Let T be a transaction with $WSet = \{y_1, y_2, \dots, y_m\}$ on a database D . For each y_i there is a value range $I := [l, u]$ associated. T is called *quota transaction* if the success of the transaction can be guaranteed in advance if the result values y_i do not exceed the quota, i.e. $y_{i(old)} + l \leq y_{i(new)} \leq y_{i(old)} + u$.

Setting quotas is a mean to guaranty success for a transaction by reserving sufficient resources without locking the resources. Caution has to be taken when using quotas as resources are reserved that finally should be taken or given back. Therefore a time out or a cancel operation is required on the server site.

4.2. A transaction's role

We will briefly describe the idea how to apply the role pattern on transactions. Consider the situation where a transaction has to prevail against other transactions' modifications. If validation fails (e.g. a constraint was violated) there is no chance for a transaction with a higher priority to prevail against concurrent transactions. However, if we assign an owner or master role to that transaction, the TM is able to detect the role and adapt the transaction's handling. In the case of an owner, he may write modifications of the owner transaction regardless of any other transactions, and conflicting transactions have to abort.

In general, roles are a well understood concept, but transaction models do not apply this concept directly to transactions. Instead, the concept is shifted up to application level whereas our intention is to apply a role directly to the transactions. Generally, a role is represented by a logical identifier, and a set of conditions reflect the roles' intention, whereby each condition leads to activities. The role model could be implemented based on the ECA (Event, Condition, Activity) concept of Active Database Management Systems (see [28]). The event is thrown, if the TM detects a role associated with the current transaction. The conditions are validated, and the activities are executed. In our model an activity, thrown by a role, affects the TM's behaviour which isn't data centric as the ECA is.

Security aspects may complement, add or even contradict the activities implied by the transaction's role. Care has to be taken if transactions with an identical role operate on the same data, and they run into a deadlock situation, e.g. two owner roles. If this is an unwanted state, the first role is a semantic lock for other identical roles.

5. Server phase of the *ec*-model

The sections above focus on reconcilable elements, i.e. values with a linear dependency function. But, in general a transaction consists of non-reconcilable elements (not able to get corrected), too. E.g. a customer name is non-reconcilable (provided no dependency function is found). On the other hand an account balance, as in the example above, is reconcilable. Both kinds of elements may belong to the same transaction.

Reconciliation prevents only reconcilable elements from unnecessary conflicts. Therefore, we will apply the escrow model to an optimistic concurrency control (OCC) algorithm in order to handle non reconcilable elements, too. Kung and Robinson [15] describe in their paper a general approach for an OCC algorithm with three different phases; The read, validation, and write phase. The obstacle of this approach is the critical section namely validation and write. The indivisibility of these two phases leads transactions in their read phase to also interrupt their work

if other transactions are validating or writing. Unland [19] suggest a validation without critical section VAL^{-CS} . Read transactions together with currently writing transactions can validate concurrently except for a short critical section when a transaction number (counter access only) is generated and assigned. The VAL^{-CS} is extended by an additional reconcile phase in order to comply with reconcilable elements.

The VAL^{-CS} defines a transaction to be either in the (i) read, (ii) validation or (iii) write phase. The additional reconcile phase is explained later (see figure 3 for an example). During the read phase (i) a transaction has to validate against all transactions terminating in this phase (write). This is referred to as forward-oriented optimistic concurrency control [12].

If a transaction enters the validation phase (ii) a transaction number TNR is assigned (the only critical phase), and the transaction has to validate against all transactions with a smaller transaction number not finished validation yet.

In the write phase (iii) T_i 's result is published provided the write succeeds.

The *ec*-model needs an additional reconcile phase for values necessary to reconcile. A validation order within a transaction is also indispensable, since some elements need validation only, and reconcilable elements need reconciliation (see later). Generally, OCC algorithms verify that a $RSet$ of a transaction T have no intersection with another transaction's $WSet$. The intersection is determined on an entity basis, but reconciliation is on a value basis. Recall, reconciliation means to re-execute an operation with the current value(s). Thus, for non reconcilable elements validation on an entity basis is proper, whereas reconciliation is on a value basis. Therefore, a transaction may need two different validation strategies. Furthermore, we have reconcilable values with a constraint, e.g. an account is not allowed to fall below the credit limit.

Based on the observations so far we define the following (Definition 5).

Definition 4: (escrow transaction)

A transaction consists of 1) reconcilable elements $\vec{x} = (x_1, \dots, x_i)$ (see definition 2), 2) reconcilable elements with a constraint $\vec{cx} = (cx_1, \dots, cx_i)$, and 3) non reconcilable elements $\vec{nx} = (nx_1, \dots, nx_i)$. For each reconcilable value x_i and cx_i the dependency function f_i of y_i is known (see definition 2). The userinput is denoted by u .

- (1) $(\vec{y}, \vec{cy}, \vec{ny}) = T(\vec{x}, \vec{cx}, \vec{nx}, u)$
- (2) Validation of \vec{x} and \vec{cx} is on an element basis, whereas \vec{nx} is on a variable basis.
- (3) $RSet(T) = \{\vec{x}, \vec{cx}, \vec{nx}\}$
- (4) $WSet(T) = \{\vec{y}, \vec{cy}, \vec{ny}\}$

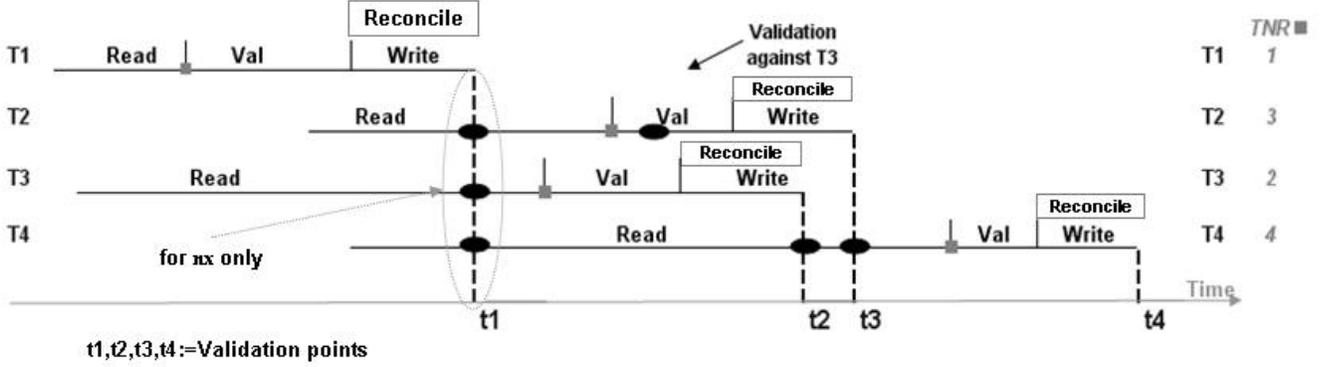


Figure 3. Modified validation without critical section (VAL^{-CS}), see [19]

$T2, 3, 4$ have to validate against $T1$ because $T1$ writes ($t1$). $T2$ has to validate against $T3$, because $T2$'s TNR < $T3$'s TNR.

Example 2: (product inventory) Several employees receive new products. Each product is stored at exactly one storage location and employees store products concurrently. Before an employee starts to stock, he reads the product's data ($id, location, quantity$) via infrared using his mobile device. An infrared access point with a limited coverage resides near each product's location. After an employee finishes work he commits the changes on the mobile device, and sends his modifications ($WSet$ or change set) back to the host via infrared. Let $quantity$ be the only reconcilable value.

$T = (r(id), r(location), r(quantity), w(quantity' := quantity + a))$, where a is the amount of new products.
 $RSet(T) = \{\vec{x} = \emptyset, \vec{c}\vec{x} = \{quantity\}, \vec{n}\vec{x} = \{id; location\}\}$ (see definition 4)

The reconcilable elements depend on the transaction. In the withdraw example the user transfers money and the current balance does not affect his decision to execute the transfer, e.g. to pay a bill. However, assume the actual balance affects the decision to pay the bill. In the first situation the balance is classified as reconcilable, whereas it's non-reconcilable in the second one.

In the next section we describe the read phase's issues. The validation, and reconcile phase are described later.

5.1. Read phase

Since the VAL^{-CS} was designed for connected architectures we analyze the read phase in order to fit the requirements for disconnected architectures and reconciliation. The support for local autonomy requires to replicate data on the mobile device. Beside this aspect, the read set contains reconcilable, as well as non-reconcilable entities (see definition 4).

In OCC, a reading transaction is not allowed to read data which intersect with the write set $WSet$ of a transaction (see

preliminaries, read phase (i)). Section 5.2 shows that reconciliation prevents from critical read anomalies. Therefore, the test in (i) reduces to $\vec{n}\vec{x}$ components only. Validation of non-reconcilable data is on an entity basis, so we denote $RSet^{NX}$ as the read set for non-reconcilable elements and $WSet^{NX}$ respectively. For the modified validation in the read phase of Unland's VAL^{-CS} algorithm see figure 4.

```

for  $\forall T_i \in T^{read}$  do
  if ( $\forall T_j \in T^{write} : RSet^{NX}(i) \cap WSet^{NX}(j) \neq \emptyset$ )
    abort  $T_i$ 
end for
// $T^{read}$ :=All reading transactions.
// $T^{write}$ :=All writing transactions.

```

Figure 4. write - read validation

5.2. Read anomalies and reconciliation

The lost update anomaly is not treated in this section because the example above (see example 1) shows that reconciliation prevents from a lost update.

Dirty read

Assume the schedule

$$S = r_2(x)w_2(x' := x - 200)ec_2r_1(x')a_2w_1(x'' := x - 100)ec_1, \text{ where } ec_i \text{ indicates an escrow commit, and } a_i \text{ an abort.}$$

S normally produces a dirty read, but it is escrow serializable. The TM will detect the conflicting transaction T_1 , replay it automatically, and will use the current value of x . The result is the following schedule which prevents from a dirty read:

$$S = r_2(x)w_2(x' := x - 200)ec_2r_1(x')a_2w_1(x'' := x - 100)ec_1a_1r_1(x)w_1(x' := x - 100)ec_1c_1$$

a_i indicates an abort by the TM, because a constraint $x > 0$ was violated, and assume x was 200 at $r_2(x)$.

Non repeatable read

Assume the following schedule:

$$S = r_1(x)r_2(x)w_2(x' := x + 200)ec_2c_2r_1(x')ec_1c_1$$

As denoted in the schedule $x \neq x'$.

The non repeatable read anomaly is predestinated to briefly sketch out the problem of read anomalies in disconnected architectures. After T_1 reads x , x is only present in T_1 's workspace. Each re-read in the workspace will produce the same result and on the mobile client's side no non repeatable read is present. But the server side handling is of interest. If the operation is replayed with x 's current value the read is still non-repeatable. Therefore only to lock x , or restrict repeatable read to workspace level is possible. But, the point is that the *ec*-model exploits a non-repeatable read for x values to replay the operation. Thus, the *ec*-model doesn't support an isolation of repeatable read.

Phantom read

Assume the following schedule:

$$S = cnt_{11} := count_1(X)insert_2(x)ec_2c_2cnt_{12} := count_1(X)ec_1c_1$$

Both $count(X)$ operations execute in the transaction's workspace (mobile client), and they have to produce the same result, unless a concurrent transaction accesses the same workspace (not provided by *ec*), or an explicit re-read was performed. If the $count$ operation is able to get reconciled and a phantom read should be prevented, T_1 has to be *ec*-aborted and replayed with the current state. The result is the following schedule which prevents from a phantom read:

$$S = cnt_{11} := count_1(X)insert_2(x)ec_2c_2ec_1a_1cnt_{11} := count_1(X')c_1$$

Summarizing, escrow serializability prevents reconcilable values from lost update, dirty read, but phantom read is possible.

5.3. Validation - and reconcile phase

Validation starts with an escrow commit and the associated change set. As described in the phase validate (ii), each transaction starting validation has to validate its change set against each (currently) validating transaction with a lower *TNR* (see figure 6).

Applying this test to transaction

$$T_w = (r(id), r(location), r(quantity), w(quantity))$$

(see example 2) means to test if $\vec{n}x = (id, location)$ or $\vec{x} = (quantity)$ intersects with another transaction's *WSet* with a lower *TNR*. If validation succeeds the modifications are written, and if either *id, location* or *quantity* intersects

with another transaction's *WSet* the transaction aborts. But, considering that *quantity* is a reconcilable value an intersection of *quantity* does not lead to an unresolvable conflict provided no constraint is violated. According to that, the test in (ii) is customized to fit the requirements for reconcilable elements (see figure 6).

There are some other conclusions. For the validation of reconcilable elements it's sufficient to validate the constraint only (on a value basis). If no constraint is present, validation is unnecessary and the algorithm is directly entering the reconciliation phase, because to reconcile means just to replay the conflicting operation with the current value. Therefore, only non reconcilable entities enter the validation and write phase, whereas constrained reconcilable elements need validation and reconciliation. And in contrast, reconcilable elements enter the reconcile phase only. Recall, that reconcile includes to write data.

Assume the validation of *location* fails. Under this circumstance the transaction aborts due to atomicity, and to prevent from unnecessary reconciliation a validation order from non-reconcile to reconcile $\vec{n}x \xrightarrow{order} \vec{x}$ has to be followed.

Following this order, to replay means that the operation which leads to a conflict is replayed only, because each write of a non-reconcilable element was still performed, and to re-write again is unwanted due to performance reasons. Thus, the *ec*-model treats with transaction splitting, because for each element of \vec{x} a new nested sub transaction is created and executed by the TM.

Conclusions:

- (1) A validation order from non-reconcilable to reconcilable prevents from unnecessary reconciliation. Denoted by $\vec{n}x \xrightarrow{order} \vec{x}$
- (2) A replay will only replay the conflicting operations.
- (3) The replay of an operation leads to a new nested transaction.

Figure 5 defines the possible states and transitions of the server phase.

Each non-reconcilable element nx_i validates first (1). If validation succeeds the value of nx_i is written (2) and committed. If validation fails, T is aborted (3).

If each nx_i was written successfully each cx_i is validated next (1). The transaction aborts if a constraint is violated (3). Provided a constraint is not violated cx_i enters reconciliation (4). In case each nx and cx commits, each x_i directly enters the reconcile phase (5). After the reconciliation and write phase a commit is only possible, since we assume a reliable and physical error free hardware (6). If hardware fails, this is a matter of recovery. Transactions, like the proposed Quota, try to prevent from constraint violation through pre-estimation. For such transactions a relaxed validation might be applicable. It could be indicated by a role, and classified by failure probabilities.

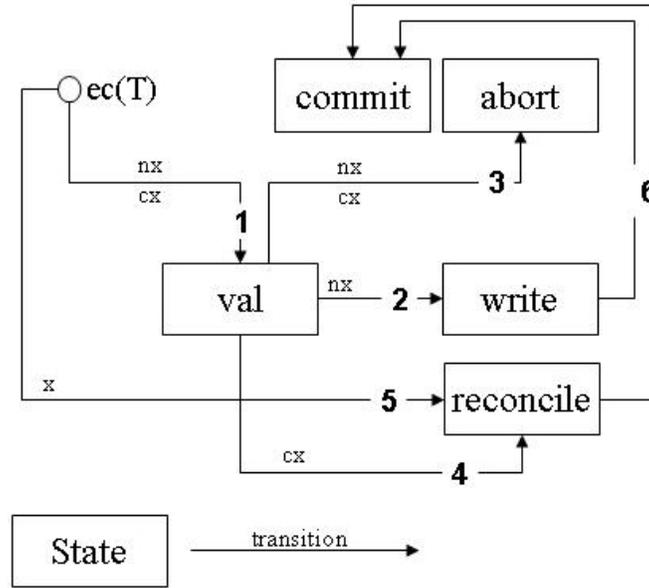


Figure 5. States of an escrow transaction

Reconciliation ensures a serial execution (see figure 6) and the performance section shows higher throughput for reconcilable transactions.

As mentioned before an escrow transaction is splitted to comply with the execution order $\vec{nx} \xrightarrow{order} \vec{x}$. Definition 5 defines how an escrow transaction is splitted by the TM.

Definition 5: (Transaction splitting)

(1) Let T' be a user transaction which spans sub transactions T_i . To ensure atomicity T' is aborted if any sub transaction $T_i \in \mathbf{T} = T_i$ fails.

(2) The set of sub transactions \mathbf{T} contains one nested transaction T^{NX} , and two nested atomic sets of transactions \mathbf{T}^{CX} and \mathbf{T}^X .

$$T' := \{T^{NX}, \mathbf{T}^{CX}, \mathbf{T}^X\}$$

(3) Each operation $op_n(cx) \in T'$ which modifies a **constrained** reconcilable entity leads to the creation of a new sub transaction in order to replay the operation within T' . Let \mathbf{T}^{CX} represent these kind of transactions.

$$\mathbf{T}^{CX} := (T_1(op_1(cx)), \dots, T_i(op_n(cx)))$$

(4) Each operation $op_n(x) \in T'$ which modifies a reconcilable entity leads to the creation of a new sub transaction in order to replay the operation within T' . Let \mathbf{T}^X represent these kind of sub transactions.

$$\mathbf{T}^X := (T_1(op_1(x)), \dots, T_i(op_n(x)))$$

(5) Let T^{NX} be the transaction which contains all operations $op_n(nx)$ that are non-reconcilable.

$$T^{NX} := (op_1(nx), \dots, op_n(nx))$$

To split a transaction requires information about the variables and their semantics. We base our model on the idea of change sets which are delivered to the TM after local (on the mobile device) modifications took place. A change set ChS represents the user transaction T' and consists of several entities e . Beside the different versions for a variable (old and new), each e must provide a key (e.g. unique type name) to enable a mapping between transaction, reconciliation rules and constraints. Usually a transaction defines which variable is reconcilable or non-reconcilable. This information, and the type information of the change set is adequate to split a transactions into its corresponding sub transactions $T^{NX}, \mathbf{T}^{CX}, \mathbf{T}^X$. In our prototyp a type handler is used in order to facilitate the mapping. The reconcilable entities are defined transaction specific on a unique type basis, and after a transaction starts, it registers immediately by the TM. This registration and the type handler enable, as long as the transaction and the TM rely on an identical type basis, to split a transaction and utilize a transaction specific reconciliation.

This section described the server phase of the ec-model and how the VAL^{-CS} algorithm is extended by an additional reconcile phase. To split a transaction is an adequate solution to handle reconcilable and non-reconcilable elements within the same transaction.

5.4. Nested transactions in the ec-model

So far, an user transaction is defined to be atomic, but there might be some dependencies within an user transaction

which allow to weaken the execution order. To obtain a weakened execution order we classify sub transactions according to the open nested transaction model first (see [29], [12], [30]). Generally, the nested transaction model allows to relax atomicity and isolation which is often required in mobile, transactional workflow scenarios, or other so called Advanced Transaction Models (ATM) (see [31]). In the open nested transaction model a sub transaction, or child, is:

- 1) open ($iso = false$), if the results are published to all transactions. It is closed ($iso = true$), if results are published to the parent transaction only (ISOLATION).
- 2) It is vital ($vit = true$), if to abort leads its parent to abort, too. And, non-vital ($vit = false$), if an abort does not affect its parent (Atomicity, $abort \overrightarrow{T_c T_p}$).
- 3) T depends on its parent ($dep = true$), if the parents abort leads T to abort, too; independent ($dep = false$) if not (Atomicity, $abort \overrightarrow{T_p T_c}$).

