Project no.:	**IST-FP6-STREP- 26979**
Project full title:	**Highly dependable ip-based networks and services**
Project Acronym:	**HIDENETS**
Deliverable no.:	**D1.3**
Title of the deliverable:	**Final evaluation, consolidated results and guidelines**

Contractual Date of Delivery to the CEC:	31st December 2008
Actual Date of Delivery to the CEC:	30th January 2009
Organisation name of lead contractor for this deliverable	Carmeq
Authors: Björn Könning (ed), Manfred Reitenspiess, Irene de Bruin, Sonia Hemstra de Groot, Tom Lippmann, Inge-Einar Svinnset, Zoltan Égel, Gábor Huzerl, Mohamed Kaâniche, Marc-Olivier Killijian, Nicolas Rivière, Matthieu Roy, Hélène Waeselynck, Andrea Bondavalli, Alessandro Daidone, Felicita Di Giandomenico, Lorenzo Falai, Paolo Lollini, Hans-Peter Schwefel, Erling M Møller, Anders Nickelsen, Jimmy Nielsen, Antonio Casimira Costa	
Participants: AAU, BME, Carmeq, FSC, FCUL, LAAS-CNRS, Telenor, WMC, UniFi	
Work package contributing to the deliverable:	WP1
Nature:	R
Version:	1.00
Total number of pages:	120
Start date of project:	1st Jan. 2006 **Duration:** 39 months

Project co-funded by the European Commission within the Sixth Framework Program (2002-2006)

Dissemination Level

PU	Public	**x**
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This document serves as a summary document of the end-to-end resilience solutions that were developed and assessed in the course of the HIDENETS project. Solutions to important challenges and remaining and newly identified research topics are described including lessons learned from the HIDENETS activities. As the document provides a generalised view on the HIDENETS results, it also contains the final version of the HIDENETS Reference Model. The resulting architectural solutions and resilience services as well as the development and assessment methods are not restricted to the HIDENETS use-case domain of car-to-car and car-to-infrastructure applications and use-cases, but can be applied in a wide variety of scenarios with mobility and ad-hoc requirements (e.g. RFID applications or web services involving mobile devices).

Starting from a condensed summary of the identified use cases and applications in the vehicular setting, the network architecture containing the ad hoc and infrastructure domain and the definition of the main network elements are presented. The software architecture of the mobile nodes employs the concept of architectural hybridization, which allows executing a subset of so-called oracle services with stronger guarantees on timeliness and availability. Additional complex resilience middleware and communication level services are introduced and learnings on their design and implementation are reported.

The quantitative analysis of HIDENETS use-cases and HIDENETS resilience solutions is realised by a combination of top-down abstraction-based decomposition with bottom-up modelling in so-called point-wise evaluation models. Results of modelling approaches combining simulation, mathematical modelling, and experimental evaluation are reported and the quantitative data are used to demonstrate the benefits and explore the limitations of the HIDENETS resilience solutions. Selected use-case models prove the feasibility of end-to-end holistic modelling, and this modelling experience is used to develop a semi-automatic generic evaluation workflow which can be applied to HIDENETS like scenarios of mobile applications.

The runtime resilience support solutions and quantitative evaluation approaches are complemented by model-based development and testing tools, where specific emphasis is put on extending existing standards based approaches and tool-chains by methods to characterise and handle the dynamicity of the systems resulting from wireless communication of mobile nodes.

Essential parts of the HIDENETS solutions have been implemented as proof-of-concept prototypes. An application-development test-bed demonstrates the practical feasibility of the extended model-based HIDENETS application development approach and corresponding tool-chain. A distributed black-box test-bed verifies the cooperative data storage approaches of the complex resilience middleware in highly mobile scenarios. A platooning test-bed demonstrates the solutions for timeliness and adaptivity, which need to be employed by platooning applications to achieve efficiency without sacrifice of safety. Finally, a resilient communication test-bed demonstrates the practical benefit of selected communication level solutions. The latter two test-beds optionally can use emulated dynamic network topologies to achieve reproducible experiments, for which a dedicated open-source tool was developed in HIDENETS.

Last but not least, the results of the project are mapped to a condensed dependability process which facilitates the application of the HIDENETS dependability assessment approach by industry. The document closes with an outlook to future adaptation options of HIDENETS solutions.

Keyword list:

Reference model, network and node architectures, middleware-level and communication-level services, dependability and performance assessment (evaluation and testing), design methodologies, dependability process etc.

Version information

Version	Date	Comments

0.0	13.06.2008	First Draft with table of contents and initial structure
0.1	19.08.2008	First Draft with input from WPs
0.2	10.09.2008	Second Draft with additional input from WP 6
0.3	12.09.2008	Input from WP5
0.4	15.09.2008	Update of Fault Analysis Chapter
0.5	19.09.2008	Input from WP2
0.6	23.09.2008	Additional input for the open research points
0.7	30.09.2008	Reorganised doc structure
		Additional input from LAAS related to WP2 topics
		Update of “Proof of concept set up”
0.73	12.10.2008	Input on chapter Outlook to other Application Fields
		Update of chapter Proof of Concept
0.8	15.10.2008	Inclusion of a preliminary version of BIA
		Lesson learned for TCO
0.82	20.10.2008	Update of Chapter quantitative evaluation5
0.83	27.10.2008	New input for other application fields
0.84	11.11.2008	Input for the dependability process
0.85	13.11.2008	Refined Outlook Chapter
		Input on Contributions to standardization bodies
		Refined executive summary and abstract
0.86	14.11.2008	Update of the business impact analysis
0.87	17.11.2008	Input on Replication Manager in chapter 3.4 and 7.4
0.88	6.12.2008	Updates from WP1, WP3, WP4, partially WP7
0.89	11.12.2008	Updated BIAs,
0.9	17.12.2008	Modified document structure, Updates in Testing framework chapter and Use case and BIA, chapter
0.91	21.12.2008	Update of chapter model-based application development
0.92	15.01.2009	Comments from 1st external reviewer are finalised
0.93	21.01.2009	Minor changes in chapter 7.3 and 8.1.3

Table of Contents

Bibliography

[1] A. Casimiro et al., “Resilient Architecture (final version)”, EU FP6 IST project HIDENETS, deliverable D2.1.2, December 2007.

[2] IEEE 802.11 WG, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification”, IEEE 1999.

[3] IEEE 802.11 WG, “Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)”, IEEE 802.11e/D13.0, Jan. 2005.

[4] IEEE 802.11p draft standard, http://www.ieee802.org/11/Reports/tgp_update.htm

[5] 3GPP TS 23002-710: “Network Architecture”, V7.1.0, March 2006

[6] P. Veríssimo. Travelling through Wormholes: a new look at Distributed Systems Models, ACM SIGACT news (ACM Special Interest Group on Automata and Computability Theory), 37(1):66-81, 2006.

[7] Flaviu Cristian, Christof Fetzer. The timed asynchronous system model. In Proceedings of the 28th Annual International Symposium on Fault-Tolerant Computing, pp.140-149, Munich, Germany, June 1998. IEEE CS Press.

[8] Paulo Veríssimo, António Casimiro. The timely computing base model and architecture. IEEE Transactions on Computers, 51(8):916–930, 2002.

[9] T. Chandra, V. Hadzilacos, S. Toueg, and B. Charron-Bost. On the impossibility of group membership. In Proceedings of the 15th ACM Symposium on Principles of Distributed Computing, pages 322–330, May 1996.

[10] E. Anceaume, B. Charron-Bost, P. Minet, and S. Toueg. On the formal specification of group membership services. Technical Report RR-2695, INRIA, Rocquencourt, France, November 1995.

[11] T. Chandra, S. Toueg. Unreliable failure detectors for reliable distributed systems. Journal of the ACM, 43(2):225–267, March 1996.

[12] Lorenzo Falai et. al., “Mechanisms to provide strict dependability and real-time requirements”, EU FP6 IST project HIDENETS, deliverable D3.3, June 2008.

[13] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, F. Grandoni. Threshold-based mechanisms to discriminate transient from intermittent faults. IEEE Transactions on Computers, 49(3):230–245, 2000.

[14] M. Pizza, L. Strigini, A. Bondavalli, F. Di Giandomenico. Optimal discrimination between transient and permanent faults. In Third IEEE International High-Assurance Systems Engineering Symposium, pages 214–223, 1998.

[15] S. Porcarelli, M. Castaldi, F. Di Giandomenico, A. Bondavalli, P. Inverardi. A Framework for Reconfiguration-Based Fault-Tolerance in Distributed Systems, In R. De Lemos, C. Gacek, and A. Romanovsky, editors, Architecting Dependable Systems, LNCS. Springer-Verlag, 2004. also ICSE-WADS2003, Post-Proceeding of ICSE-WADS2003.

[16] A. Casimiro, P. Lollini, M. Dixit, A. Bondavalli, P. Veríssimo. A framework for dependable adaptation in probabilistic environments. In Proc. of the 23rd ACM Symposium on Applied Computing (SAC 2008), Dependable and Adaptive Distributed Systems (DADS) Track, pages 2192-2196, Fortaleza, Ceara, Brazil, March 16 - 20, 2008.

[17] M. Kovács, P. Lollini, I. Majzik, A. Bondavalli. An Integrated Framework for the Dependability Evaluation of Distributed Mobile Applications. In Proc. of the RISE/EFTS Joint International

Workshop on Software Engineering for REsilieNt systEms (SERENE 2008), pages 29-38, Newcastle upon Tyne, UK, November 17-19, 2008.

[18] L. Courtès, Cooperative Data Backup for Mobile Devices, PhD Thesis, LAAS-CNRS, November 2007.

[19] P. Lei, et al., 'An Overview of Reliable Server Pooling Protocols', IETF, draft-ietf-rserpool-overview-02.txt, July 2007.

[20] R. Stewart, et al., "Aggregate Server Access Protocol (ASAP)", IETF, draft-ietf-rserpool-asap-18.txt, November 2007.

[21] Q. Xie, R. Stewart, M. Stillman, M. Tuexen, A. Silverton, "Endpoint Handlespace Redundancy Protocol (ENRP)", IETF, draft-ietf-rserpool-enrp-18.txt, November 2007.

[22] L. Courtès, O. Hamouda, M. Kaâniche, M.-O. Killijian, D. Powell. Dependability Evaluation of Cooperative Backup Strategies for Mobile Devices. In Proc the 13th IEEE Int. Symp. On Pacific Rim Dependable Computing (PRDC-07), December 2007.

[23] A. Bondavalli, P. Lollini, L. Montecchi. Analysis of User Perceived QoS in Ubiquitous UMTS Environments Subject to Faults. In Software Technologies for Embedded and Ubiquitous Systems, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 5287/2008, Pages 186-197, 2008.

[24] P. Lollini, A. Bondavalli et al., "Evaluation methodologies, techniques and tools (final version)", EU FP6 IST project HIDENETS, deliverable D4.1.2, December 2007.

[25] P. Veríssimo and L. Rodrigues. Distributed Systems for System Architects. Kluwer Academic Publishers, 2001.

[26] Service Availability ForumTM - Application Interface Specification Software ManagementFramework SAI-AIS-SMF-A.01.01.

[27] Service Availability ForumTM - Distributed Systems Management for AIS-SNMP SAI-AIS-SNMP-A.01.01, 2005.

[28] Evaluation of Routing Dependability in MANETs using a Topology Emulator / N. Jensen, Morten ; Nickelsen, Anders. Elektronik og IT, Kandidatuddannelsen (Spec. Distribuerede Systemer), 10. semester. 2007

[29] Scalable emulation of dynamic multi-hop topologies. / Nickelsen, Anders ; Jensen, Morten N.; Matthiesen, Erling Vestergaard ; Schwefel, Hans-Peter. In: Proceedings of ICWMC 2008. 2008.

[30] http://air-in-a-box.sourceforge.net

[31] A. Avizienis, J.C. Laprie, "Dependable computing: from concepts to design diversity", Proceedings of the IEEE, vol. 74, no. 5, May 1986, pp. 629-638.

[32] A. Avizienis, J.C. Laprie, B. Randell, C. Landwer, "Basic Concepts and Taxonomy of Dependable and Secure Computing", IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, January-March 2004, pp. 11-33.

[33] W.C. Carter, "A time for reflection", in Proc. 12th IEEE Int. Symp. on Fault Tolerant Computing (FTCS-12), Santa Monica, California, June 1982, p. 41.

[34] J.C. Laprie, A. Costes, "Dependability: a unifying concept for reliable computing", Proc. 12th IEEE Int. Symp. on Fault Tolerant Computing (FTCS-12), Santa Monica, California, June 1982, pp. 18-21.

[35] J.-C. Laprie (Ed.), Dependability: Basic Concepts and Terminology, Springer-Verlag, Vienna, 1992.

[36] P. Lollini, A. Bondavalli et al., "Application of the evaluation framework to the complete scenario (final version)", EU FP6 IST project HIDENETS, deliverable D4.2.2, December 2008.

[37] M. Radimirsch et al., “Use case scenarios and preliminary reference model”, EU FP6 IST project HIDENETS, deliverable D1.1. September 2006.

[38] J. Rosenberg et al., “SIP: Session Initiation Protocol,” IETF, RFC3261, June 2002.

[39] I.-E. Svinnset et al., “Report on resilient topologies and routing – final version”, EU FP6 IST project HIDENETS, deliverable D3.1.2, June 2008.

[40] Manfred Reitenspieß et. al., “Experimental proof-of-concept set-up HIDENETS”, EU FP6 IST project HIDENETS, deliverable D6.3, June 2008.

[41] Z. Egel et. al., “Documentation and Evaluation of the experimental work”, EU FP6 IST project HIDENETS, deliverable D6.4, December 2008.

[42] Jean Arlat et al., “Revised reference model”, EU FP6 IST project HIDENETS, deliverable D1.2. June 2007.

[43] HIDENETS tutorial: http://www.hidenets.aau.dk

[44] J.Barton, V. Vijayaragharan. Ubiwise: A Simulator for Ubiquitous Computing Systems Design, Technical report HPL-2003-93, Hewlett-Packard Labs, 2003.

[45] de Bruin, D.; Kroon, J.; van Klaverem, R.: Nelisse, M.. Design and test of a cooperative adaptive cruise control system, Intelligent Vehicles symposium, pp.392-396, IEEE CS Press, 2004

[46] D. R. Choffnes and F. E. Bustamante. “An Integrated Mobility and Traffic Model for Vehicular Wireless Networks”, Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET), ACM Press, Germany Sep. 2005, pp 69-78.

[47] David Harel and Shahar Maoz, Assert and negate revisited: Modal semantics for UML sequence diagrams. Software and Systems Modeling, 7(2):237–253, May, 2008.

[48] Z. Micskei, H. Waeselynck, “A survey of UML 2.0 sequence diagrams' semantics”, LAAS Report no. 08389, August 2008.

[49] R. Morla and N. Davies, “Evaluating a Location-Based Application: A Hybrid Test and Simulation Environment”, IEEE Pervasive computing, Vol.3, No.2, Jul.-Sep. 2004, pp.48-56.

[50] Object Management Group, UML 2.1.1 Superstructure Specification, URL: http://www.omg.org/technology/documents/formal/uml.htm, 2007.

[51] S. Pickin and J-M. Jézéquel, “ Using UML sequence diagrams as the basis for a formal test description language”, in Proc. of 4th International Conference on Integrated Formal Methods (IFM2004), LNCS 2999, Springer, 2004, pp. 481-500.

[52] K. Sanmiglingam and G. Coulouris. “A Generic Location Event Simulator”, UbiComp 2002, LNCS 2498, Springer-Verlag Berlin Heidelberg, 2002, pp. 308-315.

[53] C. Schroth et al, “Simulating the traffic effects of vehicle-to-vehicle messaging systems”, Proc. 5th Int. Conf. on ITS Telecommunications (ITST 2005), France, Jun. 2005.

[54] K. Matheus, R. Morich, et al., „Car-to-Car Communication - Market Introduction and Success Factors“, ITS 2005: 5th European Congress and Exhibition on Intelligent Transport Systems and Services, 1 - 3 June 2005, Hannover, Germany

[55] K. Matheus, R. Morich, A. Lübke, „Economic Background of Car-to-Car Communications“, IMA 2004, Informationssysteme für mobile Anwendungen, 20.-21.10.2004, Braunschweig, Germany

[56] P. Kotler and G. Armstrong, “Principles of marketing” – New product development and life cycle strategies (chapter nine), Prentice Hall 2001

[57] G.A. Churchill Jr and D. Iacobocci, “Marketing research – Methodical Foundations”, Thomson 2005

[58] A.C. Burns and R.F. Bush, “Marketing research”, Prentice Hall 1998

[59] N.B. Holbert and M.W. Speece, “Practical marketing research – an integrated global perspective”, Prentice Hall 1993

[60] The AMBULANCE Project, http://www.biomed.ntua.gr/emergency112/ambulance.html

[61] ACEA – European Automobile Manufacturers Association: http://www.acea.be/home_page

[62] Karl F. Doerner et. all, „Heuristic Solution of an Extended Double-Coverage Ambulance Location Problem for Austria”, Central European Journal of Operations Research

[63] NRW-Offensive gegen den Verkehrsstau. Konzepte und Maßnahmen für die Zukunft. http://www.strassen.nrw.de/_down/pub_antistau-offensive.pdf

[64] Service Availability Forum: http://www.saforum.org/

[65] www.car-to-car.org: Car 2 Car Communication Consortium home page

[66] www.coopers-ip.eu: COOPERS Project home page

[67] www.comesafety.org: COMeSafety home page

[68] www.isaca.at/Ressourcen/CobiT%204.0%20Deutsch.pdf: german document of the CobiT Framework

[69] David J. Smith, Kenneth G. L. Simpson, “Functional Safety: A Straightforward Guide to Applying IEC 61508 and Related Standards“, Elsevier Butterworth-Heinemann, 2004

[70] http://www.itil-officialsite.com/AboutITIL/WhatisITIL.asp: IT Infrastructure Library

[71] COMeSafety Newsletter 5 – Newsletter for European ITS Related Research Projects, http://www.comesafety.org/uploads/media/COMeSafety_Newsletter_Issue-5.pdf, July 2008

[72] PRE-DRIVE C2X Project Description by CORDIS, http://cordis.europa.eu/fetch?CALLER=FP7_PROJ_EN&ACTION=D&DOC=10&CAT=PROJ&QUERY=011cb00077ee:06a9:689b95a0&RCN=87604

[73] COMeSafety architecture document, “European ITS Communication Architecture - Overall Framework Proof of Concept Implementation“, http://www.comesafety.org/index.php?id=109

[74] A. Daidone, F. Di Giandomenico, A. Bondavalli. Hidden Markov Models as a support for diagnosis: formalization of the problem and synthesis of the solution, In 25th IEEE Symposium on Reliable Distributed Systems (SRDS 2006), Leeds, UK, October 2006.

[75] J. Nielsen et al., “Cross-Layer Resilience Optimization in the Ad-Hoc Domain”, EU FP6 IST project HIDENETS, deliverable D3.2. June 2008.

[76] H. Waeselynck et al. “Mobile Systems from a Validation Perspective: a Case study”, Proc. of the 6th International Symposium on Parallel and Distributed Computing (ISPDC’07), IEEE CS Press, Austria, Jul. 2007.

[77] Z. Micskei, H. Waeselynck, M. D. Nguyen, and N. Riviere. “Analysis of a group membership protocol for Ad-hoc networks,” LAAS Technical Report no. 06797, November 2006.

[78] M.D. Nguyen, H. Waeselynck, N. Rivière, “Testing mobile computing applications : towards a scenario language and tools, 6th Workshop on Dynamic Analysis (WODA 2008), ACM Press, Washington D.C, USA, July 2008.

[79] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, F. Grandoni. Threshold-based mechanisms to discriminate transient from intermittent faults. IEEE Transactions on Computers, 49(3):230–245, 2000.

[80] M. Pizza, L. Strigini, A. Bondavalli, F. Di Giandomenico. Optimal discrimination between transient and permanent faults. In Third IEEE International High-Assurance Systems Engineering Symposium, pages 214–223, 1998.

[81] S. Porcarelli, M. Castaldi, F. Di Giandomenico, A. Bondavalli, P. Inverardi. A Framework for Reconfiguration-Based Fault-Tolerance in Distributed Systems, In R. De Lemos, C. Gacek, and A.

Romanovsky, editors, Architecting Dependable Systems, LNCS. Springer-Verlag, 2004. also ICSE-WADS2003, Post-Proceeding of ICSE-WADS2003.

[82] S. Porcarelli, F. Di Giandomenico, A.Chohra, A. Bondavalli. Tuning of database audits to improve scheduled maintenance in communication systems, in Computer Safety, Reliability and Security, Proc. of the 20th International Conference SAFECOMP 2001, Budapest, Hungary, pages 238–248. Lecture Notes in Computer Science 2187. Springer, 2001.

[83] Andrea Bondavalli, Andrea Ceccarelli, Lorenzo Falai. A Self-Aware Clock for Pervasive Computing Systems. 15th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2007), 7-9 February 2007, Naples, Italy. IEEE Computer Society 2007, pages 403-411.

[84] Bondavalli, A. Ceccarelli, L. Falai. Assuring Resilient Time Synchronization. SRDS2008. October 2008, Naples, Italy.

[85] Bondavalli, A. Ceccarelli, L. Falai, and M. Vadursi. Towards making NekoStat a proper measurement tool for the validation of distributed systems. In Proceedings of The 8th International Symposium on Autonomous Decentralised Systems, pages 377–386, March 2007.

[86] L. Falai. Observing, Monitoring and Evaluating Distributed Systems. PhD thesis, University of Florence, 2008.

[87] L. Falai, A. Bondavalli. RODS: General Framework for Rigorous Observation of Distributed System. DSN 2008 Workshop on Resilience Assessment and Dependability Benchmarking. Anchorage (USA), June 2008.

[88] Bondavalli, A. Ceccarelli, L. Falai, M. Vadursi. Enhancing the NekoStat Tool with Uncertainty, Resolution and Intrusiveness Evaluation Capabilities. DSN 2008 Workshop on Resilience Assessment and Dependability Benchmarking. Anchorage (USA), June 2008.

[89] Henrique Moniz, Nuno F. Neves, Miguel Correia, António Casimiro and Paulo Verissimo. Intrusion Tolerance in Wireless Environments: An Experimental Evaluation. Proceedings of the 13th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'07),

[90] Hans P. Reiser and António Casimiro. Optimizing Byzantine Consensus for Fault-Tolerant Embedded Systems with Ad-Hoc and Infrastructure Networks. 4th International Workshop on Dependable Embedded Systems (WDES'07), Bejing, China, October 2007.

[91] Hugo Ortiz, António Casimiro and Paulo Veríssimo. Architecture and Implementation of an Embedded Wormhole. In Proceedings of the 2007 Symposium on Industrial Embedded Systems (SIES'07), Lisbon, Portugal, July 2007.

[92] António Casimiro, Odorico Mendizabal and Paulo Veríssimo. On the development of dependable embedded applications using specialised wormholes. 3rd International Workshop on Dependable Embedded Systems (WDES'06), Leeds, UK, October 2006.

[93] T. Chandra, S. Toueg. Unreliable failure detectors for reliable distributed systems. Journal of the ACM, 43(2):225–267, March 1996.

[94] T. Cicic, A. F. Hansen, and O. K. Apeland, “Redundant trees for fast IP recovery”, IEEE Broadnets 2007, North Carolina, US, 2007

[95] A. F. Hansen, O. Lysne, T. Cicic, and S. Gjessing, “Fast Proactive Recovery from Concurrent Failures”. In: ICC 2007, June 2007

[96] A. F. Hansen, G. Egeland and P. Engelstad, “Could Proactive Link-State Routed Wireless Networks Benefit from Local Fast Reroute?” CNSR 2008, Halifax, Canada

[97] Y. Liu, H.-P. Schwefel, ”Algorithms for Efficient Broadcasting in Wireless Multi-hop Networks”. In: Proc. of IEEE Globecom 2006

[98] J.Wu and H. Li, ”On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks”. In: Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Aug. 1999

[99] Y. Liu, H-P Schwefel, “Localised Algorithms for Virtual Backbone Formation in Wireless Multi-hop Networks with unidirectional links”, In: Proceedings of IST mobile summit, July 2007.

[100] Y. Liu, “Virtual Backbone and Mobility-based optimizations for wireless multi-hop networks”, PhD thesis, Aalborg University, September 2007

[101] J. Grønbæk, J. Nielsen, “Cross-Layer Optimization of Message Broadcast”, In MANETs, Master thesis, Aalborg University, Jul. 2007.

[102] Pintér G., Micskei Z., Kövi A., Égel Z., Kocsis I., Huszerl G. and Pataricza A.: Model-Based Approaches for Dependability in Ad-Hoc Mobile Networks and Services. In R. de Lemos, F. Di Giandomenico, C. Gacek, H. Muccini and M. Vieira (Eds.) Architecting Dependable Systems V (LNCS-5135) pp. 150-174. 2008, Springer

[103] Szatmári Z., Kövi A., and M. Reitenspiess: Applying MDA approach for the SA forum platform. In Proceedings of the 2nd Workshop on Middleware-Application interaction: Affiliated with the Discotec Federated Conferences 2008 (Oslo, Norway, June 03 - 03, 2008). MAI '08, vol. 306. ACM, New York, NY, 19-24. DOI= http://doi.acm.org/10.1145/1394272.1394278

[104] Luís Marques, António Casimiro and Paulo Veríssimo, Proof-of-concept Platooning Application Using the HIDENETS Architecture, The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN’09), to be submitted.

[105] M-O. Killijian, N. Rivière, M. Roy. Experimental evaluation of resilience for ubiquitous mobile systems. Workshop on Ubiquitous Systems Evaluation (USE), UbiComp 2007, Innsbruck (Autriche), Sept 16-19 2007, pp.283-287.

[106] M-O. Killijian, D. Powell, M. Roy, G. Séverac. Experimental Evaluation of Ubiquitous Systems. Why and how to reduce WiFi communication range. DEBS 2008 (2nd International Conference on Distributed Event-Based Systems). July 2008, Rome

[107] A. Casimiro, P. Martins, and P. Veríssimo, How to build a timely computing base using real-time linux. In Proceedings of the 2000 IEEE International Workshop on Factory Communication Systems, pages 127–1343, Porto, Portugal, September 2000, IEEE Industrial Electronics Society.

[108] M. Correia, P. Veríssimo, and N. F. Neves, The design of a COTS real-time distributed security kernel, In Fourth European Dependable Computing Conference, October 2002.

[109] P. Sousa, A. Bessani, M. Correia, N. F. Neves and P. Veríssimo, Resilient Intrusion Tolerance through Proactive and Reactive Recovery, In PRDC '07: 13th IEEE Pacific Rim International Symposium on Dependable Computing, pages 373–380, Melbourne, Australia, December 2007.

[110] C. Weinhold, H. Härtig. VPFS: Building a Virtual Private File System with a Small Trusted Computing Base, Proceedings of ACM SIGOPS/EuroSys European Systems Conference - EuroSys'08, Glasgow, Scotland, April 2008.

[111] B. G. Chun, P. Maniatis, S. Shenker and J. Kubiatowicz. Attested append-only memory: making adversaries stick to their word. Symposium on Operating Systems Principles - SOSP 2007, pages 189-204. 2007.

[112] EU FP6 IST project HIDENETS, Project Proposal Annex I – Description of Work, http://rcl.dsi.unifi.it/projects/HIDENETS-DoW.pdf

[113] András Kövi, András Pataricza, Bálint Rákosi, Gergely Pintér, Zoltán Micskei, “UML profile and design patterns library”, EU FP6 IST project HIDENETS, deliverable D5.1, March 2007, http://www.hidenets.aau.dk/Public+Deliverables

[114] András Kövi, Dániel Varró, Zoltán Németh: Making Legacy Services Highly Available with OpenAIS: An Experience Report. ISAS 2006: 206-216

[115] Z. Micskei, I. Majzik, F. Tam: Comparing Robustness of AIS-Based Middleware Implementations, In Proceedings of International Service Availability Symposium (ISAS 2007), LNCS 4526, Durham, New Hampshire, USA, May 21-22, 2007.

[116] Zoltan Szatmari, Andras Kovi and Manfred Reitenspiess. Applying MDA for SA Forum AIS based application development. MAI2008 workshop at DisCoTec2008

[117] Z. Szatmári, "Model-driven development for highly available services", MSc Diploma thesis, BME, 2008

[118] Gábor Huszerl, Hélène Waeselynck (eds.), Zoltán Égel, András Kövi, Zoltán Micskei, Minh Duc N'Guyen, Gergely Pintér and Nicolas Rivière, "Refined design and testing framework, methodology and application results", EU FP6 IST project HIDENETS, deliverable D5.3, December 2008, http://www.hidenets.aau.dk/Public+Deliverables

[119] IBM Rational Software Architect official home page, http://www-01.ibm.com/software/awdtools/swarchitect/websphere/

[120] András Kövi, Dániel Varró: An Eclipse-Based Framework for AIS Service Configurations. ISAS 2007: 110-126

[121] Gábor Urbanics, András Kövi, Zoltán Égel, András Pataricza, Introducing dynamism to SA Forum cluster, DNCMS08 workshop at SRDS2008.

[122] G. Urbanics, "Introducing dynamism to SA Forum cluster", MSc Diploma thesis, BME, 2008

[123] M. Reitenspiess et al., "Final evaluation, consolidated results and guidelines-Annex", EU FP6 IST project HIDENETS, deliverable D1.3, Jan 2009.

[124] A. Bondavalli, I. Mura, S. Chiaradonna, R. Filippi, S. Poli, and F.Sandrini. "DEEM: a tool for the dependability modeling and evaluation of multiple phased systems". In *DSN-2000 IEEE Ing. Conference on Dependable Systems and Networks* (FTCS-30 and DCCA-8), pages 231-236, June 25-28 2000.