By definition 5 T' has to be atomic, thus each $T \in T'$ is dependent, vital, and closed in order to avoid cascading rollback (see section 2). For independent, non-vital and open children however, it's possible to change the execution order, and to execute this new class of transactions separately (possibly pooled or even delayed). E.g. in example 2 assume an additional transaction which monitors the time an employee needs to complete an order. A *time* variable is incremented locally and synchronized at the location with the time of all employees in order to calculate the average delivery time for that product. This is an example for an independent, non-vital and open transaction on a reconcilable value.

The drawback of intra transaction dependencies is that the TM needs to know them. Our implementation (see 6) as well as our model requires a change set reflecting local modifications. If the change set provides the information needed to classify a transaction a new execution order is applicable. Based on the classifications for sub transactions definition 6 extends definition 5.

Definition 6: (*ec*-independency)

- (1) Let \mathbf{T}^{IN} be the set of all independent, non-vital, and open transactions.
- (2) T' is defined as an user transaction with one nested transaction T^{NX} , two nested atomic sets of transactions \mathbf{T}^{CX} and \mathbf{T}^X , and one atomic, and *ec*-independent set of sub transactions \mathbf{T}^{IN} . Each $T \in \mathbf{T}^{IN}$ is *ec*-independent.
 $T' := \{(T^{NX}, \mathbf{T}^{CX}, \mathbf{T}^X), \mathbf{T}^{IN}\}$
- (3) A transaction T_c is *ec*-independent if the dependency function dep of T_c is known for all other transactions $T_i \in T' : T_i \neq T_c$, and for each $T_i \in T'$ dep validates to true, whereas true indicates the *ec*-independency of T_c .
- (4) *ec*-independency: $iso = 0 \wedge vit = 0 \wedge dep = 0$

Although mobile transaction models deal with (non-) substitutable, compensable, temporal, and spatial transactions¹, none of them have been considered here. Such intra transaction dependencies often originate from the use case's semantics and less from the value's or operation's semantics. In an interaction scenario (e.g. workflow based) the state and transition model is able to provide the information about such intra dependencies. In mobile computing, service selection and composition is context aware and an (formal) interaction model might not given. Hahn's model [33] exploits transactional properties (non-functional) defined in the interface to determine which workflow pattern (XOR, AND, and SEQUENCE) is proper for a service interaction.

Our focus is to exploit the semantics on a data and operations level first, and not to exploit the semantics of complex interaction scenarios. The *ec*-model may provide a reliable foundation other transaction models could rest upon. To benefit from both kinds of semantics (data & value and interaction) seems to be a worthwhile objective.

6. Example Implementation of the *ec* Model with SDO

Service Data Objects (SDO [34], [35]) are a platform neutral specification and disconnected programming model, which enables dynamic creation, access, introspection, and manipulation of business objects.

Our implementation (see Lessner [36]) of the transaction manager (TM) uses SDO graphs and resides between the data access service (DAS) and the client. This way, the TM fits well into SDO's vision of being independent of the data source.

A snapshot of each delivered graph is taken by the TM and each SDO graph associates a change summary that complies with the requirements for optimistic concurrency control (see Section 3). To assert "escrow serializability" (provided by the reconciliation algorithm) an association between a transaction and the semantic of this transaction is needed². This association results in a classification of the transaction (e.g. "linear dependent").

An association between a classification and a verifier (see Figure 8) enables a semantic transaction level (e.g. quota verifier, escrow concurrency (EC)).

Assume that we have an incoming transaction (T-Level=EC) with a changed data graph. The transaction handler delivers the transaction to the verifier. To ensure EC, optimistic concurrency control (OCC) is checked first. If OCC is passed there is no serialization conflict. In case of a conflict the semantic level concurrency control (T-Level=EC) is invoked. This means, two verifier implemen-

1. Temporal and spatial describe transactions subject to temporal and spatial restrictions, respectively.

2. In heterogeneous environments an additional data object could be used to describe the semantic of a transaction, SDO uses XML as protocol


```

ensure: set of transactions  $\mathbf{T} = \{T_1, T_2, \dots, T_n\}$ 
ensure: actual database state  $D^s$ , set of constraints  $C(D)$ 
ensure: only committed data in read set  $RSet(i)$  of  $T_i$ 
ensure:  $T_i = (op_{ik}, i = 1, 2, \dots, k_i)$ 
ensure:  $T' := \{T^{NX}, T^{CX}, T^X\}$ 
for  $\forall ec_i \in \{ec_1, ec_2, \dots, ec_n\}$  received do
T-number:  $TNR_i := TNC; TNC++$ 
for  $\forall T_j \in \mathbf{T}^{val} : TNR_j < TNR_i$  do
  // test 1 for NX only
  if  $(RSet^{NX}(i) \cap WSet(j) \neq \emptyset)$ 
    abort  $T'$ 
  else
    // test 2 for CX only
    ensure:  $RSet(T') \subseteq D^s$ 
    ensure: serial execution
    if  $(\exists cx: (c = false))$  then
      abort  $(T')$ 
    else
      for  $\forall T_i(op_n(cx)) \in \mathbf{T}^{CX}$ 
        //reconcile
        do  $T_i(op_n(x))$ 
      end for
      for  $\forall T_i(op_n(x)) \in \mathbf{T}^X$ 
        //reconcile
        do  $T_i(op_n(x))$ 
      end for
    end if
  end if
end for

```

\mathbf{T}^{val} each validating transaction.

Figure 6. VAL^{CS} with reconciliation for ec serializability

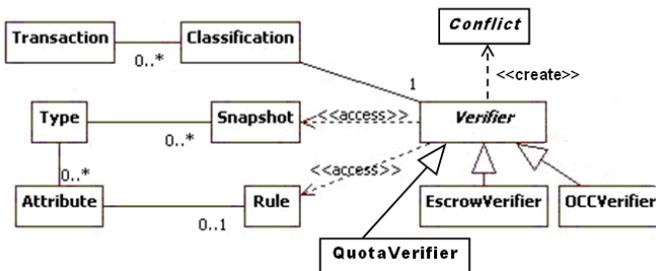


Figure 8. Abstract design of the transaction manager

detection of changes is required.

Assume the row (tuple) $(id, location, quantity, v)$, where v is the version. Now, two transactions T_1 and T_2 update $quantity$ concurrently, where T_1 writes first. Thus, the version is incremented and if T_2 tries to write, the TM has to abort T_2 . To facilitate the TM to support a reconciliation

the TM (1) has to know the row's current version v_c in order to compare v_c with the version read by the transaction. And, (2) if a conflict is detected the TM must be able to indicate a modification of $quantity$ only; Assumed $quantity$ is reconcilable. To detect fine grained modifications if a conflict was detected, requires to re-read the row with its current version and to analyze the row. Analyzing means that the TM has to correlate the changes with the matching tuple's component. If such fine grained modifications are detectable reconciliation is possible. Nevertheless, the drawback of (1) and (2) is that during the phases re-reading of the version and conflict detection, or in the case of a conflict also the phases re-read the row, analyzing it, and reconciliation, the row has to be consistent.

Now, assume the row above is divided into $(id, location, ref_q, v)$ and $(quantity, v^q)$, where ref_q is a reference on quantity. Now two versions have to be verified, but v^q directly indicates modifications for $quantity$, and prevents the analysis of the whole row.

Another concern is, if modifications, represented by an atomic change set, belong to several rows with corresponding versions. Then version verifying has to be atomic and each version has to be compared, or another mechanism is provided (e.g. *intent versioning*). Furthermore, to enable reconciliation for each modified row the TM has also to re-read each conflicting row.

In general, the main difference is RVV tries to write on the database in order to detect conflicts. Whereas the snapshot handler detects conflicts and ensures ec serializability before any data is written (sent) into the DB. The snapshot handler's drawback is to ensure a synchronous and consistent state between middleware and DB level. Its advantage is a common representation for the change set, and the snapshots which alleviates conflict detection. As mentioned, RVV's advantage is a fast conflict detection at row level (DBMS level), and even modifications of non ec - transactions are detectable. The overhead for a fine grained error detection and the required consistence of a row have to be still analyzed.

Another solution like an event driven one, which triggers an event to notify each participating node of a modification is also conceivable. Each solution has to break through the obstacle to keep versions or snapshots synchronous across layers and during concurrent access.

In summary, RVV is an easy to understand solution and it's generally applicable on the ec -model, especially for nx entities. RVV itself is a more architectural aspect which may be needed in some scenarios. Versioning in principle is a well known discipline.

8. Performance of the ec -model

We ran a series of simulations of concurrent withdraw transactions accessing the same account. The transaction configuration parameters were as following: Reading the

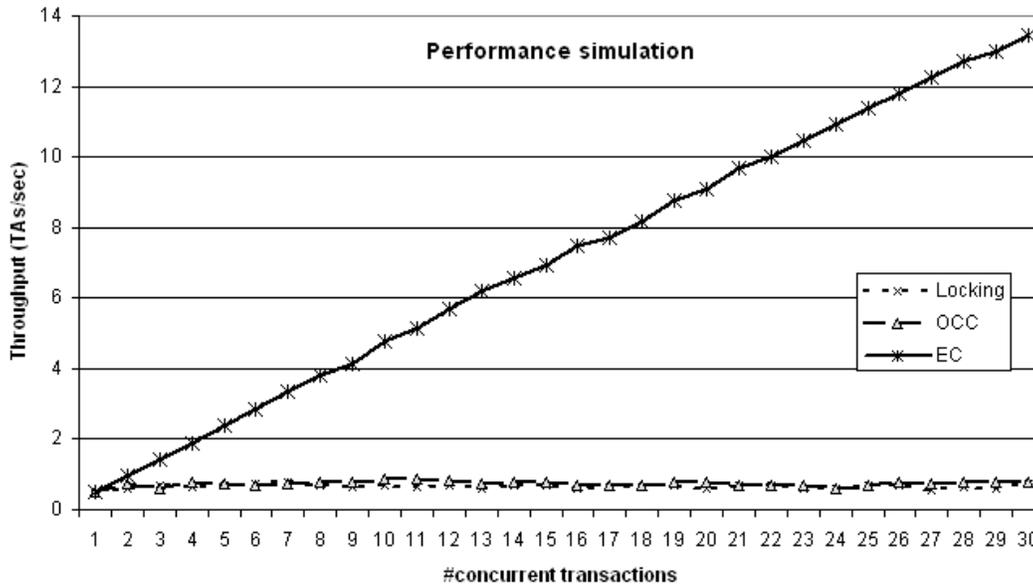


Figure 9. Performance of EC compared with OCC and locking

balance took less than 10 ms, the user's thinking time was randomly chosen between 1 and 2 seconds, and the write time needed about 10 ms. The throughput results for up to 30 concurrent transactions is shown in Figure 9.

Running 30 transactions in parallel generated 23 serialization conflicts which triggered the replay mechanism. The net processing time for a transaction or a replay was approximately 20 ms. The total elapsed time for all 30 transactions was $t = 2.1$ sec which is consistent with the minimum thinking time (1 sec) plus the time for processing 30 transactions ($(30 + 23) \times 20$ ms = 1.06 sec) in *escrow*-serialization mode. The results show that we achieved an elapsed time close to the theoretical limit considering the number of replays necessary.

The nearly linear growth of the throughput when using the *escrow* concurrency control indicates that we have not reached the throughput limit. Given a processing time of 20 ms, the theoretical limit for this scenario ("hot spot" on the balance) would reach 50 transactions per second.

In contrast, the traditional OCC and locking schemes could not interleave the transactions and resulted in essentially serial processing. Therefore the performance saturated at $1/1.5 = 0.66$ transactions per second, where 1.5 sec are the average transaction duration.

In order to have a more complex example than the withdraw transaction we used the popular TPC-C benchmark [37] and analysed the *New_Order* (NOrder-Ta) and the *Payment* (Pay-Ta) transactions. The NOrder-Ta exhibits two "hot spots" with linear dependency semantics defined in section 4. One is the update of the next order id (*d_next_o_id*) in the district table and the other is the update of the quantity on

stock (*s_quantity*) for each line item. The Pay-Ta contains "hot spots" in the tables *Warehouse*, *District*, and *Customer*. Again, the semantics of the transaction is linear dependent with gradient 1 (see section 4) as it deals with updating three balances with a fixed amount, updating the year to date payment by the same amount, and incrementing the payment count.

In total we have identified 7 situations where the *escrow*-serialization mechanism could be beneficial for performance. First tests indicate a substantial improvement over traditional locking mechanism.

9. Conclusions

Mobile transactions have special demands for the transaction management. We propose a transaction model that is non-blocking and is reconciling conflicting transactions by exploiting the semantic of the transaction. A simple abort-replay mechanism can produce reconciliation in the sense of *escrow* serializability. The abort-replay algorithm detects conflicts by rereading the data. The mechanism is easy to implement and can make use of update operation when read - and write set overlaps.

If all writes are postponed until the commit is issued and the reread and write operations during reconciliation are executed serialized or serial, then no inconsistent data will be read. A further option is to use consistent snapshots. Independent from the mechanism the read phase should be executed with optimistic concurrency control.

In contrast, the reconciliation phase should run in a pre-claiming locking mode. This ensures efficient sequential

processing of competing transactions without delays as user input is already available and starvation is avoided. With this marginal condition the escrow serialization algorithm has the potential to outperform other mechanisms.

For the class of linear dependent transactions it is sufficient for reconciliation to know the client state at begin of transaction, the state produced by the client transaction on the client site, and the database server state at commit time.

10. Future work

The following outlines some important points of future research. Even if first simulations show higher transaction throughput, they need to become re-engineered. More complex business scenarios with several devices involved and an implementation based at databases driver level are some aspects. Such an implementation will simplify benchmarks and ease the existing SDO-implementation. Concerning the development of software, developers should be supported by methods to define or classify dependency functions for transactions which intend to use escrow serializability. In general, to define and detect a transaction's semantics manually is a drawback of many other transaction models using semantic properties. Regarding our model, an investigation of the opportunities to derive dependency functions based on common interaction patterns or on a data type base is intended. A not mentioned, but relevant concern is recovery in the escrow model, and more research has to be done in this field.

Acknowledgements

This paper was inspired by discussions with the members of the DBTech network and the ideas presented during the DBTech Pro workshops. The DBTech Project was supported by the Leonardo da Vinci programme during the years 2002 - 2005.

References

- [1] Fritz Laux and Tim Lessner. Transaction processing in mobile computing using semantic properties. *The First International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA, Cancun, Mexico*, 169, March 2009.
- [2] J.H. Abawajy and M. Mat deris. Supporting disconnected operations in mobile computing. *Computer Systems and Applications, ACS/IEEE International Conference on*, 0:911–918, 2006.
- [3] Shirish Hemant Phatak and Badri Nath. Transaction-centric reconciliation in disconnected client-server databases. *Mob. Netw. Appl.*, 9(5):459–471, 2004.
- [4] Joanne Holliday, Divyakant Agrawal, and Amr El Abbadi. Disconnection modes for mobile databases. *Wirel. Netw.*, 8(4):391–402, 2002.
- [5] Hector Garcia-Molina. Using semantic knowledge for transaction processing in a distributed database. *ACM Trans. Database Syst.*, 8(2):186–213, 1983.
- [6] Abdel Aziz Farrag and M. Tamer Özsu. Using semantic knowledge of transactions to increase concurrency. *ACM Trans. Database Syst.*, 14(4):503–525, 1989.
- [7] Shi-Ming Huang and Chien-Ming Huang. A semantic-based transaction model for active heterogeneous database systems. *Systems, Man, and Cybernetics*, 3:2854–2859, 1998.
- [8] Dimitrios Georgakopoulos, Marek Rusinkiewicz, and Amit P. Sheth. On serializability of multidatabase transactions through forced local conflicts. In *Proceedings of the Seventh International Conference on Data Engineering*, pages 314–323, Washington, DC, USA, 1991. IEEE Computer Society.
- [9] Sharad Mehrotra, Rajeev Rastogi, Henry F. Korth, and Abraham Silberschatz. Non-serializable executions in heterogeneous distributed database systems. In *PDIS '91: Proceedings of the first international conference on Parallel and distributed information systems*, pages 245–252, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.
- [10] Shi Ming Huang, Irene Kwan, and Chih He Li. A study on the management of semantic transaction for efficient data retrieval. *SIGMOD Rec.*, 31(3):28–33, 2002.
- [11] Peter Graham and Ken Barker. Effective optimistic concurrency control in multiversion object bases. In Elisa Bertino and Susan Darling Urban, editors, *ISOOMS '94: Proceedings of the International Symposium on Object-Oriented Methodologies and Systems*, volume 858 of *Lecture Notes in Computer Science*, pages 313–328, London, UK, 1994. Springer-Verlag.
- [12] Gerhard Weikum and Gottfried Vossen. *Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.
- [13] Fritz Laux, Tim Lessner, and Martti Laiho. Semantic transaction processing in mobile computing. In Cherif Branki, Brian Cross, Gregorio Daz, Peter Langendrfer, Fritz Laux, Guadalupe Ortiz, Martin Randles, A. Taleb-Bendiab, Frank Teuteberg, Rainer Unland, and Gerhard Wanner, editors, *TAMoCo*, volume 169 of *Frontiers in Artificial Intelligence and Applications*, pages 153–164. IOS Press, 2008.
- [14] Qi Lu and M. Satyanarayanan. Isolation-only transactions for mobile computing. *Operating Systems Review*, 28:81–87, 1994.
- [15] H. T. Kung and John T. Robinson. On optimistic methods for concurrency control. *ACM Trans. Database Syst.*, 6(2):213–226, 1981.
- [16] Jim Gray, Pat Helland, Patrick O'Neil, and Dennis Shasha. The dangers of replication and a solution. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 173–182, New York, NY, USA, 1996. ACM.

- [17] G. D. Walborn and P. K. Chrysanthis. Supporting semantics-based transaction processing in mobile database applications. In *SRDS '95: Proceedings of the 14TH Symposium on Reliable Distributed Systems*, page 31, Washington, DC, USA, 1995. IEEE Computer Society.
- [18] Gary D. Walborn and Panos K. Chrysanthis. Transaction processing in pro-motion. In *SAC '99: Proceedings of the 1999 ACM symposium on Applied computing*, pages 389–398, New York, NY, USA, 1999. ACM.
- [19] Rainer Unland. Optimistic concurrency control revisited. Technical report, Department of Business Informatics, University of Muenster, 1994.
- [20] Martti Laiho and Fritz Laux. Data access using rrv discipline and persistence middleware. *e RA - 3, Greece*, 2008.
- [21] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database Systems: The Complete Book*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
- [22] Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts, 5th Edition*. McGraw-Hill Book Company, 2006.
- [23] K. A. Momin and K. Vidyasankar. Flexible integration of optimistic and pessimistic concurrency control in mobile environments. In Julius Stuller, Jaroslav Pokorný, Bernhard Thalheim, and Yoshifumi Masunaga, editors, *ADBIS-DASFAA '00: Proceedings of the East-European Conference on Advances in Databases and Information Systems Held Jointly with International Conference on Database Systems for Advanced Applications*, volume 1884 of *Lecture Notes in Computer Science*, pages 346–353, London, UK, 2000. Springer-Verlag.
- [24] Stefano Ceri and Susan S. Owicki. On the use of optimistic methods for concurrency control in distributed databases. In *Berkeley Workshop*, pages 117–129, 1982.
- [25] Ho-Jin Choi and Byeong-Soo Jeong. A timestamp-based optimistic concurrency control for handling mobile transactions. In Marina L. Gavrilova et al., editor, *ICCSA (2)*, volume 3981 of *Lecture Notes in Computer Science*, pages 796–805. Springer, 2006.
- [26] Adeniyi A. Akintola, G. Adesola Aderounmu, A. U. Osakwe, and Michael O. Adigun. Performance modeling of an enhanced optimistic locking architecture for concurrency control in a distributed database system. *Journal of Research and Practice in Information Technology*, 37(4), 2005.
- [27] V. Krishnaswamy, D. Agrawal, J. L. Bruno, and A. El Abbadi. Relative serializability: An approach for relaxing the atomicity of transactions. *SIGMOD/PODS 94*, 1994.
- [28] Klaus R. Dittrich and Stella et al. Gatzju, editors. *Aktive Datenbanksysteme*. dpunkt Verlag, Heidelberg, BW, GER, 2000.
- [29] J. E.B. Moss. *Nested transactions: an approach to reliable distributed computing*. Massachusetts Institute of Technology, Cambridge, MA, USA, 1985.
- [30] Kyong-I Ku and Yoo-Sung Kim. Moflex transaction model for mobile heterogeneous multidatabase systems. In *RIDE '00: Proceedings of the 10th International Workshop on Research Issues in Data Engineering*, page 39, Washington, DC, USA, 2000. IEEE Computer Society.
- [31] Sushil Jajodia and Larry Kerschberg, editors. *Advanced Transaction Models and Architectures*. Kluwer, 1997.
- [32] Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.
- [33] Katharina Hahn and Heinz Schweppe. Exploring transactional service properties for mobile service composition. In Markus Bick, Martin Breunig, and Hagen Höpfner, editors, *MMS*, volume 146 of *LNI*, pages 39–52. GI, 2009.
- [34] M. Adams and C. Andrei et al. Service data objects for java specification. *OSOA (BEA Systems, IBM, et al.)*, 2.1, 2006.
- [35] J. Beatty, S. Brodsky, M. Nally, and R. Patel. Next-generation data programming. *BEA Systems, IBM*, 2003.
- [36] Tim Lessner. Transaktionsverarbeitung in disconnected Architekturen am Beispiel von Service Data Objects (SDO) und prototypische Implementierung eines Transaktionsframeworks. Diploma thesis (ger), 2007.
- [37] TPC. Tpc benchmark c, standard specification, revision 5.9, 2007.

A FAMILY OF RECURSIVE LEAST-SQUARES ADAPTIVE ALGORITHMS SUITABLE FOR FIXED-POINT IMPLEMENTATION

Constantin Paleologu, Silviu Ciochină, and Andrei Alexandru Enescu

Telecommunications Department, University Politehnica of Bucharest, Romania
e-mail: {pale, silviu, aenescu}@comm.pub.ro

ABSTRACT

The main feature of the least-squares adaptive algorithms is their high convergence rate. Unfortunately, they encounter numerical problems in finite precision implementation and especially in fixed-point arithmetic. The objective of this paper is twofold. First, an analysis of the finite precision effects of the recursive least-squares (RLS) algorithm is performed, outlining some specific problems that could appear in fixed-point implementation; consequently, we present a modified version of the RLS algorithm suitable for fixed-point implementation, using an asymptotically unbiased estimator for the algorithm's cost. Second, we extend the procedure for the case of QR-decomposition-based least-squares lattice (QRD-LSL) adaptive algorithm, a "fast" member of RLS family, with good numerical properties. The reduced dynamics of the algorithm's parameters leads to facility for fixed-point implementation. The simulations performed on a fixed-point digital signal processor (DSP) sustain the theoretical findings. Also, as a practical aspect of this work, we illustrate the performance of the proposed QRD-LSL algorithm for noise reduction.

Index Terms— Adaptive filters, fixed-point implementation, noise reduction, QR-decomposition-based least-squares lattice (QRD-LSL) algorithm, recursive least-square (RLS) algorithm.

1. INTRODUCTION

The Recursive Least Squares (RLS) algorithm is one of the most popular adaptive algorithms, mainly due to its fast convergence rate [1]. Nevertheless, there are some major drawbacks related to the high computational complexity and the large dynamic range of the algorithm's variables. The first issue could be overcome by using a fast RLS algorithm, in the meaning that the computational cost increases linearly with the number of adjustable parameters. The last drawback is more severe and could cause unwanted effects in a fixed-point arithmetic context,

such as overflow or stalling phenomena [2]. In this paper we focus on some numerical problems of the RLS algorithm and present a modified version of this algorithm, which is more suitable for fixed-point implementation. For practical reasons, the proposed procedure is applied to the QR-decomposition-based least-squares lattice (QRD-LSL) algorithm, which is a fast member of the RLS family with robust numerical behavior.

The QRD-LSL algorithm [3] combines the good numerical properties of QR-decomposition and the desirable features of a recursive least-squares lattice. Whereas the recursive QR-decomposition-based recursive least-squares (QRD-RLS) algorithm [1] requires a high computational load on the order of L^2 (where L is the filter order), in terms of both the number of processing cells and the computation per iteration, the QRD-LSL implementation is fast in the sense that these numbers are reduced to a linear dependence on L . This algorithm exploits the shifting property of serialized input data (the Toeplitz structure of the data matrix) to perform joint-process estimation in a fast manner. By virtue of these facts, the QRD-LSL algorithm is endowed with a highly desirable set of operational and implementation characteristics such as good numerical properties (inherited from QR-decomposition), good convergence properties (due to the RLS nature), and a high level of computational efficiency (resulted from the modular, lattice-like structure). The combination of these characteristics makes the QRD-LSL a powerful adaptive algorithm, suitable for a wide range of applications, e.g., echo cancellation, interference rejection, or noise reduction [4]–[10].

Another implication of the modular structure of the QRD-LSL algorithm is that it lends itself to the use of very large-scale integration (VLSI) technology for its hardware implementation. Of course, the use of this sophisticated technology can be justified only if the application of interest calls for the use of VLSI chips in large number. Otherwise, a digital signal processor (DSP) implementation represents a proper solution. In this case, an important practical aspect is related to the dynamic range of the algorithm's parameters. It is known that in two-complement fixed-point implementation context the

absolute values of all involved parameters have to be smaller than 1. In the case of the classical QRD-LSL algorithm the cost functions asymptotically increase; theoretically, they are upper bounded by $1/(1-\lambda)$, where λ is the exponential weighting factor ($0 < \lambda \leq 1$) [1]. When dealing with a value of λ very close to 1 (which is the case in most of the applications due to stability reasons [11]), very large values of the cost functions are expected. In order to prevent any unwanted overflow phenomenon it is necessary to scale the cost function. As a consequence, the major drawback is the precision loss because of these factors.

In the first part of this work we analyze the behavior of the RLS algorithm in fixed-point arithmetic, revealing some specific problems that could appear in this context. In order to overcome these potential issues we present a modified version of the RLS algorithm that is suitable for fixed-point implementation. The main idea is to use an asymptotically unbiased estimator for the algorithm's cost function, in order to reduce the dynamic range of this parameter. The idea can be applied to other RLS-based algorithms. In this paper we extend the procedure for the case of QRD-LSL algorithm. Consequently, a modified version of this algorithm is obtained. The reduced dynamic of the parameters leads to facilities for fixed-point implementation.

The paper is organized as follows. Section 2 contains certain backgrounds of the classical RLS algorithm, outlining several specific problems that could appear in fixed-point implementation. In Section 3 we establish a connection between the dynamic range of variables and the initial convergence rate, and we present a modified version of the RLS algorithm suitable for fixed-point implementation. The modified version of the QRD-LSL algorithm is developed in Section 4. The simulation results are presented in Section 5. A summarized discussion of the main results is given in Section 6. Finally, Section 7 briefly concludes this work.

2. RLS ALGORITHM BACKGROUND AND ANALYSIS

The well-known RLS adaptive algorithm [1] uses a cost function defined as an estimate of the mean square-error, i.e.,

$$J(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 = \lambda J(n-1) + |e(n)|^2, \quad (1)$$

where $0 < \lambda \leq 1$ is the exponential weighting factor and $e(i)$ is the difference between the desired response $d(i)$ and the output $y(i)$ produced by an adaptive transversal filter. That is,

$$e(i) = d(i) - y(i) = d(i) - \mathbf{w}^H(n) \mathbf{x}(i), \quad (2)$$

where $\mathbf{x}(i)$ is the tap-input vector at time i and $\mathbf{w}(n)$ is the tap-weight vector at time n . The superscript H denotes Hermitian transposition (transposition and complex conjugation).

This estimate of the cost function induces similar estimates for the correlation matrix $\Phi(n)$ and the cross-correlation vector $\theta(n)$, i.e.,

$$\Phi(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^H(i) = \lambda \Phi(n-1) + \mathbf{x}(n) \mathbf{x}^H(n), \quad (3)$$

$$\theta(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) d^*(i) = \lambda \theta(n-1) + \mathbf{x}(n) d^*(n), \quad (4)$$

where superscript $*$ denotes complex conjugation. The optimum value of the tap-weight vector $\mathbf{w}(n)$, for which the cost function $J(n)$ from (1) attains its minimum value is defined by the normal equation written in matrix form:

$$\Phi(n) \mathbf{w}(n) = \theta(n). \quad (5)$$

The regular procedure is to apply the matrix inversion lemma in (3) in order to solve (5). Denoting

$$\mathbf{P}(n) = \Phi^{-1}(n) \quad (6)$$

and defining the Kalman vector

$$\mathbf{k}(n) = \frac{1}{\lambda} \cdot \frac{\mathbf{P}(n-1) \mathbf{x}(n)}{1 + \frac{1}{\lambda} \mathbf{x}^H(n) \mathbf{P}(n-1) \mathbf{x}(n)} \quad (7)$$

the inverse of the estimate of the correlation matrix is computed in a recursive manner as [1]

$$\mathbf{P}(n) = \frac{1}{\lambda} \mathbf{P}(n-1) - \frac{1}{\lambda} \mathbf{k}(n) \mathbf{x}^H(n) \mathbf{P}(n-1). \quad (8)$$

Finally, the recursive equation for updating the tap-weight vector is

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n) \alpha^*(n), \quad (9)$$

where $\alpha(n)$ is the a priori estimation error defined by

$$\alpha(n) = d(n) - \mathbf{w}^H(n-1) \mathbf{x}(n) \quad (10)$$

The initial value of $\mathbf{P}(n)$ is chosen

$$\mathbf{P}(0) = \delta^{-1} \mathbf{I}, \quad (11)$$

where δ is the regularization parameter (a positive constant) and \mathbf{I} is the identity matrix. This initial value assures the non-singularity of the correlation matrix $\Phi(n)$. In the case of a stationary environment or a slowly time-varying one, the parameter δ should be assigned a small value for high signal-to-noise ratio (SNR) and a large value for low SNR [12].

Next, in order to analyze the behavior of the RLS algorithm in finite precision implementation, let us examine some of its main parameters. Following (1) and (3), the expectations of the cost function $J(n)$ and of the matrix $\Phi(n)$ are

$$E\{J(n)\} \cong \frac{1-\lambda^n}{1-\lambda} E\{|e(n)|^2\}, \quad (12)$$

$$E\{\Phi(n)\} \cong \frac{1-\lambda^n}{1-\lambda} \mathbf{R}, \quad (13)$$

where \mathbf{R} is the correlation matrix of input data. It can be noticed that $J(n)$ is a biased estimate of $E\{|e(n)|^2\}$ and similarly $\Phi(n)$ is a biased estimate of \mathbf{R} , i.e.,

$$E\{J(n)\} \Big|_{n \rightarrow \infty} \cong \frac{1}{1-\lambda} E\{|e(n)|^2\}, \quad (14)$$

$$E\{\Phi(n)\} \Big|_{n \rightarrow \infty} \cong \frac{1}{1-\lambda} \mathbf{R}. \quad (15)$$

Some classes of applications, e.g., [6], [7] require a high memory algorithm, which means that the value of the exponential weighting factor λ is very close to 1. In this case very large values for the parameters from (14) and (15) could result, causing unwanted finite precision effects in a practical implementation. Apparently, the RLS algorithm avoids this problem by using the inverse of the matrix $\Phi(n)$. Therefore, the maximum values of the elements of the matrix $\mathbf{P}(n)$ result in the initialization phase of the algorithm, according to (11). Nevertheless, the “reverse” problem persists because the values of the elements of the matrix $\mathbf{P}(n)$ decrease towards very small values close to zero, when λ is close to 1.

For example let us consider the following scenario. The fixed point two’s complement arithmetic with a word length of $B + 1$ bits is used and the input signal is a white Gaussian noise, so that $\mathbf{R} = \sigma_x^2 \mathbf{I}$, where σ_x^2 is the input signal variance. We assume the input signal power upper bounded, so that

$$\sigma_x^2 \leq a, \quad (16)$$

where a is a positive constant. In addition, for the RLS algorithm to work, a persistent excitation condition [13] must be imposed, i.e.,

$$\sigma_x^2 \geq b, \quad (17)$$

where b is a positive constant. Following (13) results

$$\Phi(n) \Big|_{n \rightarrow \infty} = \frac{\sigma_x^2}{1-\lambda} \mathbf{I}, \quad (18)$$

$$\mathbf{P}(n) \Big|_{n \rightarrow \infty} = \frac{1-\lambda}{\sigma_x^2} \mathbf{I}. \quad (19)$$

Consequently, the elements of the main diagonal of $\mathbf{P}(n)$, denoted here by $\mathbf{P}_{(i,i)}(n)$, are asymptotically bounded by

$$\frac{1-\lambda}{a} \leq \mathbf{P}_{(i,i)}(n) \leq \frac{1-\lambda}{b}. \quad (20)$$

On the other hand, according to (11), it results that

$$\mathbf{P}_{(i,i)}(0) = \frac{1}{\delta}. \quad (21)$$

Therefore, a scaling procedure is required to avoid overflow phenomenon. The scaling factor $0 < s < 1$ has to be chosen such that

$$sM < 1, \quad (22)$$

where

$$M = \max \left\{ \frac{1}{\delta}, \frac{1-\lambda}{b} \right\}. \quad (23)$$

Nevertheless, reducing the values of the elements of $\mathbf{P}(n)$ by scaling may lead to a stalling phenomenon. This phenomenon appears when $\mathbf{P}(n)$ becomes a zeros matrix, so that, according to (8), the RLS algorithm is “frozen”. To avoid this situation it is necessary that

$$s \frac{1-\lambda}{a} > 2^{-B}. \quad (24)$$

Consequently,

$$\frac{a \cdot 2^{-B}}{1-\lambda} < s < \frac{1}{M} \quad (25)$$

and that implies

$$2^B > \frac{aM}{1-\lambda}. \quad (26)$$

In (23) the case $(1-\lambda)/b > 1/\delta$ is improbable for a value of λ very close to 1, so that usually $M = 1/\delta$ and the algorithm is not sensitive to decrease of excitation.

3. MODIFIED RLS COST FUNCTION

Taking into account the previous discussion it would be very helpful to use an unbiased estimator of the matrix $\Phi(n)$. For this reason, the cost function from (1) can be modified as follows [14]:

$$\bar{J}(n) = (1-\lambda) \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 = \lambda \bar{J}(n-1) + (1-\lambda) |e(n)|^2. \quad (27)$$

In this case

$$E\{\bar{J}(n)\} \cong (1-\lambda^n) E\{|e(n)|^2\} \quad (28)$$

is an asymptotically unbiased estimator of the mean square-error.

Following this idea we have to perform the same modification in (3) and (4) obtaining

$$\begin{aligned} \bar{\Phi}(n) &= (1-\lambda) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^H(i) = \\ &= \lambda \bar{\Phi}(n-1) + (1-\lambda) \mathbf{x}(n) \mathbf{x}^H(n) \end{aligned} \quad (29)$$

$$\begin{aligned} \bar{\theta}(n) &= (1-\lambda) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) d^*(i) = \\ &= \lambda \bar{\theta}(n-1) + (1-\lambda) \mathbf{x}(n) d^*(n) \end{aligned} \quad (30)$$

According,

$$E\{\bar{\Phi}(n)\} \cong (1-\lambda^n) \mathbf{R} \quad (31)$$

is an asymptotically unbiased estimator of the correlation matrix.

As a consequence of these modifications, the Kalman vector from (7) has to be evaluated as [14]

$$\bar{\mathbf{k}}(n) = \frac{\bar{\mathbf{P}}(n-1) \mathbf{x}(n)}{\frac{\lambda}{1-\lambda} + \mathbf{x}^H(n) \bar{\mathbf{P}}(n-1) \mathbf{x}(n)}, \quad (32)$$

where $\bar{\mathbf{P}}(n)$ is the inverse of the matrix $\bar{\Phi}(n)$. The others algorithm's relations remain the same [i.e., equations (8), (9), and (10)].

Next, let us perform a brief convergence analysis of this modified RLS algorithm. First we assume that the desired response $d(i)$ and the tap-input vector $\mathbf{x}(i)$ [see (2)] are related by the linear regression model

$$d(i) = \mathbf{w}_0^H \mathbf{x}(i) + e_0(i), \quad (33)$$

where \mathbf{w}_0 is the regression parameter vector of the model and $e_0(i)$ is the measurement noise, assumed to be white with zero mean and variance σ_0^2 , and independent of $\mathbf{x}(i)$. As a result of the initialization procedure, (29) becomes

$$\bar{\Phi}(n) = \lambda^n \delta \mathbf{I} + \bar{\Phi}_0(n), \quad (34)$$

where $\bar{\Phi}_0(n)$ is a particular solution, i.e.,

$$\bar{\Phi}_0(n) = (1-\lambda) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^H(i). \quad (35)$$

Using (34), (35), (33), and (30) in (5) we get

$$\begin{aligned} (1-\lambda) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^H(i) \mathbf{w}(n) + \lambda^n \delta \mathbf{w}(n) = \\ = (1-\lambda) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^H(i) \mathbf{w}_0 + (1-\lambda) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) e_0^*(i) \end{aligned} \quad (36)$$

Taking the expectation of both sides of (36) and taking into account that

$$E\left\{\sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) e_0^*(i)\right\} = \mathbf{0}, \quad (37)$$

$$E\left\{\sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^H(i)\right\} = \frac{\mathbf{R}}{1-\lambda}, \quad (38)$$

we obtain

$$(\mathbf{R} + \lambda^n \delta \mathbf{I}) E\{\mathbf{w}(n)\} = \mathbf{R} \mathbf{w}_0. \quad (39)$$

Therefore,

$$E\{\mathbf{w}(n)\} \Big|_{n \rightarrow \infty} = \mathbf{w}_0 \quad (40)$$

so that $\mathbf{w}(n)$ is an asymptotically unbiased estimator of \mathbf{w}_0 . The initial convergence rate of the modified algorithm depends on how "fast" the product $\lambda^n \delta$ decreases to zero.