[125] G. Clark, T. Courntey, D. Daly, D. Deavours, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. Webster. "The Möbius modelling tool". In *Proceedings of the 9th International Wrokshop on Petri Nets and Performance Models*,pages 241-250, Aachen, Germany, September 11-14 2001.

[126] J . Nielsen, J. Grønbæk, T. Renier, T. Toftegaard, HP Schwefel, "Cross-Layer Optimization of Multipoint Message Broadcast in MANETs",To appear in Proceedings of IEEE WCNC 2009.

[127] A. Nickelsen, J. Grønbæk, HP Schwefel, "Probabilistic Network Fault-Diagnosis using Cross-Layer Observations",To appear in Proceedings of AINA 2009.

[128] E. Matthiesen, O. Hamouda, M. Kaaniche, HP Schwefel, "Dependability Evaluation of a Replication Service for Mobile Applications in Dynamic Ad-Hoc Networks", International Service Availability Symposium (Proceedings to appear in Springer LNCS), Japan, 2008.

[129] Y.Liu, F. Li, HP Schwefel, "Reliable Broadcast in Error-Prone Multi-hop Wireless Networks: Algorithms and Evaluation",Proceedings of IEEE Globecom 2007.

[130] Y. Liu, F. Li, A. Nickelsen, HP Schwefel, "A New Link State Routing Protocol for Mobile Ad-hoc Networks", 4th IEEE International Symposium on Wireless Communication Systems (ISWCS), October 2007.

[131] E. Matthiesen, T. Renier, HP Schwefel, "A new selection metric for backup group creation in inter-vehicular networks",Proceedings of IST mobile summit, July 2007.

[132] Y. Liu, HP Schwefel, "Localised Algorithms for Virtual Backbone Formation in Wireless Multi-hop Networks with unidirectional links",Proceedings of IST mobile summit, July 2007.

[133] I. Antonos, L. Lipsky, HP Schwefel, “Performance-relevant network traffic correlation”, [with I. Antonios, L. Lipsky]14^th International Conference on Analytic and Stochastic Modelling Techniques and Applications, ASMTA June 2007.

[134] HP Schwefel, I. Antonios, “Performability Models for Multi-Server Systems with High-Variance Repair Durations”, Dependable Systems and Networks (DSN), June 2007.

[135] J. Grønbæk, HP Frejek, T. Renier, HP Schwefel, “Client-Centric Performance Analysis of a High-Availability Cluster”, Proceedings of International Service Availability Symposium, published in Springer LNCS, May 2007.

[136] Y. Liu, HP Schwefel, “Algorithms for Efficient Broadcasting in Wireless Multi-hop Networks”, Proceedings of IEEE Globecom, Nov. 2006.

[137] RL Olsen, MB Hansen, HP Schwefel, “Quantitative analysis of access strategies to remote information in network services”, Proceedings of IEEE GLOBECOM, November 2006

[138] T. Renier, E. Matthiesen, HP Schwefel, “Inconsistency Evaluation in a Replicated IP-based Call Control System”, In D. Penkler, M. Reitenspiess, F. Tam (eds.) ‘Service Availability’, LNCS 4328, pp.177-192. Springer, 2006.

Abbreviations

AO: Authentication Oracle

AMF: Application Management Framework

AP: Access Point

API: Application Programming Interface

BGP: Border Gateway Protocol

BIA: Business Impact Analysis

C2C: Car-to-Car

C2CCC: Car to Car Communication Consortium

C2I: Car-to-Infrastructure

CA: Certification Authority

CAC: Connection Admission Control

COTS: Commercial Off-The-Shelf

CRC: Cyclic Redundancy Coding

DFCD: Decentralised Floating Car Data

DM: Diagnostic Manager

DoS: Denial of Service

ETSI: European Telecommunications Standards Institute

FCD: Floating Car Data

FEC: Forward Error Correction

GMP: Group Membership Protocol

GPRS: General Packet Radio Service

GPS: Global Positioning System

GSM: Global System for Mobile communication

GSPN: Generalised Stochastic Petri Nets

HW: Hardware

IEEE: Institute of Electrical and Electronics Engineers

IFIP: International Federation of Information Processing

IM: Intermediate Model

IMS: IP Multimedia Subsystem

IP: Internet Protocol

ISO: International Organization for Standardization

J2SE: Java 2 Standard Edition

JVM: Java Virtual Machine

LLC: Logical Link Control

MAC: Medium Access Control

MDA: Model Driven Architecture

MIP: Mobile IP

MSC: Message Sequence Chart

MW: Middleware

NeMo: Network Mobility

ODP: Open Distributed Processing

OS: Operating System

OSI: Open System Interconnection

OTS: Off-the-Shelf

PCO: Points of Control and Observation

PHY: Physical layer

PLCP: Physical Layer Convergence Protocol

PKI: Public-Key Infrastructure

QoS: Quality of Service

RACS: Resource and Admission Control Subsystem

RFID: Radio Frequency Identification

RM: Reference Model

RecM: Reconfiguration Manager

RepM: Replication Manager

R&SA Clock: Reliable and Self-Aware Clock

RSU: Road Side Unit

SA Forum: Service Availability Forum

SCTP: Stream Control Transmission Protocol

SD: Streaming Data

SDL: Specification and Design Language

SIL: Safety Integrity Level

SINR: Signal to Interference-plus-Noise Ratio

SIP: Session Initiation Protocol

SME: Small and Medium sized Enterprise

SNMP: Simple Network Management Protocol

SOP: Start Of Production

SRN: Stochastic Reward Nets

SW: Software

TAI: International Atomic Time

TCO: Trust and Cooperation Oracle

TCP: Transmission Control Protocol

TPH: Tamper Proof Hardware

TTP: Trusted Third Party

UDP: User Datagram Protocol

UML: Unified Modelling Language

UMTS: Universal Mobile Terrestrial Access

VoIP: Voice on Internet Protocol

V&V: Verification and Validation

WIMAX: Worldwide Interoperability for Microwave Access

WLAN: Wireless Local Area Network

1. Executive summary

HIDENETS addresses the provision of highly available and resilient distributed applications and mobile services in highly dynamic environments characterized by unreliable communications and components due to the occurrence of accidental and malicious faults (attacks and intrusions).

From an applicatory perspective the HIDENETS project combines two ambitious challenges: the development of car-to-car communication solutions and the development of quantitative evaluation techniques to predict dependability properties. The first one has to cope with new requirements derived from high dynamicity and uncertainties caused by environmental influences but also involves runtime support, application development and prototyping, while the second has to handle the very high complexity of the system by means of analytical, simulative and experimental evaluation techniques to improve the prediction of a system's dependability properties and to identify systematic faults as early as possible.

Our investigations include networking scenarios consisting of ad hoc/wireless multi-hop domains as well as infrastructure network domains. Applications and use case scenarios from the automotive domain, based on car-to-car communication with additional infrastructure support are used as driving examples to identify the key features (challenges, threats, and resilience requirements) that are relevant in the context of the project. Based on these features, the project developed appropriate fault tolerance mechanisms, at the middleware and communication layers, as well as methodologies to support their design, development, evaluation and testing.

The HIDENETS Reference Model synthesises the main solutions that are promoted by the project for the design, development support, evaluation and testing of resilient mobile and ad hoc based applications and services, based on the results and achievements obtained in the course of the project. The terminology from the dependability related community is included in the annex of this document, as is used as a starting point for the concepts.

Figure 1 illustrates the scope of the technical work and solutions developed in the context of HIDENETS with respect to a typical layered communication model. In particular, the results covering resilient architecture and communication, and methodologies to assist in design, testing, and quantitative evaluation provide the main input for the HIDENETS Reference Model. It is noteworthy that HIDENETS does not develop new technologies for the physical layer.

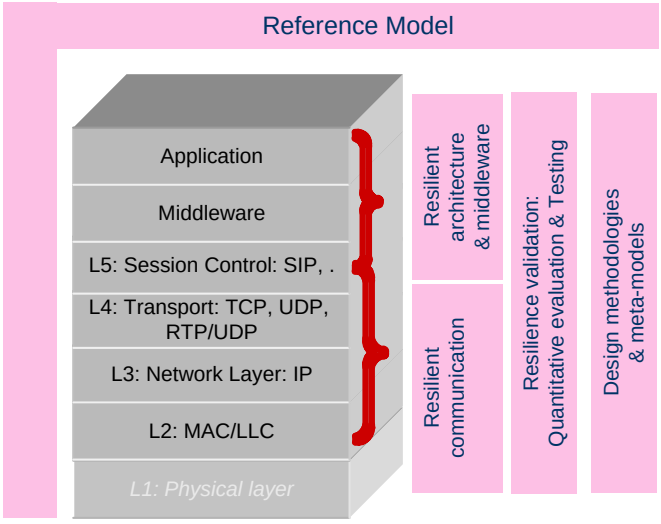


Figure 1: Scope and reference model of HIDENETS with respect to OSI model

This deliverable serves as an introductory and at the same time as a summary document for the entire HIDENETS project. It contains a finalised version of the HIDENETS Reference Model, preliminary versions of which were introduced in D1.1 [37] and D1.2 [42]. The deliverable points to the most relevant research results obtained during the 36 months of the HIDENETS project and the achievements of the project. Solutions to important challenges and remaining or newly identified research topics are described including lessons learned from the HIDENETS activities. This will allow a better technical understanding of the modifications and adaptations of all HIDENETS work packages.

The remaining part of this deliverable is structured into 9 chapters. Chapter 2 describes the most important HIDENETS use cases and applications and provides a summary of the business impact analysis. Basic dependability related terminology that is used throughout HIDENETS is provided in Annex I in this document. The details of the business impact analysis are provided in a second Annex; in order to keep the main document compact, this second Annex for the business impact analysis is provided as a separate document.

Chapter 3 presents an overview of the HIDENETS run-time dependability solutions and deals with the network and node architecture, which includes an identification of the different layers investigated in the project. It describes the HIDENETS node architecture and introduces the concept of architectural hybridization underlying the HIDENETS architecture solution. Then architectural features like the timeliness and trustworthiness oracles are explained. The chapter closes with sections on middleware resilience services and resilience communications together with the challenges and activities addressed at these levels. Chapter 4 deals with quantitative evaluation techniques by explaining the holistic approach of HIDENETS towards a point-wise description of all related challenges and activities. Model based application development is addressed in chapter 5 and deals with the creation of meta models and automatic configuration generation to provide basic notations and modelling facilities required for quantitative evaluation and testing. The description of the testing framework is contained in chapter 6 focusing on specific challenges raised by mobility. The four HIDENETS test beds are handled in chapter 7 as well as the topology emulator. The HIDENETS test beds are developed for experimental set-ups to prove the feasibility and relevance of HIDENETS results by practical evidence.

The overall rationale of HIDENETS is to contribute dependability solutions to the scientific community and industry. Especially for industrial use generalised knowledge from the project’s experiences into a process description is derived. Therefore, chapter 8 contains learning of the project described in a condensed dependability process which facilitates the application of the HIDENETS dependability assessment approach in an industrial environment where the application of specific engineering standards and processes has been widely established a long time ago.

Further on, chapter 9 gives an outlook which aims at a more generic applicability of HIDENETS results. They are meant to be applicable beyond the context of car-to-car and automotive scenarios, applications, and use-cases. Additionally, it gives an overview on future adaptation opportunities of HIDENETS solutions and describes newly arising open research points to motivate and help the reader to use the achieved results in other application fields and domains.

2. HIDENETS use case and business impact analysis

This chapter describes the use cases defined within HIDENETS and names involved applications previously defined in [37]. These use cases impose numerous challenges to timeliness, integrity and freshness of data, and marks their actors and roles as well as the challenging dependability requirements. The rest of this chapter briefly summarises the HIDENETS BIA (Business Impact Analysis) which has been performed during the project. More details to both dependability terminology and the BIA [123] are attached in the annex of this document.

2.1 Selected use cases

According to the definition, a use case is a set consisting of (one or more) applications, the actors and roles involved and the identification of the affected dependability domains. The identified applications for a use case are assumed to occur in a certain context where these applications typically appear together and interact with each other. The actors and their roles represent the glue of the use case and are important for the kind of interaction between the applications. In that sense, they have an impact on architecture discussions.

Each use case is complemented with failure modes and challenges. These challenges go beyond the challenges in the descriptions of single applications in the sense that they already take into account the interaction between different applications in the use case, the actors and their roles. The purpose of the failure modes and challenges was to initially stimulate architecture discussions and resulting solutions for dependability in the HIDENETS work packages.

Within HIDENETS 17 applications were identified and 6 use case scenarios were formed out of them (see [37]). The criteria for choosing these use cases and applications have been derived from supposed user needs and new functionalities obviously useful in future traffic scenarios. Beyond that, HIDENETS use cases were carefully adjusted in accordance to current discussions and trends in the automotive domain and the Car2Car Communication Consortium.

The document focuses on three HIDENETS use cases (infotainment, platooning and car accident) which impose complementary challenges and functionalities. The patenting use cases (see section 2.1.1) imposes primarily timeliness requirements while the infotainment use cases (see section 2.1.2) addresses a classical scenario in which online content can be accessed via car2car and car2Infrastructure communication. This imposes real time and QoS needs but also trustworthiness requirements. The car accident use case (see section 2.1.3) contributes to both the motivation to avoid and alleviate fatalities and its consequences and the unique idea of a distributed black box by using car2car and car2infrastructure communication. So the car accident use case describes again a scenario which is based on both car2car and car2infrastructure but here the requirements impose a stronger focus on service differentiation and resilient rerouting mechanisms which need to be balanced based on the current phase of the use case.

2.1.1 Platooning use case

2.1.1.1 Selected application(s)

The only application involved in this use case is “Platooning”, see section 3.2.6 in [37], where the leading car is driven by a human and sends control information to the following cars, which immediately have to process the control data and act accordingly. The following cars try to follow the leading car as tight as possible. A tight distance facilitates saving energy consumption by driving in the slipstream of the car in front. Therefore, all subsequent cars of a platoon are not controlled by humans. Humans are not able to

handle such tight distances. For this, a very fast and reliable one-hop/multi-hop data transmission is indispensable.

The platooning use case imposes very strict dependability requirements. Each platoon member must transmit its control data like current position, speed and acceleration in a reliable and timely way. Depending on the size of a platoon, cars in the middle of the platoon need to forward control data between cars located at begin and end of a platoon. This imposes even stricter dependability requirements.

2.1.1.2 Failure modes and challenges

There are a number of challenges and failure modes related to platooning. They are ranked by their importance.

Very important challenges and failure modes:

- If messages arrive at a platoon member with too much delay, this may result in too late reactions to a manoeuvre in the platoon, leading to safety problems. The sources of delay in the networking part needs to be investigated and appropriate means for achieving timeliness are required.
- It is essential that all messages sent out can reach all addressed platoon members. This touches throughput but also transmission errors. Means to ensure sufficient throughput and a reliable communication link are needed.
- It is considered a worst case situation if two long platoons meet driving in opposite directions. In this case, the amount of radio resources needed by the two platoons increases in the affected geographic area and needs to be guaranteed for each of the platoons. This requires appropriate means to detect such events and to re-distribute the available radio resources quickly and efficiently.
- Any information inside the platoon needs to be trustworthy. False or faked messages may lead to traffic accidents.

Other important challenges:

- Larger platoons may not only need single but multi-hop communication. This may introduce additional delays and use up more radio resources. It is unknown whether multi-hop communication for platooning is possible and whether mechanisms for the efficient use of radio resources in this case exist.
- Multi-hop communication in large platoons may lead to increased data volumes which increases required resources. A solution may be to condense platoon data for multi-hop communication.
- Data integrity is important. The information distributed may e.g. contain information about the braking status. If a platoon member gets wrong information about a hard braking vehicle ahead, this may lead to traffic accidents.
- A fault in the system may cause false messages which are a hazard to traffic safety. Possible solutions will involve fault detection and recovery mechanisms.

The platooning requirements related to timeliness are addressed by hybrid node architecture of HIDENETS which includes wormhole services like the Reliable and Self-Aware Clock (see section 3.3.1.1); the Duration Measurement (see section 3.3.1.2) and the Timely Timing Failure Detector (see section 3.3.1.3). Point wise evaluations on these solutions can be found in section 4.3.1. Furthermore a platooning test bed has been developed which is described in section 7.2.

2.1.2 Infotainment and work with highly mobile terminals

The scenario consists of a car, taxi, bus, train etc, and may also involve pedestrians. Workers want access to their firm’s intranet to get access to or upload important documents, update their calendars, check email,

meet some deadline etc. While travelling, they may take part in a teleconference discussing important strategic decisions before an upcoming meeting later that day.

Other travellers want to play online games during while travelling, watch TV or a video, collect tourist information about the scenery passing by or shop products on the web etc.

In these cases, we envision a multitude of different devices within and outside of cars, and people with different preferences. We also expect a wide diversity in the applications requested. How the ability to use more than one access point, if possible, can improve the dependability. How the high mobility of the clients/terminals (possibly frequently changing communication channels and network locations) influence the communication delay. Multiple alternative access points and technologies to a fixed infrastructure will therefore be assumed. Possibly multi-hop communication over an ad-hoc network is used to reach the fixed-network gateway.

2.1.2.1 Selected applications

All applications involve Internet access on the move. The applications that are relevant for this use case can be listed as follows. The applications and their requirements are further described in deliverable D1.1 [37]

1. **Information**
    a. **Tourist information:** Users in cars and pedestrians may request information that is suited to their current geographical position or context. Also information about future locations and contexts may be relevant. Such information may be requested by the users on-demand or alternatively one could imagine some kind of subscription where the information is pushed down to the users as messages or streaming audio/video/data. Thus, many types of applications may be used for delivering tourist information, and location/context awareness may be a particular (add-on) aspect of all these applications.
    b. **Floating car data:** Floating Car Data is not included in this use-case, since this application is covered in the other use-cases. However, it might be possible to merge them if necessary.
    c. **Hazard warning:** Hazard warnings are warnings about road safety information which is relevant over multiple hops. The source of information may be one or more cars. It includes road conditioning warning, traffic jam warning, and cooperative forward collision warning and emergency vehicle alert warning.
2. **Entertainment**
    d. Music, radio, video, TV and on-line gaming
3. **Office**
    e. Calendar and email synchronization, documents upload and download, document sharing and teleconference and voice call
4. **Web-shopping**.

All these mentioned applications are described in detail in [37].

2.1.2.2 Failure modes and challenges

This use case consists of different applications which will experience and be affected by different modes of failures. For example tourist information is not a very time critical application, but it will not work properly if messages are too much delayed and therefore out of interest for the user. Another risk is that the service may deliver messages that are not relevant for the current position. For the music, radio, video and TV applications a large receiver buffer may solve some of the failure modes. However, too many messages out of order and too high a content error will be perceived by the users as noise and poor audio and video quality. For the online gaming application, a buffer may not be appropriate due to possible real-time requirements. For the document uploads and email/calendar synchronization, it may be critical to make the application terminate with a completed status. Inconsistency between the different versions may cause

difficulties. Teleconferences share some of the same failure modes as online gaming due to real time requirements. Web-shopping share some of the failure modes with office synchronisation, like importance to complete. It is also important to deliver messages within a threshold, so that the application will not terminate due to timeout.

From the communication level requirements described for each application in [37], it is already concluded that different applications require different treatment with respect to e.g. delay, throughput, jitter and packet loss. To manage the resources in the network, mechanisms for differentiated service delivery and routing is necessary. This might also be necessary due to different capabilities in routers and gateways. Radio resource management must also consider this heterogeneity in application and equipment requirements and capabilities.

Since ad hoc networks are based on wireless communication and mobile nodes, the service delivery may be interrupted due to several reasons. Resilient routing techniques are necessary to ensure that traffic can be quickly rerouted due to loss of neighbours or congestion. It should also be possible to roam between different gateways. Since applications usually require different treatment, prioritizing between applications should also be considered in the case of rerouting and roaming.

Wireless and highly mobile networks might also operate with tailored transport layer and media layer mechanisms (e.g. packet resending) to improve the throughput and reduce the delay in the wireless environment. Some applications like web shopping do not require very strict delay bounds; however they require hard guarantees with respect to session completion and correctness. Communication delays in dynamic multi-channel/multi-protocol/mobile environments and effective store and forward mechanisms in the ad hoc domain will be important in that respect.

A short list of challenges and candidate research topics that have been addressed in HIDENETS are:

- Highly resilient routing techniques
- Differentiated resilience
- Effective RRM, possible across different radio technologies

These challenges and its requirements are investigated by the HIDENETS solutions on resilient communications (in chapter 3.5) which e.g. embraces the development efficient routing (section 3.5.1.3), IP resilient routing (section 3.5.1.2), reliable broadcast mechanisms (section 3.5.1.4) and a multi-radio multi-channel architecture (section 3.5.1.1) to more efficiently use the bandwidth. Point wise evaluations on these solutions can be found in section 4.3.1. Furthermore, a resilient communication test bed has been developed which is described in section 7.4.

2.1.3 Car accident (including distributed black-box)

This use-case evolves around a scene with an accident on a road, involving cars and other road users. The use case covers mainly what happens after the accident, but also involves some issues directly before and during the accident. General driver assistance and collision avoidance are not part of this use case, they are treated in the use case “Assisted Transportation” in section 3.3 in [37].

Directly before the accident, the distributed black-box functions of the cars in the area collect time-stamped information. This information is backed up to other cars as they pass, as well as to fixed-network servers whenever access to the fixed infrastructure is available. The higher the percentage of cars is equipped, the lower is the risk of losing data.

Right after the accident, many people may try to call the emergency services, call home, and send text and multimedia messages, at least in motorways with high traffic density. This may cause congestion in the radio access network, both in WLAN-type technologies and in mobile networks. It is a challenge for the networks to be able to prioritise between the requested services according to some pre-determined parameters (service type, user class, …).

Some time after the accident, an ambulance is approaching and the cars along the road are notified either directly through the ad-hoc network or via a central unit that broadcasts the message to the cars along the particular road as it is approaching. The alarm centre personnel may already at this stage know the names of the persons involved in the accident, and even pictures may have been transferred from the accident scene. Essential data on the injured, along with possible pictures of the crash scene, may be transferred to the ambulance while on its way.

Arriving at the place in question, there may be a need to communicate with medical expertise at the local or a central hospital by use of voice, video and data transmission (multi-media application). There may also be a need for group communication with other emergency teams at the site. Heading back to the hospital with the injured there will be a need to transmit information on the positioning of the ambulance to communicate that it is approaching the hospital and at the same time maintain the multimedia connection with the medical expertise.

Afterwards, what really happened in the accident can be found by investigating data collected by the distributed black box application.

Traffic information on the basis of floating car data that could guide cars around the accident by presenting alternate roads is not included in this use-case, since this application is covered in another use-case (assisted transportation). However, it might be possible to merge these use cases if necessary.

2.1.3.1 Selected applications

The applications will in this case require high dependability due to the emergency nature and some will also have real-time requirements:

1. **Emergency communication**
    f. **Emergency vehicle warning**
    g. **Online notifications to hospital:** Due to the emergency nature of this use case, the dependability and security requirements of this application will be higher in this use case than in the normal case. Requirements regarding transmission delay are loose, but messages should have a very high probability of reaching the receiver.
    h. **Access to medical expertise (multimedia):** Due to the emergency nature of this use case, the dependability and security requirement of these application components will be higher in this use case than in the normal case.
    i. **Group communication at crash site:** Due to the emergency nature of this use case, the dependability and security requirement of this application will be higher in this use case than in the normal case. Further it is essential that these applications work when terminals are connected via a single hop to the infrastructure, but also when no connection to infrastructure exists (ad-hoc only).
2. **Distributed black box:**
    j. Automatic backup in neighbouring cars in the ad-hoc domain,
    k. Automatic backup via the fixed network whenever possible

All these mentioned selected applications are described in detail in section 3.2 in [37]. Generally, application components used in emergency applications will have the same or higher requirements on quality/performance versus the “internet access on the move” applications. Further, they will have higher requirements regarding dependability, integrity and stability, for instance having priority over other applications when communication resources are scarce, as is often the case at crash scenes, or keeping a session while on the move.

2.1.3.2 Failure modes and challenges

The applications in this use case are associated with slightly different failure modes. Emergency vehicle warning and online notification to a hospital will not function properly if the messages are too much delayed or contain too many errors to be interpreted. The delay bound is exceeded and the messages are useless if the ambulance is ahead or very short behind the message. The multimedia applications associated with access to medical expertise will consider message delay, errors and out-of-order messages as a failure mode due to bad video and audio quality. Group communication at a crash site can contain both notifications and multimedia communication, and will share failure modes with the application mentioned above. In addition, there may be failure modes with respect to group connection and management. Messages may exist that only make sense if all members of some specific group can receive them. Some details about failure modes for the distributed black-box can be found in Section 3.2.7 in [37].

The overall challenge for emergency communication applications is to give the ambulance a dependable and secure network connection to the central site (hospital or alarm central) while driving. This may be achieved for instance by being able to utilise and seamlessly switch between several different network technologies and access points. Further, the emergency applications should be given priority over other applications. Techniques for differentiated service quality, differentiated resilience, highly resilient routing and differentiated rerouting may help to achieve this. The radio network is a particular important area for studying these issues, and effective radio resource management techniques should be studied.

For some applications, it must be possible to roam between different gateways/access points. For the ambulance, particular requirements exist as to keeping the session alive while on the move. For the distributed black-box application, ensuring timeliness is very important for opportunistic communication both with cooperating cars when in the ad-hoc domain and with access points when in the infrastructure domain. Fast and seamless handover mechanisms, possibly between different access network technologies, should be studied. Differentiation should also be considered in the case of roaming and handover.

Wireless and highly mobile networks might also operate with tailored transport layer and media layer mechanisms (e.g. packet resending) to improve the throughput and the delay in the wireless environment. Server response time and effective store and forward mechanisms in the ad hoc domain will be important in that respect.

The following list names challenges analyzed in HIDENETS but also candidate research topics interesting for further research in the future:

- Highly resilient routing techniques
- Differentiated routing
- Differentiated resilience
- Effective and stable mechanisms for roaming
- Effective RRM, possibly across different radio technologies
- Tailored transport layer and media layer mechanisms
- Effective store and forward in the ad hoc domain
- Server response time.

Considering the case of the distributed black-box application, switching between the ad-hoc and the infrastructure domain should be notified to the application in order to trigger reconfiguration. The amount of information produced and stored by this application is low but its freshness is critical, i.e. reducing communication latency is more important than ensuring throughput.

Thus, in general solutions for the distributed black-box should be designed to ensure the consistency, integrity, availability and confidentiality of the data itself.

- Availability of the black box information: this is the main goal of the application - information availability must be maximised despite faults.

- Confidentiality and privacy: the black box information should be accessible only from authorised parties. Only the original car should be allowed to write data, and only the data owner (or its delegates, e.g. the insurance company) should be allowed to read it.
- Integrity of the black box information: the original information produced by the car at the provider site should not be modifiable, neither by the driver nor by the other cars hosting copies of the original data, nor by any third party.
- Logical consistency: the backed up data must be consistent, i.e. the data reaching the fixed server, or the cooperating cars, should not be modified. Only the producer must be able to write the data, the other entities have no need for write (or even read) access on the data.
- Temporal consistency: it is important that the information disseminated in the ad-hoc domain and stored in the infrastructure domain reflects the real situation. Only most up-to-date data need to be backed-up and restored in the infrastructure domain.

These challenges and their requirements are investigated by the HIDENETS solutions on resilient communications (in chapter 3.5). Beside the development of efficient routing (section 3.5.1.3), IP resilient routing (section 3.5.1.2), reliable broadcast mechanisms (section 3.5.1.4) and a multi-radio multi-channel architecture (section 3.5.1.1) the investigations in section 3.5.1.5 on resilient infrastructure connections and on ABC and differentiated resilience in section 3.5.2.4 can be mentioned .Further investigations on point wise evaluations on these solutions are described in section 4.3.1. More specific evaluations on the car accident use case and the distributed black box are addressed in section 4.3.2.2 and in 4.3.2.1 which deal with the development of assessment methods and tools for complex, dynamic scenarios. In addition a distributed black box test bed has been developed which is described in section 7.3.

2.2 Business impact analysis

The Business Impact Analysis study (BIA study or BIA in short) is part of HIDENETS. It is written as a report on the market research effectuated for two realistic applications of HIDENETS dependability related research results: Navisave and Medigate. Both services have been selected as they reflect the Hidenets use cases Streaming Data and Floating Car Data. The detailed study is available in the annex of this document which is complied in a separate annex document [123].

One important aspect of the research was the analysis of market and business issues of dependability work. Unfortunately it is a typical story for technological developments – from the technological point of view, the improvements are a big step ahead, but they are not always marketable (e.g. consumers/users/customers are not willing to pay for them). Therefore, a market study was planned as an integral part of the research work to find out whether the results are marketable and what could be the economic effects. The market study was organised as follows:

- Initially, a business model was developed to analyse the flow of data and dependencies between the involved market players (software manufacturers, service users, car owners…)
- Then a questionnaire was defined to be sent to a limited number of experts in the field for completion. To make the feedback as specific as possible, two specific applications were referenced in the questionnaire: Navisave and Medigate.
- Finally, the responses were analysed and generalised for their implications on dependability work and dependability requirements in mobile, ad-hoc applications.