Performing the same analysis for the classical RLS algorithm we get

$$(\mathbf{R} + (1-\lambda)\lambda^n \delta \mathbf{I}) E\{\mathbf{w}(n)\} = \mathbf{R}\mathbf{w}_0. \quad (41)$$

Comparing (39) to (41) it is obvious that the classical RLS algorithm has a faster initial convergence rate than the modified algorithm for the same λ and δ . Anyway, if we use for the classical RLS algorithm a value of the regularization parameter equal to δ and for the modified algorithm the value

$$\delta' = \delta(1-\lambda) \quad (42)$$

both RLS algorithms achieve the same initial convergence rate.

Finally, let us analyze the dynamic range of the elements of the matrix $\bar{\mathbf{P}}(n)$. In a similar manner as in the case of the classical RLS algorithm we find

$$M' = \max\left\{\frac{1}{\delta'}, \frac{1}{b}\right\}, \quad (43)$$

$$2^{-B} a < s < \frac{1}{M'}, \quad (44)$$

so that

$$2^B > aM'. \quad (45)$$

For $\delta' = \delta$, the probability of stalling phenomenon becomes significantly lower. Nevertheless, if M' is chosen as $M' = 1/\delta'$, the probability of overflow due to small signal becomes higher.

It can be concluded that the fast initial convergence rate, being dependent on δ , is conditioned by the numerical resolution. Even in the modified version, the estimate of the mean square error is biased at the beginning of the process and similar property results for the matrix $\bar{\Phi}(n)$. In addition, the initial value $\mathbf{P}(0)$ from (11) is an arbitrary starting point, without any relation with the input signal, but important for the initial convergence rate. We propose a more natural way for initialization of the cost function, suggested by the well-known averaging algorithm

$$\tilde{J}(n) = \begin{cases} \frac{n-1}{n} \tilde{J}(n-1) + \frac{1}{n} |e(n)|^2, & 1 \leq n \leq N \\ \frac{N-1}{N} \tilde{J}(n-1) + \frac{1}{N} |e(n)|^2, & n > N \end{cases} \quad (46)$$

In the first N steps, $\tilde{J}(n)$ is the sample mean of $|e(n)|^2$, i.e.,

$$\tilde{J}(n) = \frac{1}{n} \sum_{i=1}^n |e(i)|^2 \quad (47)$$

Alternatively, the above cost function can be written as

$$\tilde{J}(n) = \lambda(n) \tilde{J}(n-1) + (1-\lambda(n)) |e(n)|^2 \quad (48)$$

where

$$\lambda(n) = \begin{cases} \frac{n-1}{n} & \text{for } 1 < n \leq \frac{1}{1-\lambda} \\ \lambda & \text{for } n > \frac{1}{1-\lambda} \end{cases} \quad (49)$$

For $n = 1$, $\lambda(1) = 0$ is not acceptable so that the iterations start from $n = 2$, considering the initial value

$$\bar{\mathbf{P}}(1) = x(0)^{-2} \mathbf{I} \quad (50)$$

In order to agree assumption (17) the algorithm starts only when $x(0) > b$. The initial convergence rate depends on the starting value $x^2(0)$ but this dependence is considerably reduced because of the low memory behavior (i.e., small λ) in the initial part of the process. The main advantage of the algorithm consists of the fact that $\bar{\Phi}(n)$ is an unbiased estimator of \mathbf{R} , almost every time.

4. QRD-LSL ALGORITHMS WITH REDUCED DYNAMICS OF PARAMETERS

The classical RLS is not very frequently used in practical application mainly due to its high computational complexity (on the order of L^2). For this reason, the fast RLS algorithms are preferred in practice (e.g., [15]). Among these, the QRD-LSL algorithm represents one of the most attractive choices, mainly due to its robust numerical features [9].

In order to extend the idea of the modified cost function from Section 3 to the case of the QRD-LSL algorithm, let us consider the time series $x(1), x(2), \dots, x(n)$ (i.e., the input signal) that occupies the time interval $1 \leq i \leq n$, assuming that $x(i) = 0$ for $i \leq 0$. Most of the notations from [1] will be involved in the following development. The data matrix used in a least-squares estimation problem can be expressed as

$$\mathbf{A}_{m+1}(n) = \begin{bmatrix} x^*(1) & 0 & 0 \\ \mathbf{d}_{f,m-1}(n-1) & \mathbf{A}_{m-1}(n-2) & \mathbf{d}_{b,m-1}(n-2) \\ x^*(n) & \mathbf{x}_{m-1}^H(n-1) & x^*(n-m) \end{bmatrix}, \quad (51)$$

where subscript $m = 1, 2, \dots, L$ is the prediction order and

$$\mathbf{A}_{m-1}(n-2) = \begin{bmatrix} x^*(1) & 0 & \dots & 0 \\ x^*(2) & x^*(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x^*(n-2) & x^*(n-1) & \dots & x^*(n-m) \end{bmatrix}, \quad (52)$$

$$\mathbf{x}_{m-1}(n-1) = [x(n-1), \dots, x(n-m+1)]^T, \quad (53)$$

$$\mathbf{d}_{f,m-1}(n-1) = [x^*(2), \dots, x^*(n-1)]^T, \quad (54)$$

$$\mathbf{d}_{b,m-1}(n-2) = [0, \dots, x^*(n-m-1)]^T. \quad (55)$$

where superscript T denotes transposition. Let us denote by $\mathbf{Q}_m(n)$ an n -by- n unitary matrix and by $\mathbf{R}_m(n)$ an $(m+1)$ -by- $(m+1)$ upper triangular matrix. The exponential weighting matrix from the classical QR-decomposition is

$$\mathbf{\Lambda}_c(n) = \text{diag}\{\lambda^{n-1}, \lambda^{n-2}, \dots, 1\} = \begin{bmatrix} \lambda \mathbf{\Lambda}_c(n-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (56)$$

In order to approach the cost function from (27), a modified form of the previous matrix will be used in our development, i.e.,

$$\mathbf{\Lambda}(n) = (1-\lambda) \text{diag}\{\lambda^{n-1}, \lambda^{n-2}, \dots, 1\} = \begin{bmatrix} \lambda \mathbf{\Lambda}(n-1) & \mathbf{0} \\ \mathbf{0}^T & 1-\lambda \end{bmatrix}. \quad (57)$$

Using (57) and following the QR-decomposition we have:

$$\begin{bmatrix} 1 & \mathbf{0}^T & 0 \\ \mathbf{0} & \mathbf{Q}_{m-2}(n-2) & \mathbf{0} \\ 0 & \mathbf{0}^T & 1 \end{bmatrix} \mathbf{\Lambda}^{1/2}(n) \mathbf{A}_{m+1}(n) = \begin{bmatrix} (1-\lambda)^{1/2} \lambda^{(n-1)/2} x^*(1) & \mathbf{0}^T & 0 \\ \lambda^{1/2} \mathbf{p}_{f,m-2}(n-1) & \lambda^{1/2} \mathbf{R}_{m-2}(n-2) & \lambda^{1/2} \mathbf{p}_{b,m-2}(n-2) \\ \lambda^{1/2} \mathbf{v}_{f,m-2}(n-1) & \mathbf{0} & \lambda^{1/2} \mathbf{v}_{b,m-2}(n-2) \\ (1-\lambda)^{1/2} x^*(n) & (1-\lambda)^{1/2} \mathbf{x}_{m-1}^H(n-1) & (1-\lambda)^{1/2} x^*(n-m) \end{bmatrix} \quad (58)$$

Let $\mathbf{B}(n)$ denote the matrix on the right-hand term of (58). We use a unitary matrix $\mathbf{P}(n-2)$ to annihilate the vector $\mathbf{v}_{b,m-2}(n-2)$, except for its first element, denoted by

$\sqrt{J_{m-1}^b(n-2)}$. A new element is generated, namely, $\pi_{f,m-1}(n-1)$, in the first column. Next, we use the unitary matrix $\mathbf{T}_{m-2}(n-1)$ to update the vectors $\mathbf{p}_{f,m-2}(n-1)$, $\mathbf{p}_{b,m-2}(n-2)$, and the matrix $\mathbf{R}_{m-2}(n-2)$. Angle-normalized forward and delayed backward prediction errors, $\varepsilon_{f,m-1}(n)$ and $\varepsilon_{b,m-1}(n-1)$, are generated in complex conjugate forms and all the elements of the vector $\mathbf{x}_{m-1}^H(n-1)$ are annihilated, i.e.,

$$\begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{T}_{m-2}(n-1) \end{bmatrix} \begin{bmatrix} \mathbf{P}(n-2) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \mathbf{B}(n) = \begin{bmatrix} (1-\lambda)^{1/2} \lambda^{(n-1)/2} x^*(1) & \mathbf{0}^T & 0 \\ \mathbf{p}_{f,m-2}(n) & \mathbf{R}_{m-2}(n-1) & \mathbf{p}_{b,m-2}(n-1) \\ \lambda^{1/2} \pi_{f,m-1}(n-1) & \mathbf{0}^T & \lambda^{1/2} \sqrt{J_{m-1}^b(n-2)} \\ \lambda^{1/2} \mathbf{v}_{f,m-1}(n-1) & \mathbf{0} & \mathbf{0} \\ (1-\lambda)^{1/2} \varepsilon_{f,m-1}^*(n) & \mathbf{0}^T & (1-\lambda)^{1/2} \varepsilon_{b,m-1}^*(n-1) \end{bmatrix} \quad (59)$$

Let $\mathbf{C}(n)$ denote the matrix on the right-hand term of (59). We can write:

$$\begin{bmatrix} 1 & \mathbf{0}^T & 0 & \mathbf{0}^T & 0 \\ \mathbf{0} & \mathbf{I}_{m-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0}^T & c_{b,m-1}(n-1) & \mathbf{0}^T & s_{b,m-1}^*(n-1) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{n-m-1} & \mathbf{0} \\ 0 & \mathbf{0}^T & -s_{b,m-1}(n-1) & \mathbf{0}^T & c_{b,m-1}(n-1) \end{bmatrix} \mathbf{C}(n)$$

$$\begin{bmatrix} (1-\lambda)^{1/2} \lambda^{(n-1)/2} x^*(1) & \mathbf{0}^T & 0 \\ \mathbf{p}_{f,m-2}(n) & \mathbf{R}_{m-2}(n-1) & \mathbf{p}_{b,m-2}(n-1) \\ \pi_{f,m-1}(n) & \mathbf{0}^T & \sqrt{J_{m-1}^b(n-1)} \\ \lambda^{1/2} \mathbf{v}_{f,m-1}(n-1) & \mathbf{0} & \mathbf{0} \\ (1-\lambda)^{1/2} \varepsilon_{f,m}^*(n) & \mathbf{0}^T & 0 \end{bmatrix} \quad (60)$$

where \mathbf{I} denotes identity matrices. Furthermore, the following updates result:

$$c_{b,m-1}(n-1) = \frac{\lambda^{1/2} \sqrt{J_{m-1}^b(n-2)}}{\sqrt{J_{m-1}^b(n-1)}}, \quad (61)$$

$$s_{b,m-1}(n-1) = \frac{(1-\lambda)^{1/2} \varepsilon_{b,m-1}^*(n-1)}{\sqrt{J_{m-1}^b(n-1)}}, \quad (62)$$

$$J_{m-1}^b(n-1) = \lambda J_{m-1}^b(n-2) + (1-\lambda) |\varepsilon_{b,m-1}(n-1)|^2, \quad (63)$$

$$\begin{aligned} \varepsilon_{f,m}(n) &= c_{b,m-1}(n-1) \varepsilon_{f,m-1}(n) \\ &\quad - s_{b,m-1}^*(n-1) \pi_{f,m-1}^*(n-1) \left(\frac{\lambda}{1-\lambda} \right)^{1/2}, \quad (64) \end{aligned}$$

$$\begin{aligned} \pi_{f,m-1}^*(n) &= c_{b,m-1}(n-1) \lambda^{1/2} \pi_{f,m-1}^*(n-1) \\ &\quad + s_{b,m-1}(n-1) \varepsilon_{f,m-1}(n) (1-\lambda)^{1/2} \quad (65) \end{aligned}$$

In the same manner, another two transformations over the matrix $\mathbf{B}(n)$ are performed, i.e.,

$$\begin{aligned} &\begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{T}_{m-2}(n-1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{K}(n-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \mathbf{B}(n) \\ &= \begin{bmatrix} \lambda^{1/2} \sqrt{J_{m-1}^f(n-1)} & \mathbf{0}^T & \lambda^{1/2} \pi_{b,m-1}(n-1) \\ \mathbf{p}_{f,m-2}(n) & \mathbf{R}_{m-2}(n-1) & \mathbf{p}_{b,m-2}(n-1) \\ \mathbf{0} & \mathbf{O} & \lambda^{1/2} \mathbf{v}_{b,m-1}(n-1) \\ (1-\lambda)^{1/2} \varepsilon_{f,m-1}^*(n) & \mathbf{0}^T & (1-\lambda)^{1/2} \varepsilon_{b,m-1}^*(n-1) \end{bmatrix} \quad (66) \end{aligned}$$

Let $\mathbf{D}(n)$ denote the matrix on the right-hand term of (66). We can write:

$$\begin{aligned} &\begin{bmatrix} c_{f,m-1}(n) & \mathbf{0}^T & s_{f,m-1}^*(n) \\ \mathbf{0} & \mathbf{I}_{n-2} & \mathbf{0} \\ s_{f,m-1}(n) & \mathbf{0}^T & c_{f,m-1}(n) \end{bmatrix} \mathbf{D}(n) \\ &= \begin{bmatrix} \sqrt{J_{m-1}^f(n)} & \mathbf{0}^T & \pi_{b,m-1}(n) \\ \mathbf{p}_{f,m-2}(n) & \mathbf{R}_{m-2}(n-1) & \mathbf{p}_{b,m-2}(n-1) \\ \mathbf{0} & \mathbf{O} & \lambda^{1/2} \mathbf{v}_{b,m-1}(n-1) \\ 0 & \mathbf{0}^T & (1-\lambda)^{1/2} \varepsilon_{b,m}^*(n) \end{bmatrix} \quad (67) \end{aligned}$$

Similarly, a set of recursive relations for the forward prediction part of the algorithm are obtained, i.e.,

$$c_{f,m-1}(n) = \frac{\lambda^{1/2} \sqrt{J_{m-1}^f(n-1)}}{\sqrt{J_{m-1}^f(n)}}, \quad (68)$$

$$s_{f,m-1}(n) = \frac{(1-\lambda)^{1/2} \varepsilon_{f,m-1}^*(n)}{\sqrt{J_{m-1}^f(n)}}, \quad (69)$$

$$J_{m-1}^f(n) = \lambda J_{m-1}^f(n-1) + (1-\lambda) |\varepsilon_{f,m-1}(n)|^2, \quad (70)$$

$$\begin{aligned} \varepsilon_{b,m}(n) &= c_{f,m-1}(n) \varepsilon_{b,m-1}(n-1) \\ &\quad - s_{f,m-1}^*(n) \pi_{b,m-1}^*(n-1) \left(\frac{\lambda}{1-\lambda} \right)^{1/2}, \quad (71) \end{aligned}$$

$$\begin{aligned} \pi_{b,m-1}^*(n) &= c_{f,m-1}(n) \lambda^{1/2} \pi_{b,m-1}^*(n-1) \\ &\quad + s_{f,m-1}(n) \varepsilon_{b,m-1}(n-1) (1-\lambda)^{1/2}. \quad (72) \end{aligned}$$

Finally, for the joint-process estimation part of the algorithm we have

$$\varepsilon_{m+1}(n) = c_{b,m}(n) \varepsilon_m(n) - s_{b,m}^*(n) p_m^*(n-1) \left(\frac{\lambda}{1-\lambda} \right)^{1/2}, \quad (73)$$

$$p_m^*(n) = c_{b,m}(n) \lambda^{1/2} p_m^*(n-1) + s_{b,m}(n) \varepsilon_m(n) (1-\lambda)^{1/2}. \quad (74)$$

Summarizing, the proposed algorithm uses (61)–(65) for the backward prediction part, together with (68)–(72) for the forward prediction part, and (73), (74) for the joint-process estimation. Let us call this modified algorithm by QRD-LSL-m1. According to the discussion from the end of Section 3, this type of algorithm achieve a slower initial convergence rate than the classical one (for the same initialization parameters) but we may use a variable exponential weighting factor according to (49) in order to speed up the initial convergence of the algorithm. Therefore, it results a second modified algorithm, which we called QRD-LSL-m2.

The computational complexity of the proposed algorithms is similar with the complexity of the classical QRD-LSL (i.e., around $20L$), while the computational amount of the RLS algorithm is around $3L^2$.

5. SIMULATION RESULTS

For the first set of experiments we consider an adaptive “system identification” configuration [1]. In this class of applications an adaptive filter is used to provide a linear model that represents the best fit (in some sense) to an unknown system. The adaptive filter and the unknown system are driven by the same input; the unknown system output supplies the desired response for the adaptive filter. These two signals are used to compute the estimation error, in order to adjust the filter coefficients. Our input signal is

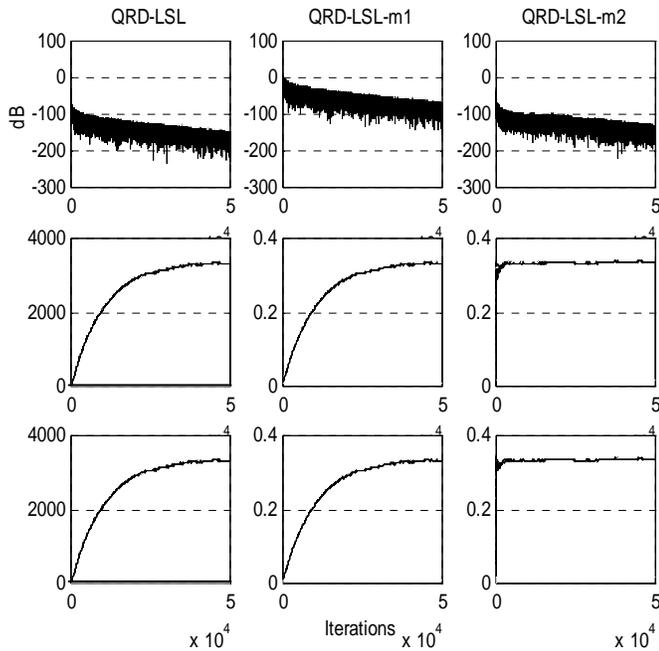


Fig. 1. Square errors [dB] and the cost functions of the classical QRD-LSL algorithm (column 1) and the modified versions QRD-LSL-m1 (column 2) and QRD-LSL-m2 (column 2), in a system identification setup. Row 1 – Square errors [dB]; Row 2 – J^b cost functions; Row 3 – J^f cost functions.

a random sequence with an uniform distribution in the interval $(-1;1)$. The order of the adaptive filter is $M = 64$. In Fig. 1 are presented the convergence curves and the evolution of the cost functions for the classical QRD-LSL algorithm and its modified versions, QRD-LSL-m1 and QRD-LSL-m2, using $\lambda = 0.9999$. In the case of both modified algorithms the values of the cost functions can not exceed 1 [according to (63) and (70)]. Hence, due to the reduced dynamic range of these parameters, the “effort” for scaling procedures is significantly reduced. On the other hand, the cost functions of the classical algorithm will be upper bounded (theoretical) by $1/(1 - \lambda)$, which leads to large values when λ is close to 1. Also, it can be noticed that the QRD-LSL-m2 achieve the same initial convergence rate as the classical QRD-LSL algorithm.