The answers from the questionnaire sustain the need for more dependable services on the market (both Navisave and Medigate were perceived as fulfilling relevant service qualities by the respondents to the questionnaire). They are described as relatively unique and the implementation and development costs are characterised as acceptable. However, expectations are high regarding the market entrance and acceptability on the market (38% of the market for Navisave and 25% for Medigate). The study is to be seen as an initial step to trigger further research beyond the HIDENETS project. It is strictly bound due to the available time

span, due to the bounded number of recipients of the questionnaire, due to few market data regarding the service quality (specifically dependability aspects) and regarding the economic aspects of existing products on the market. Both Navisave and Medigate are fictitious services. Current service deployments do not fulfil the high dependability requirements as assumed for Navisave and Medigate, but were used as reference points to understand the importance of dependability services as developed in the course of HIDENETS.

Finally the study contains a preliminary study of the present products on the market and possible influences by introducing HIDENETS dependability tools.

3. Run-time dependability solutions

In the previous chapter we described several use cases that were identified as relevant in the context of HIDENETS due to their characteristics, involving mobility, criticality requirements and communication requirements, among others. In order to address these requirements and deal with the involved challenges, HIDENETS considered the development of appropriate architectural solutions and resilience services, to be deployed at the middleware level and provide run-time support to the applications.

Regarding the HIDENETS architecture, the developed solution incorporates the concept of architectural hybridization, a paradigm to construct systems with wormholes, i.e., special components that present improved characteristics with respect to the remaining components of the system. There are a number of potential advantages in using hybrid distributed system models, as will be reviewed in Section 3.2.

In this Chapter we first revisit the HIDENETS architecture, including the aspects of the networks envisaged in HIDENETS and their main components, and including also the aspects concerning the node part, the software building blocks and their organization. Then we describe the assumed system model, in particular referring to the consequences of adopting a hybrid model in terms of temporal and security properties.

The remaining parts in this chapter are devoted to the HIDENETS services, devoted to the improvement of resilience aspects of distributed applications, and addressing, in particular, some of the challenges that were previously identified in the definition of the use cases. The final HIDENETS node architecture includes two kinds of services: generic middleware services and timeliness and trustworthiness oracles.

The timeliness and trustworthiness oracles are fundamental because of their improved properties regarding timeliness and security (when compared to the remaining middleware services), and because they are instrumental to ensure that some strict timeliness and security requirements of the applications can be addressed. They are addressed in Section 3.3.

The generic middleware services contribute to the overall resilience improvement because of the functionality they offer. The idea is that these middleware services can incorporate arbitrarily complex tasks or be based on the execution of complex protocols, as needed to achieve some dependability objective or to increase the resilience of applications using them. In fact, these services address some of the challenges identified in HIDENETS, like the need to provide support for adaptive applications or the need to maximise information availability despite faults. The middleware resilience services are addressed in Section 0.

3.1 HIDENETS architecture overview

This section presents first an overview of the HIDENETS network architecture and application context to clarify the various types of scenarios and interactions investigated by the project. Then, we introduce the basic models and assumptions underlying the design of the HIDENETS architecture. Finally, we present a simplified and high-level description of the architecture itself, considering the architecture of the nodes that will be implementing the basic dependability services and mechanisms needed to provide the level of resilience required for the HIDENETS applications.

3.1.1 HIDENETS network architecture and application context description

The HIDENETS network architecture introduces the relevant network elements and domains as illustrated in Figure 2. We distinguish two fundamentally different domains: 1) the ad hoc domain in which service access and service deployment are performed in a wireless setting, and 2) the infrastructure domain that consists of a back-bone IP network connecting both service providers as well as service clients. Parts of the ad hoc

domain may be connected to the infrastructure domain via cellular access (GPRS/UMTS) or via WLAN RSU (Road Side Unit).

As illustrated in Figure 2, mobile nodes communicate with other mobile nodes directly, or via the infrastructure domain. In the HIDENETS scenarios, these nodes will typically be cars (or terminals in cars, either integrated or portable), but they may also be car-external devices. Mobile nodes may also communicate with nodes in the infrastructure domain. Three main classes of scenarios are studied:

1) All communicating entities are located in the ad hoc domain. Note that this includes scenarios in which the infrastructure domain is needed for connectivity, when the entities may not be within ad hoc connectivity of each other.
2) The service accessing entities are located in the ad hoc domain and the service provisioning entities are in the infrastructure domain.
3) The service accessing entities are in the infrastructure domain and the service provisioning entities are in the ad hoc domain.

The before mentioned car accident use case is an typical example for the first class of scenarios but also for the third class as stationary medical experts and car insurance agencies require data directly from the road. The infotainment use case is an example for the second class of scenarios as typically online data is downloaded to the passengers. The platooning use case in its simplest form does not need any infrastructure but can be seen as an example of scenario class one as extended when supported by RSU.

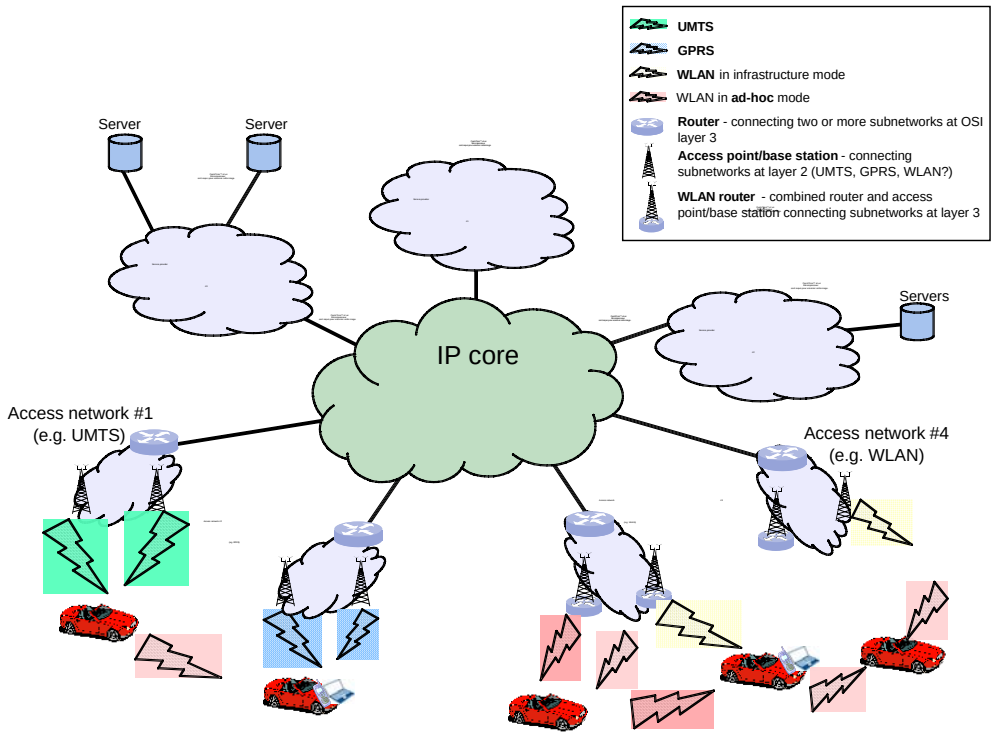


Figure 2: HIDENETS network architecture – infrastructure and ad hoc domains

A mobile node is a node communicating via wireless technologies and protocols so that it can potentially move without losing connection. In this figure we have a set of mobile nodes (OBU in cars) that are communicating directly with other mobile nodes, or via a fixed network, via different types of wireless links (Access Link or Ad hoc Link).

When mobile nodes communicate or are ready to communicate directly without an infrastructure, i.e., within the ad hoc domain, we call it an ad hoc network. The nodes may then run applications that are peer-to-peer in nature, or where the server-part of the application is implemented in a wireless node. They may also communicate via an access network connecting the mobile nodes to the infrastructure domain. We assume that several service providers are connected to the IP core part of the infrastructure domain. These provide,

besides applications/services running in the ad hoc network, additional applications/services for the ad hoc nodes.

When an ad hoc network is connected to the infrastructure domain or acting as an extension to the infrastructure domain, there will be one or more devices functioning as gateways between the two domains. There are several technologies that could be used for such a gateway. An important example is a WLAN access point that connects hosts with WLAN interfaces operating in Infrastructure mode together, forming a wireless network (WLAN). In the case that the WLAN access points are connected to the infrastructure domain (normally the case, and making it a gateway), they also forward data between the wireless hosts and servers or hosts connected to the wired network. A WLAN access point operates at OSI layer 2, but it can also be integrated with a router, in which case it is called a WLAN router.

Another important gateway technology is a GSM/GPRS/UMTS base station. This is more specifically a network element in the radio access network responsible for radio transmission and reception to and from the user equipment. It is as such always connected to the infrastructure domain, and communication between the wireless hosts is transmitted via the mobile core network. The operation of GSM/GPRS/UMTS base stations is defined by 3GPP standards (e.g., see [5]). The coverage area of a GSM/GPRS/UMTS base station is termed a cell. A device moving from one cell to another will automatically be handed over from one base station to another.

Note that, according to our definitions, a car can be an ad hoc node even if it is not connected to other cars in an ad hoc manner. Second, an ad hoc node can also function as an ad hoc gateway at the same time, i.e., the ad hoc node acts as an interface to the infrastructure domain (Fixed-Wireless Ad hoc Gateway or Wireless-Wireless Ad hoc Gateway). In summary, an ad hoc node can be in several states: ad hoc connected only; ad hoc disconnected, but infrastructure connected; gateway (both ad hoc and infrastructure connected); both ad hoc and infrastructure disconnected.

Wireless technologies: Except for the Layer-2 mechanisms, most of the dependability solutions that are introduced as part of the HIDENETS reference model in subsequent sections are independent of the underlying link-layer technology that is used for ad hoc connectivity and for the connection to the infrastructure domain. The link-layer technology, however, strongly influences the communication properties (as expressed by neighbour discovery and link establishment delays, link throughput, L2-frame delays, L2-frame loss probability, availability of L2 broadcast functions) and hence will influence the quantitative performance and dependability metrics on and above the link-layer. Therefore, it is important to identify relevant candidate technologies, so that they can be used in the quantitative analysis and testing. For dependability functionality placed on the link-layer (such as multi-channel MAC), candidate technology selection is even mandatory, as it directly influences the conceptual design of such functionalities.

The main candidates for the ad hoc link-layer connectivity are the Wireless Local Area Networks (WLAN) described by the IEEE 802.11 standards [2]. Several varieties are likely candidates in HIDENETS: common 802.11a/b/g networks where unlicensed frequency bands are available or 802.11p [4] networks for vehicular communication (draft standard). Due to the presence of an additional control channel in 802.11p and the use of licensed spectrum, 802.11p can show advantages in particular for safety-critical applications. The original WLAN standards provide a best effort service, and when the offered traffic load is too high, the overall network performance drops. The extended 802.11e [3] provides functions for differentiated (not guaranteed) QoS.

For non-vehicular ad hoc communication beyond the scope of HIDENETS, also short-range technologies such as Bluetooth and the IEEE 802.15 family or upcoming Ultra-Wide-Band communication can be interesting candidates. This is in particular the case, if small distances, low mobility, and scarceness of battery-energy are present; relevant example scenarios include Personal (Area) Networks and sensor network scenarios. In HIDENETS, these technologies are not considered.

For the connection to the infrastructure, in addition to the packet-switched transport services of cellular networks GPRS and UMTS, WIMAX (mobile versions of the IEEE 802.16 family) and WLAN 802.11-like link-layer protocols in so-called infrastructure mode are the most interesting candidates. While the long-range cellular technologies operate through the already installed radio access networks, dedicated road-side access points will need to be deployed in most cases for the WLAN-based infrastructure access.

For more general, non-vehicular, communication scenarios, short-range technologies can also be used for the infrastructure connectivity, e.g., via the use of Bluetooth access points, sometimes even deployed using meshed networks technology. Although such scenarios are out of scope for HIDENETS, the solutions as presented in this reference model could be relevant and should be adapted and tuned to take into account the specific requirements inherent to these scenarios.

3.1.2 HIDENETS node architecture – simplified description

In this section, we present a simplified description of the proposed architecture of a HIDENETS mobile node that will include the software and hardware components and services needed to run a HIDENETS application in an ad-hoc based mobile environment and to satisfy its dependability and resilience requirements. A more detailed description of the services and building blocks of the proposed architecture is presented in Sections 4, 5 and 6.

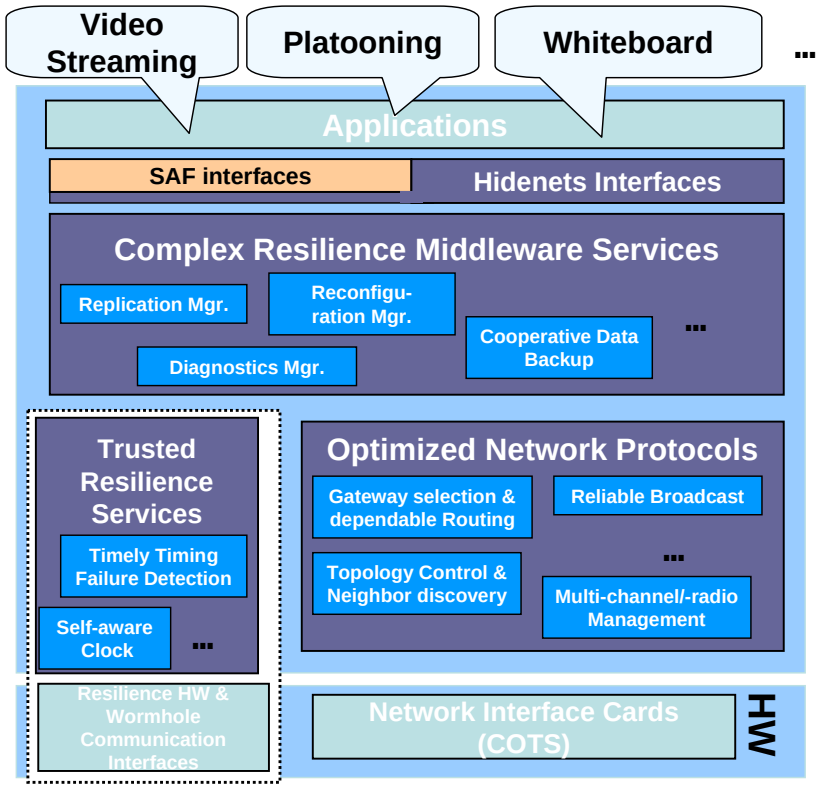


Figure 3: Simplified HIDENETS node architecture

A simplified view of the HIDENETS node architecture is shown in Figure 3.

The node consists of some hardware (HW) that may be installed in a mobile node (e.g., a car) or be part of a separate terminal, and some software running on it. One particular piece of hardware is the network interface card that allows the transmission of information out on the network. Other relevant hardware parts may for instance be GPS devices (possibly included in the more resilient part of the system, described ahead).

Regular applications that may be installed and run by users (e.g. Video Streaming, Platooning, Whiteboard) are always implemented in user space. Applications are allowed to access the HIDENETS services or operating system functions through well-defined Application Programming Interfaces (SAF interfaces or HIDENETS interfaces). Complex Resilience Middleware Services include a range of services developed in HIDENETS and meant to implement quite complex functionalities to support and improve application resilience. These services can be implemented in several ways, in user space as middleware libraries directly linked to applications, or user space middleware services accessed by the applications, or in the operating

system. At the operating system level, HIDENETS considers Optimised Networking support, including, for example, multi-channel/-radio management functions, beside the general communication related functions provided by the operating system, typically implementing OSI layers 2 to 4.

We note that Complex Resilience Middleware Services do not necessarily need to be developed on top of the "standard" network layers implemented in the operating system, or may not necessarily use them. These Complex Resilience Middleware services may rely on other resources for which low-level access must be granted. These resources can be located in a special part of the system, a "Resilience kernel", which provides Trusted Resilience Services.

This special part is clearly separated from the remaining system. In fact, this part can be implemented as a subsystem with its own computational resources, possibly dedicated communication channels and a clearly defined interface to the rest of the system, namely to the Complex Resilience Services. From a modelling perspective, this resilience kernel is a subsystem that has better properties than the rest of the system (user space and operating system). Typically, this means that it can be timelier, more secure and/or more reliable than the rest of the system. These better properties represent a potential for the improvement of the overall node resilience.

Looking at the node as a whole, the existence of these two parts with different sets of properties prefigures a system that is well characterised by the Wormholes model [6], in contrast with other distributed system models that assume homogeneous properties for the entire system. Therefore, in Section 3.2 we discuss the wormhole model and the implications of adopting such a hybrid system architecture in HIDENETS, in particular concerning resilience improvements.

3.1.3 Middleware interfaces and standardization

HIDENETS applications will run on different HIDENETS nodes that – because of the possibly different HW platforms — may have different implementations of the HIDENETS services. In order to support the execution of applications in this environment without modifications, specific interfaces are created for each service. The applications access the services through these interfaces.

Furthermore, it is important for the application programmers to create code that is reusable for different solutions, thus, reduce the development time. Reusability is highly facilitated by the application of standards. The specifications of the Service Availability Forum (SA Forum)[1] define a complete set of standard interfaces for accessing the services of highly available middleware implementations. In addition, the development of HIDENETS applications can be based on the best practices and experiences of SA Forum application development projects.

By taking the above mentioned advantages, in HIDENETS, we decided to provide services on the basis of SA Forum's interfaces. However, there are additional HIDENETS service interfaces that are made directly accessible for applications since the functionality they provide is out of the scope of the SA Forum specifications.

The structure of the HIDENETS middleware is depicted in Figure 4. It is shown that the applications access the services through the standardised SA Forum interfaces and the predefined HIDENETS interfaces and do not directly use the HIDENETS service implementations.

[1] http://www.saforum.org.

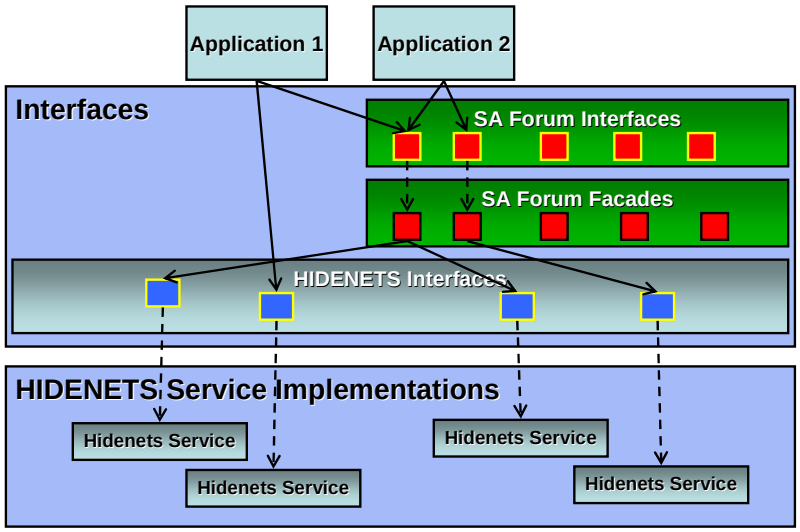


Figure 4: Structure of the HIDENETS middleware

Relation between SA Forum services and HIDENETS services. Although, applications use the SA Forum interfaces, those services do not have separate implementations. Rather they are implemented as façade objects that use the existing HIDENETS services to carry out a specific operation. Generally, a façade object is an object that provides a simplified interface to a larger body of code, such as a class library. In our case, we use the façade objects to hide the HIDENETS service calls and to provide higher-level operations to applications. We also plan to propose HIDENETS developed or extended services to the SA Forum for possible impact on related standardization activities.

3.2 Architectural hybridization

This section is intended to focus and discuss the distributed systems model that is followed in the definition of the HIDENETS architecture. The definition of a system model implies making assumptions about a certain number of aspects. In the context of HIDENETS, we are interested in modelling distributed systems, and we focus in particular in synchrony aspects.

In this section we address architectural hybridization as a proposed approach for the design of systems with improved resilience. Despite the several dimensions in which it may be possible to characterise a system model, architectural hybridization has particular implications concerning synchronicity assumptions. A detailed discussion of synchrony aspects was presented in Deliverable D3.3 [12].

3.2.1 Modelling the synchronicity of the system

Computing systems and computer networks are inherently heterogeneous. This is true both in terms of the functional and non-functional properties they exhibit. In fact, we reason about systems in terms of their functional attributes, such as speed, power consumption or memory capacity, and also in terms of non-functional ones, such as their reliability, security, dependability or resilience, which, as we know, are typically not homogeneous in a distributed system, i.e., the required properties and the level of resilience and dependability might not be the same for all system components and subsystems. On the other hand, it is interesting, and perhaps peculiar, to note the contrast with distributed system models, which are generally homogeneous, that is, which assume the same properties for the whole system. This is particularly noticeable in what concerns synchrony properties, where two typical system models can be distinguished: a) purely asynchronous models that do not make any time-related assumption or b) purely synchronous models

assuming that timeliness bounds will always be satisfied throughout the system. Some partially synchronous system models also exist, which assume some local synchrony (e.g., reliable clocks) or that synchrony is eventually achieved and will remain for sufficiently large amounts of time to finalise computations.

Differently from previous approaches, in HIDENETS we follow a hybrid distributed system model approach, in which different parts of the system have different sets of properties and can rely on different sets of assumptions, namely in terms of faults and synchronism. This has a number of advantages when compared to approaches based on homogeneous models, as detailed in [6].

One example of a hybrid distributed systems model is the Wormholes mode, named in this way after the astrophysics theory. This theory argues that one could take shortcuts, through, say, another dimension, and re-emerge safely at the desired point, apparently much faster than what is allowed by the speed of light. Those shortcuts received the inspiring name of Wormholes. In essence, Wormholes prefigure an ancillary theory which coexists with the classical theory, predicting subsystems which present exceptional properties allowing overcoming fundamental limitations of the systems under the classical theory.

From a more practical point of view, a Wormhole can be instantiated by following the architectural hybridization paradigm. This paradigm not only explores the inherent heterogeneity of practical systems, when possible, but also defines a few principles that pave the way for the construction of the required hybrid and modular systems.

In fact the architectural hybridization paradigm is based on the following principles:

Heterogeneity principle: Systems can be composed of several realms with different properties, in particular non-functional ones, such as quality-of-service, synchronism, security, etc.

Construction principle: In order to follow the architectural hybridization paradigm for building distributed systems it is possible to do more than just simply exploit the natural hybrid nature of the underlying infrastructures. The designer can be proactive in ensuring that a set of desired properties is provided by each of the subsystems that compose the overall system. This means that specific engineering measures can be taken to create realms in the system with (typically better) properties (e.g., multiple communication channels, multiple priority levels for the execution of application processes, mechanisms for differentiating the access to some specific system functions, etc), or it may be even possible to add new components as necessary to create these realms (e.g., additional network interfaces, watchdog cards, additional embedded processors or controllers, etc.).

Encapsulation and interfacing principle: Given the existence of realms with different properties, it is fundamental to ensure the encapsulation of these realms and the provision of well-defined interfaces, as the only mean through which each of the realms can manifest their properties.

We may say that architectural hybridization is an enabler for the construction of realistic hybrid distributed systems. In fact, the HIDENETS node architecture presented in Figure 3 provides an illustration of the architectural hybridization paradigm: the Resilience kernel corresponds to the subsystem with the better properties, which should be constructed according to the above-mentioned construction and encapsulation principles.

3.2.2 Architectural hybridization and the wormholes model

The HIDENETS architecture is a hybrid architecture, thus implementing the Wormholes distributed systems model [6]. It is divided into two (logical) subsystems:

- Wormhole subsystem: in HIDENETS this corresponds to the Resilience kernel, on which some simple but critical services will reside. We call to these services timeliness and trustworthiness middleware oracles (MW oracles).
- Payload subsystem: in HIDENETS this corresponds to the rest of the system. The services residing in the payload subsystem will be complex middleware services, OS services and network services.

MW oracles are trusted components ("more" trusted relative to the Payload subsystem). They offer services useful for applications with strict real-time and/or with strict security requirements. The behaviour of these components must be very strict and predictable: to not impact reliability and safety of the whole system their services have to be provided with certain guarantees at the interfaces. The identification of the role of all MW oracle services is important: what kind of guarantees exactly they provide, and what are the requirements from the hardware/operating system/network in order to maintain such guarantees with a given probability.

The communication between payload and wormhole parts of the architecture is based on a wormhole gateway (see Figure 5). The only way for payload processes to communicate with the MW oracles is through these wormhole gateways, with well-defined interfaces (thus following the encapsulation and interfacing principle). Likewise, for payload processes the properties offered by any wormhole are defined and enjoyed at a wormhole gateway. The payload processes do not have to know how wormholes are implemented, and vice-versa.

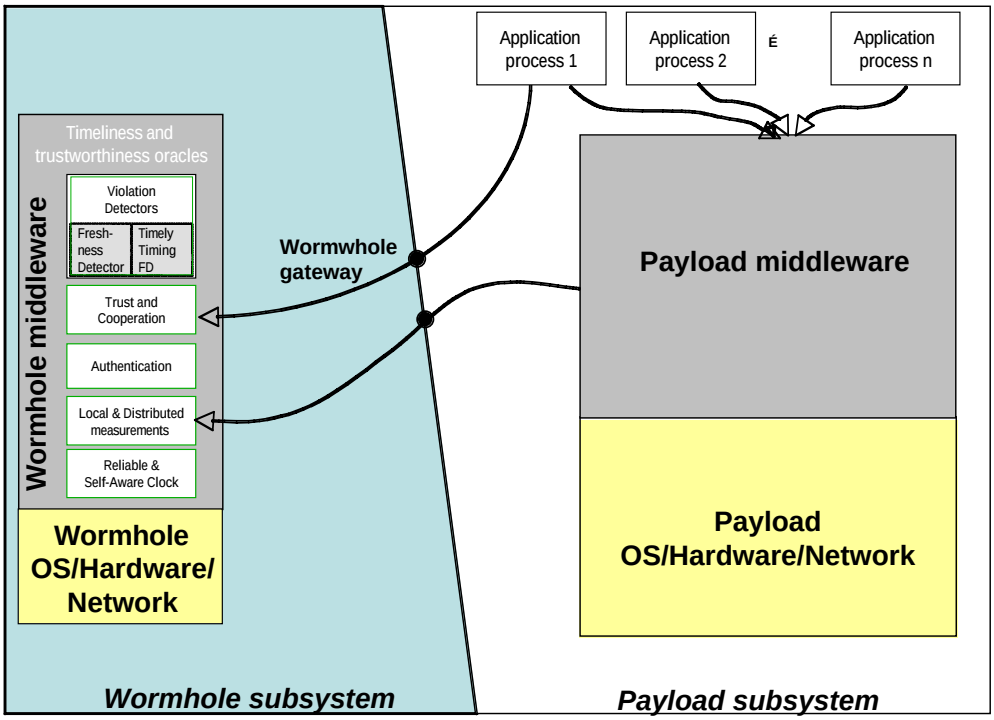


Figure 5: Wormhole and Payload subsystems in HIDENETS and their interconnections

The wormhole part of the architecture provides services with different quality with respect to the payload part. In particular, it assists the execution of fault-tolerant algorithms, offering a minimal set of trusted and timely services, implemented by the MW oracles. In order to be able to offer such services in the wormhole subsystem (with a certain probability), we have more stringent requirements from the execution environment of the wormhole subsystem.

In more practical terms, and although it is possible to classify middleware oracles in different manners, as we explained in Deliverable D3.3 [12], we should say that in HIDENETS we have followed the approach of implementing wormholes with the following characteristics:

1) *Scope*: In HIDENETS we implemented local oracles, providing services with a local scope only, although we considered the use of distributed oracles, namely the services provided by GPS, which can be considered distributed;
2) *Resilience*: we considered some oracles with high level of resilience requirements, namely the Authentication and the Trust and Cooperation oracle.
3) *Timeliness*: we considered some oracles with high level of timeliness requirements, namely the Timely Timing Failure Detection and the Reliable and Self Aware Clock oracles.

Regarding the architectural solution to address (local) resilience and timeliness requirements (see a description of the possible approaches in Deliverable D3.3 [12], the solution that we adopted in the particular case of HIDENETS was based on the use of different hardware for the wormhole and the payload subsystems. Other approaches could also have been used, as exemplified in [108][107][110][111][109]. The differences between our approach and the remaining ones concern:

- **The implementation efforts that are involved**, which tend to be higher when completely separate hardware, operating system and development environment is used. The use of virtualization can slightly reduce the involved efforts, since it avoids dealing with physical interconnection problems between the wormhole and the payload. Finally, using the same OS makes implementation tasks easier from the perspective that a common development environment can be used for the implementation of the whole system (wormhole and payload parts).
- **The coverage of the assumptions that may be achieved**, which tends to be higher also when completely separate hardware is used. In fact, this is pretty obvious, since in this case it is easier to enforce the construction and the encapsulation and interfacing principles (see Section 3.2). On the other extreme, when sharing the same operating system (even if with different containment regions), there is an increased risk of interference or contamination and therefore the coverage of synchrony or security assumptions will be typically smaller.

Clearly, since in HIDENETS we address some applications that have safety-critical requirements, it is important that the stronger properties of the wormhole environment can be satisfied with high coverage. This is one of the main reasons why we have adopted the choice of using different hardware for implementing the oracles.

Finally, it should be mentioned that it may be possible to classify applications according to certain characteristics, which make them more suitable to be implemented and make use of the HIDENETS hybrid architecture and the oracle services. For instance, fail-safe applications, which concerns applications that can switch at any moment to a fail-safe state and remain there permanently to ensure that safety properties remain valid, are a relevant class. The Platooning application is an example of a fail-safe application, since it has a safe-state, in which the car in the platoon simply stops and therefore will never crash into the front car. This kind of applications cannot be implemented in generic distributed system with uncertain timeliness, since in these systems it is not possible to limit the time it takes for the application to switch to the fail-safe state. However, with the existence of a MW oracle service devoted to the timely detection of failures and execution of handler functions, this would become possible and, therefore, this class of application could be implemented with the necessary dependability guarantees.

Other classes of applications include the time-elastic class and the time-safe class. The former concerns applications that rely on timing parameters that may be adapted dynamically. In HIDENETS some applications can be considered to be part of this class, in particular all those related to the transmission of multimedia content. The later concerns applications, the safety of which does not depend on any timeliness property. This is a relevant application class in the sense that it characterises applications in which it is possible to fully separate logical safety properties from timeliness properties. In this kind of applications the code implementing logical safety properties does not depend on time (e.g., timeouts), which means that it is possible to guarantee that if any timeliness property is violated (e.g., due to a timing fault), this will have no impact on any of the safety properties, which will always be secured. If an application of the time-safe class is also of the time-elastic class (see above), it will be able to cope with faults during the process of adaptation, and still ensure that some other important properties (usually QoS-related properties) still remain valid as an outcome of adaptation.

3.3 Timeliness and trustworthiness oracles

3.3.1 Challenges and activities

There have been a number of challenges that were addressed in the context of WP2, both by defining an appropriate architecture and by specifying a set of services that can be used, in isolation or in combination.

The hybrid structure and the services in the wormhole part of the system are meant to address the following challenges:

- Provide means to address timeliness needs despite uncertainty
- Develop trust (including authentication) mechanisms
- Develop confidentiality and privacy mechanisms.

These challenges are addressed by the timeliness and trustworthiness oracles, forming the wormhole part of the architecture of a HIDENETS node. In particular we defined and developed the following services: Reliable and Self-Aware Clock, Duration Measurement, Timely Timing Failure Detection, Authentication and Trust and Cooperation.

On the other hand, the services in the payload part of the system are meant to address the following challenges:

- Maximise information availability despite faults
- Provide support for adaptation (through improved monitoring)
- Provide support for coordination

In order to address these challenges we have considered the following complex middleware services: Diagnostic Manager, Reconfiguration Manager, QoS Coverage Manager, Replication Manager, Inconsistency Estimation, Proximity Map and Cooperative Data Backup.

In what follows, we describe the specific challenges that are addressed by all these services services (additional detailed descriptions can be found in Deliverable D3.3 [12]).

3.3.1.1 Reliable and Self-Aware Clock

It tries to provide means to address timeliness needs despite uncertainty.

In distributed, open, dynamic pervasive systems like in HIDENETS, many applications and services may have critical aspects to deal with, in order to provide a dependable (e.g.: safe) service. Examples of such aspects are: i) temporal order delivery (for example, the physical time of sensor readings in data fusion process); ii) temporal consistency; iii) reduced and reliable transmission delay. These applications and services are usually time-dependent and use timestamps intensively. Timestamps can be obtained by reading the local clocks of the nodes of the distributed system. Time measurements can be obtained through these timestamps. So, distributed time measurements are performed using clocks of local nodes, and any difference between the clock time views affects the measurement results. Thus, temporal deadlines, distributed estimations, monitoring activities are affected by discrepancies between clocks of involved nodes. As a consequence, it is easy to see that in this kind of systems, the distributed applications and services (especially real-time) require a common view of time. In order to have a common view of time, it is required that nodes keep their clock synchronised with respect to a global time (it is TAI[2] reference time in HIDENETS): this

[2] International Atomic Time (TAI), Bureau International des Poids et Mesures. Time, Frequency and Gravimetry Section. Website: http://www.bipm.org/en/scientific/tai/.

global time is required to correctly execute a large set of applications and protocols, especially real-time ones. Clock synchronization is thus a fundamental process.

However, in complex systems like HIDENETS systems, it is impossible to ensure a-priori that nodes will have a “reasonably good” common view of time: despite the use of synchronization protocols, the time view that a local clock imposes to its node may deviate from global time. Clock synchronization protocols, the clocks themselves and consequently distance from global time are influenced by unpredictability and unreliability factors: distance from global time may vary due to factors related both to the distributed system and network behaviour and to the node internal behaviour. To overcome these problems, usually systems assume worst-case bounds that are necessary constraints that allow distinguishing unreliable biased data due to poorly-synchronised clocks, from reliable data collected when clock synchronization is good. However these bounds are usually pessimistic values, far from the medium case and are of little practical use.

In HIDENETS we decided to include in the software architecture of the node a new component, the R&SA Clock. It is a specific software component usable to obtain time values: instead of a time value composed only by a temporal indicator (i.e., time value c(t) read from local clock), this clock allows obtaining both local clock time value and information on synchronization uncertainty that indicates the quality of the temporal value collected. In this way the components of the system are aware of the quality of the time value, and consequently can use this information when creating timestamps and collecting/using/analyzing timing information.

3.3.1.2 Duration measurement

Duration measurement tries to provide means to address timeliness needs despite uncertainty. The purpose of the duration measurement service is to offer a method to measure local or distributed durations; the particular characteristic of the duration measurement service that we included in the HIDENETS node architecture is that the measurements offered by the service are with bounded precision.

Duration measurement is an important building block for other services, such as different kinds of failure detectors. It is particularly important for timely timing failure detection, in which case special care must be taken during the design, to secure that measurements are done in a timely manner. In fact, this is one of the reasons why we consider this service as an oracle, since it must be in a part of the system with specific properties (with respect to synchronicity and security, for instance) in order to behave in a timely and trustworthy way.

3.3.1.3 Timely timing failure detector (TTFD)

It tries to provide means to address timeliness needs despite uncertainty. In HIDENETS we consider several applications with timeliness requirements. These include applications whose QoS requirements are specified in terms of temporal bounds or applications the correctness and safety of which depend on the timely exchange of information. Because of that, it is necessary to monitor the timeliness of relevant local activities (involving the execution of local tasks) or distributed activities (involving communication activities between different nodes), as a first step to detect timing failures and react to them. Timing failure detection is an instance of the more general problem of failure detection in which timing failures are considered (instead of more traditional class of crash failures, [11]). Moreover, freshness detection is a special case of timing failure detection. Several distributed systems with real-time requirements need to use fresh information and, in addition, they need to detect the freshness level of available data. The level of freshness of the data received has to be assessed to guarantee the safety. In this perspective, the Timely Timing Failure Detector, when applied to the detection of message freshness, is a fundamental building block of the HIDENETS architecture.

We decided to include this service in the wormhole part of the architecture since its service must be offered with real-time guarantees and its correct behaviour is critical for the correct behaviour of the applications.

Provided that an adequate programming model is employed, which allows applications to exploit the ability of this service to detect timing failures in a timely manner (and execute associated fault handling functions), this service can be fundamental to allow the implementation of safety-critical applications despite the fact that these applications communicate and execute in environments with uncertain timeliness.

3.3.1.4 Authentication

The need for secure communication channels is common to several of the applications considered in HIDENETS. For instance, when retrieving information from a server in the infrastructure (e.g., to upgrade a software module in the car), it is fundamental to ensure that the correct server is being contacted, and therefore some kind of authentication is needed. Moreover, an authentication service should be implemented in a trusted part of the system, as close as possible to the sources of the information that must be authenticated. In particular, if sensory information concerning vehicle data is to be exchanged in a trustworthy way between applications, this information should be authenticated as soon as it is collected. Therefore, we decided to include the authentication service on the wormhole part of the system, assuming that any critical information will pass through this service before being handed to applications on the payload part of the system.

3.3.1.5 Trust and Cooperation Oracle

The Trust and Cooperation Oracle (TCO) is a basic service for cooperative applications. A cooperative service emerges from the cooperation of entities that are generally unknown to one another. Therefore, these entities have no a priori trust relationship and may thus be reluctant to cooperate. In cooperative systems without cooperation incentives, entities tend to behave in a rational way in order to maximise their own benefit from the system. The goal of the Trust and Cooperation Oracle is therefore to evaluate locally the level of trust of neighbouring entities and to manage cooperation incentives.

The solution proposed in HIDENETS for the TCO proposes a cooperation policy based on the use of a smartcard. The smartcard is used for generating, storing and verifying cryptographic signatures. The TCO thus uses the smartcard for verifying that: 1) a device follows the cooperation policy defined, and 2) that the software it runs is genuine, i.e., that the software is signed using the appropriate signature.

3.3.2 Conclusions and lesson learned

The work done in HIDENETS demonstrated the feasibility and usefulness of the concept of architectural hybridization, which was used as basic idea behind the definition of the HIDENETS node architecture. Deliverable D3.3 [12] contains the detailed description of the “oracle part” of the HIDENETS resilient architecture and it describes in details lesson learned and conclusions. An important open research problem addressed in HIDENETS was the timely detection of timing failures (e.g. needed in order to provide safety in several HIDENETS reference applications, like platooning). This work was mainly related to the specification, design and implementation of the following middleware oracles: Reliable and Self-Aware Clock, Duration Measurement and Timely Timing Failure Detector. With respect to trustworthiness aspects, the research was mainly done in the specification, design and implementation of the Authentication oracle and of the Trust and Cooperation oracle.

With respect to the timeliness and trustworthiness oracles, we summarise here lessons learned and conclusions about their role in the architecture and about their development.

3.3.2.1 Reliable and Self-Aware Clock

In HIDENETS we developed two versions of R&SAClock:

1. A C++ implementation of the R&SAClock for the Network Synchronization Protocol (NTP) and Linux operating system.

2. An implementation made on top of an embedded device with a Real Time OS, using a GPS receiver as time source. This implementation aims to realise an R&SAClock in an environment without support of clock–oriented system primitives and using a low level programming language for the typical HIDENETS mobile node. The implementation was done on the device (Lantronix device) really used as wormhole device in the platooning prototype.

Also thanks to these implementations, in HIDENETS we demonstrated the feasibility of the usage of the new concept of R&SAClock in distributed, open, dynamic and pervasive systems; in particular we demonstrated the usage of this component in a typical application in which we have critical aspects to deal with, in order to provide a dependable (e.g., safe) service: the platooning application. The Reliable and Self-Aware Clock increases the resilience of the system through providing to each node of the system a view of the time that is aware of its quality (in particular, in terms of the uncertainty of the provided timestamps). This quality awareness can deliver important information for a decision function which has to handle safety critical situations. The implementation of R&SAClock that we developed in HIDENETS use a simple method to evaluate the uncertainty interval of the timestamp; the uncertainty is evaluated using a linear envelope based on the knowledge of a maximum drift rate of the hardware clock. This method is applicable in most cases and it is safe (assumed that we are able to define a maximum drift rate of the clock).

3.3.2.2 Duration measurement

The duration measurement service is simple, by design, and because of the functionality it implements. Because of that, an important outcome of the work is that the duration measurement service can be easily integrated with other services. In HIDENETS, duration measurement is tightly connected to the TTFD service. When a request for monitoring a timed action is received (and directed to the TTFD service), the duration measurement service is used to do the measurement. On the other hand, the R&SA Clock is used to obtain timestamps of the bounding events, which are then used to calculate a duration. Clearly, if the appropriate formulations are used, it is possible to obtain a measurement with a known precision. The most important conclusion is that for an application that uses this service it is possible to be aware of the precision of measurements (of time intervals), which is particularly important when the application correctness and performance is also dependent, among other things, on the precise characterization of time intervals.

From a practical perspective, a relevant issue, which is in fact a difficulty that we have observed, is that timer and scheduling services (e.g. provided by some operating system or run-time executive) access the internal clock directly. Therefore, in this case it is necessary to make changes to third party code to implement the calculations that provide the guarantees on the measurement errors. Nevertheless, since we considered a modular approach, where the R&SA clock is an independent service, the existing code only needs to be updated to integrate the duration measurement calculations.

The duration measurement service was demonstrated and used in the scope of the platooning test-bed. It was used to measure the duration of timed actions that are monitored by the Timely Timing Failure Detection service, and the resulting measurements were provided to the payload part of the platooning application (which implements a decision making algorithm in which these measurements are used). In this sense, we were able to illustrate the practical relevance of the service.

3.3.2.3 Timely timing failure detector (TTFD)

The timely detection of failures of timed executions is very important, namely for safety reasons. This has been explicitly shown through the Platooning application, in which the service is used to ensure that the application secures some safety-critical properties, while providing an improved service when compared to the service that would be provided by some classical implementation, on a homogeneous, synchronous

system. Note that because of the hybrid architecture assumed in HIDENETS, it is possible to enjoy better synchronism properties for the oracles than for the rest of the system. Therefore, not only the detection of the timing failures can be made in a more timely fashion, but also the reaction can be done more promptly and timely, if necessary, by implementing it in tight connection to the failure detection service, separately from the rest of the system. In fact, this is what we have done in the context of the implemented platooning test-bed [40], and this was instrumental to show the effectiveness and usefulness of this service and of the architectural hybridization approach.

The timely timing failure detector can also support the detection of the freshness of the exchanged messages. Freshness detection is an important task in several typical HIDENETS applications: this service detects if fresh-enough data are available, in order to allow the application/the middleware to react to the absence of fresh-enough information. The functionalities of freshness detection are provided by TTFD in HIDENETS.

In order to exploit the (improved) temporal properties provided by the TTFD service, in HIDENETS we studied and proposed a set of alternative programming styles for the interaction between payload applications and the TTFD service. It is important to mention that handling the interface between synchronous and asynchronous systems is a difficult problem. A subtle but crucial issue is that a time chain must always be constructed in order to never lose track of time and always be able to detect lack of timeliness. We have demonstrated the effectiveness of the proposed solutions by means of the platooning test-bed.

3.3.2.4 Authentication

We implemented the authentication oracle as part of an embedded wormhole, more specifically, a Lantronix UDS 100 device, with its own processor and interfaces. Given the time consuming (cryptographic) operations involved in this service, and the fact that a general purpose 16 bit (thus not very performant) processor was used, we were not able to explore the typical capabilities of hardware for high performance cryptography. In fact, perhaps the major lesson learned in the implementation of the authentication service is that in order to implement a versatile and general purpose authentication service, special hardware is required instead of low performing general purpose one.

For the specific case of HIDENETS, and even more particularly, considering the requirements of the platooning test-bed, we initially opted to use a system of hybrid symmetric-asymmetric cryptography. A mechanism of symmetric message signing can be implemented as a message authentication code (MAC). This works by producing a cryptographic digest of both the data to sign and a secret text, resulting in a hash value which can be used as a signature. The secret text is private to each pair of interlocutors, generally kept only for a single session, and can be exchanged securely by first being encrypted using the public key of the recipient entity.

The implementation devised for the wormhole was based on this mechanism. For the secure hashing function, we selected the MD5 function. To generate a secret key, we chose the wormhole's random number generator functions (in non proof-of-concept implementations the source of random numbers should be chosen with great care, to assure sufficient entropy). The computational power of the adopted wormhole hardware revealed still not to be sufficient with this optimization. In particular, the processing time to encrypt the secret key did not fit within the bounds allowed by the hardware's watchdog. Solving this would require further optimizing the cryptography algorithm's implementation or partitioning the algorithm in several chunks.

In order to solve the problem in a practical way, and not compromising the objectives of the authentication service, we opted for an implementation, the final one, based on pre-sharing a key between the participants, to use with the described hybrid cryptographic system. Such solution would not be the most appropriate in the generic case, namely in automotive scenarios, in which each vehicle should typically have its own private key, and be able to dynamically negotiate secure channels with other vehicles. However, from the perspective of illustrating the architectural benefits of having an authentication oracle service, our practical approach was sufficient.

3.3.2.5 Trust and Cooperation Oracle

When designing a trust and cooperation oracle, many cooperation incentive schemes could have been used. These schemes are diverse not only in terms of the applications for which they can be employed, but also in terms of the features they provide, the type of reward and punishment they use, their operation over time and the technological impact they have on the hardware platform. The choice of using trusted hardware, a smart-card in our case, is definitely a good choice in an automotive context. Indeed, when the various entities participating in a cooperative service belong to the same administrative domain, or to a limited number of domains as it is the case for the automotive industry, the question of trust establishment can be answered using trusted hardware, in a simpler manner than with other solutions. In this case, a trusted authority within the infrastructure domain can easily certify automobile software and thus generate and distribute these certificates. We consider a smart-card as sufficiently tamper-proof, i.e. modifying the code embedded and obtaining the cryptographic keys that are stored on the smart-card is very difficult. Notice that current smart-card systems (e.g. JavaCards) are considered robust against attacks on private keys. It is worth noting that contact-less smart-cards are not usable yet as in the solution we advocate for, software is run on the card, raising energy consumption too much for this type of cards.

3.3.3 Relevant publications

[16] António Casimiro, Paolo Lollini, Mônica Dixit, Andrea Bondavalli and Paulo Veríssimo. A framework for dependable QoS adaptation in probabilistic environments. Proceedings of the 23rd ACM Symposium on Applied Computing, Dependable and Adaptive Distributed Systems Track (SAC'08), Fortaleza, Ceara, Brazil, March 2008.

[79] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, F. Grandoni. Threshold-based mechanisms to discriminate transient from intermittent faults. IEEE Transactions on Computers, 49(3):230–245, 2000.

[80] M. Pizza, L. Strigini, A. Bondavalli, F. Di Giandomenico. Optimal discrimination between transient and permanent faults. In Third IEEE International High-Assurance Systems Engineering Symposium, pages 214–223, 1998.

[81] S. Porcarelli, M. Castaldi, F. Di Giandomenico, A. Bondavalli, P. Inverardi. A Framework for Reconfiguration-Based Fault-Tolerance in Distributed Systems, In R. De Lemos, C. Gacek, and A. Romanovsky, editors, Architecting Dependable Systems, LNCS. Springer-Verlag, 2004. Also ICSE-WADS2003, Post-Proceeding of ICSE-WADS2003.

[82] S. Porcarelli, F. Di Giandomenico, A.Chohra, A. Bondavalli. Tuning of database audits to improve scheduled maintenance in communication systems, in Computer Safety, Reliability and Security, Proc. of the 20th International Conference SAFECOMP 2001, Budapest, Hungary, pages 238–248. Lecture Notes in Computer Science 2187. Springer, 2001.

[83] Andrea Bondavalli, Andrea Ceccarelli, Lorenzo Falai. A Self-Aware Clock for Pervasive Computing Systems. 15th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2007), 7-9 February 2007, Naples, Italy. IEEE Computer Society 2007, pages 403-411.

[84] Bondavalli, A. Ceccarelli, L. Falai. Assuring Resilient Time Synchronization. SRDS2008. October 2008, Naples, Italy.

[85] Bondavalli, A. Ceccarelli, L. Falai, and M. Vadursi. Towards making NekoStat a proper measurement tool for the validation of distributed systems. In Proceedings of The 8th International Symposium on Autonomous Decentralised Systems, pages 377–386, March 2007.

[86] L. Falai. Observing, Monitoring and Evaluating Distributed Systems. PhD thesis, University of Florence, 2008.

[87] L. Falai, A. Bondavalli. RODS: General Framework for Rigorous Observation of Distributed System. DSN 2008 Workshop on Resilience Assessment and Dependability Benchmarking. Anchorage (USA), June 2008.

[88] Bondavalli, A. Ceccarelli, L. Falai, M. Vadursi. Enhancing the NekoStat Tool with Uncertainty, Resolution and Intrusiveness Evaluation Capabilities. DSN 2008 Workshop on Resilience Assessment and Dependability Benchmarking. Anchorage (USA), June 2008.

[89] Henrique Moniz, Nuno F. Neves, Miguel Correia, António Casimiro and Paulo Verissimo. Intrusion Tolerance in Wireless Environments: An Experimental Evaluation. Proceedings of the 13th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'07), Melbourne, Victoria, Australia, December 2007.

[90] Hans P. Reiser and António Casimiro. Optimizing Byzantine Consensus for Fault-Tolerant Embedded Systems with Ad-Hoc and Infrastructure Networks. 4th International Workshop on Dependable Embedded Systems (WDES'07), Beijing, China, October 2007.

[91] Hugo Ortiz, António Casimiro and Paulo Veríssimo. Architecture and Implementation of an Embedded Wormhole. In Proceedings of the 2007 Symposium on Industrial Embedded Systems (SIES'07), Lisbon, Portugal, July 2007.

[92] António Casimiro, Odorico Mendizabal and Paulo Veríssimo. On the development of dependable embedded applications using specialised wormholes. 3rd International Workshop on Dependable Embedded Systems (WDES'06), Leeds, UK, October 2006.

[93] T. Chandra, S. Toueg. Unreliable failure detectors for reliable distributed systems. Journal of the ACM, 43(2):225–267, March 1996.

3.4 Complex resilience middleware Services

3.4.1 Challenges and activities

3.4.1.1 Diagnostic Manager and Reconfiguration Manager

The challenges addressed by Diagnostic Manager (DM) and Reconfiguration Manager (RecM) are the following:

- Provide support for adaptation (through improved monitoring). The system is in fact required to adapt to the following situations: i) faults affecting both hardware and software components, which are inherently subjected to faults and, eventually, those faults could lead to components failures; ii) changes in the environment affecting the QoS provided by system services/components and hence perceived by their users, which could be no more satisfied; iii) changes in the requirements of the users of the component/service, so that users are no more satisfied of the same service.
- Maximise information availability despite faults.

The DM identifies faulty components online and diagnoses the occurred faults, triggering proper alarms; the RecM, is in charge of selecting/triggering the reconfiguration actions, possibly as a reaction to alarms triggered by DM, QoS Coverage Manager or directly from application needs. Other middleware services and oracles are involved in the process, some as information providers, and others as actuators of system reconfigurations. The joint work of the DM and RecM, together with some oracles and other middleware services, allows the system to adapt to the above mentioned situations.

The second challenge addressed by Diagnostic Manager and Reconfiguration Manager is to “Maximise information availability despite faults”. In general, by enhancing system/service resiliency and survivability, also information availability is improved, when redundancy of information providers is in place. Somehow,

this can be seen as a specialization of the above discussed adaptation challenge, because maximizing availability of information can be seen as a sort of adaptation when replicated information sources are considered. This aspect is particularly evident in the implementation of the DM and RecM in the scope of the platooning test-bed (see deliverable D6.4 [41]).

In order to perform the best adaptation possible, the following conditions must hold: i) diagnosis has to be as accurate as possible, ii) reconfiguration has to be as clever as possible, meaning that it has to perform the best reconfiguration with respect to the scenario depicted by the available diagnosis judgements.

In consideration of the above challenges, two approaches are adequate to implement the diagnosis: i) a heuristic approach (alpha-count, [13]) or ii) a probabilistic approach (diagnosis based on Hidden Markov Models, [74]). The choice between the two alternatives can be guided by the following considerations. Promptness[3] and accuracy[4] are conflicting goals [13]: the probabilistic approach is the most accurate ([14]), but it requires more computational resources both in space and time (so it is well suited for the infrastructure domain); the heuristic approach, instead, is computationally lighter (so it looks more appropriate especially in the ad-hoc network domain, where limited computational resources are available).

The high dynamicity of the HIDENETS system leads to interactions among dynamic groups of "reachable" nodes (mobile and fixed ones) along time. Each node has to be able to assess the status of nodes inside its group, because problems inside a node could negatively propagate into other nodes of the group. Some "global diagnosis" seems thus necessary, possibly organised in such a way that it is resilient to Byzantine faults. This is the reason why diagnosis in HIDENETS is designed to be organised as follows:

i) Local Diagnosis: diagnosis performed internally to the node on local resources/services (node auto-control), in order to react to local problems.

ii) Private Diagnosis: diagnosis performed inside a node on a remote node, inferred through the existing interactions between the two nodes when performing cooperative activities.

iii) Distributed Diagnosis: diagnosis performed in a distributed way among a set of collaborating nodes when an agreement about the healthy status of each node (Byzantine resilient) is necessary.

Reconfiguration can be performed according to different approaches, where extremes are i) a static approach, in which both the set of envisioned reconfiguration strategies and the static association between reconfiguration strategies and patterns of faults/deviations of system components are defined at system design time; ii) a dynamic approach, in which many strategies are applicable for the same diagnosed scenario, and the choice of the reconfiguration to be applied is performed on line through a proper evaluation support (fed with the specific system and environment conditions at the time the reconfiguration action is triggered).

Reconfiguration may span a single system component or major compounds up to the entire system. To cope with scaling issues in such a context, a hierarchical approach is suggested [15], with: i) reconfiguration local to a node (to either resist to local diagnosed faults or to better exploit local available resources); ii) reconfiguration at multi-node level (to better manage faults and resources at system level).

In HIDENETS concrete instances of DM and RecM were implemented in the platooning test-bed, focusing on the local diagnosis and reconfiguration aspects, where the mobile node monitors its resources and (possibly) reconfigures them. Each vehicle used in the test-bed is assumed to have two GPS receivers; the DM and RecM are used to manage the (active) redundancy of GPS receivers, so maximizing availability of location and timing information.

3.4.1.2 QoS Coverage Manager

The challenge addressed by QoS Coverage Manager is the following:

- Provide support for adaptation (through improved monitoring)

[3] how quickly the diagnosis mechanism reveals the problem, given that there is the problem.
[4] the ability to correctly discriminate each detected problem among all the envisioned ones

The QoS Coverage Manager is concerned with evaluating if time-related requirements of an application are satisfied, or else if it is necessary to provide an indication that some desired QoS may no longer be guaranteed and may need to be renegotiated. The service considers that requirements must be expressed in terms of a desired temporal bound to be secured for a communication link with a given probability. In fact, it is assumed that the service is to be used in probabilistic environments.

In order to do that, the QoS Coverage Manager will need information concerning the performance of the network and, in particular, it will need to use histories of measured parameters to perform statistical calculations and characterise the probability distribution functions of these parameters. In addition, other information concerning the state and configuration of the system and the networks (e.g., information concerning the amount and periodicity of transmitted data) may be used to better characterise the actual state and QoS delivered by the communication subsystem.

Given that this service needs to perform some relatively complex operation and it may be difficult to ensure that all these operations are done in a timely way (because this may depend on the amount of data and on the particular state of the environment), it was clear that this service should be provided to HIDENETS applications as a complex middleware service.

3.4.1.3 Intrusion-tolerant agreement

The challenges addressed by Intrusion-Tolerant Agreement service are the following:

- Maximise information availability despite faults
- Provide support for coordination

The idea of this service is to support HIDENETS applications in reaching agreement on specific data despite the possible presence of some faulty processes, acting in a Byzantine manner. Clearly, this is a fundamental service when coordination is needed. On the other hand, given the fault-tolerance characteristics of the service (which tolerates a fixed number of faults, more specifically, less than one third of the total number of participants that are coordinating their actions), it is clear that the service also contributes to maximise information availability despite faults.

3.4.1.4 Cooperative Data Backup

The challenges addressed by the Cooperative Data Backup service are essentially related to the improvement of the dependability of the data stored on participating nodes with or without the help of an infrastructure. The Cooperative Data Backup service does so by providing nodes with mechanisms to cope with hardware or software faults, including permanent faults such as loss, theft, or physical damage. To handle permanent faults, the service provides mechanisms to store the user’s data on alternate storage nodes using the available communication means.

The problem of cooperative backup of critical data consists essentially in discovering storage resources in the vicinity (i.e. on the neighbouring nodes), establishing trust with the neighbouring cars for the use of their resources, handling a stream of data chunks to backup, and assigning these chunks to the negotiated resources according to some data encoding scheme (and with respect to desired properties like dependability, privacy, confidentiality, etc). Of course, a pending service provision is the counter-part of this one: offering storage resource to other participating nodes. The service must also take care of the recovery phase, i.e., the data restoration algorithm: discovery and collection of the various data chunks disseminated in the neighbourhood.

3.4.1.5 Proximity Map

The challenge addressed by the Proximity Map is the provision of a precise and up-to-date view of the location of surrounding mobile devices, despite the absence of a communication infrastructure and mobility of the nodes.

The goal of the Proximity Map service is therefore to provide an abstraction of the map of neighbouring nodes, along with estimated position, speed and direction. This is done by computing a hybrid representation of the local environment. In this context, hybrid means that the map presents to application and services an abstraction that mixes physical and computational information (i.e. respectively geographical and communication related). Thus, the Proximity Map represents the local knowledge a node can compute about its vicinity in collaboration with its neighbours. This map is parameterised by its wideness (the number of communication hops represented on the map, which is related to the maximal physical distance of a node on the map) and its accuracy, i.e. how often the map is updated.

3.4.1.6 Replication Manager

The Replication Manager (RM) is a service that is used to handle replication in the middleware. A shared memory area is provided to the application, with standard memory operations available like read and write. The RM service must be available to all nodes, both user nodes, replica nodes and server nodes. The RM is responsible for where to store the replica and how to access the replica. The RM aims to hide details about change of replica servers and topology changes from the application. Furthermore the RM tries to optimise the way the replicas are selected in order to optimise the application user experience. The RM makes use of a distributed cluster formation algorithm which will select replica candidates so that the state replication can be done satisfying the desired replication scheme. Examples of such replication schemes can be to spread the replicas over a large area or only using one hop neighbours as replicas.

The scenario where the RM will be useful is in the ad-hoc domain. It is intended to be used with a service provided in the ad-hoc domain by other ad-hoc nodes. Each car is running a set of services. Some of these services are state-full and need to share their states with failover candidate services in other cars to gain dependability. The RM is supporting the application by sharing the state of the service with peers in the network. Backup servers are selected based on the relevant communication metrics like end-to-end delay, position and speed to reduce inconsistency and keep reconfiguration at a minimum. Whenever a new peer-server is selected the RM updates its list of associated replicas both in the application server middleware and in the application user middleware. The Replication Manager will contribute to minimization of state inconsistency and improving dependability.