The previous simulation was performed using the full precision of Matlab programming environment. Next, the algorithms are implemented in fixed-point precision, using a fixed-point DSP with a word length of 16 bits (15 bits for the magnitude and one sign bit). The usage of a higher precision (e.g., 24 or 32 bits) could lead to better performances but also increases the implementation costs. As a practical aspect of this work we choose to illustrate the algorithms performance in a noise reduction scenario (Fig. 2) [1]. In this type of application, the adaptive filter is use to synthesize at its output a replica of the perturbation that corrupts the voice signal.

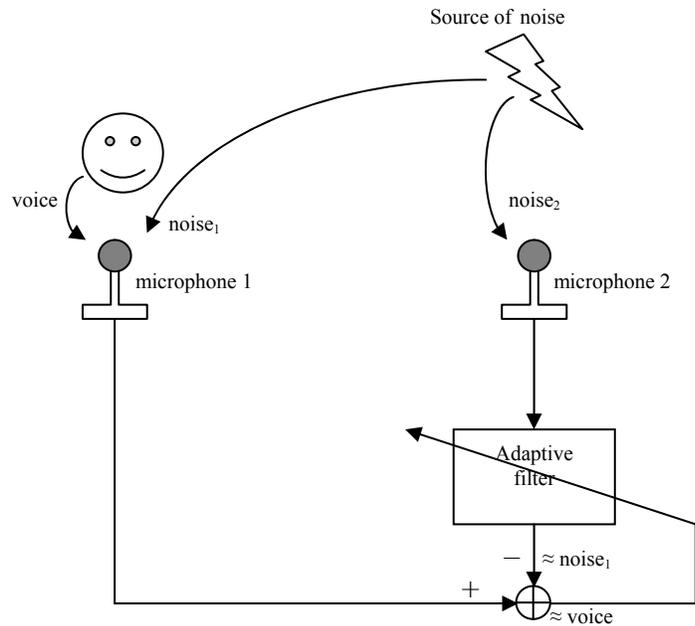


Fig. 2. Adaptive noise reduction scheme.

In the original QRD-LSL algorithm the asymptotic value for the cost functions are $1/(1 - \lambda)$. Since λ is very close to 1, the original algorithm will certainly produce overflow and thus needs to be scaled, i.e., the cost functions must be right-shifted by a number of bits such chosen as to avoid the overflow in the convergence state. A simple calculus shows that the optimum number of bits to shift-right the cost functions is $B_s = \lceil -\log_2(1 - \lambda) \rceil$, where $\lceil \bullet \rceil$ denotes superior integer round. Nevertheless, this further leads to the reduction of the effective number of bits, especially when λ is very close to 1 and eventually to a low signal to quantization noise ratio, altering the algorithm performances. For this reasons we choose to compare the QRD-LSL-m2 algorithm (since it has a faster initial converge rate as compared to QRD-LSL-m1) with the normalized least-mean-square (NLMS) algorithm [1], which is one of the most common solution for noise reduction [16], [17]. Since the computational amount of the NLMS is around $3L$, this algorithm is “cheaper” (in terms of complexity) as compared with the proposed QRD-LSL algorithm. Nevertheless, the performances of the NLMS algorithm are strongly reduced when high order adaptive filters and non-stationary inputs (e.g., speech) are used. In these cases, the RLS-based algorithms rule.

The results of the noise reduction experiment are presented in Figs. 3 and 4, using two type of noise, i.e., a white Gaussian noise with $SNR = 10\text{dB}$ (Fig. 3) and a highway noise (Fig. 4). The last one is more severe because it is a non-stationary signal. In both cases the QRD-LSL-m2 algorithm outperforms the NLMS algorithm.

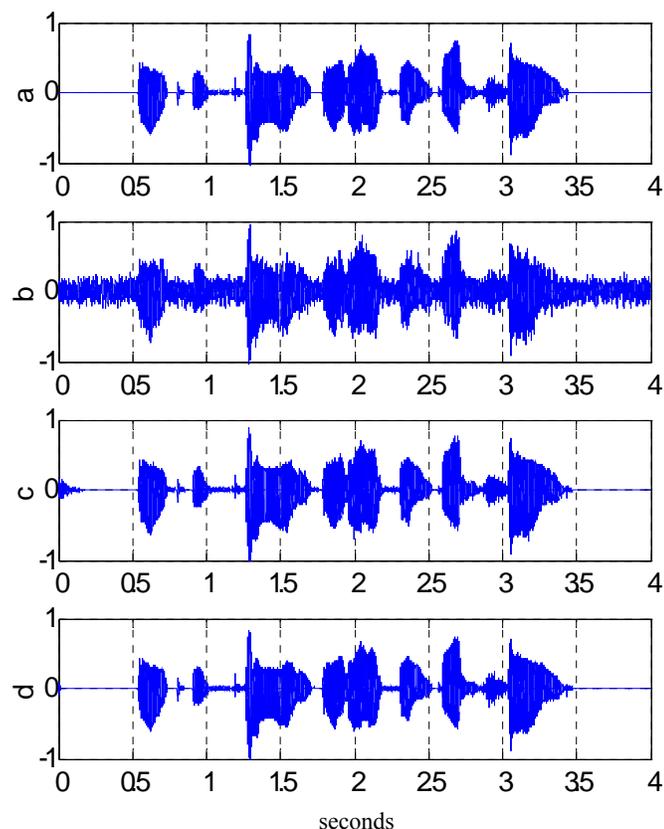


Fig. 3. (a) original signal; (b) corrupted signal (with white Gaussian noise); (c) recovered signal using the NLMS algorithm; (d) recovered signal using the QRD-LSL-m2 algorithm.

In the first case (Fig. 3) the subjective tests indicate a mean opinion score (MOS) of 3.9 for the NLMS algorithm and 4.5 for the QRD-LSL-m2 algorithm. In the second case the difference becomes more apparent, i.e., 3.2 for the NLMS algorithm and 4.1 for the QRD-LSL-m2. Note that the MOS scale is from 1 to 5, where 1 means very poor and 5 means excellent quality. This was evaluated in a subjective manner, as the average of the scores given by 20 listeners.

6. DISCUSSION

A first goal of this paper was to present and analyze a modified version of the RLS adaptive algorithm with improved features for fixed-point implementation. The basic idea was to use an asymptotically unbiased estimator for the cost function. In this manner we try to prevent the stalling phenomenon which may appear when a high memory RLS algorithm is implemented using fixed-point arithmetic. A brief convergence analysis of the RLS algorithms was performed, together with a discussion concerning the proper scale factor, which has to be chosen in order to avoid the overflow and stalling effects.

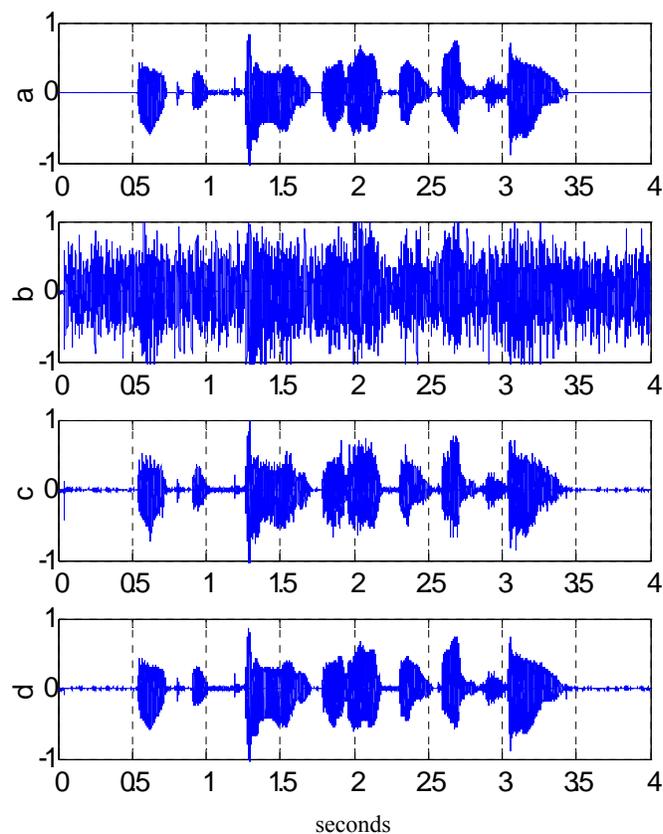


Fig. 4. (a) original signal; (b) corrupted signal (with highway noise); (c) recovered signal using the NLMS algorithm; (d) recovered signal using the QRD-LSL-m2 algorithm.

In the case of the modified RLS algorithm only the initial convergence rate is affected when it operates with the same value of the regularization parameter as the classical RLS algorithm. Choosing the value of this parameter according to (42), the modified algorithm achieves the same initial convergence rate as the classical one. Moreover, the variable exponential weighting factor from (49) speeds up the initial convergence rate of this algorithm, leading to a reasonable compromise between the convergence rate and dynamic range of the algorithm's parameters.

The procedure presented in the case of the RLS algorithm was developed and applied in the case of the QRD-LSL algorithm, which is a fast member of the RLS family. Two modified versions of the QRD-LSL algorithm were proposed. Based on the asymptotically unbiased estimator for the cost functions, we improve the behavior of these algorithms when dealing with fixed-point arithmetic. As expected, only the initial convergence rate of the QRD-LSL-m1 algorithm is affected when it operates with the same parameters as the classical QRD-LSL algorithm. Also, the variable exponential weighting factor used for QRD-LSL-m2 algorithm speeds up its initial

convergence rate. The simulations performed in both Matlab and fixed-point DSP support the theoretical findings.

7. CONCLUSIONS

A class of RLS algorithms suitable for fixed-point implementation was presented in this paper. The proposed approach was applied in the case of the QRD-LSL algorithm. The performance of the resulted algorithm was evaluated in a noise reduction scenario, obtaining promising results.

8. REFERENCES

- [1] S. Haykin, *Adaptive Filter Theory – Fourth Edition*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, 2002.
- [2] T. Adali and S. H. Ardalan, “Convergence and error analysis of the fixed point RLS algorithm with correlated inputs,” *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP 1990*, vol. 3, pp. 1479-1482.
- [3] I.K. Proudler, J.G. McWhirter, and T.J. Shepherd, “QRD-based lattice filter algorithms,” *Proc. SPIE*, vol. 1152, pp. 56-67, 1989.
- [4] J.-T. Yuan, “A modified QRD for smoothing and a QRD-LSL smoothing algorithm,” *IEEE Trans. Signal Processing*, vol. 47, no. 5, pp. 1414-1420, May 1999.
- [5] J.-T. Yuan and J.-N. Lee, “Narrow-band interference rejection in DS/CDMA systems using adaptive (QRD-LSL)-based nonlinear ACM interpolators,” *IEEE Trans. Vehicular Technology*, vol. 52, no. 2, pp. 374-379, Mar. 2003.
- [6] C. Paleologu, S. Ciochină, and A.A. Enescu, “A network echo canceler based on a SRF QRD-LSL adaptive algorithm implemented on Motorola StarCore SC140 DSP,” *Lecture Notes in Computer Science, Springer-Verlag*, vol. 3124, pp. 560-567, June 2004.
- [7] G. Rombouts and M. Moonen, “Fast QRD-lattice-based unconstrained optimal filtering for acoustic noise reduction,” *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 6, pp. 1130-1143, Nov. 2005.
- [8] J.-T. Yuan, C.-A. Chiang, and C.-H. Wu, “A square-root-free QRD-LSL interpolation algorithm,” *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP 2008*, pp. 3813-3816, Apr. 2008.
- [9] C. Paleologu, A. A. Enescu, S. Ciochină, and F. Albu, “QRD-LSL Adaptive Algorithms Suitable for Fixed-Point Implementation,” *Proc. IEEE Advanced International Conference on Telecommunications (AICT)*, pp. 163-167, Venice, Italy, May 2009.
- [10] C. Paleologu, F. Albu, A. A. Enescu, and S. Ciochină, “Modified SRF-QRD-LSL Adaptive Algorithm with Improved Numerical Robustness,” *IARIA International Journal on Advances in Systems and Measurements*, vol. 2, no. 1, pp. 56-65, 2009.
- [11] J. Benesty and Y. Huang, Eds, *Adaptive Signal Processing--Applications to Real-World Problems*. Springer-Verlag, Berlin, Germany, 2003.
- [12] G. V. Moustakides, “Study of the Transient Phase of the Forgetting Factor RLS,” *IEEE Transactions on Signal Processing*, vol. 45, no. 10, pp. 2468-2476, Oct. 1997.
- [13] M. H. Verhaegen, “Round-off Error Propagation in Four Generally-Applicable, Recursive Least-Squares Estimation Schemes,” *Automatica*, vol. 25, no. 3, pp. 437-444, 1989.
- [14] S. Ciochină, C. Paleologu, and A.A. Enescu, “On the Behaviour of RLS Adaptive Algorithm in Fixed-Point Implementation,” *Proc. of IEEE Int. Symp. on Signals, Circuits and Systems, SCS 2003*, vol. 1, pp. 57-60, July 2003.
- [15] C. Paleologu, A. A. Enescu, and S. Ciochină, “Recursive Least-Squares Lattice Adaptive Algorithm Suitable for Fixed-Point Implementation,” *Proc of IEEE International Conference on Electronics, Circuits and Systems, ICECS 2006*, pp.1105-1108, Dec. 2006
- [16] E. Haensler and G. Schmidt, Eds., *Topics in Acoustic Echo and Noise Control*. Springer-Verlag, Berlin, Germany, 2006.
- [17] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Noise Reduction in Speech Processing*. Springer-Verlag, Berlin, Germany, 2009.

Adaptive Rate Voice over IP Quality Management Algorithm

Eugene S. Myakotnykh

Centre for Quantifiable Quality of Service in
Communication Systems (Q2S)¹,
Norwegian University of Science and Technology,
Trondheim, Norway
E-mail: esm@q2s.ntnu.no

Richard A. Thompson

Telecommunications and Networking Program,
Department of Information Sciences and
Telecommunications, University of Pittsburgh,
Pittsburgh, USA
E-mail: thompson@mail.sis.pitt.edu

Abstract—The quality of voice-over-IP communication relies significantly on the network that transports voice packets because this network does not usually guarantee available bandwidth, delay, and loss that are critical for real-time voice traffic. The solution proposed here is to manage a voice-over-IP stream dynamically, changing encoding parameters as needed to assure quality. The paper proposes an adaptive-rate control algorithm that establishes interaction between a VoIP sender and a receiver, and manages voice quality in real-time. Simulations demonstrate that the system provides better average communications quality than traditional fixed-rate VoIP.

Keywords—adaptive VoIP; E-model; Packetization; Speech quality; Voice-over-IP (VoIP)

I. INTRODUCTION

A packet-switched network does not provide reliable transport of real-time data: it does not guarantee available bandwidth, end-to-end delay and packet loss parameters, which are critical for real-time voice traffic. Most of the previous research in the VoIP quality has concentrated on networking issues of QoS management. Many different algorithms were developed to improve the transport of packetized voice traffic, including traffic classification (Differentiated Services technology [1, 2]), bandwidth reservation (Integrated Services architecture [3], Resource Reservation Protocol [4]), congestion avoidance, Multi-Protocol Label Switching technology [5], and others. These approaches use different techniques to decrease transmission delay and/or probability of voice packet congestion in the network and make the Internet more suitable for real-time traffic transmission. However, these methods often do not provide acceptable results nor do they solve the problem completely because (1) not all equipment and service providers support same QoS protocols and quality standards, and (2) the Internet is a dynamic media; the technologies often cannot react to changing network conditions and manage the quality of every communication session in real-time.

The alternative approach is to adaptively manage voice encoding parameters on the sender side depending on the network conditions. A proper choice of a voice payload size or compression may enhance the quality of VoIP because it can dynamically change a configuration of a VoIP system, so that the system matches a current state of the network. This approach proposes to adjust a voice stream to the network and change parameters of the stream in real-time, depending on the network state.

This paper proposes an adaptive quality management mechanism that changes speech encoding parameters on the sender side in real-time depending on network impairments. This area is in the early stage of its development and investigation of many questions related to VoIP quality measurement and management, dependencies between multiple parameters and the resulting quality, is necessary to develop intelligent and efficient adaptive VoIP codecs.

When designing an adaptive quality management algorithm, several questions must be answered:

- (1) Which factor (or factors) should be used to make a decision that a change of certain speech encoding parameters is required or not required at a given moment of time?
- (2) How the end-user speech encoding parameters (packet duration and compression) affect VoIP quality under different network conditions. What encoding parameters should be changed and how to do it? How often should such a decision be made (per talkspurt, periodically, etc.)?
- (3) How should feedback from the receiver be sent to the sender side?

Although this paper speaks about adaptive VoIP quality management and uses a set of narrowband voice codecs for analysis and as an example, the result and the approach can potentially be extended to a wider set of narrowband codecs, to wideband encoding and to IP-based audio in general.

This article is an extended version of paper [6] and it is organized as follows: the next Section provides an overview of related research in the area of adaptive VoIP. Section III describes network scheme and assumptions used in our simulation studies. Section IV investigates the effect of speech encoding parameters (packet size and compression/encoding variation) on quality of VoIP communications; Section V describes decision-making parameters for the proposed algorithm; Section VI shows the actual adaptive voice quality management mechanism. The results of the simulation study are presented in Section VII. Conclusion is drawn in Section VIII.

¹ “Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence” appointed by The Research Council of Norway, funded by the Research Council, NTNU and UNINETT. <http://www.q2s.ntnu.no>

II. RELATED WORK

A. Adaptive Quality Management

A number of studies investigating the idea of adaptive voice quality management are available. Multiple papers (for example, Qiao et al. [7], Seo et al. [8], Matta et al. [9], and others) adopt the GSM/UMTS Adaptive Multi-Rate codec (AMR) [10] for the IP-network. The AMR codec was originally developed for wireless networks and the decision about adaptation of its encoding parameters is based on channel interference. The philosophy behind AMR is to lower a codec rate as the channel interference increases and thus enabling more error correction to be applied. Evidently, the process of adaptive quality management in the IP-network is different than that in wireless communications: there is no channel interference, there are IP packets instead of radio signals, and the threshold choice and management process will be different. The papers above present ideas of how to use the existing encoding scheme in the IP network.

A real-time change of speech encoding parameters can be achieved through variation of voice packet size or compression (encoding scheme). As the voice packet travels through the Internet, an overhead with control information is added to the voice payload. The size of overhead added to a voice packet is 40 bytes (the application layer Real-Time Transport Protocol (RTP) [11] header is 12 bytes; the transport layer UDP header is 8 bytes; and the IP header is 20 bytes), and it is significant compared to a typical voice payload size. If the G.711 codec [12] is taken as an example, the length of one 10-ms packet is 80 bytes. So, the RTP/UDP/IP overhead is 50% of the payload size and the total bandwidth required for the voice stream transmission is 96 kbps.