3.4.2 Conclusions and lesson learned

3.4.2.1 Diagnostic Manager and Reconfiguration Manager

Giving a detailed specification for the DM and the RecM deeply depends on several factors: the scenario in which diagnosis and reconfiguration are going to be implemented or used (local, private, global), the criticality of the supported application which may lead to specific performance requirements (in terms of promptness and accuracy), the characteristics of the supporting environment (infrastructure or ad-hoc domains). In HIDENETS high level descriptions of the DM and RecM were given, identifying the interfaces with the other services, selecting the proper mechanisms and discussing some design organizations w.r.t. some application contexts or interaction scenarios (ad-hoc, infrastructure). Based on the above factors, the DM and RecM could be instantiated as very complex or very simple services.

Starting from the high level specification, a very simple implementation of both DM and RecM was developed within the platooning test-bed (see HIDENETS D6.4 [41]). In this scenario, the DM and RecM are implemented as a single DM+RecM middleware service which is in charge of managing the redundancy

of GPS information: the DM+RecM has to discriminate between three possible fault classes (“crash”, “omissive”, “assertive”) for the GPS receiver, performing a simple merge of the available information, based on the diagnosed status of the receivers. For example, when both GPS receivers are diagnosed as “correct” (in this case they give consistent information), GPS information is merged and forwarded to the application level; if one receiver is diagnosed as affected by an “omissive” or an “assertive” fault (in this case there are discrepancies between the received information) and the other is diagnosed as “correct”, the information produced by the correct receiver is forwarded.

3.4.2.2 QoS Coverage Manager

The QoS Coverage Manager was designed to support adaptive systems and applications in probabilistic environments, from a dependability perspective: maintaining correctness of system properties after adaptation. The correctness rule consists in securing the coverage of some time bound, keeping this coverage stable during the entire operation.

To implement the service we assumed that i) a system alternates stable periods, during which the environment characteristics are fixed, and unstable periods, in which a variation of the environment conditions occurs, and ii) that the mode changes can be detected. Based on that, we proposed and evaluated a general framework for adaptation, which allows to dynamically set optimistic time bounds when a stable phase is detected, while it provides conservative but still dependable bounds during transient phases.

Several mechanisms can be considered for phase detection and for estimation of the parameters describing the probabilistic distributions of the monitored stochastic variables. In fact, the actual mechanisms depend on the possible behaviours of the environment.

The main lessons learned and conclusions that may be pointed out as an outcome of our work are manyfold:

We verified that some mechanisms are more adequate when observing some probabilistic distributions, where other mechanisms would not work so well.

We have also verified that despite the uncertainty of behaviours, when applying our framework and mechanisms to real data traces (of round-trip communication in ad-hoc environments) it is possible to correctly detect probabilistic behaviours and achieve the desired objectives for the QoS coverage service.

We concluded, both through a theoretical complexity analysis and in practice (by using the service in the scope of the platooning test-bed), that the potential complexity and impact of this service on the system performance is acceptable and that the proposed solutions and their implementation present a good performance and satisfactory execution times.

3.4.2.3 Intrusion-Tolerant Agreement

The Intrusion-Tolerant Agreement (ITA) service was demonstrated in the scope of the platooning test bed. It was used for a decentralised decision process in which the whole platoon agrees on a common speed to which all cars will converge.

Regarding lessons learned, perhaps the most relevant issue to mention is that we found, and showed through evaluations, that despite the potential complexity and indeterminism of randomised based solutions for solving consensus, in practical scenarios, and in ad-hoc communication scenarios in particular, the approach presents an acceptable performance. For instance, we discussed a simple case, in which the platoon is stable and, at some point, the leader vehicle slows down, reducing its speed by 40 Km/h (we can assume this is done instantaneously). In this situation a new agreement must be reached and the ITA service is used. Moreover, the performance of the service must be good enough to ensure that the new speed value is agreed before the follower car gets too close. In this case, agreement should be reached within about one second, and we verified that this bound could be satisfied in most of the tested cases.

Despite the promising results from the point of view of performance, one important issue to mention concerns some of the assumptions that had to be made in order to use the implemented version of the ITA service. In fact, the service was implemented assuming a static number of participants (cars, in the case of the

platooning test-bed), which might be reasonable in several cases, but is a limitation. As future work, it would be interesting to consider more dynamic solutions, based on group membership protocols, for instance, on which the ITA service could be based.

3.4.2.4 Cooperative Data Backup

The design of a Cooperative Data Backup service requires the exploration of a variety of different domains. While we did not explore all the issues that relate to this goal, we tried to explore several different tracks, namely: the dependability of such a service, distributed storage techniques suitable for data fragmentation-redundancy-scattering.

We identified key requirements of the storage layer of our cooperative backup service, namely: storage efficiency, scattering of small data blocks, backup atomicity, protection against accidental and malicious data modifications, encryption, and backup redundancy. We analyzed techniques described in the literature that meet these requirements. We described our storage layer implementation and used it to implement a prototype of the distributed blackbox. We described the algorithms and techniques used for opportunistic replication, data retrieval and storage contribution, and discussed their parameterization. The cooperative backup is readily usable in a number of contexts, not limited to mobile devices or to vehicles. The end result is a practical cooperative backup tool readily usable in several different contexts, including the HIDENETS context.

3.4.2.5 Proximity Map

The Proximity Map service provides an abstraction of the map of neighbouring nodes along with estimated position, speed and direction. The Proximity Map represents the local knowledge a node can compute about its vicinity. It provides an abstraction that correlates both geographical and communication related information.

It is worth noticing that the implementation of the Proximity Map service is truly a “Best-Effort” one, since the information it provides is the most accurate possible, but no real quality of service can be guaranteed.

When several nodes run in the same direction at comparable speeds, the level of details of the Proximity Map service is dictated by the accuracy parameters (Beacon Interval and NbHops), provided all nodes use the same parameters, which is assumed in our test scenario. Furthermore, it is worth noting that increasing these parameters beyond a certain threshold could be extremely costly and would imply many collisions at the network level. An optimization of the service could be to piggyback Proximity Map messages on top of regular communication-induced messages but that would strongly impact the quality of the proximity information.

When a node is seen only a small amount of time, the accuracy of the Proximity Map depends on the following hardware parameters: the relative speed of cars, the time necessary to open a connection, and the throughput of the connection (those last two being related to the network interface).

In our laboratory setup, relatively small speeds are attained, and a traditional WiFi (with reduced range) is used. In a real life setup, the performance of the network interface will have to match the physical constraints of a car-to-car scenario, which shall be the case with the new 802.11p interface.

3.4.2.6 Replication Manager

The goal in developing the RM is to develop methods for: The selection of replica peers while keeping reconfiguration overhead low and at the same time keeping inconsistency between the service and its peers low.

With these three goals, reaching them all is not a trivial task. Finding a trade-off between keeping reconfiguration overhead (signalling) low and keeping inconsistency low is depending on the scenario. The selection of replicas is performed based on four metrics:

- Location, by limiting the scope of candidate peers to peers in the vicinity the measurement overhead of obtaining the remaining metrics is lowered
- Speed, to keep the reconfiguration overhead low the candidate server must travel in the same direction and with the same speed as the replicated server.
- Direction, same as above
- End-to-end delay, to keep inconsistency low the end-to-end delay must be as low as possible.

The underlying dynamics of the network makes it challenging and sometimes impossible to accommodate the requirements of all the users of the system. In case the users are too far away from the server cluster, the connection will be lost or the cluster group will be split up.

3.5 Resilient communication

Resilient communication in HIDENETS addresses the lower protocol layers, up to Layer 4, and also networking issues as will be described in the following.

3.5.1 Challenges and activities

Various aspects have been addressed with the objective to contribute to the design of a cost-efficient highly reliable wireless network in combinations with their interconnecting (fixed) networks. Emphasise has been put on analysing and improving resilience properties and related performance properties and on the improvement of protocol design with the aim of increasing the overall efficiency.

The main activities have been centred in the following areas:

- Multi-channel multi-radio architecture
- IP resilient routing
- Efficient routing
- Efficient and reliable broadcast
- Resilience in connecting to the infrastructure domain
- Cross-layer optimisation

These areas are detailed further below.

3.5.1.1 Multi-channel multi-radio architecture

Although multiple frequency channels are available for WLAN communication, most systems use only one frequency channel. Using IEEE 802.11a/b/g are based on a single channel architecture. On the other hand the draft standard on vehicular communication, IEEE 802.11p, makes use of multiple channels by defining a mandatory common channel for safety applications and using other channels for data communication.

In HIDENETS the idea has been to use multiple channels to increase network resources by reducing interference between links, and use multiple radios to allow communication on different channels in parallel and to allow for fast recovery in cases of failure of an ad-hoc node or link. This is an option to make even more efficient use of multiple channels and to provide a more dependable system. The focus has been on

improving network resources for multi-hop ad-hoc networks, but multiple radios could also be used in WLAN access points, where it might be a possibility to increase the redundancy in access to infrastructure.

When using multiple frequency channels a management module is needed to assign channels to radio interfaces. This channel assignment module can operate stand-alone, i.e. it may be a simple fixed assignment of different channels in the case of multiple radios per node or implement a distributed protocol to assign channels to radios. Optionally, the channel assignment could be influenced by upper layers, for instance to enable topology-based or routing-based channel assignment. This cross-layer optimization possibility is discussed in section 3.5.1.6.

The channel assignment module may contribute to increased dependability and capacity in the ad-hoc domain, by optimizing the channel assignment in an ad-hoc network in a distributed manner. The challenge is to conserve connectivity while reducing the interference between links to a minimum.

3.5.1.2 IP resilient routing

The term routing refers to selecting paths in a computer network along which to send data. Prior to selecting these paths, three main steps are normally fulfilled: Neighbour discovery, information dissemination (using a routing protocol) and path calculations. In ad-hoc networks, topology control is also needed to select a connected topology among the potentially mobile nodes (see above). This process (topology control) will determine what topology information will be disseminated to other nodes in the network. The result of these processes is routing tables.

In a proactive link state protocol, each router might obtain a complete picture of the IP network topology and all possible paths to each destination. This information is inserted into the routing table. Shortest path calculations are performed on this IP network topology, and the resulting best path(s) towards each destination is inserted into the forwarding table. The forwarding table is used by the IP forwarding and route resilience service. Maintaining and calculating backup routes is also an important part of this module, too. The result of this calculation will also be used by the IP forwarding and route resilience service in case of failure.

For different purposes it could be beneficial to maintain several routing topologies, e.g. for QoS routing and resilience. This is highly relevant in the car accident (section 2.1.3) and infotainment (section 2.1.2) use cases. The main focus in these use cases and in HIDENETS in general has been resilience, so the challenge here has been to develop robust methods, i.e. construct such backup topologies for handling situations with link or node failures in the ad-hoc network (topologies that can be used for fast recovery of connectivity) when a link or node degrades, goes down or moves out of reach.

3.5.1.3 Efficient routing

Routing in ad-hoc networks has challenges related to the traffic overhead it creates and the occurences of broadcast storms. Keep in mind that wireless links in wireless multi-hop networks are resource constrained. The reduction of routing overhead can actually reduce the network load and leave more resources to the data traffic, which may improve the end-to-end performance. So how can we reduce the routing overhead in ad-hoc networks?

In infrastructure networks, nodes establish an association and subsequently exchange routing information with all neighbouring nodes. In ad-hoc networks, however, the number of nodes within physical reach may be very high and dynamically changing. Ad-hoc Topology Control can be defined as the problem of computing and maintaining a connected topology among the ad-hoc network nodes, i.e. for each node selecting which neighbours to connect to based on some predefined criteria. The topology can be optimised with respect to different parameters like for instance energy consumption, total network capacity, or network stability.

Ad-hoc Topology Control is particularly hard in networks with a high degree of node mobility, where associations would potentially only last for a short period of time. The major challenge is to do this without a central overview of the full topology. For handling the dynamics of the topology, the ad-hoc topology

control service issues link-state update messages to surrounding neighbours. A link-state update message contains the view of the surrounding topology from a single node. Based on receiving local topology information, the ad-hoc topology control service can calculate its view on the connected topology.

The HIDENETS approach is the design and analysis of a mechanism implementing circuit-elimination based connected dominating set formation. This is an efficient technique for reducing routing overhead in mobile ad hoc networks. The main focus has been on link state routing. This problem is related to the general problem of efficient broadcast and the main contribution is increased dependability due to the reduced probability of overload (i.e. broadcast storms).

3.5.1.4 Efficient and reliable broadcast

Broadcasting is the transmission of data to all stations connected to a particular network segment. While this is straight-forward in infrastructure networks, ad-hoc networks need extra functionality to handle this. Based on the local topology and the past reception of the same broadcast message, a node decides locally whether this message should be broadcasted further or not. This decision can be influenced by several requirements, e.g. to optimise efficiency or reliability of the broadcasting. Efficiency optimization regards decreasing the overhead in the network when broadcasting messages to every node in the topology. Reliable broadcasting regards guaranteeing that all nodes – eventually – receive a broadcast. To fulfil this objective, the past reception of all broadcast messages should be recorded. Ensuring broadcast reliability with minimised communication overhead is a challenging task when error-prone links are considered.

The challenge is to be able to eliminate unnecessary broadcast messages and thereby contribute to:

- Increased efficiency in communication (spectrum, energy)
- Reduction of the collision probability as unnecessary broadcasts are eliminated.
- Ensure that all targeted nodes receive the broadcast.

3.5.1.5 Resilience in connecting to the infrastructure domain

The main objective is how to increase the Internet availability for a node connecting directly to the infrastructure IP domain. This is highly relevant in the car accident (section 2.1.3) and infotainment (section 2.1.2) use cases. Multihoming improves the reliability of the connection to infrastructure by removing single-point-of-failures in the link to the infrastructure network. From a topology point of view it is similar to the methods of the previous section in that we have backup routes to the Internet in case an Internet gateway fails. This will enable fast recovery/ handover and as such increase the dependability of the Internet connection.

In HIDENETS multihoming is combined with Differentiated Resilience. In this way the scarce resources in this part of the network may be utilised more efficiently, and the load on the signalling system may be reduced in failure situations.

The challenge is then how to combine information about available technologies, gateways and networks, and information about QoS and resilience requirements, to find the best working path and possible backup path for a given communication need and simultaneously utilise the network resources efficiently.

Different strategies have been investigated by simulations. Further details can be found in [39] .

3.5.1.6 Cross-layer optimisation

Here we study the redundancy and overhead in the different protocol layers over the wireless domain, and suggest improvements of the protocol design with the aim of increasing the overall efficiency.

This has been handled in two parts:

- Firstly, a specific cross-layer resilience optimization solution has been described in detail. This solution concerns cross-layer optimization of message broadcast in a specific use case which is an aggregated derivate of the HIDENETS use cases Floating Car Data, platooning and Hazard warning.
- Secondly, interactions between components in the HIDENETS node architecture are discussed to determine whether additional opportunities for cross-layer optimization are available.

The challenge is to handle such optimisation in a robust way. This is especially important due to the rapid changing conditions in wireless networks, and not the least in a dynamic setting with moving cars (or mobile terminals in general).

3.5.2 Conclusions and lessons learned

3.5.2.1 Multi-channel multi-radio architecture

Based on a state-of-the-art study and an extensive simulation study, a multi-channel multi-radio architecture has been proposed as a means to enhance resilience by increasing the overall performance of the network and the availability of radio resources. Simulation results show the benefit of using multi-channel multi-radio with a variable, but realistic number of channels and radios for IEEE 802.11 based systems. The analysis has shown the usability of the proposed architecture in mobile (vehicular) environments, especially when using multi-hop communication. It has been shown that the optimal number of radios per node depends on network traffic patterns and node density. Also, practical considerations have been discussed and the multi-channel multi-radio architecture has been implemented in the resilient communication test-bed (see section 7.4). The proposed architecture is based on using multiple COTS IEEE 802.11 network interface cards.

The main lessons learned in the process are:

- By using multiple frequency channels in a mesh network, it is possible to increase the capacity and fault-tolerance of the network.
- By using multiple radios per nodes, it is possible to increase the performance gain of multi-channel and this allows for robust channel assignment algorithms that do not depend on synchronization or communication between nodes.
- It is feasible to build MCMR nodes with COTS products (see section 7.4), although node-internal interference needs to be taken into account when designing such nodes.

3.5.2.2 IP resilient routing

We have developed and evaluated a set of IP fast reroute schemes for scenarios with proactive link-state routing. The results showed that the source-destination connectivity improves considerably compared to having no fast reroute. We have also seen that planning for both node and link failures can be beneficial. Taking all the functional requirements and important performance metrics into consideration, we have found that IPRT and DMRC [16] are the most viable candidates for IP fast reroute in the scenarios addressed. See deliverable 2.1.2 [1] for further details.

In HIDENETS, we have identified brigade communication as the use case that can benefit strongly from fast reroute. On the accident scene, the emergency crews can establish a fairly stable ad hoc network for internal communication and information exchange. However, since such a network is based on wireless links, failures may occur. This network will carry critical real-time communication, and hence some recovery guarantees are required. It is also important that the bandwidth is not consumed by looping packets as is possible with the deflection, FIFR and LFA schemes [16]. The evaluation has pointed out IPRT and DMRC as the most viable candidates. They both provide high recovery success rates and acceptable path lengths. They also offer a great flexibility for different optimizations.

Although IPRT and DMRC seem to be the best candidates, they are not perfectly designed to fit any wireless network that uses proactive routing. Currently, they both rely on full topology knowledge to calculate the alternative next hops. As a future task it is required to optimise their performance in cases where only limited

neighbour information is provided. In addition, one should also address the case where the different nodes have inconsistency in their topology view. There is also a great potential for improving the state requirements and forwarding complexity of both schemes.

3.5.2.3 Efficient and reliable broadcast and routing

We have developed efficient and reliable broadcasting - a communication level component to guarantee delivery of message broadcast in a network by using information local to the nodes in the network. By using a distributed approach, where the nodes only use local information, and creating a virtual back-bone (called a connected dominating set (CDS)) out of this local information, the amount of information sent between nodes can be kept low.
The broadcasting component can be used by itself, utilised by other, higher-layer services, or it can be integrated into other components. This is the case in HIDENETS, where an efficient optimised link-state routing has been devised. This is a so-called proactive routing protocol, where nodes periodically broadcast link-state update messages to neighbours. These messages contain the nodes' current view of the 1-hop topology. The effect of a guaranteed delivery of broadcast messages is increased confidence in the state of the links in the topology and thus a more efficient routing protocol.

The main lessons learned in the process are

- By using information local to each node instead of continuously building a large topology, and by using this local knowledge to build a back-bone of nodes, it is possible to guarantee delivery of messages with a reduced number of messages compared to traditional reliable broadcasting.
- By carefully combining different algorithms, the performance of the reliable broadcasting component can be optimised to several types of scenarios, e.g. static and mobile nodes.
- It is possible to utilise reliable broadcasting in link-state routing and thereby make the routing use less overhead and still be able to provide as high quality routes as traditional link-state routing in terms of packet delivery ratio.

3.5.2.4 Always Best Connected (ABC) and differentiated resilience

To address parts of the challenge described in the previous section on "Resilience in connecting to the infrastructure domain", we have studied how resource availability and reliability for Mobile Terminals (MTs) is affected when using QoS differentiation and selected resilience mechanisms (protection). This has provided us with knowledge on how well these schemes are suited for different densities of Best Effort (BE) and High Priority (HP) MTs when nodes are mobile (and given the characteristics of our simulation environment). For our simulations the Access Points (APs) have a fixed total capacity and a fixed capacity limit for high priority reservations, and MTs with high resilience requirements may be connected to 2 APs simultaneously.

We have learned that:

- Using reservations for high priority traffic, potentially with protection, provides significantly increased availability and reliability compared to BE - when the number of HP nodes is relatively low. With high number of HP nodes using protection, the use of extra resources by these nodes may result in starvation.
- The choice of protection scheme for HP nodes significantly impacts availability and reliability for the BE nodes.
- As wireless access scenarios with mobile nodes tend to be highly dynamic, the usage of protection should probably be made more dynamic than what has been simulated, with different strategies for different densities of mobile nodes. When resources are scarce, the most resource hungry mechanisms should be avoided as lack of resources has negative impact on reliability. This is supported by results showing the effect of using the existing mechanisms for different densities of MTs.

- According to our simulations, providing increased availability and reliability by just increasing the total AP capacity and using BE resource allocation for all nodes will require much more total capacity than what is required to provide this for a relatively few nodes when using differentiation and protection.

3.5.2.5 Cross-layer optimisation

As documented in [75] HIDENETS has considered the development of a framework for cross-layer resilience optimization and to develop a specific optimization scheme, which chooses the optimal values for protocol parameters, such as transmission power, modulation scheme and forward error correction code rate for a flooding broadcast message dissemination service. The optimal layer parameters are determined from a probabilistic model of the protocol stack, which is composed of several sub models corresponding to relevant layers in the protocol stack. The physical and Media Access Control (MAC) layers are based on probabilistic models from existing literature, whereas the flooding broadcast model in the network layer has been developed within HIDENETS. During the development of the optimization models and by means of simulation studies we have learned that:

- The proposed optimization approach works well in some cases, but is too optimistic since correlated losses are not considered. Furthermore it was found that a queuing model is necessary to predict when buffer congestion occurs. This should be added to complete the optimization model.
- Transmission power and bit-rate/modulation adaptations worked well as mechanisms for increasing performance and thus adding a higher degree of resilience to the system.
- A broadcast model for ad-hoc wireless networks should take correlated losses due to collisions into account. An assumption of independent losses does not apply for high-contention scenarios when no coordination of transmission is possible as is the case for Layer 2 broadcasts.
- A more dependable broadcast protocol such as the efficient and reliable broadcast developed within HIDENETS (see [75]) should be used for this type of message dissemination to minimise the amount of overhead.
- Realization of the proposed optimization is possible by use of IP and IEEE 802.11e traffic classification in combination with existing HIDENETS components. This would require a modest extension of the multi-channel management considered in the multi-channel multi-radio architecture to recognise the targeted traffic flows.

Besides considering the specific message broadcast cross-layer resilience optimization, potential benefits of cross-layer interactions have been discussed in relation to jointly considering the efficient and reliable broadcast, fast-reroute, and multi-channel multi-radio functionalities. Most noticeable result was that:

- Cross-layer interactions between multi-channel multi-radio and fast re-route parts potentially can give performance enhancements greater than what is achievable by use of the multi-channel multi-radio architecture alone.

3.5.3 Conclusions

Various aspects of the communication layer have been addressed with the objective to contribute to the design of a cost-efficient highly reliable wireless network in combinations with their interconnecting (fixed) networks. Emphasise has been put on analysing and improving resilience properties and related performance properties and on the improvement of protocol design with the aim of increasing the overall efficiency.

The current MCMR architecture provides a robust generic solution for mobile ad-hoc car-to-car mesh networks. The idea has been to use multiple channels to increase network resources by reducing interference between links, and use multiple radios to allow communication on different channels in parallel and to allow for fast recovery in cases of failure of an ad-hoc node or link.

In the resilient routing part the challenge has been to develop methods that are robust with respect to changing topology and failures, and this is accomplished by constructing backup topologies for fast recovery of connectivity when a link or node degrades, goes down or moves out of reach.

Routing in ad-hoc networks has challenges related to the traffic overhead it creates and the occurrences of broadcast storms. By reducing the routing overhead we can reduce the network load and leave more resources to the data traffic, which may improve the end-to-end performance. In HIDENETS this is done by analysing and reducing the topology used for distribution of routing messages. This can be done in a reliable way.

When connecting to Infrastructure network multihoming can be used to increase the redundancy and as such make this part more robust with respect to access point failures. Due to the increased overhead and limited capacity in this part of the network this has to be done with care. The HIDENETS solution is to combine multihoming with Differentiated Resilience, giving priority to those applications requiring it.

The fact that we can find redundancy between different protocol layers in the wireless domain implies a possible gain by doing cross-layer optimisation. We have suggested improvements in the protocol design to increase the overall efficiency.

For each of these items we have findings that may improve the overall dependability of ad-hoc networks and their interconnection to Infrastructure networks. Of course there are other problems that fell out of the scope of HIDENETS, but overall we have addressed essential problems within this domain

3.5.4 Relevant publications

[39] I-E. Svinnset et al., “Report on resilient topologies and routing – final version”, EU FP6 IST project HIDENETS, deliverable D3.1.2. June 2008.

[75] J. Nielsen et al., “Cross-Layer Resilience Optimization in the Ad-Hoc Domain”, EU FP6 IST project HIDENETS, deliverable D3.2. June 2008.

[12] A. Bondavalli et al., “Mechanisms to provide strict dependability and real time requirements”, EU FP6 IST project HIDENETS, deliverable D3.3. June 2008.

[94] T. Cicic, A. F. Hansen, and O. K. Apeland, “Redundant trees for fast IP recovery”, IEEE Broadnets 2007, North Carolina, US, 2007

[95] A. F. Hansen, O. Lysne, T. Cicic, and S. Gjessing, “Fast Proactive Recovery from Concurrent Failures”. In: ICC 2007, June 2007

[96] A. F. Hansen, G. Egeland and P. Engelstad, “Could Proactive Link-State Routed Wireless Networks Benefit from Local Fast Reroute?” CNSR 2008, Halifax, Canada

[97] Y. Liu, H.-P. Schwefel, ”Algorithms for Efficient Broadcasting in Wireless Multi-hop Networks”. In: Proc. of IEEE Globecom 2006

[98] J.Wu and H. Li, ”On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks”. In: Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Aug. 1999

[99] Y. Liu, H-P Schwefel, “Localised Algorithms for Virtual Backbone Formation in Wireless Multi-hop Networks with unidirectional links”. In: Proceedings of IST mobile summit, July 2007.

[100] Y. Liu, “Virtual Backbone and Mobility-based optimizations for wireless multi-hop networks”, PhD thesis, Aalborg University, September 2007

[101] J. Grønbæk, J. Nielsen, “Cross-Layer Optimization of Message Broadcast”. In MANETs, Master thesis, Aalborg University, Jul. 2007.

3.6 Fault analysis at the communication level

In terms of the HIDENETS project and the work related to the communication layer, resilience is about managing failures and implementing mechanisms that make the system resilient and robust despite possible occurrences of failures. In this context we distinguish between a failure that is the loss of ability to function as intended and an error that is a detected deviation from the correct service state or agreed specification. An error is caused by a fault or due to outside interference. The error may propagate to a failure. On the other hand a fault is the adjudged or hypothesised cause of an error or a defect that gives rise to an error. So in short a fault is what is built into the system, an error is an incorrect state and a failure is the observable phenomenon.

From a communication point of view it seems most natural to contribute to failure management by analyzing possible faults, errors and failures and their consequences. In this section we first classify the faults that are most relevant for the communication layer and then discuss the main causes of such faults. An analysis of possible faults and errors and their propagation at the communication level can be done following the traditional layered communication model. We have therefore outlined a fault hierarchy that has been used as a framework in the HIDENETS project. This is further described. Finally we describe the delimitations of faults that are addressed and give the motivation for the methods that are more deeply analysed in the remaining sections of this deliverable. This is done with emphasis on the methods ability to enhance the resilience and dependability of the network.

3.6.1 Fault classification and analysis

At the communication layer we may divide the faults in two main categories.

- Timing faults: The message is reaching its destination too late (for a given deadline/QoS requirement)
    - Omission faults are a special case, namely the message is dropped
- Value faults: A message reaches the destination, but with wrong content. This case includes
    - Packet reordering, if it leads to a changed message content
    - Packet insertion
    - Message modifications

This applies to both unicast, multicast and broadcast message transmission, but for multicast and broadcast a finer distinction may need to be applied as multiple destinations are involved.

Main causes for timing faults include

- Retransmission mechanisms that handle omission faults on lower layers at the price of additional delay
- Congestion that leads to additional delays due to buffering and MAC delays. Congestion occurs if the available (multi-hop) transmission resources are not sufficient to carry the desired traffic load (including signalling traffic). This can have
    - Accidental causes: e.g., high traffic volumes created by simultaneously starting applications, high node density due to the mobility model, poor propagation environment reducing the available transmission resources (including accidental interference)
    - Malicious causes: reducing the available transmission resources via interference, increasing the traffic volumes via flooding/DoS attacks
- Wireless link conditions (e.g., increased path-loss) that trigger physical (PHY) layer adaptations such as adaptive modulation and coding and hence reduce PHY data rates (without leading to congestion in the sense of resource bottlenecks)
- Processing delays

Omission faults can result from

- The protocol stack in the source node does not operate properly (due to HW or SW faults) and 'swallows' the message (note that a crash fault of the full source node would disable the application/middleware layer to generate any messages, hence it is out of scope of HIDENETS)
- Intermediate nodes fail to relay the message, including the causes
    - The node crashes due to HW or SW problems
    - The node shows malicious/egoistic behaviour and does not forward the message (parts)
    - The intermediate node becomes disconnected from the ad-hoc network due to mobility or changing propagation environment
    - Poor link-properties. The intermediate node fails to receive the message due to multiple unsuccessful transmissions and limits in the retransmission functions
- The protocol stack in the end-node 'swallows' the message (due to HW or SW faults)

Finally, value faults can result from the following

- Malicious modification/insertion/reordering: either by an intermediate node or malicious SW code in the source/destination protocol stacks (man-in-the-middle attack). In case of insertion, also external nodes (not in the multi-hop transmission chain) can create a wrong value using masquerading
- Accidental modifications of the Message
    - Bit errors after the PHY decoding which are not corrected or detected by Layer 2 mechanisms
    - Software errors (e.g., buffer overflow overwriting data content) in the source, destination, or intermediate nodes

3.6.2 A fault hierarchy

The analysis of possible faults and errors and their propagation at the communication level can be done following the traditional layered communication model. Error contention means and resilience mechanisms can be implemented at each layer. Additionally, cross-layer optimizations can be used also to enhance resilience.