Changing end-user parameters may significantly affect bandwidth requirements per call and, as a result, its quality. For example, increasing a voice stream IP-rate will lead to increased quality, but the probability of quality degradation due to potential congestion may also increase because of higher channel capacity requirements. Several papers (for example, [13], [14], [15]) studied how changing of the encoding parameters affects VoIP communication quality.

Various parameters can be used to detect congestion in the network and make a decision about encoding parameters adaptation. For example, Bolot et al. [16] and Mohamed et al. [17] perform adaptive rate control based on packet loss statistics. The computational quality model called the E-model [18] is used in [9], [19] and [20]. Ngamwonwattana, [21] makes decision about codec rate adaptation based on moving average thresholds of delay and packet loss, and proposes sending control messages from the receiver "on demand".

The recent paper of F. Sabrina and J. Valin [22] describes an adaptive mechanism using Speex codec [23]. The authors propose a novel criterion to get feedback about the network condition and a mechanism for adjusting the encoding bit rate based on the feedback and on instantaneous speech properties.

B. Voice-over-IP Quality Assessment

This paper uses several metrics to estimate quality of voice-over-IP communications. Voice quality measurement methodologies include subjective testing (involves human subjects and considered as the ultimate way of quality evaluation), and objective techniques (signals comparison or computational methods). The leading subjective criterion of voice quality measurements is the Mean Opinion Score (MOS), which is defined in the ITU-T Recommendation P.800 [24]. MOS is a score of voice quality as perceived by a large number of people listening to speech over a communication system. This Recommendation uses the scale from one to five and the MOS of some session of voice transmission is the average estimate of voice quality rates assigned by individual listeners (1 – bad, 2 – poor, 3 – fair, 4 – good, 5 – excellent). Subjective tests are usually complex and time-consuming and cannot be used for real-time quality assessment. Objective mechanisms are needed for this purpose.

Here in the project, the real-time decision about parameters adaptation is based on two computed metrics: an instantaneous quality level, which is measured per talkspurt using the computational E-model [18], and, in addition to this, a change of integral perceptual quality level, which is estimated based on a model developed by AT&T [25]. The original version of the E-model is relatively complex. It includes about 20 input parameters representing various terminal, network and environmental quality impairment factors. In the narrowband voice-over-IP area, the simplified version of this model is often used with default values for all but a few parameters - delay and packet loss. The model computes speech quality rating on a 100-point scale as:

$$R = R_0 - I_d - I_{e\text{-eff}} \quad (1)$$

R is the resulting indicator of voice quality; R_0 is the maximum score, achievable by codecs in the absence of loss and significant delay; I_d is the impairment factor caused by end-to-end delay (a function of delay); $I_{e\text{-eff}}$ is the effective equipment impairment factor, which depends on used codec, and also on packet loss rate and effectiveness of packet loss concealment algorithms. The E-Model is based on the concept that "psychological factors on the psychological scale are additive" [18]. It does not imply that the factors are uncorrelated, but only that their contributions to the estimated impairments are independent and each impairment factor can be computed separately. Numerical characteristics for different codecs and more details about the model can be found in [18], [26], [27]. A similar model for wideband telephony is proposed in [28].

The E-model uses a special mapping function to establish a relationship between the 100-point R -scale and the traditional MOS scale (Equation 2).

$$MOS = 1 + 0.035R + R(R - 60)(100 - R) \cdot 7 \cdot 10^{-6} \quad (2)$$

This model is not a perfect tool to calculate absolute quality level, but it is acceptable for measuring variations in quality.

Detailed investigations of this question by NTT Lab (Japan) [29] concluded that correlation coefficient of results provided by the E-model with subjective human testing results is about 80%. Since, goal of these algorithms is to achieve noticeable relative quality improvement, the E-model can be used to track changes in quality.

The E-model estimates average quality during a certain period of time. But quality, as perceived by the end-user, depends not only on significance of a quality distortion, but also when this distortion happened during a communication session. The effect, which reflects the way that a listener remembers call quality, is called “recency” effect. This effect implies that periods of low or high quality positioned at the end of a speech sample have a stronger influence on the overall session quality than when such periods are positioned in the beginning of the sample. In tests conducted by AT&T [25], a burst of noise was created and moved from the beginning to the end of a 60-second call. When the noise was at the start of the call, users reported a higher MOS score than when the noise was at the end of the call. Tests reported by France Telecom [30] showed a similar effect. The effect is believed to be due to the tendency for people to remember the most recent events or possibly due to auditory memory, which typically decays over a 15-30 second interval [30].

Further discussion about these parameters will be provided below. Using these metrics, the novel mechanism to adaptively manage speech encoding parameters is proposed.

III. SIMULATION DESIGN

A. Network Model

The proposed adaptive speech quality management algorithm is tested using a simulation implemented in Matlab. The network topology for the simulation is shown in Fig. 1. A simplified scenario with a single place of potential congestion is analyzed. The congestion may be caused by bursty background traffic through the router. The link capacity is 5 Mbps, but, actually, this number is not very important. But, a portion of voice and data traffic in the network is important: a significant difference in VoIP quality is seen over the same network, with the same total (voice plus data) average traffic load, but with different voice-to-data traffic ratios. The presence of large data packets with bursty behavior creates “instability” in the voice transmission process, which causes additional delay variation (jitter) and, as a result, higher delay and/or packet loss. The

propagation and network processing delay is assumed fixed at 50 ms (it may be noticeably higher in real networks). The bottleneck router uses FIFO queuing and the drop-tail mechanism in case of overflow. The router has a finite queue size (64 Kbyte), enough to keep packets in queue for about 100 milliseconds.

B. Voice Encoding

This project uses parameters of three narrowband codecs for developing and simulating the proposed adaptive quality management scheme: the G.711 [12] (PCM encoding with no compression), the G.726 [31] (ADPCM encoding with 2:1 compression), and the G.729a [32] (CS-ACELP encoding with 8:1 compression). While codec may have different voice payload sizes, the discrete values 10 ms, 20 ms, 30 ms will be used. So, nine different sets of encoding parameters are used. These codecs and parameters are chosen for analysis because: 1) they provide a relatively high level of quality (higher than or close to the toll-grade); 2) their quantitative characteristics are known in terms of maximum encoded speech quality in the absence of packet loss and significant delay [18]; and 3) the difference in channel capacity consumption in these codecs is significant: for example, the G.711 codec with 10 ms voice payload size requires 96 kbps per stream (64 kbps of audio bit rate and 32 kbps to send the RTP/UDP/IP overhead); the 30 ms G.729 codec needs just 18.7 kbps channel (8 kbps audio rate and 10.7 kbps overhead bit rate). Selecting one of the nine sets of encoding parameters will be based on metrics calculated on the receiver side. These codecs are chosen as examples; similar adaptive mechanisms can be used with a different set of narrowband codecs, with some set of wideband codecs or with a combination of narrowband and wideband codecs.

C. Call Characteristics

It is assumed that speech codecs with variable parameters (packet size, compression) are used. Delay and packet loss statistics is calculated on the receiver side. The E-model [18] is applied to get a quality metrics based on these parameters. The E-model parameters for the selected codecs are defined in [18] and [27]. Simulated call duration is 120 seconds. Silence suppression is not used (for simplicity). Voice stream may include a single call or a group of calls (aggregated voice traffic). All calls in the group use the same speech encoding algorithm; calls can be managed simultaneously; and all have the same behavior (the quality of all calls degrades equally).

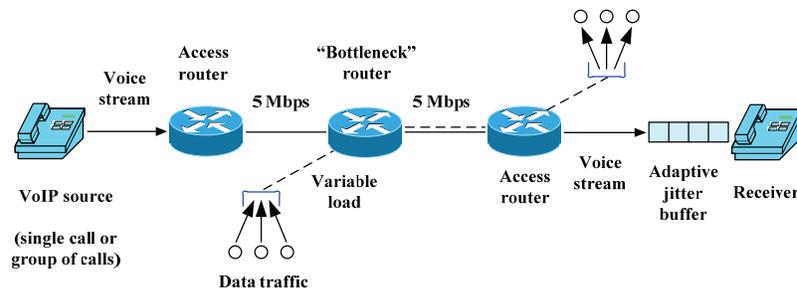


Figure 1: Simulated network structure

D. Background Traffic Modeling

It is desirable to generate background traffic with characteristics similar to traffic patterns in the Internet. Studies revealed that Internet traffic exhibits properties of (1) self-similarity, (2) burstiness and (3) long-range dependency (LRD) [33]. The self-similar process shows that short-time traffic behavior patterns are close to long-time patterns. LRD means that there is a statistically significant correlation across large time-scales [34]. In [35, 36], it is suggested to use multiple Pareto aggregation sources with a Pareto index parameter $\alpha < 2$. Our study uses this approach by generating Pareto On/Off traffic from 10 different sources. The traffic from each source consists of sending packets at a fixed rate only during the On periods, whereas the Off periods are idle. The packet sizes are: 64 bytes (60% of packets), 550 bytes (25% of packets) and 1500 bytes (15% of packets). The aggregated traffic will have all the required characteristics. The Network Simulator NS-2 [37] also uses this model of background traffic generation.

It is important that the model does not separate the generated traffic into TCP and UDP flows. But the approach is good to model Internet traffic behavior, even in congested networks, ignoring nonlinearities arising from the interaction of multiple traffic sources because of network resource limitations and TCP's feedback congestion control algorithm [38]. One possible reason is that more than 90 percent of TCP sessions in the Internet are very short (1-2 seconds) and exchange less than 10 Kbytes of data [39].

According to recent research [40, 41], the nature of traffic in the Internet changes because of a significant increase of peer-to-peer traffic and assumptions used in this section may not be true in future. The proportion of peer-to-peer traffic in the network increased significantly during the last several years and achieves 50% of the total traffic. This fact may change two assumptions: a) the packet-size distribution of the Internet traffic will change; b) TCP session will exchange more data, will be longer and the TCP back-off mechanism should be simulated. This study uses the On/Off Pareto model and the TCP-based model will be analyzed in future.

E. Jitter Buffer Management

Jitter is a variation in packet transit delay caused by queuing and serialization effects on the path through the network. It is eliminated by jitter buffers, which temporarily store arriving packets and send them to a receiver in equal intervals. The buffer may have a fixed size, but if is too small, a lot of packets may be discarded because of a significant delay variation. This will negatively affect speech quality. Increasing the jitter buffer allows waiting longer for delayed packets but increases the overall end-to-end delay, which also negatively affects speech quality. A lot of research focuses on adaptive jitter buffer strategies to find some optimal point in the tradeoff between the end-to-end delay and packet loss and to optimize speech quality dynamically.

The basic adaptive playout algorithm of Ramjee *et al.* [42] is used in the simulation. It calculates two statistics to make a jitter buffer adaptation decision. For each arriving

packet, it computes the expected mean and variation in the end-to-end delay (d_i and v_i , respectively). Specifically, the end-to-end delay estimate for packet i is computed as

$$d_i = \alpha \cdot d_{i-1} + (1-\alpha) \cdot n_i \quad (3)$$

where n_i is the i -th packet delay. The variation is: as:

$$v_i = \alpha \cdot v_{i-1} + (1-\alpha) \cdot |d_i - n_i| \quad (4)$$

Packet playout time in this algorithm is calculated as:

$$p_i = t_i + d_i + \beta v_i \quad (5)$$

where t_i is the time the packet was sent. $\alpha = 0.875$ and $\beta = 4$.

While these equations estimate d_i and v_i for each packet, playout time p_i is adjusted only in the intervals between talkspurts (periods of speech). Different papers use periods between 200 and 700 ms to describe talkspurt durations and there is no agreement about the "best" number. The ITU P.59 [43] recommendation specifies an artificial on/off model for generating human speech with the talkspurts and silence intervals of 227 ms and 596 ms. Jiang and Schulzrinne [44] reported mean spurts and gaps of 293 ms and 306 ms in experiments with the G.729 codec. The durations of 300 ms both for silence and active speech periods are used in the simulation.

IV. THE EFFECT OF END-USER PARAMETERS ON VOIP QUALITY

A. Effect of Background Traffic on VoIP Quality

Before investigating the effect of the parameters on speech quality, it is important to demonstrate that the quality is affected by not only high link utilization, but also by the proportion of voice and data traffic in the network. This hypothesis is rather evident: the data traffic is bursty and it may cause a sudden congestion in the network, which results in higher delays and/or losses of voice packets.

The simulation design is based on the network scheme and the assumptions described in the previous section. The narrowband G.711 codec with 64 kbps of audio bit rate is used in the example. Speech quality is measured in MOS depending on the total link utilization U (voice and data traffic; average during the call) and D (ratio of data traffic and total mixed traffic). Note that U is not an average ISP network utilization; this is a utilization of the bottleneck link during the considered 120-second simulation. The results and the standard deviations are presented in Fig. 2.

The results show that the behavior of voice traffic significantly depends on the presence of, and load of, the data traffic in the network. Even a relatively small volume of data traffic can cause a significant degradation of voice traffic. While this conclusion is intuitively clear, the simulation study gives numerical estimates. If the average link utilization exceeds 70%, adaptive quality management mechanisms can potentially be used and change of packet size and/or speech compression may decrease the quality degradation effect. But, this number (70%) depends on the

assumptions and will be slightly different with other adaptive jitter buffer algorithms, different approaches for background traffic generation, and assumptions about talkspurt duration

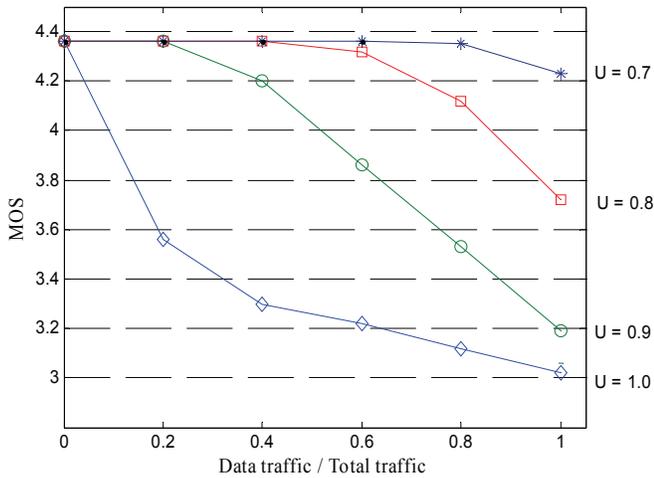


Figure 2: Speech quality depending on traffic structure

It is necessary to understand that, in most cases, the structure of traffic in the network (in the congested router) is not known and not possible to measure (unless a user or a provider controls all hardware devices along a path of a given call). But the results of the study presented below still helpful in understanding of how variations of voice stream parameters affect its quality and in which scenarios the end-user codec adaptation may be beneficial.

B. Effect of Packet Size Variation on VoIP Quality

The effect of packet size variation is difficult to describe theoretically because many of the parameters affecting quality (delay, loss, jitter) are not independent and improving one parameter may cause a decline in another. Some effects of packet size on speech quality are very clear, others are less evident. Four main relationships are identified:

- (1) Increasing packet size leads to an increase of end-to-end delay. If the delay is not too large, the direct impact of packet size increase is very small and not perceptually noticeable [18]. But, if the delay is significant, an additional increase of packet duration may be noticeable.
- (2) Increasing packet size leads to decrease of the IP-rate per call. This may decrease congestion in the network and improve the quality of communication. The dependency is also evident and the question about the effectiveness of voice transmission was briefly discussed in Section 2.
- (3) There are two less evident, but also important effects of packet size variation on VoIP quality: (3) A loss of one “long” packet has more significant negative effect on speech quality than a random loss of several “small” packets. Mathematical representation of this effect in the E-model can be found in [18].
- (4) In presence of data traffic in the network, increase of voice packet size decreases link utilization, but increases the data-to-voice traffic ratio. As demonstrated in the previous section, this may cause additional “instability” in the

network, which may result in higher jitter, delay or loss. This factor may affect the resulting speech quality, but it is not clear how significant the effect might be.

It is seen that increasing packet size causes different effects on the speech quality. Since, the result is difficult to predict theoretically, it is investigated using simulation. The portions of voice (number of calls) and data traffic in the network were changed, and the average call quality was measured for different packet sizes. Speech quality is measured in MOS depending on the portion of voice traffic in the network V , the total link utilization U (average during the call) and the packet size PS (changes from 10 to 50 ms). For each combination of $\{V, U, PS\}$, 100 experiments were run. The standard deviation of MOS scores does not exceed 0.15 MOS. The simulation results for several specific values of V (portion of voice traffic) are shown in Fig. 3.

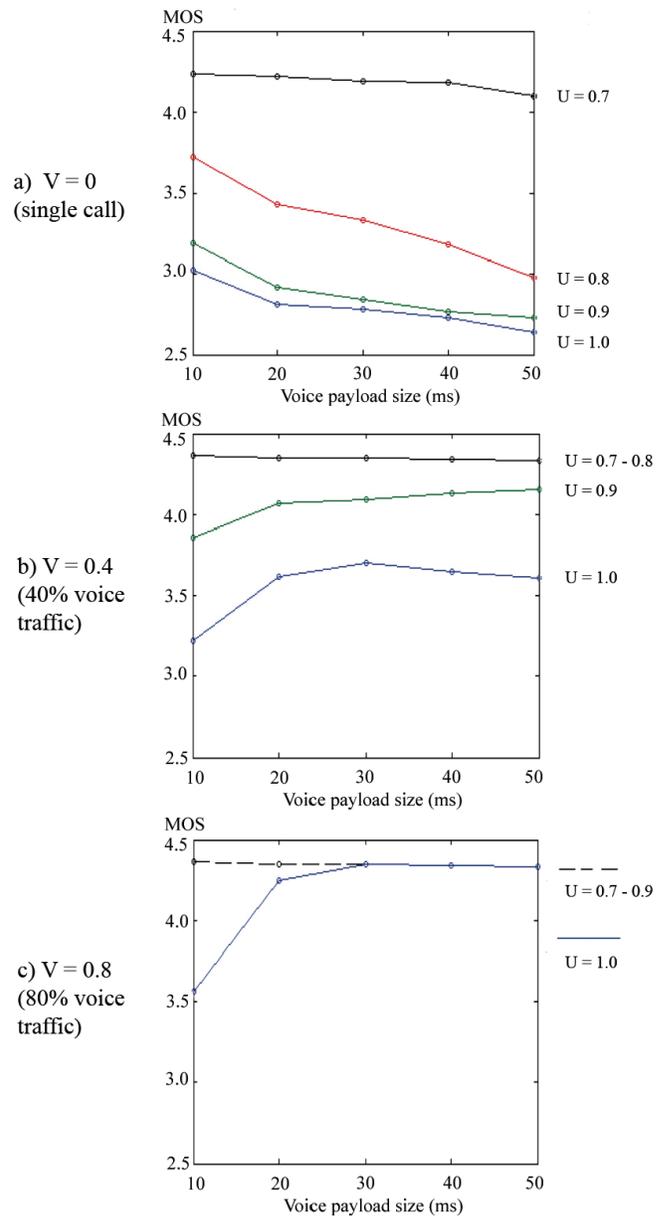


Figure 3: Effect of packet size on VoIP quality

In the case of a single call or a relatively small number of calls (Figure 3a), change of packet size does not provide any quality improvement because it does not influence the situation in the network. Increase of packet size in this scenario makes the situation even worse; it is recommended to keep it small (10 ms).