With a layered model, each layer implements mechanisms that handle faults and errors that occur within the layer. An error that is treated and contained within the layer of that escaped lower layers protections is not observable from outside, so that the layer does not fail from a higher layer view. Errors that are not fixed inside the layer could propagate as failures till the layer boundaries. A fault affecting a lower level service when observed at the interface offered to the layer above will be referred to as a failure that the layer above has to treat. In many cases, the failure in a lower layer eventually reaches a higher layer in the networking stack where it can be fixed, and hidden for the remaining upper layers.

First, we note that different faults may occur in different layers. It is therefore natural to introduce the term "fault hierarchy" as a framework for fault analyses. The fault hierarchy illustrates in which layer(s) different types of faults may occur. It also illustrates the interaction between faults at different layers. For example, buffer overflow at one layer might translate into a packet loss at the layer above; a bit error at one layer might translate into a lost packet at the layer above; or network contention/congestion at one layer might translate into increased packet delay at a higher layer.

Finally, we note that there is a high degree of resilience and a large number of resilience mechanisms already implemented in today's layered networking model. Most routing protocols of IP (including the exterior routing protocol of IP, BGP, which binds the Internet together) were designed with robustness in mind. If one router is removed, the routing protocols adapt and — if possible — find alternative routes around the failed router. For applications needing fast recovery these mechanisms may nevertheless not be satisfactory.

Figure 6 gives an outline of a fault hierarchy that is used as a framework in the HIDENETS project. The faults are described by the *yellow* (i.e., lightly coloured or blank) boxes, while the built-in resilience mechanisms are illustrated with *orange* (darker) boxes. "OK" refers to the case where the resilience mechanism is able to fix the error, while "NOK", i.e., Not OK, refers to the opposite. The boxes are placed relative to the different layers, i.e., the physical layer, link layer, network layer, transport layer and session/application layer. The main focus here is on faults caused by failures in the layer below, and these are placed on the boundaries between two layers. The boxes are further explained in [39]. Several typical contextual factors that are relevant for the analysis are also identified (see the ellipses shown at the bottom of the diagram).

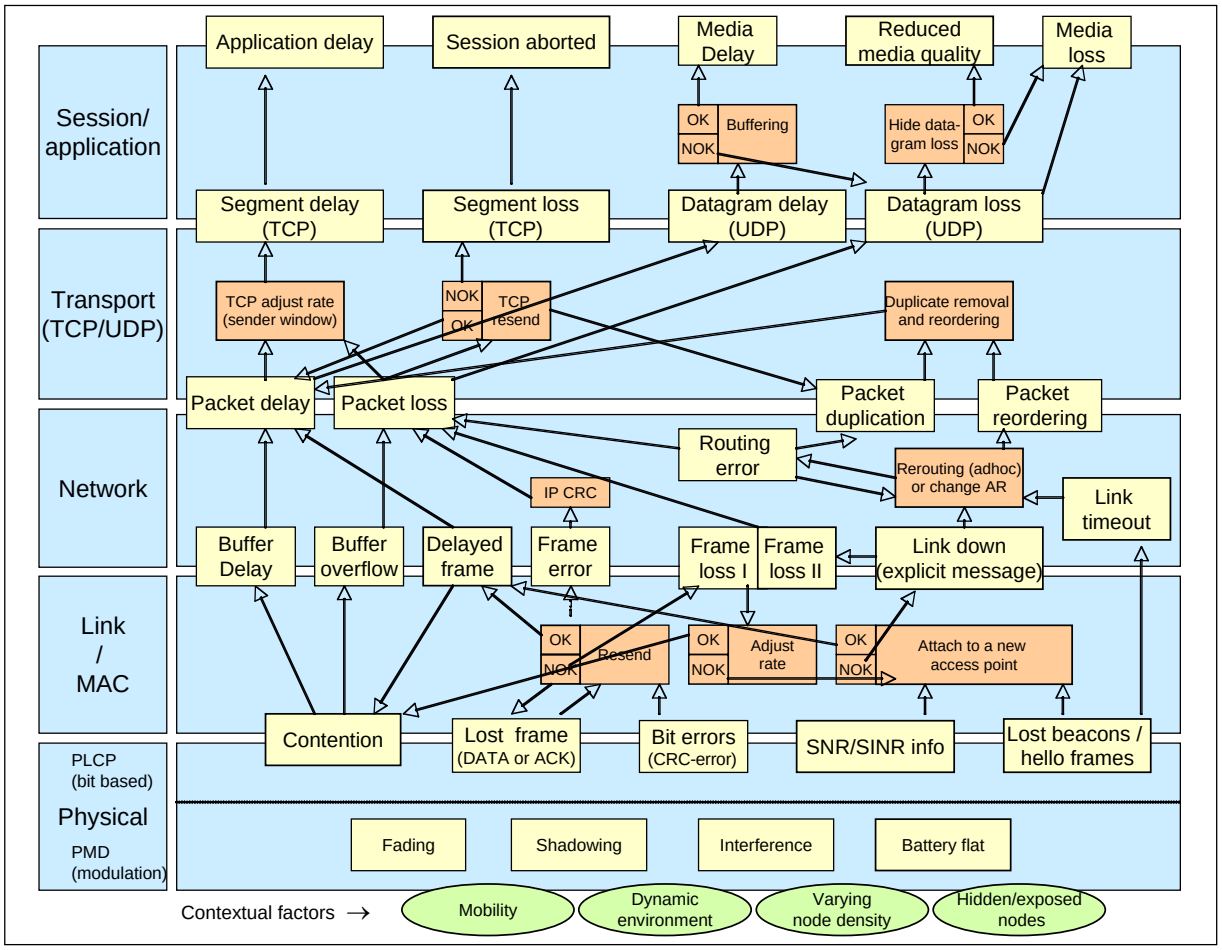


Figure 6: The Fault Hierarchy for fault analysis at communication level

As illustrated in the picture a contention, or more generally congestion, will cause buffering at e.g., layer 2 or layer 3. This may then again cause a timing fault if the delay gets too high or it may cause a buffer overflow and thereby an omission fault. Another example could be that lost beacons or hello frames may induce a link timeout and thereby a recovery action. This may again cause packet reordering or possible packet duplication and thereby a value fault.

Our model might conceal the fact that faults can have different degrees of severity. For example, the "packet delay" fault box in Figure 6 says nothing about the extent of the delay. The packet delay might be so low that it does not affect the functionality of the higher layers, and the effects of this delay might for example be removed by a streaming buffer at the application layer (Figure 6). On the other hand, the streaming buffer has a finite size and can only compensate for a certain packet delay, and the packet might be delayed to such an extent that it is dropped by the end system.

The same argument goes with packet loss. In some cases only a single packet is lost, which might for example not influence the quality of a voice conversation considerably. In other cases, there might be a large number of consecutive packets that are lost, in which case the consequences are much more serious.

Figure 6 illustrates the fact that some of the faults are forwarded to the layer above without being fixed there, e.g., a lost frame at the link layer might result in a packet loss error at the networking layer. Other examples are also found in the figure. For example, contention (/congestion) at the link layer, might translate into buffer delay or buffer overflow at the lower part of the networking layer (e.g., in the device driver or in the socket). These are perceived by the transport layer as packet delay and packet loss, respectively, and as datagram delay and datagram loss by the application or session above if UDP is being used. Finally, a lost datagram might for example result in a loss of VoIP signal (referred to as "media loss" in Figure 6).

Furthermore, the figure also shows that if a TCP segment is delayed, it might delay the execution of the application above, while a TCP segment loss might eventually result in a TCP reset in which the TCP session is aborted. Finally, loss of a multicast hello packet (in ad hoc mode) or of a beacon frame (in infrastructure mode), might result in link timeout at the networking layer, in which the networking layer assumes that the link is down.

The orange box labelled "Rerouting or change of AR" (AR = Access Router) addresses the cases where rerouting or change to a new access point is necessary due to a link failure. Also in the case that the node is in ad hoc mode, a "Link down" notification might lead to changes in the routing protocol. This change (or "reroute") is also carried out in the orange "Rerouting or change AR" box. Alternatively, the reroute might also be triggered by the loss of a number of hello frames. In order to capture this type of fault, a link timeout is implemented in the routing protocol at the network layer, as illustrated in Figure 6.

If the node is in ad hoc mode and is a part of an ad hoc network, the functionality carried out in the "Rerouting or change of AR" box might be considerably more complex than the functionality carried out if the node is in infrastructure mode. Thus, for a node that is in ad hoc mode the "Rerouting or change of AR" box could probably easily be divided into a large number of additional yellow fault-boxes and additional orange resilience-boxes that try to fix these additional errors within the context of the routing protocol. Thus, a fault hierarchy could probably be constructed for the routing protocol separately. However, this is out of the scope of this document.

3.6.3 Scope and delimitations

First, we note that it is very difficult to provide a complete picture of the fault hierarchy with every possible fault described in detail. It is also difficult to describe every interaction between faults and between the mechanisms in a layer used to fix faults. As seen in Figure 6, the description is already quite complex, and describing the fault hierarchy in more detail than what is done in the figure might not serve its purpose well.

The fact that a model will not be able to cover all possible faults is easy to realise by mentioning some extreme cases of faults. For example, the hardware might be smashed in a car accident or hit by lightening. Likewise, there might be hackers that purposely try to launch DoS attacks, e.g., by transmitting bogus 802.11 management frames that dissociate all nodes attached to an access point / base station. Finally, nodes might be misconfigured in all sorts of ways, leading to nearly all sorts of possible errors. The main point here is that the potential types of faults are quite many and too many to be included in the framework presented in this document. It is though our opinion that the most relevant faults have been dealt with in the discussion and that the remaining faults do not undermine the resilience and dependability achievable by HIDENETS because of their very low frequency or non severe consequences.

As a natural consequence we have made some delimitations of the type of faults to be addressed within the scope of HIDENETS:

- All accidental value faults are mapped to omission faults via the use of CRC methods
- Processing delays as causes for timing faults are not considered
- SW errors in the protocol stacks (source, destination, intermediate) are not considered

On the other hand, the scope of HIDENETS and the focus of our work are:

- Fault-prevention by avoiding congestion via

  - Increasing the available resources (Multi-channel multi-radio management (section 3.5.1.1), always best connected ABC (section 3.5.1.5)
  - Reducing the traffic volumes (efficient broadcast (section 3.5.1.4), routing (section 3.5.1.3))
- Fault-tolerance to omission faults
  - In case of persistent faults of links or intermediate nodes → resilient routing (section 3.5.1.2) and ABC (section 3.5.1.5)
  - ACK schemes for reliable broadcast (section 3.5.1.4)
  - Optimistic approaches to increase broadcast reception probability via cross-layer optimization (see D3.2[75])

The rationale for that scope definition is

- One of the main HIDENETS challenges is the dynamicity of the ad-hoc domain. This dynamicity mainly results in congestion and in omission faults.
- The dynamicity also leads to changing topologies that may be experienced as node failures.
- A promising method to handle the changing topologies and keep-alive communication is by increasing the network redundancy and the awareness of the mobile nodes about alternate communication paths
- Malicious cases could have been dealt with within HIDENETS, but in order to restrict the scope these aspects were not prioritised by the partners.
- SW errors on the communication layer leading to value faults are seen as part of MW and as such not treated within the analysis of communication faults.

3.7 Implication of communication fault hierarchy on the MW Oracles

In the previous section, we carried out an analysis of faults and errors propagation and containment at the communication level. However, this analysis focused on the "normal" or payload communication services, which are used by middleware services and applications located in the payload part of the system, be it at the operating system level or in user space (see Figure 7).

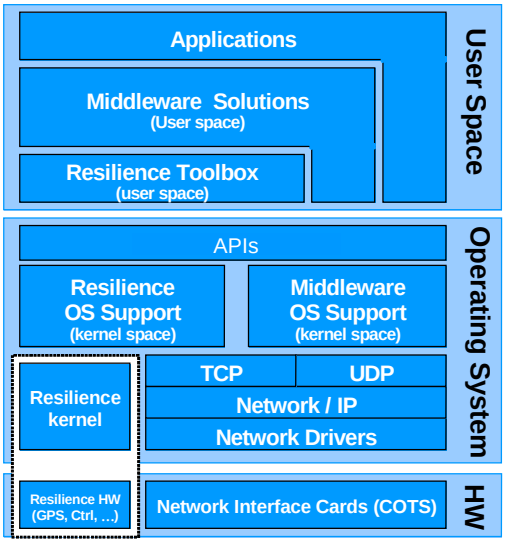


Figure 7: Simplified node architecture – hybrid system perspective

We now address the fault model and error propagation and containment from the point of view of the oracles included in the resilience kernel, according to the hybrid system perspective.

It should be noted that this discussion is only relevant when considering the existence of MW oracles with a distributed nature (see Section 4.3.1 of [42]). In this case, the adoption of a hybrid system perspective must

also be extended to the communication subsystem. A distributed MW oracle, like, for instance, the timely timing failure detection MW oracle, typically requires the timely execution of a distributed protocol, which must be done through a communication system with better (synchronicity) properties than the payload communication system. For other MW oracles, which provide strictly local services or which do not need the execution of distributed protocols with timeliness requirements, there is no need for a separate communication system and so the following discussion is not relevant.

On the other hand, the need to ensure communication channels with better properties when distributed oracles are to be implemented turns out to be an easier task than it would be if those better properties had to be ensured for general purpose communication channels. This is because, by construction, these better communication channels are only used for specific purposes, for transmitting typically small and a priori known pieces of data, which are thus, by comparison, easier to schedule in a predictable way.

Quite clearly, it is not always possible to provide strict guarantees for the timeliness or trustworthiness of the communication channels. It is a matter of coverage of assumptions. We argue that by using the proposed hybrid architecture it may be easier to secure some better properties for the channels serving the oracles with higher coverage than it would be possible to achieve for general purpose channels. We also note that this is the case, independently of the techniques or mechanisms that are used to achieve improvements of the communication channel properties. Some specific techniques might also be considered that exploit the fact that communication can be better controlled (by knowing how the oracles exchange information) to achieve the improved properties.

The diagram of the fault hierarchy for the communication system serving the MW oracles (see Figure 8) is based on the fault diagram represented in Figure 6. In fact, by design this communication system should be simpler than a general purpose communication system, as required to ensure increased resilience and synchronicity. Nevertheless, we use the diagram presented in the previous section in order to better illustrate the differences in the fault analysis and what could be done to secure the required fault model for the communication between MW oracles.

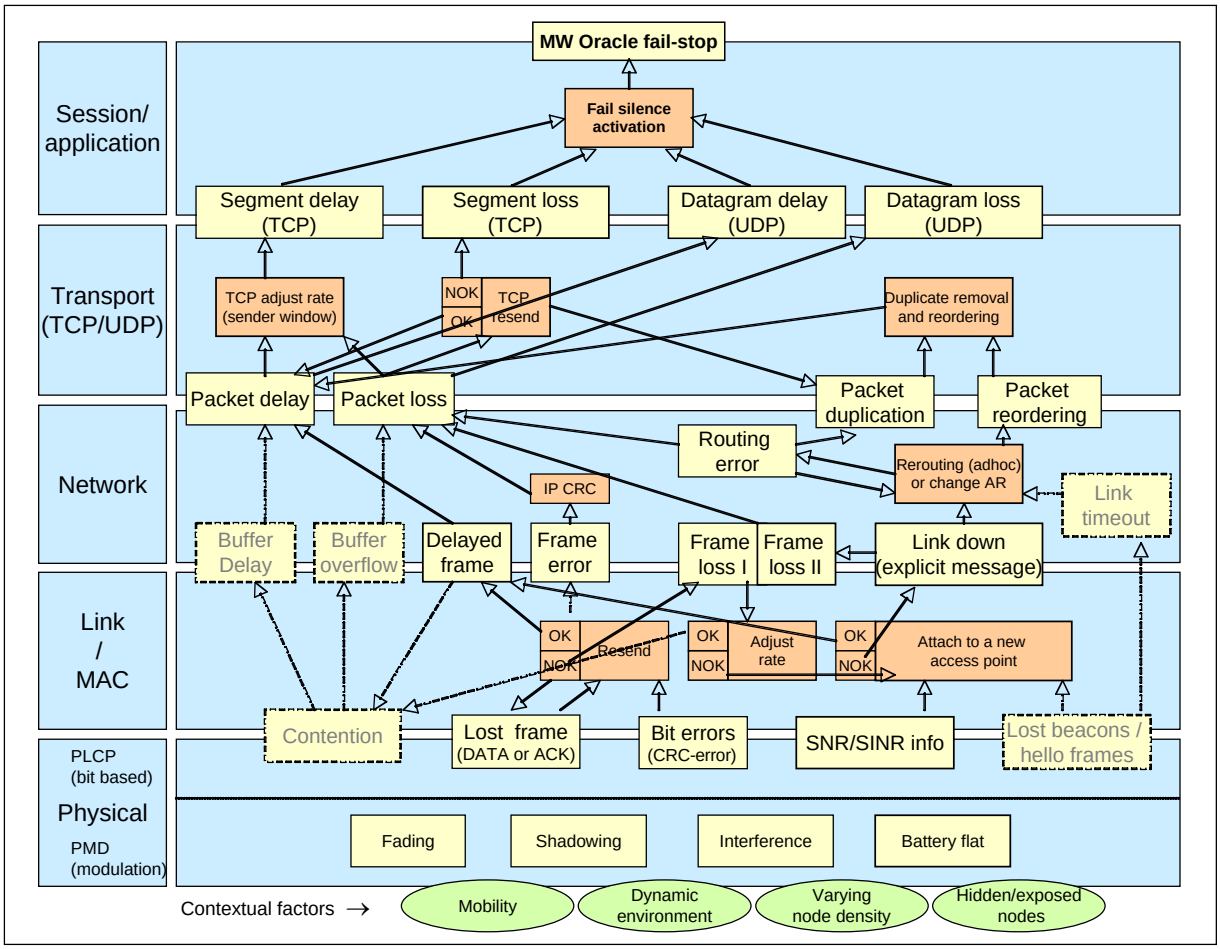


Figure 8: Fault analysis for the communication between MW oracles.

In general, when compared to the fault model assumed in the payload part of the system, the fault model assumed for the construction of MW oracles must be stricter. This stronger model is necessary in order to be able to provide improved services and secure better properties to be presented by the MW oracles. This is true not only in general terms, but in particular concerning fault assumptions for the communication.

For the sake of illustration, the diagram for general-purpose communication has therefore been modified in two different ways, assuming the following assumptions are enforced by the design:

- First, some faults have been considered as less probable, and therefore have been partly removed from the assumed ones, which can be supported assuming that fault prevention techniques can be used.
- Second, the behaviour of the communication subsystem has been transformed into a synchronous system with crash behaviour, by transforming every timing or omission fault into a crash failure, which we assume to be possible by using adequate fail-stop mechanisms.

To be more specific, for this example, the faults with virtually zero occurrences (with realistic assumption coverage) are:

- contention, and
- lost beacons / hello frames.

The corresponding boxes are identified by means of dashed lines, as well as the related links and directly linked downstream boxes.

For what concerns contention, this could be accomplished by the use of *private channels*. Due to this, the following situations are not expected to occur: buffer delay and buffer overflow. A simple technique to avoid the buffer overflow is simply to define buffers with a size that makes it impossible to fill them completely. This of course requires the knowledge or the definition of upper bounds for some transmission parameters,

which might be done more easily in the scope of the MW oracles (for instance with the implementation of admission control techniques).

Concerning lost beacons/hello frames, other techniques will have to be used. One possibility is increasing the redundancy of these beacons, with a higher number of transmissions, as high as possible to not increase the probability of collisions. Another solution might come from the use of alternative techniques and standards, such as the 802.11p standard, which defines the possibility of using a more restricted control channel that could be assigned to the MW oracles and, in that sense, would probably allow reducing the possibility of collisions.

With respect to the transformation of the timing and omission faults into crash faults, the affected services in the diagram are those on the top layer and include:

- Segment/Datagram delay, and
- Segment/Datagram loss.

These faults provoke a deadline miss so the MW oracle must ensure that this is not perceived at the application level. Otherwise, the assumed properties for the oracle would not be satisfied. Therefore the MW oracle transforms these failures into crash failures, and the node where the MW oracle resides goes into a fail-silence state. This transformation is highlighted in Figure 9, in which it is possible to observe the referred transformation of timing and omission faults into crash faults. Note that we distinguish between detected and non-detected timing/omission faults, because it might not be possible to implement a detector with the perfect properties, one that would perfectly detect and distinguish all the kinds of faults. Any implementation will necessarily have to be based on some assumptions, which might turn out to be not fulfilled.

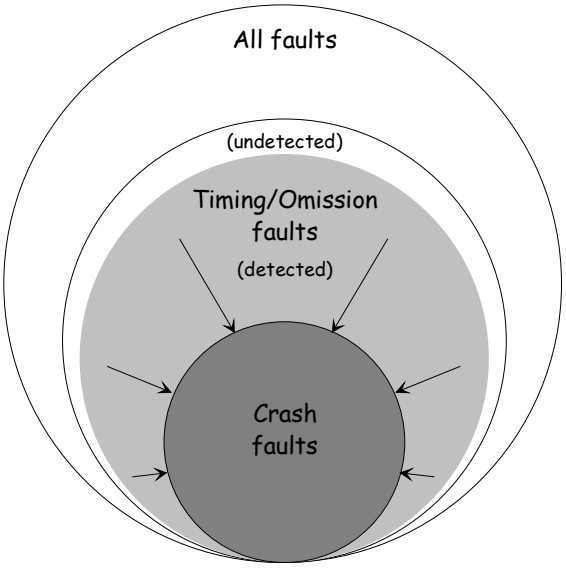


Figure 9: Fault classes and their treatment for the communication in distributed MW oracles.

4. Quantitative evaluation

Complex software and hardware systems are widely used in different applications and they have become pervasive in many fields of human activity. Each system demands some specific properties, such as a certain level of availability, reliability, performance or quality of service (QoS), the quantitative evaluation of which has become a key issue in several application fields.

The quantitative evaluation of such dependability-related properties is performed following three basic approaches: analytical stochastic modelling, simulation**,** and experimental measurements. Each approach shows different characteristics, which determine the suitability of the method for the analysis of a specific system aspect. The most appropriate method for quantitative assessment depends upon the complexity of the system, its development stage, the specific aspects to be studied, the attributes to be evaluated, the accuracy required, and the resources available for the study (see D4.1.2 [24]) for a more detailed discussion on this topic).

The HIDENETS activities related to quantitative evaluation are aimed at developing methodologies and techniques that can be used to quantify and analyse dependability and performance characteristics of HIDENETS architectural solutions and protocols sketched in Chapter 3, and applications and use-case scenarios reported in Chapter 2.

In the following Section 4.1 we will discuss the challenges (from a quantitative evaluation perspective) related to the specific HIDENETS characteristics. In Section 4.2 we present an overview of the HIDENETS holistic evaluation approach, while Section 4.3 summarises some of the main evaluation activities performed within HIDENETS, together with their main outcomes.

4.1 Main challenges

The assessment of the dependability-related attributes of the HIDENETS applications and use-cases is a very challenging topic due to their characteristics, like:

- Heterogeneity of the network domains, including wireless ad-hoc networks, wireless infrastructure-based networks, and also wired networks.
- Use of (inherently unreliable) off-the-shelf components (OTS), which exhibit little information on their architecture and their actual failure behaviour.
- Large number of interacting components involved in a single use-case, with a large number of failure modes and recovery and maintenance scenarios to be taken into account.
- System dynamicity and evolvability in terms of topology, connectivity, and channel conditions.
- Strong interdependencies between different system parts, which can result from functional or structural interactions between system components related to the system architecture, or from fault tolerance and maintenance strategies, leading to stochastic dependencies that need to be captured by the models and evaluation techniques used to assess dependability.
- Variety of threats, including both accidental and malicious faults.

Actually such characteristics are not strictly related to HIDENETS systems only, but they are common to many other contemporary application fields having a high interconnectivity between different infrastructures with seamless interactions.

The system complexity previously described leads to several problems that need to be addressed in performing a quantitative evaluation.

Concerning **analytical modelling**, the main challenges are the following:

- Modelling complexity. The overall description of critical and distributed complex systems can be a very tedious and difficult task. The modelling complexity is generally related to the level of detail considered for describing the main phenomena and behaviours captured by the models and to the quantitative measures that need to be evaluated. The complexity could also be exacerbated when multiple interactions and interdependencies exist among the components.
- State-space explosion. The state-space methods construct a structure (the state-space) that consists of all states that a system can reach, and all transitions that the system can make between those states. The problem is that the size of a state space of a system tends to grow exponentially with the number of its processes and variables.
- Stiffness. A problem is stiff if the numerical solution has its step size limited by the stability of the numerical technique. Therefore, a symptom of the potential presence of stiffness is the existence of components that change much faster than others.
- Parameters' estimation. Another problem is the determination of the values to assign to the parameters required by the models. Actually these values can be difficult to obtain (usually by way of measurement or experimental tests), and they might not be available during the preliminary design phases of the system. Since even slight variations of critical parameter values may result in relevant changes of system dependability attributes, a thorough calibration of such parameters is necessary to increase the level of confidence that can be put on the dependability evaluation itself.

Concerning **simulation**, the main challenges are the following:

- Development of simulation models with adequate system abstractions. Although simulation models can be rich in features, runtime constraints typically require to determine the adequate level of detail of the simulation model and to abstract the remaining parts of the system.
- Scalability of the simulation scenarios. Scalability is related to adequate system abstractions, in the sense that the complexity of particular components puts restrictions on attainable simulation scenarios and also puts requirements on the scalability of the simulation tool and implementation. In particular, the simulation of representative mobility scenarios to evaluate some connectivity parameters needed for the resilience assessment of HIDENETS systems and applications is a challenging task.
- Output Analysis and Rare event problems. One of the most severe problems that can affect simulation is the so-called rare events problem. The rare events problem occurs when there are events that occur at very different time scales, so the computational time needed to obtain a statistically significant solution becomes unacceptable. For any simulation analysis performed in HIDENETS, simulation results have to be inspected carefully and standard methods to evaluate statistical significance of the simulation output have to be applied. It will depend on the specific assumptions considered (especially with respect to fault models), whether actual rare event problems occur in the HIDENETS simulation systems.

Concerning **experimental evaluation**, the main challenges are the following:

- Cost. Experimental evaluation is usually a quite expensive activity since it requires having access to real systems or prototypes that need to be evaluated.
- Intrusiveness. The experimental evaluation activity can be seen as performed by monitoring probes that are "plugged" into the target system (a real system or a prototype). It is evident that such probes should be non invasive, in the sense that they could not affect the behaviour of the target system (besides the interactions they have been designed to perform). Rather, the target system should evolve as in absence of the probing devices.
- Semantic gap. Frequently there is a semantic gap between the data collected during experimental evaluation and the relevant measures which are of interest for the user or evaluator. Due to the physical constraints of the available interfaces for non-intrusive observation, monitoring systems often collect low-level data (e.g., interactions at the physical communication layer, memory accesses of a processor), that shall be 'translated' to higher level quality of service characteristics (e.g., expected service delay) or dependability measures.

- Reproducibility. Conducting field measurement or controlled experiments with real wireless interfaces may make it difficult to reproduce results of experiments. Even if the geographic mobility of the nodes is very carefully described and followed during the course of the experiment, the results of the experiments may not be the same since the propagation conditions are strongly influenced by changing environment conditions, e.g., interference, moving objects etc. It is also important to note that, for evaluation purposes, reproducibility requirements relate to the ability to obtain statistically equivalent results as explicitly recognised for example in the experiments carried out for dependability benchmarking.

4.2 The holistic approach

All the evaluation activities have been carried out in the context of the HIDENETS holistic approach, which allows defining a "common strategy" using different evaluation techniques, applied to the different components and subsystems, thus exploiting their potential interactions. In the evaluation of systems like HIDENETS a single technique is not capable to tackle the whole problem. The idea underlying the holistic approach follows a "divide and conquer" philosophy: the original problem is decomposed in more simpler sub-problems that can be solved using appropriate solution techniques; then the solution of the original problem is obtained from the partial solutions of the sub-problems, exploiting their interactions.

The vision of the holistic procedure to pursue a common objective is the following:

1. Identification of a common objective (i.e. the end-to-end evaluation of some system dependability attribute), that can not be adequately solved using a single evaluation technique.
2. Decomposition of the entire problem in a set of simpler sub-problems.
3. Solution of the single sub-problems using an appropriate evaluation technique (partial solutions).
4. Reconstruction of the solution of the original common problem, exploiting the interactions among different evaluation techniques (complete solution).

Some of the possible interactions among different evaluation techniques are the following:

- Cross validation. A partial solution validates some assumptions introduced to solve another sub-problem, or validates another partial solution (e.g. a simulation model can be used to verify that the duration of an event in an analytic model is exponentially distributed).
- Solution feedback. A partial solution (or a part of it) obtained by applying a solution technique to a sub-problem is used as input to solve another sub-problem possibly using a different technique (e.g. a critical parameter in an analytic model is obtained using experimental evaluation).
- Problem refinement. A partial solution gives some additional knowledge that leads to a problem refinement (e.g. the architecture of a component changes since it is recognised to be a system bottleneck).

4.3 Activities and main results

In this section we provide an overview of the main activities concerning the evaluation of the architectural solutions proposed in Chapter 3 and of the HIDENETS applications and scenarios identified in Chapter 2. A detailed description of the used evaluation techniques, methods and tools can be found in D4.1.2 [24], and the final results of their applications are reported in D4.2.2 [36].

The central part of Figure 10 corresponds to Figure 3 and it represents a simplified view of the final HIDENETS node architecture.

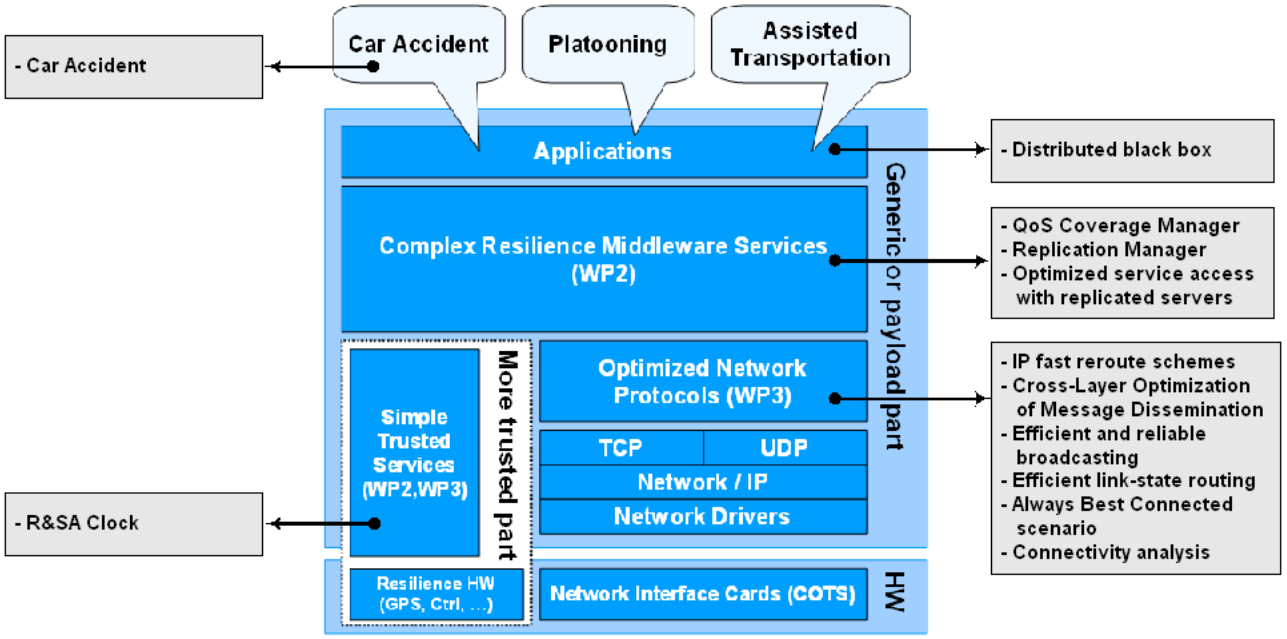


Figure 10: Simplified HIDENETS node architecture, and performed evaluations

On the left and right part of the picture are listed the different types of evaluation activities that are addressed in this chapter, mapping them to the main building-blocks of the architecture. As we can note the evaluations concern all the layers of the node architecture, including optimised network protocols, trusted and complex middleware services, as well as applications and use-cases.

The activities addressing the pointwise evaluations of specific HIDENETS aspects, like middleware components and network-level characteristics/properties will be detailed in Section 4.3.1, while those dealing with the analysis of HIDENETS applications and use-cases will be addressed in Section 4.3.2. Finally, Section 4.3.3 will present the definition of an evaluation workflow encompassing several tools and model transformation steps, showing the feasibility of the holistic approach discussed in Section 4.2.

4.3.1 Pointwise evaluations of specific HIDENETS aspects

A first set of experimental techniques addresses specific middleware components among those identified in D2.1.2 [1]: “R&SA clock” (in Section 4.3.1.1), “QoS Coverage Manager” (in Section 4.3.1.2), and “Replication Manager” (in Section 4.3.1.1). Another set of techniques deals with different networks-related aspects, like the optimised service access with replicated servers (in Section 4.3.1.4), the connectivity analysis (in Section 4.3.1.5), the efficient and reliable broadcasting and link-state routing (in Section 4.3.1.6), the IP fast reroute schemes (in Section 4.3.1.7), the cross-layer optimization of message dissemination (in Section 4.3.1.8), and the “Always Best Connected scenario” (in Section 4.3.1.9).

4.3.1.1 R&SA Clock

The goal of the analysis of R&SAClock (performed through experimental evaluation) is to verify if the requirements of the R&SAClock are fulfilled by a specific implementation of the service (a prototype of the service for NTP clock synchronization mechanisms in Linux, and what are the level of QoS attainable by R&SAClock in different scenarios. In order to obtain this, we have used an approach to the experimental evaluation composed by the following sequence of steps:

1. Description of the component and identification of its requirements;
2. Definition of metrics and purposes of the evaluation;

3. Definition of Database tables (DataWarehouse approach) able to contain results of experimental evaluations;
4. Definition of evaluation scenario(s);
5. Definition of experimental setup;
6. Instrumentation of the system;
7. Experiment execution, results collection, and analysis.

The experimental evaluation activities of R&SAClock are described in details D4.2.2 [36]. Here we report only a summary and lesson learned thanks to this evaluation. In this experimental evaluation the prototype of R&SAClock met all its requirements. We however noticed that current implementation of R&SAClock has improvable performance level, mainly in terms of uncertainty interval size that can be improved. Moreover, the procedure described in D4.2.2 of the R&SAClock prototype can be used for future different (e.g., improved) versions of the service implementation. The only step to do a new experimental evaluation campaign of a different version of R&SAClock is the instrumentation of the source code in order to get the log files; the remaining parts of the evaluation process are in common with this experimental campaign.

4.3.1.2 QoS Coverage Manager

An evaluation activity has been carried out to analyze the effectiveness of the current implementation of the "QoS Coverage Manager" middleware component, sketched in Section 3.4.1.2 and detailed in D2.1.2 [1]. Such a middleware component uses probabilistic methods for the recognition of the "state" of the environment, or at least of some of its characteristics, and then it dynamically adapts the time-bounds of the (time-elastic) applications to maintain the required coverage.

A number of simulation experiments have been performed, both based on synthetic data flows generated from well-known probabilistic distributions and on real Round-Trip Time (RTT) traces collected in different environments. Based on these results, some of them presented in [16], we are able to conclude that:

- The assumptions at the base of the "QoS Coverage Manager" definition (e.g., the alternation between stable and transient periods during the lifetime of a system) are met in real systems.
- It is possible to define effective mechanisms to detect environment changes (in particular, detecting stable and transient phases) and, for the stable periods, correctly characterise the observed probabilistic distribution.
- It is possible to achieve dependable adaptation and, at the same time, obtain improved (tighter) time bounds than those previously obtained with a more conservative approach (as done in [8]). This improvement is relevant in the implementation of many other practical systems, for instance in the configuration of timeouts in failure detectors, where the objective is to use the smallest possible time bound (to improve the detection time) without compromising the failure detector accuracy (mistakes due to timing faults).

Concluding, the proposed evaluation activity has provided significant contribution for the design refinement of the analyzed middleware component, as well as for the fine tuning of some of its critical parameters.

4.3.1.3 Replication Manager

The Replication Manager supports server replication in dynamic clusters within the ad-hoc domain. Basic functionalities thereby include cluster membership assignment, optimistic state replication after transaction finalization (implemented via a shared memory approach), and client access to the dynamic cluster. The pointwise analysis focused on quantitative insights into the tradeoff between service availability, replica consistency, and overhead for state synchronization. Optimization strategies involving also fault-detection and server selection policies as described in Section 4.3.1.4 can in principle also be transferred to the

dynamic clusters in the ad-hoc domain; however, the scope of the analysis during HIDENETS was rather on simple (heuristic) membership selection strategies in combination with different mobility scenarios.

Main analysis steps and results were:

- A simulation-based analysis of the impact of different subset division heuristics was performed. These heuristics aimed at dividing the available set of nodes into disjunct replica groups. The heuristics aimed at increasing replica consistency and server set stability. Abstract cost metrics based on geographic distance and speed were investigated as replacement for overhead intensive measurements of pair wise end-to-end (multi-hop) communication delays. These heuristic cost metrics lead to substantial enhancement mainly of pool stability (less cluster reconfigurations needed), while consistency enhancements were less substantial.
- A stochastic model for inconsistency analysis was developed, which allowed calculating quantitative results for the so-called mismatch probability for replica updates driven by state change events in comparison to periodic replica updates. Using matrix-algebraic methods, the model allows comparing the mismatch probability for different network delay processes and server state change processes. The considered scenarios lead to higher mismatch probability when the variance of the delays or inter-change event times was becoming lower when keeping the same first moments. Such qualitative behaviour was then extrapolated into more complicated replication scenarios as analyzed under exponential assumptions in the next item.
- A Petri-Net model with Markov representation was developed to jointly describe the number of servers in a replica set and the consistency set size among these servers in settings of exponential network delays, exponential application state changes, and exponential assumptions on the process of newly entering replica candidates. The latter assumption was analyzed further for different freeway mobility models in Section 4.3.1.5. The Markov model allows obtaining service availability and reliability in dependence of properties of the mobility model. The numerical results showed that for a relevant range of mobility parameterizations and for communication delays corresponding to 802.11 type ad-hoc connectivity under low to medium load, there is a substantial benefit from dynamic replication in the ad-hoc domain.

The quantitative models were developed for relatively simple replica selection strategies and for optimistic replication strategies. Also the mobility models reflect mainly single-lane highway scenarios. An extension of the models and analysis along all three of these dimensions would be useful for work beyond HIDENETS.

4.3.1.4 Optimised service access with replicated servers

Server replication is often used to improve dependability, because it provides resilience against node failures and/or network failures that both cause server unavailability. By retransmitting the request to the same server, the client might be able to finally reach this server in case of temporary network failure. When the original server does not reply after a few attempts, the client can send the request to a backup replicated server that offers the same service as the original server.

Retransmissions and failovers (i.e., retransmitting the request to a backup server) are costly time wise. Therefore, the average service response time for given traffic and failure models is closely linked with failure detection. Clients can minimise the service response time of a given application by ideally contacting an available server every time they send their requests. Therefore, the more accurately the client sees the status of the replicated servers, the more likely the request will be replied back to the client quickly because there would be fewer re-transmissions on average. Accurate failure detectors most often rely on exchanging frequent packets to check on the status of all servers; hence, server replication usually comes with worsened performance in terms of service response time and overhead.

It is important to analyze the tradeoff between dependability (successful application-level transactions), performance (service response time) and cost (overhead) of a system when designing a fault-resilient solution. The scenario investigated deals with replicated servers which deploy the same service in the wired domain and clients emit their requests from both the wired and the ad-hoc domain. Analytically, this translates into a simple model where link errors and delays are more frequent/longer in the ad-hoc domain

than in the wired domain. The replicated server set is managed by the Reliable Server Pool framework (RSerPool [19, 20, 21]).

The evaluation work applies to client-server applications in general, but we use the specific example of the Session Initiation Protocol (SIP [38]), which:

- is used for managing IP-based multimedia sessions,
- is transaction-based (request-response),
- specifies basic application-level standard retransmission and failure detection schemes.

The objective is to evaluate how replicated services perform in their standard settings, and investigate to which extent performance can be optimised, while maintaining dependability as high as possible (or even increased) and keeping the overhead induced by higher failure detection traffic as low as possible. Several parameters and schemes of the failure detection framework and recovery mechanisms (both implemented in the client, in the server part of the Replication Manager, and in the client part of the Replication Manager) are looked into, in order to propose optimal setting and enhancements to the standard mechanisms. In particular, the following parameters are studied:

- Number of request retransmissions: this parameter is implemented at the application layer in the considered scenario (i.e., SIP) and can be dynamically adapted during the application run-time by the communication adaptation manager (see deliverable D2.1.2 [1]) in the application client, according to the current network performance and failures.
- Number of replicated servers needed: this can be a static parameter in the Replication Manager of the Name Server, which is equivalent to the server logic of the Replication Manager. The parameter behaves differently in a dynamic ad-hoc network and in a static infrastructure network. In ad-hoc network, the parameter would be dynamic and be adjusted by Replication Manager logic and in infrastructure networks it would be constant. As we here mainly consider infrastructure (i.e. IMS), the number of servers is considered constant.
- Timeout per request: this parameter is implemented at the application layer in the considered scenario (i.e., SIP) and can be dynamically adapted during the application run-time by the communication adaptation manager in the application client, according to the current network performance and failures.
- Heartbeat frequency (ASAP heartbeats sent to the server from the name server): this parameter is implemented at the ASAP layer in the considered scenario and can be dynamically adapted during the application run-time, according to the current network performance and failures.

Protocol extensions are studied, in order to bring additional failure detection support, e.g., by using failure detection schemes between users and servers (instead of name server-server only in the standard case) and by introducing new error notification schemes from the name server to the users. Even though some schemes introduce additional overhead, failure detection will hopefully be sensitively increased and therefore users will have more chances to access an available server. In turn this might lead to i) shorter service access time, ii) higher dependability, and iii) by avoiding retransmissions, this compensates for the additional overhead due to the new failure detection schemes.

4.3.1.5 Connectivity analysis

Applications and communication protocols in dynamic ad-hoc networks are exposed to physical limitations imposed by the connectivity relations that result from geographic mobility. For vehicular scenarios, i.e. applications that utilise car-to-car and or car-to-infrastructure applications, the dynamics of the mobility model and hence of the resulting node (multi-hop) connectivity are particularly pronounced, while at the same time the applications need to fulfil certain dependability or safety requirements. A set of stochastic models was developed in HIDENETS that allow analyzing resulting connectivity metrics for a single stretch of long freeway with multiple lanes. The nodes thereby are assumed to have a circular connectivity range of fixed size. The metrics are computed for static snap-shots of the mobility model:

- The node degree, corresponding to the number of single-hop neighbours of a mobile node;

- The connectivity number, expressing the number of nodes reachable via multi-hop paths of arbitrary hop-count;
- the connectivity distance, expressing the geographic distance that a message can be propagated in the network on multi-hop paths
- the connectivity hops, which corresponds to the number of hops that are necessary to reach all nodes in the connected network.

HIDENETS developed analytic expressions for the distributions and moments of these random variables for general stationary Markov Arrival Processes on a one dimensional space. The numerical results compare bursty vehicular traffic with independent movement scenarios described by a Poisson process. Depending on the parameterization of the burstiness (corresponding e.g. to the distribution of number of cars stuck behind slower trucks), the connectivity properties in some cases can be mainly determined by individual bulks of cars.

The static snap-shots of the mobility model were subsequently extended to dynamic scenarios in which cars move independently. Under certain preconditions, it was shown that the moments of meeting new single-hop communication neighbours is a Poisson process, whose rate can be obtained from the car density and from the average absolute relative speed of the cars (see D4.2.2 [36]). For multi-lane high-ways, the latter depends on the communication range as well, but this dependence could be expressed with an integral representation, which has a simple closed form in case of uniform speed distributions. A comparison with scenarios of more complex mobility models showed that the Poisson assumption and the resulting relation between the rate of meeting new communication neighbours and mobility parameters represents an in practice useful approximation also in a large set of the cases.
The connectivity results were subsequently used to parameterise Markov models for the distributed black-box analysis (Section 4.3.2.1) and for the Replication Manager analysis (Section 4.3.1.3).

4.3.1.6 Efficient and reliable broadcasting and link-state routing

Evaluation by simulation has additionally been carried out to evaluate the performance of efficient and reliable broadcasting – a communication level component developed to guarantee delivery of messages broadcast in a network by using information local to the nodes in the network. The broadcasting component can be used by itself, utilised by other, higher-layer services, or it can be integrated into other components. This is the case in HIDENETS, where an efficient optimised link-state routing protocol has been devised. The E-OLSR is a so-called proactive routing protocol, where nodes periodically broadcast link-state update messages to neighbours. These messages contain the nodes' current view of the 1-hop topology. The effect of a guaranteed delivery of broadcast messages is increased confidence in the state of the links in the topology and thus a more efficient routing protocol.

Through the simulation studies of the reliable broadcasting, the performance has been evaluated of both the broadcasting itself and the routing. The performance of the broadcasting was in terms of packets and acknowledgements sent, called transmission overhead. The performance of the routing was in terms of packet delivery ratio.

Based on the results, we were able to conclude that:

- The efficient and reliable broadcasting performs better than regular reliable broadcasting in terms of messages sent for acknowledgments.
- The efficient and reliable broadcasting provides reliable message delivery in networks up to 50 nodes and still performs better than regular algorithms.
- The algorithms perform well in static networks, yet it is a challenge to evaluate them in dynamic networks, as the evaluation methods are very dependent on the simulation run. Two traces of mobility may yield two very different conditions for the algorithms and therefore not be directly correlated for evaluation.

4.3.1.7 Simulating IP fast reroute

In D3.1.2 [39] we have described and developed a large set of IP fast reroute schemes for wireless ad hoc networks. All kinds of applications are envisioned to be offered in these networks. Some might have stringent requirements on the robust delivery of data, either due to a real-time nature of the application or due to the importance of the content. These applications bring challenges to the routing protocols due to the failure frequency of wireless links and mobility of ad-hoc nodes. Integrating all these fast reroute schemes into a public complex simulator tool like NS-2[5] would be too time-consuming, and hence we have developed a general routing simulator in Java that gets topology inputs from a python-based[6] topology generator (see D4.2.2 [36]).

Since our goal was to evaluate a large set of routing schemes, the proprietary Java-based routing simulator was expected to be the best choice to come to some general conclusions on the schemes. These results have given us some indicators of what could be the most appropriate scheme in certain scenarios. However, the lack of per packet simulations for various traffic demands in our simulator has camouflaged some of the most severe effects from some of the routing schemes. Some schemes are not able to drop packets that cannot be recovered, and our simulator could not capture that a portion of the packets would loop around in the networks consuming resources. However, since we are aware of the fact that some schemes give such looping, it can be addressed in the functional evaluation, but the effect cannot be quantified. Still, our experience is that by combining knowledge about functional properties with our proprietary simulator, it is possible to identify the most viable routing schemes for the different scenarios.

4.3.1.8 Cross-layer optimization of message broadcast

A cross-layer optimization, which chooses the most optimal values for protocol parameters, such as transmission power, modulation scheme and forward error correction code rate of a flooding message broadcast service has been developed. In order to verify the developed models used in the optimization, a NS-2 based simulation framework that incorporates a physical layer simulation model that accurately models the influence of the selectable parameters on bit and frame errors has been used as the basis for the analysis.

By simulation analysis of static scenarios where nodes were placed along a line corresponding to a stretch of road for different inter-node distances and settings of protocol parameters, we were able to conclude the following:

- The proposed optimization scheme can be used to improve performance in terms of the end-to-end metrics message delivery probability and delay until first arrival. In the investigated scenarios, the optimal parameters were quite different and thus a static default setting of protocol parameters would not yield optimal results. More specifically we found that the use of the 11 Mbit/s transmission rate can improve the global end-to-end performance compared to 1 or 2 Mbit/s even though it is less robust.
- When the level of contention is high, as is the case in some scenarios with the high-frequency broadcast application that has been considered in this work, the flooding broadcast protocol cannot deliver high coverage due to correlated collisions. Therefore a more efficient and reliable broadcast protocol should be used for this type of applications.
- In the investigated scenarios, where the transmission power is increased stepwise, the experienced end-to-end delays are well below the acceptable limit until at a certain point buffer overflows start to occur. This means that when determining optimal parameter values the end-to-end delay should be used first to determine the valid region in which the optimal transmission power setting can be determined afterwards.

4.3.1.9 Always Best Connected scenario

This study has focused on the usage of QoS differentiation and resilience mechanisms to improve the quality and reliability of the Internet access in an Always Best Connected (ABC) context for Mobile Terminals (MTs) with high requirements when it comes to QoS and service availability. The main idea is to use Differentiation and Protection to increase resilience for those MTs that require it. Simulations have been

[5] http://www.isi.edu/nsnam/ns/
[6] http://www.python.org/

carried out where MTs are divided in “High Priority” (HP) and “Best Effort” (BE) MTs. When the resource demands for an AP are higher than a certain level, the BE nodes will receive congestion signals, whereas the HP nodes will not as they have reserved resources for their communication (but with the resource pool for HP MTs being less than the total resources available to avoid starvation of BE nodes). The HP nodes may also use “Protection”, with resource allocations at 2 different Access Points (APs) at the same time. For 1+1 Protection, both these APs resources are marked as “used”, to mimic the situation where traffic is duplicated and arriving at the MT via both the APs. This may be used for emergency services, where reliability requirements are very high. For 1:1 Protection, on the other hand, HP capacity is reserved at 2 APs, but one of them is used as backup and with no duplicate traffic stream, so the resources are available to BE. If the primary AP fails, however, the HP MT will use the resources at the secondary “backup” AP.

Using these mechanisms comes at a cost. The usage of QoS differentiation provides increased service quality for the “favoured” MTs, but at the expense of the other “Best Effort” MTs. The usage of protection mechanisms may result in a higher total resource demand (when using 1+1 protection) and will always require careful planning to avoid lack of resources for these nodes as both protection mechanisms require that extra capacity is allocated. It is clear that both QoS differentiation and protection must be employed with care.

We have investigated these issues by using simulations where APs have a fixed total capacity and a fixed capacity limit for high priority reservations. For some of the simulations we let the APs fail according to a probability distribution to see the effect on the connectivity of MTs of different types. We have studied how the usage of QoS Differentiation and Protection affects the connectivity and QoS for the MTs employing these mechanisms, and also how the other MTs are affected. The mechanisms studied includes the basic Differentiation and Protection (1+1 and 1:1) schemes mentioned above, HP nodes not using protection, and BE nodes not tolerating congestion (will try to find another AP if there is congestion – if no AP is found it will be disconnected).

The main results coming from the evaluation of differentiation and protection mechanisms for ABC have been sketched in Section 3.5.2.4 and are detailed in D3.1.2 [39].

4.3.1.10 Multi-channel multi-radio

A multi-channel multi-radio (MCMR) architecture has been proposed as a means to enhance resilience by increasing the overall performance of the network and the availability of radio resources. An extensive simulation study has been performed to evaluate the proposed architecture with a variable, but realistic number of channels and radios for IEEE 802.11 based systems in mobile (vehicular) environments. An existing network simulator has been modified to support both multiple mesh radios per node and dynamic channel switching algorithms. Based on the evaluation, it can be concluded that:

- The proposed use of MCMR will decrease congestion-induced faults by increasing the network capacity; also it will improve end-to-end delay in multi-hop communication.
- The robust channel assignment method proposed, guarantees connectivity without putting any requirement on synchronization or communication between nodes has been demonstrated.
- The guaranteed connectivity between nodes is especially useful in networks with dynamic topologies/mobile nodes, which is the case in car-to-car communication.
- The optimal number of radios per node depends on network traffic patterns and node density.

4.3.2 Experiences in evaluation of applications and use-cases

In this section the focus moves to the analysis of some selected applications and use-cases among those introduced in D1.1 [37]. In particular, Section 4.3.2.1 deals with the dependability analysis in the context of

the “distributed black box” application, while the QoS analysis of a subset of the “car accident” use-case is briefly discussed in Section 0.

4.3.2.1 Distributed black box

The distributed black application consists of recording at regular intervals informational data about the state of participating vehicles and their environments. This data is temporarily replicated on participating vehicles encountered in the ad-hoc domain. Permanent backups are created when the participating devices are able to access the fixed infrastructure. This application implements three main HIDENETS services: 1) Proximity Map, 2) cooperative data-backup, and 3) trust & cooperation. The evaluation activities concerned the elaboration of analytical dependability models allowing: 1) the comparison of various data replication strategies and 2) the estimation of the expected dependability gain compared to scenarios where cooperative backup in the ad-hoc domain is not used (i.e., the data recorded from the source mobile node is stored at the infrastructure domain only when an access to the infrastructure is possible). The main dependability properties that we have analyzed are data availability and confidentiality.

The dependability modelling is based on generalised stochastic Petri nets (GSPN) and Markov chains. We have carried out several sensitivity studies to analyze the dependability gain provided by this application as a function of (i) the various environmental parameters (frequency of Internet access, mobile nodes encounter rate, and node failure rate) and (ii) different replication strategies. The results allowed us: 1) to determine under what circumstances the cooperative backup is the most beneficial, compared to solutions that do not replicate data in the ad-hoc domain, and 2) to choose among different replication strategies (e.g., simple replication vs. erasure coding based replication), depending on a given scenario’s parameters and user preferences (e.g., target data availability, confidentiality requirements). As an example, we have shown that the cooperative backup service can decrease the probability of data loss by a factor that can be as large as the ad-hoc network to Internet connectivity ratio. Also, we have observed that erasure codes provide an advantage (dependability-wise) over simple replication only in narrow scenarios. More details are available in [18] and [22].

4.3.2.2 Car accident use-case

A model-based evaluation approach has been developed for the analysis of a subset of the car accident use-case scenario sketched in Section 2.1.3. The analyzed network scenario is composed by a set of overlapping UMTS mobile communication networks covering a high-way, and a set of mobile users (cars and emergency vehicles) moving in the high-way and requiring different UMTS services (e.g., conversational, interactive, and background). The final goal was twofold: to quantitatively evaluate some QoS measures regarding both the users (e.g., the probability that an ongoing service request is interrupted) and the mobile operators (e.g., the load factor of the UMTS cells), and to show the feasibility of the analysis of a complex and comprehensive HIDENETS use-case scenario among those identified in Chapter 2.

To do this, as detailed in [23], we have adopted a modular, hierarchical modelling approach (using Stochastic Activity Networks formalism) based on composition, replication and parameterization, which facilitates the model construction process as well as the model reusability. The produced numerical results provide a useful insight in the relationships between the selected QoS measures, the users’ behaviour and the users’ mobility. The classical QoS analysis is enhanced by taking into account the congestion both caused by outage events reducing the available network resources, and by the varying traffic conditions. In addition, they show the effectiveness of the modelling approach considering the computational time required to solve the overall model by simulation. The modularity of the proposed modelling approach has been also used to enable a more refined representation of the users mobility aspects through the integration of mobility traces generated by an external user mobility simulator (VanetMobiSim[7], in particular) into the SAN model (see D4.2.2 [36] for the details).

[7] http://vanet.eurecom.fr/

4.3.3 A holistic evaluation workflow to analyze high-level measures in dynamic HIDENETS environment

This activity has been carried out to demonstrate how to solve a complex evaluation problem by integrating the results of partial solutions. An evaluation workflow was proposed that supports the inclusion of several tools and solution steps. The problem selected for demonstration was the computation of the probability that a series of user activities is successfully executed in a dynamic HIDENETS environment. Here the users try to execute collaborative activities with other users relying on applications built upon the services of the HIDENETS infrastructure and ad-hoc domains. The environment is characterised by changes including the movement of users, the scheduling of collaborative activities, and the failure and recovery of resources that influence the availability and quality of the services.

The overall evaluation follows the approach of multilevel modelling addressing phases (this approach is described in D4.1.2 [24]). Hierarchical multilevel modelling is applied by considering user, application, architecture (services) and communication levels, while the multi-phase approach is used to separate time periods in which the users' activities and the environment conditions can be considered unvarying. Accordingly, existing evaluation solutions integrated in the evaluation workflow belong to specific hierarchy levels (e.g., simulation of movements and construction of topology models on the network level, stochastic dependability modelling and evaluation covering the application and resource levels) and handle the phases (e.g., identification of phases, solution of phased analysis models).

The input and output of the evaluation workflow are sketched in Figure 11. Parallelograms represent models; multi-layered parallelograms stand for multiple models. The rectangles illustrate modelling, model transformation and model solution steps. The three sets of *input* models (views of the scenario being evaluated) are as follows:

- Each *user workflow* specifies the user activities in terms of application usage (the mobility aspects are not addressed in the user workflow, they are included in the topology model). There is a user workflow for each participant of the scenario to be evaluated. The information required for the evaluation can be extracted from UML based workflow models (activity models extended with time information). There is also the possibility to use a domain-specific workflow editor that was implemented on the Eclipse platform.
- Each *topology model* represents the information on the evolving ad-hoc topology of network connections. In case of multiple networking technologies there is a separate topology model for each technology that can be used for communication. The topology model can be constructed utilizing an existing mobility trace generator and network topology generator tool-chain reported in D4.1.2. As a current application example, VanetMobiSim is supported as mobility trace generator and the topology model is constructed from its output traces (assuming a WLAN technology).
- The *application-service dependency models* define how the given applications depend on the services, hardware or software components of nodes. These dependencies are described by UML architecture diagrams.

The concept of hierarchical multi-level modelling is directly applied when the application-service dependency models are constructed, as the already computed dependability parameters are used at the highest level of the application hierarchy. For example, if dependability parameters of a service are computed by a specific evaluation tool or technique then the internal structure of this service is not considered. If there is no specific evaluation technique then the lower level software structure and its dependencies on the hardware and communication resources are taken into account by using the UML based dependability model construction approach described in D4.1.2 (note that this latter approach provides generic and rough evaluation in comparison with the specific evaluation that considers the precise semantics of the services).