Congested links with dominated voice traffic are more stable (Figures 3b, 3c). They can handle higher utilization (for example, 80% or even 90%) without noticeable quality degradation (again, these numbers depend on the assumptions about the network structure, traffic generation, etc). In these scenarios, quality can be improved even further using higher rate codecs.

Based on the study, the “lower bound” of managed voice traffic in a congested link when change of packet size provides noticeable improvement in quality is about 30%. With higher voice load, packet size variation improves quality despite the multiple negative effects discussed in the previous section. In these cases, a 20-ms packet size is enough to increase average call quality; 30 ms can provide even better quality under some scenarios.

C. Effect of Compression on VoIP Quality

In addition to the packet size variation, multiple encoding algorithms with compression can be used. Compressed speech often has even smaller IP-rate, but there is a more significant loss in codec quality due to compression. In addition to the G.711 codec, this study considers the G.726 codec with 2:1 compression and the G.729a codec with 8:1 compression. The goal of this section is to answer the question: if a number of simultaneously managed calls does not change, would it be better to use the codec with higher compression under certain network conditions or packet size variation would be more effective.

Similar to the previous section, change of compression causes opposing effects on VoIP quality:

(1) Increasing compression generally leads to a decreased codec quality. For example, the maximum quality, which can be achieved by the G.729a codec under ideal conditions, is noticeably less than that of the G.711 codec.

(2) Increasing compression leads to decreased IP-rate per call. This may decrease congestion in the network (especially if a group of calls is managed) and improve the quality of communication.

(3) Increasing compression not only decreases codec quality, but also the effectiveness of packet loss concealment algorithms. Concealment of compressed packets is less effective. For example, the loss of 1% of the G.729a packets has more significant negative impact on speech quality than the same loss of the G.711 packets.

(4) As in the previous section, if there is data traffic in the network, voice compression increases the data-to-voice traffic ratio, which may negatively result on speech quality.

Mean VoIP communications qualities in several scenarios: 1) the G.711 codec is used (no compression, 20 ms packet size); 2) the G.726 codec (2:1 compression; 20 ms packet size); 3) the G.729a codec (8:1 compression; 20 ms packet size). Notations: V - portion of voice traffic in the network, U - total bottleneck link utilization (voice and data

traffic; average during a call). U and V are measured based on the G.711 codec, 10-ms packet size. Fig. 4. shows the simulation results for several specific values of V (portion of voice traffic).

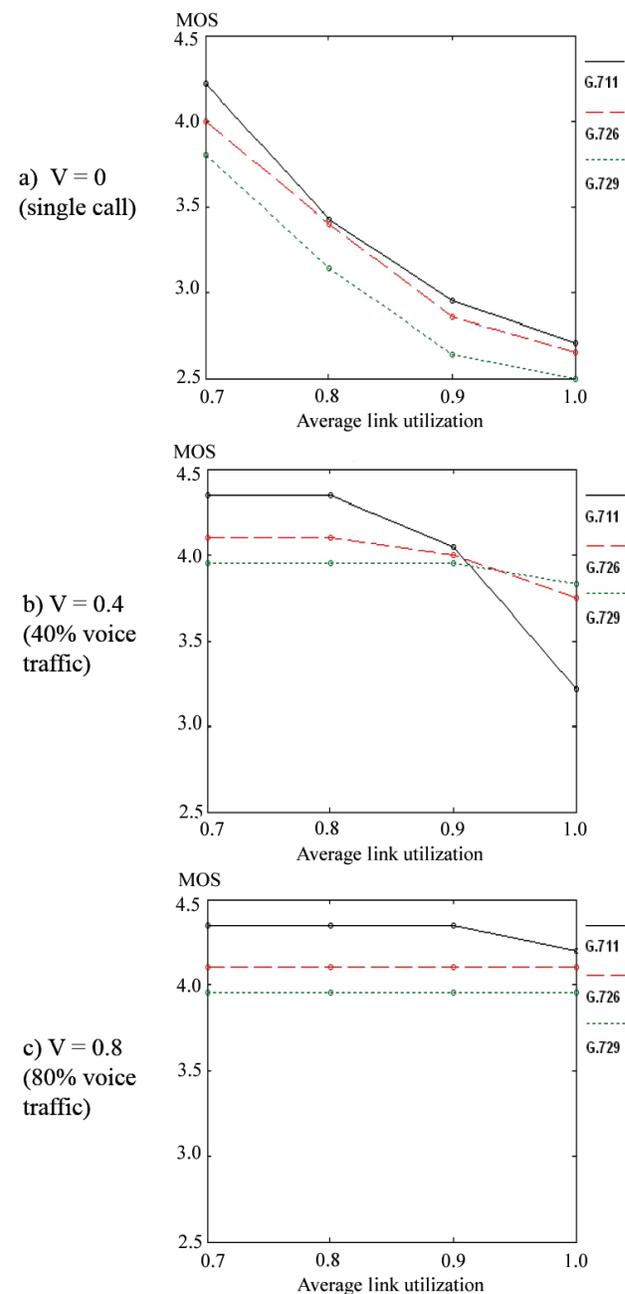


Figure 4: Effect of compression on VoIP quality

The results demonstrate that, when a portion of managed voice traffic in the network is relatively small (Fig. 4a), neither the packet size variation nor the increase of compression (or even both of them together) provide improvement in quality. For best achievable quality under the given scenario, it is necessary to use the best available codec (the G.711 or even wideband codecs) and small voice payload size (10 ms; smaller packet sizes make the

communication process very inefficient). If portion of managed voice traffic is more significant, changing packetization provides better resulting speech quality than compression variation (until a certain level of the bottleneck link utilization) despite the fact that the compressed speech uses less channel capacity (Fig. 4c). Only when the bottleneck link is very heavily congested (more than 90%), the compressed encoding provides better quality (Fig. 4b).

D. Summary

The presented results are based on assumptions about network structure, background traffic pattern, speech characteristics (talkspurt duration), which are described in the previous section. Other jitter buffer algorithms or, for example, a different model of speech representation (talkspurt and silence periods) will change numerical results, but the general conclusion will remain the same: in congested networks try to manage voice payload size first, because it may provide higher resulting quality than codecs with compression; if this does not help – change a voice stream bit rate using both higher compression and packet size variation. The results are consistent with other studies mentioned in Section 2, but provide approximate quantitative criteria when end-user variation of encoding parameters may be effective for average call quality improvement.

This study had assumed that average background traffic load in the network is known. In real networks this information is not available to the end-points, so mechanisms would be needed to estimate the effect of the network on communication quality. Monitoring a call or a group of calls on the receiver side, we do not know average link utilization and average volume of background traffic in the network. Number of simultaneously managed calls is known, but this information is not too important: the quality depends on the proportion of data traffic in the network and, even if there are many calls, often it cannot be said with high confidence whether the network is voice-only or even if voice traffic dominates in the network. For this reason, the conclusions from this part cannot be used directly.

It is important to remember that two adaptive mechanisms are going to work simultaneously: a) the proposed variable sender-based encoding mechanisms and b) the receiver's adaptive jitter buffer. The adaptive jitter buffer mechanism is used to improve short-term quality (its fast reaction does not change the encoding characteristics, but manages the delay-loss tradeoff). Sender-based management is designed to improve long-term voice flow characteristics (to choose the encoding scheme that best matches the given network conditions).

V. DECISION-MAKING PARAMETERS FOR ADAPTIVE SPEECH QUALITY MANAGEMENT

Which parameters should be taken into account to make a decision about quality adaptation? One variant is to use, for example, the mean delay, the moving average delay, or loss statistics. This approach has already been used in several papers mentioned above. It would not be acceptable to analyze these parameters separately: high packet loss definitely means a significant degradation in quality but low

(or absence of) packet loss does not mean an absence of degradation because the adaptive jitter buffer size can be very significant and we can get high end-to-end delay instead of the loss. So, it would be better to use these parameters together. In other words, one must measure quality, which depends on end-to-end delay, loss, and codec characteristics. This project does not analyze some "less evident" parameters affecting speech quality like echo, attenuation, noise in a channel, etc. The quality can be measured using the computational E-model [18].

How can this model be used? The adaptive algorithms proposed here will measure and manage quality-per-talkspurt. Human communication consists of periods of active speech and periods of silence. The adaptive jitter buffer algorithm changes its buffer size in periods between talkspurts (during silence periods). So, it would be logical to analyze speech quality behavior, and to make decisions about adaptive quality management at the end of a talkspurt (at the end of an active speech period). The E-model can be used to calculate the quality of each talkspurt (referred to as "instantaneous quality"). Packets within a talkspurt have the same end-to-end delay; network loss and jitter buffer loss can easily be counted. The difference between instantaneous quality levels in two consecutive talkspurts can be very significant because of the bursty nature of the background traffic.

But, knowledge of instantaneous call quality is not enough to make a decision about changing the speech encoding parameters. Measurable voice quality can change significantly and immediately but it takes some time for users to understand that the quality has changed. Perceptual (real) speech quality is different from an instantaneous computational quality. Perceptual quality takes into account factors not only during the last short period of time (last second or several seconds), but all quality values and quality variation history starting at the beginning of a call [45]. So, in addition to the computational quality model, it would be useful to estimate (1) instantaneous perceptual speech quality at any moment of time during a call, and (2) integral quality at any moment of time during a call.

The E-model can also be used to measure the average call quality at a given moment of time during a conversation. This metric would probably be acceptable, but this project tries a different approach, using a metric, that describes integral (total) speech quality. Integral quality that is calculated as a mean of instantaneous qualities is not a very good metric. This model does not take into account the history of previous quality variations (frequent variations may result in a relatively high average quality but noticeably smaller real perceptual quality).

Instead of using just a mean MOS metric, integral quality is calculated from the beginning of a call using the perceptual model of Rosenbluth [25]. This model presents a call as a sequence of 8-second intervals. Quality within each interval is calculated as a mean of instantaneous qualities. Integral call quality is calculated as a weighed sum of the qualities of the longer intervals and reflects an opinion that quality levels at the end of a conversation have higher weights on the overall perceived call quality than quality

levels at the beginning of a session. Perceptual quality is usually lower than the average computational quality if there are frequent variations of instantaneous quality levels. This model was justified by subjective experiments performed by AT&T. It is proposed to use the weighting average with weights

$$W_i = \max \left[1, 1 + (0.038 + 1.3 \cdot L_i^{0.68}) \cdot (4.3 - MOS_i)^{\{0.96 + 0.61 \cdot L_i^{1.2}\}} \right] \quad (6)$$

$$MOS_I = \frac{\sum_i W_i \cdot MOS_i}{\sum_i W_i} \quad (7)$$

MOS_I is the integral perceptual call quality; MOS_i is the MOS during the shorter measurement period; L_i is a location of a degradation period (measured on 0-to-1 scale; 0 indicates the beginning of a conversation, 1 is the end of a conversation; the parameter changes proportionally to time starting from the beginning of a call).

This perceptual call quality metrics will be used as one of decision parameters for adaptive speech quality management. If there are concerns about this model, it is possible to calculate the integral quality using weighted average with exponentially distributed weights described in [45]. The idea in Equations 6 and 7 is similar; the representation is a little bit more complex than just the exponentially distributed weighting.

VI. ADAPTIVE VOIP QUALITY MANAGEMENT ALGORITHM

This section describes the proposed adaptive quality management mechanism. The following section provides an example and results of a simulation study.

Step 1: Collect statistics of packet delays before the jitter buffer. If there are multiple calls, it is assumed that the quality degradation pattern of all calls to be similar (this assumption is confirmed by our preliminary simulations). So, just one call from aggregated voice traffic may be chosen for analysis.

Step 2: Calculate packet playout time (Section 3E). This parameter is calculated continuously for each arrived packet but, since the jitter buffer is adjusted only in pauses between talkspurts (between active speech periods), the end-to-end delay is constant within a talkspurt.

Step 3: Calculate the quality of a talkspurt based on the E-model. This calculation includes: (1) calculating packet loss within a talkspurt (network and jitter buffer loss); (2) measuring end-to-end delay, which is constant within a talkspurt. This "quality per talkspurt" is referred to as "instantaneous quality" and denoted as Q_i . The difference between instantaneous quality levels in two consecutive talkspurts can be very significant because of the bursty nature of the background traffic. But, knowledge of this instantaneous call quality is not enough to make a decision about adaptation of speech encoding parameters.

Step 4: Calculate the maximum achievable quality level for a given codec under the given network conditions. This calculation is based on zero packet loss and minimum network delay (the sum of transmission delay, propagation delay and minimum queuing delay). This delay is taken from the analysis of incoming packets delays. Using this minimum network delay, calculate the maximum achievable quality under the given set of encoding characteristics:

$$R = R_0 - I_{e\text{-eff}} - I_d$$

$$I_d = \text{min. network delay} + \text{min. jitter buffer delay}$$

$$MOS = 1 + 0.035R + R(R - 60)(100 - R) \cdot 7 \cdot 10^{-6} \quad (8)$$

MOS is a Mean Opinion Score on the 1-to-5 scale; R is the indicator of voice quality (the 100-point scale) from the computational E-model, $R_0 = 93.2$ is the maximum achievable narrowband quality, I_d is a function of delay; $I_{e\text{-eff}}$ describes impairments related with encoding (codec quality) and packet loss. See the E-model standard [18] for more details about voice quality calculation. The result looks something like in Fig. 5:

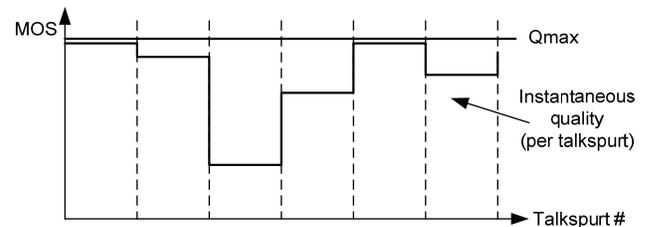


Figure 5: Instantaneous speech quality measurement

Step 5: Continuously calculate the integral voice quality level based on the model expressed by Equations 6 and 7.

Step 6: Decision. The proposed quality management scheme uses three parameters: 1) instantaneous quality level – Q_i ; 2) integral perceptual quality – Q_I ; and 3) maximum quality level achievable under the given set of speech encoding parameters – Q_M . If the number of managed calls (which is assumed to be known) is not significant (for example, one or two), voice flow is concluded to have an insignificant effect on the network, and it would be better to use the best available codec from the very beginning and not to use any quality adaptation strategies.

Similar to [7], two thresholds are used in the algorithm: 0.25 MOS and 0.5 MOS. These thresholds are used only to describe integral quality variation and these numbers are not chosen arbitrarily. A change in quality of 0.2-0.25 MOS is not too significant, but is noticeable by some people; smaller changes in quality are noticeable only by a relatively small percentage of listeners. A change in quality of about 0.5 MOS is very significant and is noticeable by almost everybody. If the best narrowband G.711 codec is used and its quality decreases by 0.5 MOS, the resulting quality will be lower than the toll-grade quality level. If the G.729 codec

is used and its quality degrades by 0.5 MOS, the resulting level of speech quality (about 3.4-3.5 MOS) is considered to be low by most people.

The situation with thresholds for instantaneous quality level is a little more complex. Talkspurt quality can decrease for two reasons: 1) high jitter buffer size, which results in high end-to-end delay and, usually, not significant loss, or 2) packet loss (usually not in the network but on a receiver side caused by significant delay variation and insufficient jitter buffer size). The effect of delay is generally lower: for example, with 150 ms of network and packetization delay and an additional 80 or 100 ms of jitter buffer size, the decrease in quality is about 0.3-0.4 MOS. But, if a bursty loss of packets causes a loss of, for example, only 3 out of 30 packets in a given talkspurt, the decrease in the instantaneous quality equals to 0.9 MOS. If 5 packets out of 30 are discarded, the resulting quality (for the G.711 codec) will be only 2.65 MOS (with a maximum level of 4.3-4.4 MOS). As in the case of integral quality, two thresholds are used: 0.3 and 1.0. If the difference between maximum and instantaneous quality levels does not exceed 1.0 on the MOS scale, the observed packet loss is considered reasonable.

This model does not use quality adaptation mechanisms during the first several seconds of conversation because the perceptual quality model expressed by Equations 7 and 8 is very sensitive to quality variations in the beginning of a call. In the simulation, this period is set to 8 seconds.

The details of the algorithm follow. Consider the differences between two parameters: 1) between the maximum and integral qualities ($Q_M - Q_T$), and 2) between the maximum and instantaneous quality levels ($Q_M - Q_I$). The first difference quantifies total quality variation; the second one describes instantaneous quality changes.

Condition 1:

- if $Q_M - Q_T > 0.5$ // Low or unacceptable level of quality. Something has to be done immediately
 - if $Q_M - Q_I > 1.0$ // Instantaneous quality level is also very low. Not too much can be done in this situation. Switch to the G.729 codec with 30 ms packet size (the worst codec using the minimum IP-rate)
 - ⇒ Action: switch to the G.729 codec, 30 ms packet size
 - if $0.3 < Q_M - Q_I < 1.0$
 - ⇒ Action: keep current settings expecting that adaptive buffer will compensate the degradation
 - if $Q_M - Q_I < 0.3$ // Total quality is very low, but instantaneous quality level is close to maximum. Quality degradation is not seen, so start slowly to improve the codec quality by decreasing packet size
 - ⇒ Action: decrease packet size by 10 ms if a current size is higher

Condition 2:

- if $0.2 < Q_M - Q_T < 0.5$ // Degradation of quality is noticeable, try to improve the situation
 - if $Q_M - Q_I > 1.0$ // Have a long bursty loss of packets. The network is significantly congested.
 - ⇒ Action: use codec with higher compression: if current codec is G.711, switch to G.726; if current codec is G.726, switch to G.729
 - if $0.3 < Q_M - Q_I < 1.0$ // Also, the situation is not good and bursty packet loss is observed
 - ⇒ Action: increase packet size by 10 ms or change codec if current packet size is 30 ms (maximum)
 - if $Q_M - Q_I < 0.3$ // Instantaneous quality is good; expect integral quality improvement
 - ⇒ Action: decrease packet size by 10 ms if a current size is higher

Condition 3:

- if $Q_M - Q_T < 0.2$ // Degradation of quality is not significant but might be noticeable; try to improve the situation
 - if $Q_M - Q_I > 1.0$ // Significant quality degradation. Total quality is good but one more bursty loss can significantly drop overall quality. Try to avoid.
 - ⇒ Action: increase packet size by 10 ms up to 30 ms
 - if $0.3 < Q_M - Q_I < 1.0$ // Assume that this decrease of quality is temporal and due to single loss or increase of end-to-end delay
 - ⇒ Action: keep current settings
 - if $Q_M - Q_I < 0.3$ // Everything is fine: both total and instantaneous qualities are high.
 - ⇒ Action: decrease packet size up to minimum or switch to a better codec

The algorithm is summarized in Table 1.

Step 7: To change current encoding algorithm according to the decision.

The actions defined above cannot be executed immediately. The collected information about instantaneous and integral quality levels has to be transmitted to the sender. The transmission delay can be significant in congested networks. Assume that three consecutive talkspurts on the sender side (TS1, TS2, TS3) are separated by periods of silence (S1, S2). According to assumptions in Section 3, the mean durations of the active speech and silence period are

TABLE I: THE ALGORITHM SUMMARY

| Q(M)-Q(I) | Q(M)-Q(T) | | |
|---------------------|--|---|---|
| | ≤ 0.2 | $0.2 < \dots < 0.5$ | ≥ 0.5 |
| ≤ 0.3 | - if current packet size is higher than 10 ms, decrease it - if current size is 10 ms but used codec is not the G.711, switch to a better codec | - decrease packet size up to 10 ms | - if current packet size is lower than 30 ms, increase it |
| $0.3 < \dots < 1.0$ | - keep current settings | - if current packet size is lower than 30 ms, increase it | - if used codec is not G.729, switch to a codec with higher compression |
| ≥ 1.0 | - if current packet size is lower than 30 ms, increase it | - if used codec is not G.729, switch to a codec with higher compression | - switch to the G.729 codec with 30 ms packet size |

300 milliseconds. Assume that the receiver gets TS1 and makes a decision to send some control information to the sender. The period of time between the departure of the TS1 talkspurt and the arrival of the feedback from the receiver is equal to a round-trip delay (RTT). In congested networks this RTT might be significant and longer than the period of silence between the TS1 and TS2 talkspurts. In this case, the decision about quality adaptation will not be applied to the second talkspurt (TS2); it would be applied to TS2 only if the RTT is less than the S1 duration (300 ms). So, the receiver would not see the result of the requested changes of speech encoding parameters until the TS3 talkspurt, about one second later. The minimum reaction time of the algorithm is 300 ms (when $RTT \leq 300$ ms). If the assumptions about speech/silence duration are different, these numbers will change respectively.

This fact has to be taken into account in the adaptation scheme. So, a restriction is added that, if a receiver analyzes a talkspurt (for example, TS1) and sends a control message to the sender to change speech encoding parameters, the next control message cannot be sent after the next consecutive talkspurt.(TS2); but only after analyzing of TS3, if it is required. This restriction provides more stability to the algorithm.

One more restriction is added. If a decision is made about several consecutive improvements of speech encoding parameters (for example, to decrease the voice payload size from 30 ms to 20 ms and then to 10 ms or to replace a given codec by a better codec) these changes should not be made too quickly because each change causes noticeable increase of IP-rate per call and thus, a higher probability of degradation due to congestion. The preliminary experiments showed that the system is more stable if the receiver waits for four talkspurts (about 2 seconds) between such decisions.

In sender-based control, observations about the network and resulting speech quality must be reported back to the sender. Utilizing RTCP is a common approach: packet loss

and delay variation statistics are included in RTCP reports. These packets are sent periodically, usually every 5 seconds. But, obviously, this type of control is very slow to respond to the network. If the control mechanism must make decisions more frequently, it is necessary to use a different scheme rather than RTCP. But more frequent periodic call control may introduce additional traffic in the network. The paper assumes that the adaptive quality management mechanism sends control messages, not periodically, but on demand, that is, when a change of sender parameters is required. As the simulation results will demonstrate, a decision to vary the encoding parameters is made less frequently than per talkspurt. Also, if a group of calls is managed, just one message can be sent to deliver feedback from the receiver, but not to control every call independently. This approach will not create a significant amount of additional control traffic in the network

VII. SIMULATION RESULTS

A. Example

Fig. 6 demonstrates an example of the statistics collected in one of the simulations. The first picture shows the packet delay and the jitter buffer size calculated in Step 2. If packet delay (blue line) is higher than jitter buffer size (red line), the packet is lost (discarded) and restored using a current codec's packet loss concealment mechanism. The second figure demonstrates instantaneous (per talkspurt) speech quality (blue line, Step 3) and integral (perceptual) quality (red line, Step 5). The third picture shows the background traffic rate averaged over 10 ms intervals, and also the average long-term traffic rate.

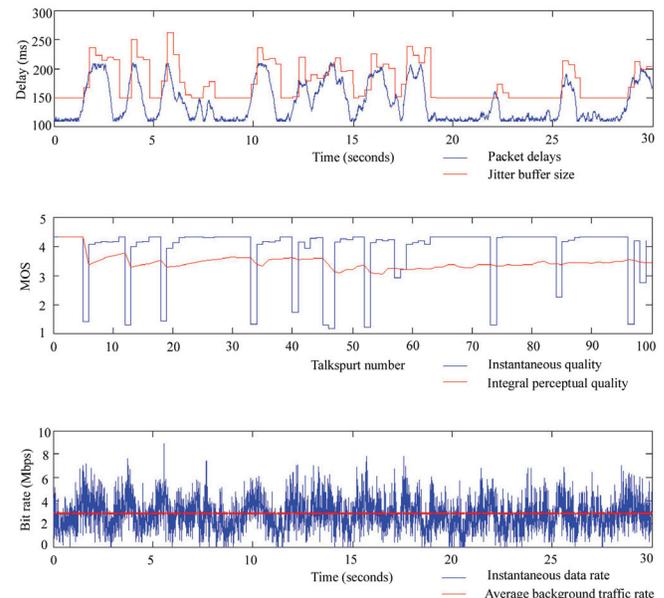


Figure 6: Measured speech quality statistics

Figures 7 shows the quality of the VoIP stream when exactly the same background traffic pattern exists and the stream is managed by the proposed adaptive quality

management mechanism. This example considers a rather congested network with 40% of voice and 50% of data traffic. Blue (upper) line is the instantaneous quality level; red (lower) line is the integral quality level. Because of the real-time adaptation, the algorithm helped to choose encoding settings and change them dynamically to improve average communication quality.

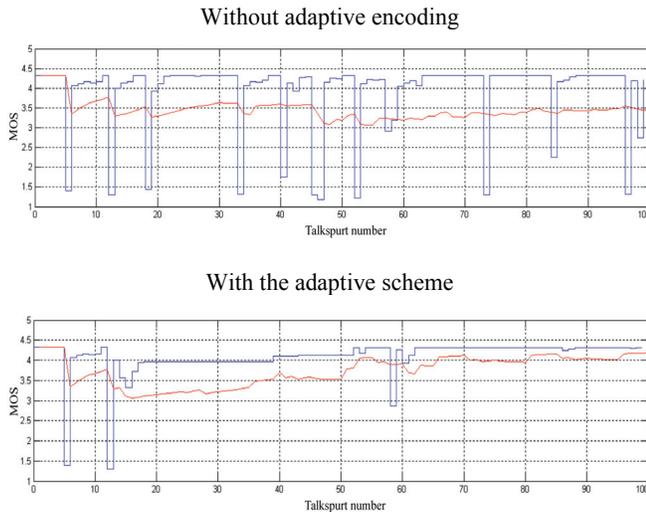


Figure 7: Speech quality comparison

B. Simulation study

The simulation described in Section 3 was used to analyze effectiveness of the proposed adaptive speech quality management algorithm. As it was mentioned in Section 4, the algorithm (and the approach of adaptive encoding in general) is not effective if a portion of managed real-time voice traffic in the network (in the bottleneck) is less than approximately 30%. The simulation was performed for 25% and 50% of average background data traffic load in the network and for different number of voice calls. The graphs in Fig. 4 show average qualities with and without adaptive encoding calculated using the E-model.

The results compare average MOS scores in the two scenarios; the simulation was executed 20 times for each set of parameters (more extensive simulation study will be performed in future). Fig. 8 demonstrates that adaptive encoding allows decreasing of quality degradation in case of network congestion. For example, having 50% average background traffic load and 20 simultaneous fixed-rate G.711 calls, average quality of these calls was around 3.4 MOS. Adaptive algorithm detects degradation in quality caused by traffic burstiness and high network utilization, adaptively changes packetization or encoding and results in better average quality (around 3.9). The average increase in quality is quite significant.

Although the increase in average quality is seen with the proposed algorithm, two important things should be emphasized. 1) In heavily congested networks, individual MOS scores for a call can be noticeably different. Running the simulation multiple times and having the same volume of

traffic and the same number of VoIP calls in the network, significant difference in call quality may be seen. This happens because of different background traffic patterns (burstiness of generated traffic in a given simulation). 2) Increase in average quality does not mean that call quality is improved in all individual simulations. For example, having 50% average background traffic load and 18 G.711 calls in the network, out of 20 runs, 2 experiments showed slight decrease in quality (less than 0.2 MOS) and two had even more significant drop in quality (about 0.4-0.5 MOS). This also happens because of background traffic burstiness: in certain cases, the algorithm expects a significant congestion in the network and future quality degradation, and switches to compressed lower quality codec. But, if situation in the network suddenly improves, the adaptation decision becomes not effective. The “intelligence” of the algorithm should be improved to decrease the number of such failures.

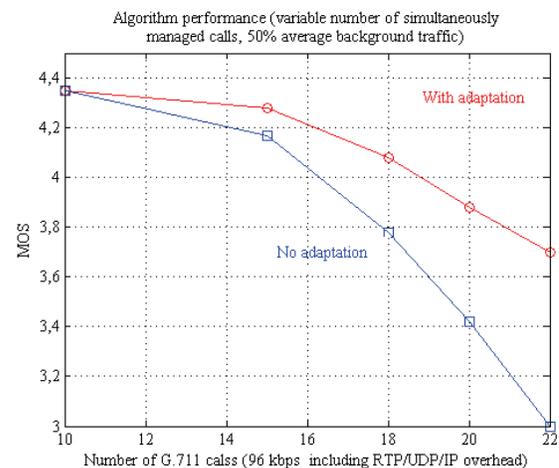
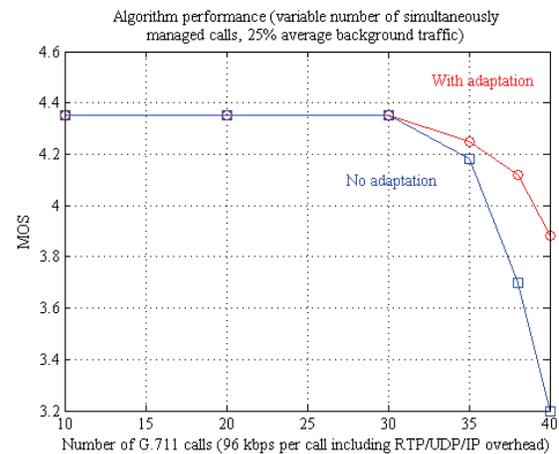


Figure 8: Simulation results

VIII. CONCLUSION AND FUTURE WORK

In this paper, an adaptive control mechanism was designed to dynamically manage and improve the average quality of VoIP communication. In this scheme, the receiver

makes a control decision based on two parameters: 1) the computational instantaneous quality level, which is calculated per talkspurt using the E-model and 2) the perceptual metric, which estimate the integral speech quality by taking into account the fact that a decrease of communication quality depends, not only on the presence of packet delay or loss in the network, but also on the position of a quality degradation period in the call.

The algorithm works together with the adaptive jitter buffer mechanism. The adaptive jitter buffer is used to manage short-term quality; the sender-based adaptation technique tries to choose encoding parameters to improve a long-term quality by decreasing network congestion and, as a result, significant instantaneous changes in quality.

The paper uses three narrowband codecs with different packet sizes for analysis but the approach can be extended to a wider set of narrowband codecs, to wideband encoding schemes and, potentially, to IP-based audio in general.

Several questions will be addresses in the future work. First, it is necessary to work more on the algorithm and to decrease a number of cases when adaptive encoding makes a situation with quality worse. Second, according to recent research, the nature of traffic in the Internet changes because of significant increase of peer-to-peer traffic and assumptions about background traffic modeling used in Section 3 may not be true in future. Other traffic models simulating TCP backoff mechanism and/or mix of TCP and UDP traffic should be studied. Third, an adaptive change of speech encoding parameters affects perceptual quality. The proposed algorithm tried to avoid rapid changes of codecs and make adaptation as smooth as possible, but the question about what users hear when encoding parameters are changed and how this variation affects user feeling about a call, should also be investigated in more detail.

REFERENCES

- [1] RFC 2474, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", 1998
- [2] RFC 2475, "An Architecture for Differentiated Service", 1998
- [3] RFC 2205, "Resource ReSerVation Protocol (RSVP)", 1997
- [4] RFC 1633, "Integrated Services in the Internet Architecture: an Overview", 1994
- [5] RFC 3031, "Multiprotocol Label Switching Architecture", 2002
- [6] E. Myakotnykh, R. Thompson, "Adaptive Speech Quality Management in Voice-over-IP Communications", Fifth Advanced International Conference on Telecommunications AICT'09, Venice, Italy, May 2009
- [7] Z. Qiao, L. Sun, N. Heilemann and E. Ifeachor, "A New Method for VoIP Quality of Service Control Use Combined Adaptive Sender Rate and Priority Marking", IEEE International Conference on Communications, 2004
- [8] J. Seo, S. Woo, K. Bae, "Study on the application of an AMR speech codec to VoIP", Acoustics, Speech, and Signal Processing, 2001 Proceedings
- [9] J. Matta, C. Pepin, K. Lashkari, R. Jain, "A source and channel arte adaptation algorithm for AMR in VoIP using the E-model", Proceedings of NOSSDAV 2003
- [10] 3G TS 26.071, "Mandatory Speech Codec speech processing functions; AMR Speech Codec; General Description", 1999
- [11] RFC 3550: "RTP: A Transport Protocol for Real-Time Applications", July 2003
- [12] ITU-T G.711, "Pulse code modulation (PCM) of voice frequencies", Geneva, Switzerland, 1988
- [13] H. Oouchi, T. Takenaga, H. Sugawara and M. Masugi, "Study on Appropriate Voice Data Length of IP Packets for VoIP Network Adjustment", NTT Network Service Systems Laboratories, 2002
- [14] L. Yamamoto, J. Beerends, "Impact of network performance parameters on the end-to-end perceived speech quality", Expert ATM Traffic Symposium, Greece, 1997
- [15] B. Ngamwongwattana, "Effect of packetization on VoIP performance", in Proc. ECTI-CON, 2008
- [16] J. Bolot and A. Vega-Garcia, "Control Mechanisms for Packet Audio in the Internet", Proceedings IEEE Infocom, San Francisco, CA, pp 232-239, 1996
- [17] S. Mohamed, F. Cervantes-Perez and H. Afifi, "Integrating Network Measurements and Speech Quality Subjective Scores for Control Purposes." IEEE Infocom, 2001
- [18] ITU-T G.107, "The E-model, a computational model for use in transmission planning", Geneva, Switzerland, 2000
- [19] Y. Huang, J. Korhonen, Y. Wang, "Optimization of source and channel coding for voice over IP", IEEE International Conference on Multimedia and Expo, 2005
- [20] S. Huang, P. Chang, E. Wu, "Adaptive voice smoothing with optimal E-model method for VoIP services", IEICE transactions on communications, vol. 89, 2006
- [21] B. Ngamwongwattana, "Sync & Sense Enabled Adaptive Packetization VoIP", PhD Dissertation, University of Pittsburgh, 2007
- [22] F. Sabrina, J. Valin, "Adaptive Rate Control for Aggregated VoIP Traffic", Globecom, 2008
- [23] J. Valin, "The speex codec manual", <http://www.speex.org/docs>
- [24] ITU-T P.800, "Methods for subjective determination of transmission quality", Geneva, Switzerland, 1996
- [25] J. H. Rosenbluth, "Testing the Quality of Connections having Time Varying Impairment", Committee contribution T1A1.7/98-031, 1998
- [26] ITU-T Rec. G.107 Amendment 1, "Provisional impairment factor framework for wideband speech transmission", 2006
- [27] ITU-T Rec. G.113, "Transmission impairments due to speech processing", Geneva, Switzerland, 2001
- [28] S. Möller, A. Raake, N. Kitawaki, A. Takahashi, M. Wältermann, "Impairment Factor Framework for Wide-Band Speech Codecs", IEEE Transactions on Audio, Speech, and Language Processing, Vol. 14, No. 6, November 2006
- [29] A. Takahashi, H. Yoshino, "Perceptual QoS assessment technologies for VoIP", IEEE Communications Magazine, July 2004
- [30] ITU-T SG12 D.139: "France Telecom study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences", Geneva, Switzerland, 2000
- [31] ITU-T G.726, "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)", Geneva, Switzerland, 1990
- [32] ITU-T G.729, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)", Geneva, Switzerland, 1996
- [33] V. J. Ribeiro, M. Coates, R. H. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation", in Proc. of ITC Specialist Seminar on IP Traffic Measurement, September 2000
- [34] T. Karagiannis, M. Molle, and M. Faloutsos, "Long-range dependence: Ten years of Internet traffic modeling", IEEE Internet Computing, 2004
- [35] M. S. Taqqu, W. Willinger, and R. Sherman, "Proof of a Fundamental Result in Self-Similar Traffic Modeling", ACM Computer Communications Review, pp. 5 – 23, April 1997
- [36] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through High-Variability: Statistical Analysis of Ethernet

- LAN Traffic at the Source Level”, Proceedings of the ACM/SIGCOMM'95, Cambridge, MA, 1995
- [37] The Network Simulator – ns-2, www.isi.edu/nsnam/ns/
 - [38] K. Park, G. Kim, M. Crovella, “On the relationship between file sizes, transport protocols, and self-similar network traffic”, Proc. IEEE International Conference on Network Protocols, 1996
 - [39] C. Williamson, “Internet traffic measurement”, Internet Computing, IEEE, 2001
 - [40] N. Basher, Aniket Mahanti, Anirban Mahanti, C. Williamson, and M. Arlitt, “A Comparative Analysis of Web and P2P Traffic”, WWW2008. Beijing
 - [41] Laird Popkin (Pando Network), Doug Pasko (Verizon), “P4P: ISPs and P2P”, 2006
 - [42] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide-area networks”, in Proc. INFOCOM, 1994
 - [43] ITU-T P.59, “Telephone Transmission Quality Objective Measuring Apparatus: Artificial Conversational Speech”, Geneva, Switzerland, 1996
 - [44] W. Jiang and H. Schulzrinne, “Analysis of On-Off Patterns in VoIP and their Effect on Voice Traffic Aggregation”, IEEE International Conference on Computer Communications and Networks, 2000
 - [45] A. D. Clark, “Extensions to the E Model to incorporate the effects of time varying packet loss and recency”, Telecommunication standards contribution, Document T1A1.1/2001-037, April 2001



www.iariajournals.org

International Journal On Advances in Intelligent Systems

✦ ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS

✦ issn: 1942-2679

International Journal On Advances in Internet Technology

✦ ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING

✦ issn: 1942-2652

International Journal On Advances in Life Sciences

✦ eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO

✦ issn: 1942-2660

International Journal On Advances in Networks and Services

✦ ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION

✦ issn: 1942-2644

International Journal On Advances in Security

✦ ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

✦ issn: 1942-2636

International Journal On Advances in Software

✦ ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS

✦ issn: 1942-2628

International Journal On Advances in Systems and Measurements

✦ ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL

✦ issn: 1942-261x

International Journal On Advances in Telecommunications

✦ AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA

✦ issn: 1942-2601