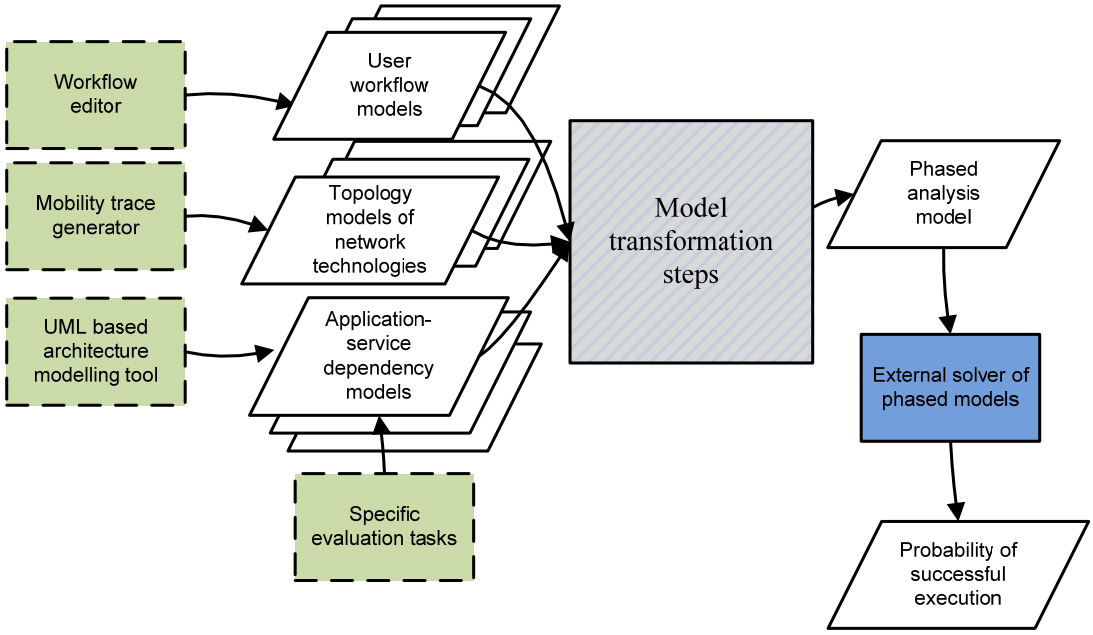


Figure 11: Inputs and output of the evaluation process

Accordingly, the "specific evaluation tasks" input in Figure 11 can represent various evaluation tasks developed in HIDENETS, depending on the applications and services included in the user workflow to be evaluated (e.g., if the user activity relies on the response from replicated servers then the results of its specific evaluation task can be included in the parameterised application-service dependency model). Potentially, all the pointwise evaluations detailed in Section 4.3.1 could be considered at this stage, like the experimental evaluation of the Reliable and Self-Aware Clock, the reliability evaluation of the Replication Manager, the results of the various network simulation techniques.

The *output* of the evaluation workflow (that is represented as the output of the internal model transformation steps in Figure 11) is a phased analysis model. This phased analysis model is solved by an external solver (e.g., Möbius [125] or DEEM [124]) which carries out the transient evaluation resulting in the success probability of the user activity sequence along the scenario.

The experiments with the evaluation workflow (presented in more detail in [17]) resulted in the following lessons learned:

- A default approach of refining the dependability models to the lowest levels of the application-resource hierarchy results in state space explosion. The complexity can be mastered by solution feedback, i.e., including in the evaluation workflow (the results of) specific evaluation techniques that consider the precise semantics of the services. This way unnecessary refinement can be avoided.
- Multiple interactions and interdependencies among the users and components (especially the number of potential multi-hop connections among the participants) exacerbate the complexity issue. To reduce complexity, our initial step was the decomposition of the problem by identifying the users that do not communicate with each other and analyzing the resulting "flocks" independently. The handling of complex scenarios will necessitate the inclusion of more powerful abstraction techniques.
- We also gained experiences from the practical implementation of the evaluation process. It includes multiple models and notations. In order to carry out model transformations on these, a common framework (model bus) was needed. In our case the VIATRA2 graph transformation framework was used. It is easily extensible with plug-ins, so manipulation, import, and export of models can be carried

out by custom components. The application of this framework for the implementation of model transformations speeded up the development process.

- Finally, the practical output of the evaluation, that is the computed probability of a successful execution of a user activity sequence, can be useful to characterise the user workflow in a best case or worst case situation, to compare different execution strategies at the user level, or to compare different environment options. This problem can be generalised for the analysis of business workflows executed in environments in which the resources are changing dynamically.

Concluding, the implementation of the evaluation workflow allowed us to have an insight into the complexity of evaluating a user level dependability measure in a HIDENETS environment (facing the challenges identified in Section 4.1), and contributed to the validation of initial ideas for complexity reduction as well as to the development of techniques for tool integration.

4.4 Relevant publications

[23] A. Bondavalli, P. Lollini, L. Montecchi. Analysis of User Perceived QoS in Ubiquitous UMTS Environments Subject to Faults. In Software Technologies for Embedded and Ubiquitous Systems, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 5287/2008, Pages 186-197, 2008.

[16] A. Casimiro, P. Lollini, M. Dixit, A. Bondavalli, P. Veríssimo. A framework for dependable adaptation in probabilistic environments. In Proc. of the 23rd ACM Symposium on Applied Computing (SAC 2008), Dependable and Adaptive Distributed Systems (DADS) Track, pages 2192-2196, Fortaleza, Ceara, Brazil, March 16 - 20, 2008.

[18] L. Courtès, Cooperative Data Backup for Mobile Devices. PhD Thesis, LAAS-CNRS, November 2007.

[22] L. Courtès, O. Hamouda, M. Kaâniche, M.-O. Killijian, D. Powell. Dependability Evaluation of Cooperative Backup Strategies for Mobile Devices. In Proc the 13th IEEE Int. Symp. On Pacific Rim Dependable Computing (PRDC-07), December 2007.

[17] M. Kovács, P. Lollini, I. Majzik, A. Bondavalli. An Integrated Framework for the Dependability Evaluation of Distributed Mobile Applications. In Proc. of the RISE/EFTS Joint International Workshop on Software Engineering for REsilieNt systEms (SERENE 2008), pages 29-38, Newcastle upon Tyne, UK, November 17-19, 2008.

[126] J . Nielsen, J. Grønbæk, T. Renier, T. Toftegaard, HP Schwefel, “Cross-Layer Optimization of Multipoint Message Broadcast in MANETs”,To appear in Proceedings of IEEE WCNC 2009.

[127] A. Nickelsen, J. Grønbæk, HP Schwefel, “Probabilistic Network Fault-Diagnosis using Cross-Layer Observations”,To appear in Proceedings of AINA 2009.

[128] E. Matthiesen, O. Hamouda, M. Kaaniche, HP Schwefel, “Dependability Evaluation of a Replication Service for Mobile Applications in Dynamic Ad-Hoc Networks”, International Service Availability Symposium (Proceedings to appear in Springer LNCS), Japan, 2008.

[129] Y.Liu, F. Li, HP Schwefel, “Reliable Broadcast in Error-Prone Multi-hop Wireless Networks: Algorithms and Evaluation”,Proceedings of IEEE Globecom 2007.

[130] Y. Liu, F. Li, A. Nickelsen, HP Schwefel, “A New Link State Routing Protocol for Mobile Ad-hoc Networks”, 4th IEEE International Symposium on Wireless Communication Systems (ISWCS), October 2007.

[131] E. Matthiesen, T. Renier, HP Schwefel, “A new selection metric for backup group creation in inter-vehicular networks”,Proceedings of IST mobile summit, July 2007.

[132] Y. Liu, HP Schwefel, “Localised Algorithms for Virtual Backbone Formation in Wireless Multi-hop Networks with unidirectional links”,Proceedings of IST mobile summit, July 2007.

[133] I. Antonos, L. Lipsky, HP Schwefel, “Performance-relevant network traffic correlation”, [with I. Antonios, L. Lipsky]14^th International Conference on Analytic and Stochastic Modelling Techniques and Applications, ASMTA June 2007.

[134] HP Schwefel, I. Antonios, “Performability Models for Multi-Server Systems with High-Variance Repair Durations”, Dependable Systems and Networks (DSN), June 2007.

[135] J. Grønbæk, HP Frejek, T. Renier, HP Schwefel, “Client-Centric Performance Analysis of a High-Availability Cluster”, Proceedings of International Service Availability Symposium, published in Springer LNCS, May 2007.

[136] Y. Liu, HP Schwefel, “Algorithms for Efficient Broadcasting in Wireless Multi-hop Networks”, Proceedings of IEEE Globecom, Nov. 2006.

[137] RL Olsen, MB Hansen, HP Schwefel, “Quantitative analysis of access strategies to remote information in network services”, Proceedings of IEEE GLOBECOM, November 2006

[138] T. Renier, E. Matthiesen, HP Schwefel, “Inconsistency Evaluation in a Replicated IP-based Call Control System”, In D. Penkler, M. Reitenspiess, F. Tam (eds.) ‘Service Availability’, LNCS 4328, pp.177-192. Springer, 2006.

5. Model based application development

From the very beginning, we aimed at technical solutions where “Resilience and availability of services deployed either in an ad-hoc domain or on dedicated servers in the Internet, have to be taken into account on a system design level, since the components are inherently unreliable.” (from “HIDENETS – Description of Work” [112])

We have defined a design methodology that can reach this goal. We have chosen a modelling based approach since modern design methodologies fit under the model-driven architecture (MDA) initiative in which applications are primarily designed and specified by their (semi-)formal model. MDA and UML have been the glue to interlock our efforts in supporting both the application development and testing.

5.1 Challenges and Activities

The main challenge that we met in our work on the model based application development was how to support the industrial utilization of the solutions elaborated during the HIDENETS project for all the challenges identified at the beginning of the project in [37]. That is, rather than providing solutions for some of the enlisted challenges, our work was focused on the application development that eventual has to deal with all of them.

Therefore our work was focused on:

- The analysis of applicability of existing modelling standards of the application field:
  - SysML as the most widespread modelling language for complex embedded systems
    It aims at supporting the specification, analysis, design, verification and validation of a broad range of large and complex systems that include hardware and software components. Since our aim in the HIDENETS project is to propose a meta-model for Platform Independent Models, which is able to describe complex, embedded, real-time systems from a high-level view, SysML fits our purposes only partly.
  - AUTOSAR as the de-facto standard of the automotive industry
    This one aims at the development and establishment of a common open industry standard for an automotive E/E (electrical/electronic) architecture, facilitating the reuse of software components between different vehicle platforms, equipment manufacturers and suppliers. However, that kind of component development lies outside the scope of the project, and additionally it would also limit the project partners to develop all HIDENETS services in a strict AUTOSAR manner.
  - UML 2.0 as standard general modelling language with its Profiling Mechanism and Testing Profile
    The Profiling Mechanism of UML2.0 supports the definition of different domain specific languages for the dependability related requirements of the applications, for the ad-hoc and infrastructure domain etc. (For further details please refer to the HIDENETS deliverable D5.1 – “Preliminary UML profile and design patterns library” [113].)
    The UML 2.0 Testing Profile is an official OMG standard UML extension defining testing related concepts that can be associated with four groups: Test Architecture, Test Behaviour, Test Data and Time. Although a good basis for testing related concepts, it lacks the necessary specifics to deal with our specific environment (see chapter 6).
- Supporting other work packages by formalising and visualising different architectural options and by supporting the different subtasks using different modelling approaches.

  With the extensive study of the above mentioned standards and approaches we learned the concepts necessary to have an established view of the application field. Based on that knowledge we created a model (view) that could support the different needs of different project activities (development of resilient communication, architecture and middleware, and the quantitative evaluation) best. We continuously kept track of changes in the planned and implemented HIDENETS services, providing

models and information on possible inconsistencies and incompleteness in the preliminary results on Resilience architecture and middleware and resilient communication, while providing data on the dependency relations of the internal services for the HIDENETS holistic evaluation approach.

- Creating meta models for the newly designed middleware services

  Cooperating with the activities on Resilience architecture and middleware we had to study the middleware service candidates as they serve as basis of any application development. The decision to build our own middleware services for the nodes in the ad-hoc domain but apply standard ones on nodes in the infrastructure domain confronted us with some extra challenges: We had to study the existing documentation of the chosen standard solution that was produced (by the Service Availability™ Forum) independently of the project and its goals (see [114], [115], [116]).

  The Service Availability™ Forum (SA Forum) aims at fostering standardised middleware solutions for making services highly available by specifying availability interface standards. The Application Interface Specification (AIS) of the SA Forum defines the standard interfaces for accessing Highly Available (HA) middleware and infrastructure services that reside logically between applications and the operating system. These specifications strictly divide physical from logical views while their (open-source) implementations provide to us a basis for examining HA functionalities in the infrastructure domain.

  We have selected the approach and the specified interfaces of the SA Forum at the beginning of the project as a possible solution for the server nodes in the infrastructure domain while keeping in mind that their High-Availability techniques might be adapted in the ad-hoc domain as well. To fulfil the first one and estimate the second, we had to build a meta-model for the easier and better understanding of the SA Forum services. When specifying the resilient middleware we have decided that our focus will be set on SA Forum's Application Interface Specification (AIS) and neglect the hardware related interface specifications (the Hardware Platform Interface Specification – HPI of SA Forum). From the existing AIS implementations, OpenAIS and OpenSAF were selected, being available as open-source implementations, with some preference towards OpenAIS for its slightly less complexity.

  After scrutinizing AIS – with a strong emphasis on the Application Management Framework (AMF) – we selected a relevant set of services for our infrastructure domain solutions – keeping in mind the prototype implementation of WP6 (see section 7.1 for further details) – we built a meta-model introducing both the physical and logical concepts of AIS and inserting the services and their relationships in the system [117].

  Creating meta-models for the middleware services of both domains was the basis for coping with the next challenge.

- Providing a suitable way of application development

  Given an established knowledge of both the HIDENETS middleware services and SAF solutions we had to find a formalised solution to incorporate it into a model-driven application development method.

  UML was designed as a general modelling language. However, instead of defining all the modelling concepts of the domains where UML could be used, the specification contains only some core elements and a standardised extension mechanism is given. We defined a HIDENETS UML profile covering the main concepts and aspects of both domains and the corresponding two middleware services. To support application development, the following multi step process was used (for a more detailed description of the process, please refer to [102] and the HIDENETS deliverable D5.3 – "Refined design and testing framework, methodology and application results" [118].)

  First we developed the meta-models that illustrate the intended organization of applications running on the platforms and identify the key concepts from an application developer's point of view with respect to various services. The meta-model introduced the corresponding meta-classes and connected these newly introduced meta-classes to core UML concepts.

Having constructed the meta-models, a UML profile was constructed on the basis of these meta-models. The profile defines stereotypes and tagged values to be used for annotating UML models with dependability and platform-related information.

Further we provided a set of design patterns to support the implementation of applications built for the development platforms using our profile. These patterns can be seen as detailed examples for implementing various dependability-related parts of applications based on the defined interfaces.

We created a Domain Specific Editor (DSE) for the HIDENETS environment incorporating the profile thus providing support for application developers to use the defined stereotypes hence making their models easier to understand while providing an opportunity for further model-based tools to enhance the process. The DSE was used to deliver the Platoon Driver Support System (PDSS) application of the Application Development test-bed.

Finally we have elaborated methods for the code and configuration generation, providing means for a more efficient and less error prone transformation of the model based design into source code and configuration data. Our implementation is based on the IBM Rational Software Architect modelling product [119] that is built on the extensible architecture provided by the Eclipse development platform as RSA supports a lightweight extension mechanism through so called plug lets. Plug lets are Java applications integrated into the RSA framework, and they are able to access the model in the model space using the Eclipse Modelling Framework (EMF) APIs.

So we implemented separate plug lets that traverse the application model, access entity attributes, identify them through the stereotypes and generate the configuration file (plain text or XML, depending on the target middleware implementation) in a uniform way [120]. The structure of all such files will be identical over the different applications, hence easier to understand, maintain and compare whenever necessary. We implemented another plug let traversing the application model and generating the source code.

5.2 Lessons learned and guidelines

First we gained new insights in defining UML based modelling languages for special application domains and to adapt such languages to the needs of different underlying middleware systems, and how to express design patterns in such languages. (See “D5.3 Refined design and testing framework, methodology and application results” [118] for more details.) The main learned lesson, how to introduce the new elements into the modelling language and how to separate the different semantic levels, are summarised there in our multi-phase design approach. We have learnt that the model based development approach is applicable even in the special field of application development for the HIDENETS middleware, and UML with its profiling methods is capable of modelling the corresponding special requirements and system components. Further we had to learn that the SysML fits our purposes only partly, and the AUTOSAR standards not yet cover the fields related to the application development for car-to-car and car-to-infrastructure communication.

From an application designer's point of view it would be preferable to have a single interface for the underlying middleware services through all the different possible executing nodes (being a highly available server node in the infrastructure domain, a complex node in the ad-hoc domain or an embedded node in a vehicle). However, our next lesson was that the “harmonisation of the interfaces of different middleware systems” (HIDENETS DoW [112]) has its limits that in our case mainly come from the differences in the requirements in the two domains and from the correspondingly different services that the middleware offer. After a deep dive investigation to find a way of harmonization between

- the HIDENETS services of the ad-hoc domain developed during the project
- and those provided by SAF specifications and implementations

The resilient middleware can be either an implementation created from scratch, built intentionally on the HIDENETS architecture, or an existing implementation, modified to use the HIDENETS services for the internal operation. Figure 12 shows the difference between the two approaches.

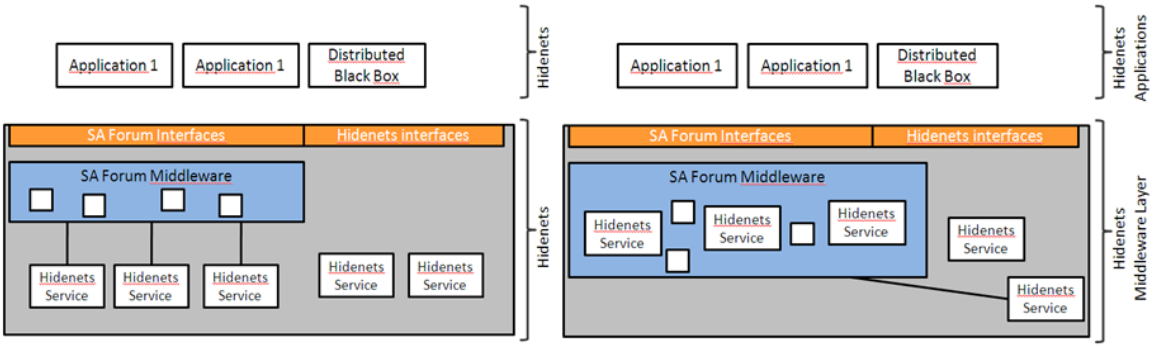


Figure 12: Relations of the SA Forum Middleware towards applications and HIDENETS

When an existing middleware implementation is adapted to HIDENETS, then the existing SA Forum middleware building blocks, e.g. the communication subsystems, are modified to use HIDENETS services (see left figure). In this approach, the middleware provides only SA Forum services to the applications. In the other case, what we chose in HIDENETS, the middleware is implemented from the ground up by using HIDENETS services. This allows the middleware to provide also those services complementary to the SA Forum ones and become an integral part of the overall architecture (see right figure). In this case, the HIDENETS applications have to be implemented on top of this architecture and use both the standard SA Forum and the established HIDENETS interfaces to access the provided services (as already shown at Figure 4 in Section 3.1.3).

Further, when examining dynamism of the SA Forum solutions to address requirements of the ad-hoc domain, we have found that current AIS implementations support mainly rather static environments and lack the ability to autonomously and frequently modify cluster configurations. After analyzing the available solutions we found 3 different approaches [121], [122]:

- One is based on an AIS Service, the Information Model Management (IMM) that is originally defined to be in charge of creating, accessing and managing the objects representing the whole system.
- The second approach is based on the modification of the system model by means of the Software Management Framework (SMF) that aims at controlling and executing the “migration from one configuration to another” [26]. As in the case of IMM, it has not yet been implemented by any SA Forum members.
- The third way is based on the Simple Network Management Protocol (SNMP) that collects and stores management information about the Managed Objects according to the definitions in the Management Information Base (MIB). The MIBs representing the information about an AIS cluster are defined in [27]; they describe the same logical entities as the Information Model and the relationships amongst them.

To create a pioneer implementation of these mobility services and to build it into an existing SA Forum implementation exceeded the scope of the project.

The third group of lessons is related to that if one has to design applications that have to run on different platforms (different hardware resources, different implementations of the middleware services, different execution environments, ...), automatic code generation is even more advantageous. Beyond the already well-known benefits of automatic code generation we have learned how to structure modelling and the modelling language to provide a solid basis for the automatic code generation for a highly heterogeneous execution platform without overloading the application developer. Because of the existing different implementations we took the configuration of the SA Forum middleware as an example, but the lessons learned here are transferable to other cases, as well.

As the HIDENETS platform is intended to run separate, distributed applications parallel on nodes with different physical properties (performance, user interfaces, dependability, ...), and even the set of applications may differ depending on the different node instances, we provide a model-based configuration generator tool that can automatically synthesise the required configuration information. This makes the development and maintenance of the applications running on the HIDENETS middleware more efficient and dependable.

SA Forum seemed a good field for experimentation, having a well-defined set of services and a clustered nature that is in many ways akin with distributedness, while with results on this specific field ”HIDENETS will advance the industrial practice of implementing highly available, resilient services” (from “HIDENETS – Description of Work” [112]) and influence and support standardization efforts.

The configuration is different in all existing middle-ware implementations (both in structure and representation) making applications practically non-portable as manual configuration of large-scale systems is not only a demanding task but a rich source of human-prone errors as well. So we aimed at developing a tool for automatic configuration generation, to invent a methodology that can easily be adapted to any future middleware implementation while its power can be demonstrated on multiple existing ones, for which purpose OpenAIS and OpenSAF were selected.

The above introduced methodology can not completely eliminate the task of manual configuration as some information for the automatic generation is not available in the application model – like the path of binary executables – but it lessens the necessary efforts of configuration considerably and can be easily extended for other AIS middleware implementations as well – the underlying AIS profile and thus the stereotypes are independent of it. Furthermore, the technique itself is completely independent of SA Forum specifications and can be easily adapted to the HIDENETS middleware for the ad-hoc domain and to other platforms/middleware implementations.

And the fourth lesson in this field is automatic code generation is another very helpful support for the application design. Automatic code generation is another one of the most important aspects of application development support. It does not only help developers in their work, but also removes a set of human-prone errors thus reducing the development time significantly and increasing the dependability level of the resulting software. Again, we took the SA Forum middleware as an example, but the learnings are more general.

Our studies of different HA middleware specifications showed that most components built on similar middleware functions have a similar structure (even over different middleware implementations), thus providing a very useful basis for model-based code generation, as all these entities could be generated from common code templates. That way we enable developers to avoid the demanding routine tasks and focus their efforts on the business functionality instead.

5.3 Relevant publications

[112] EU FP6 IST project HIDENETS, Project Proposal Annex I – Description of Work, http://rcl.dsi.unifi.it/projects/HIDENETS-DoW.pdf

[37] M. Radimirsch et al., “Use case scenarios and preliminary reference model”, EU FP6 IST project HIDENETS, deliverable D1.1. September 2006.

[113] EU FP6 IST project HIDENETS, Project Proposal Annex I – Description of Work, http://rcl.dsi.unifi.it/projects/HIDENETS-DoW.pdf

[114][András Kövi, Dániel Varró, Zoltán Németh: Making Legacy Services Highly Available with OpenAIS: An Experience Report. ISAS 2006: 206-216

[115] Z. Micskei, I. Majzik, F. Tam: Comparing Robustness of AIS-Based Middleware Implementations, In Proceedings of International Service Availability Symposium (ISAS 2007), LNCS 4526, Durham, New Hampshire, USA, May 21-22, *2007.*

[116] Zoltan Szatmari, Andras Kovi and Manfred Reitenspiess. Applying MDA for SA Forum AIS based application development. MAI2008 workshop at DisCoTec2008

[139] Z. Szatmári, “Model-driven development for highly available services”, MSc Diploma thesis, BME, 2008

[102] Pintér G., Micskei Z., Kövi A., Égel Z., Kocsis I., Huszerl G. and Pataricza A.: Model-Based Approaches for Dependability in Ad-Hoc Mobile Networks and Services. In R. de Lemos, F. Di Giandomenico, C. Gacek, H. Muccini and M. Vieira (Eds.) Architecting Dependable Systems V (LNCS-5135) pp. 150-174. 2008, Springer

[118] Gábor Huszerl, Hélène Waeselynck (eds.), Zoltán Égel, András Kövi, Zoltán Micskei, Minh Duc N’Guyen, Gergely Pintér and Nicolas Rivière, “Refined design and testing framework, methodology and application results”, EU FP6 IST project HIDENETS, deliverable D5.3, December 2008, http://www.hidenets.aau.dk/Public+Deliverables

[119] IBM Rational Software Architect official home page, http://www-01.ibm.com/software/awdtools/swarchitect/websphere/

[120] András Kövi, Dániel Varró: An Eclipse-Based Framework for AIS Service Configurations. ISAS 2007: 110-126

[121] Gábor Urbanics, András Kövi, Zoltán Égel, András Pataricza. Introducing dynamism to SA Forum cluster, DNCMS08 workshop at SRDS2008.

[122] G. Urbanics, “Introducing dynamism to SA Forum cluster”, MSc Diploma thesis, BME, 2008

[26] Service Availability ForumTM - Application Interface Specification Software ManagementFramework SAI-AIS-SMF-A.01.01

[27] Service Availability ForumTM - Distributed Systems Management for AIS-SNMP SAI-AIS-SNMP-A.01.01, 2005.

6. The testing framework

Software testing consists of executing a program with defined input values and then verifying whether the outputs conform to the expected behaviour. In this section, we address the challenges and methodologies for the verification of HIDENETS-like applications and middleware services using testing.

6.1 Challenges and activities

Our definition of a testing framework for mobile-based applications contributes to the following project goal:

“Identify development tools and mechanisms like design patterns and testing methodologies to assist in the implementation of said service qualities.”.)

Work was focused on the verification of the highest layers in the HIDENETS architecture, that is, the application layer and possibly some high-level middleware services. We consider functional (black-box) approaches to test whether applications fulfil their expected requirements. Note that quantitative evaluation, e.g., reliability or availability assessment, is not addressed here (it is studied in Section 4). Our interest is on the correctness issue.

As a first step, a review of relevant literature has been performed together with a testing case study that allowed us to gain concrete insights into validation problems. One of the conclusions was the lack of adequate formalisms to capture system-level behaviour and spatial topology in a mobile setting. Work has then been directed towards the definition of a scenario-based testing framework that covers (1) the definition of a language that describes interaction scenarios in mobile settings, and (2) some automated support to analyze and implement scenarios on a test platform with simulation facilities.

Usual scenario languages do not offer concepts to express the spatio-temporal relationships of nodes as first-class entities, nor do they offer concepts to represent broadcast communication in local vicinity. We proposed extensions to fill these gaps, and integrated them into a widely used scenario language, namely UML 2.0 Sequence Diagrams [50]. The extended Sequence Diagrams include two connected views, the spatial view (describing the topological configurations of the scenario) and the event view (describing communication events, and their causal dependencies on configuration change events).

Obviously, the automated processing of scenario descriptions requires that they are given a well-defined meaning. A difficulty is that not all available constructs in scenario languages have clean semantics. For example, UML 2.0 has introduced many new elements in the sequence diagrams, but they come with intriguing semantics problems (see e.g., [51][47]). The solution we retained is based on a restricted usage of the problematic elements in the event view, which allowed us to define semantics that suit our application domain (analysis of test traces). We also developed a graph matching tool to help the extraction of test scenarios from test traces, in accordance with the spatial view.

6.2 Lessons learned and guidelines

The lessons and guidelines from our work relate to technological issues (test platform) as well as more conceptual ones (role of scenarios in the testing framework, specificities of scenarios in mobile settings, automated processing of scenario descriptions).

6.2.1 Test platform

The test platform should be as realistic as possible with respect to the operational environment. However, this is not easy to achieve. In practice, it is difficult and expensive to test some mobile applications that

require checking the combined logic of the involved hardware and software components. For example, in the automotive domain, realistic platforms involve prototypes that are implemented into real cars (e.g., see [45]). The cost of such platforms, as well as controllability and observability constraints, implies that part of the testing activities may preferably be performed using emulation/simulation facilities.

We assume the use of three categories of facilities: Application execution support, Network simulator, and Context controller. Concrete examples of platforms built according to this generic architecture (see Figure 13) are provided in [49] and [53].

The application execution support is needed to emulate the executive support for the application code. A requirement is to provide an interface to the context controller and network simulator, so that the application can interact with them as if it would interact with a real environment. Since applications and services connect over wireless links, the simulated environment must include a model of the underlying network. The network simulator is responsible for simulating the full functionality of a real wireless network. Network simulators like ns-2[8], GlomoSim[9] or SWANS[10] can be used.

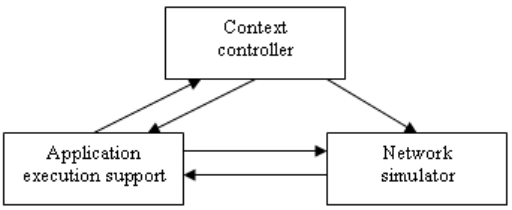


Figure 13: High-level view of the platform

The context controller is needed to simulate context information. Applications exploit context information to take different actions adaptively. Context is also needed by the network simulator to set up input parameters for imitating the real network characteristics in order to manage connectivity between nodes. Indeed, the delivery of the messages has to be done in agreement with the network topology. There have been several toolkits for simulating contexts in recent years. Some of them are built on 3D game engines and serve to simulate a first-person view of the physical environment [44]. Other examples include generic location event simulators, like [52], or traffic simulators e.g., see [46].

This simulated environment needs Points of Control and Observation (PCOs) to control and observe the test experiments. Depending on the target application and test strategy developed, PCOs can be integrated in each component or can be realised as a global component added to the architecture.

6.2.2 Role of scenarios in the testing framework

Scenario descriptions are useful to support various test-related activities, such as the representation of requirements, of test purposes (i.e., interaction patterns to be covered by testing), of test cases, or of execution traces. Accordingly, the testing framework depicted on Figure 14 shows scenario-based artefacts that may be produced during different Verification and Validation phases (V&V).

[8] http://www.isi.edu/nsnam/ns/
[9] http://pcl.cs.ucla.edu/projects/glomosim/
[10] http://jist.ece.cornell.edu/

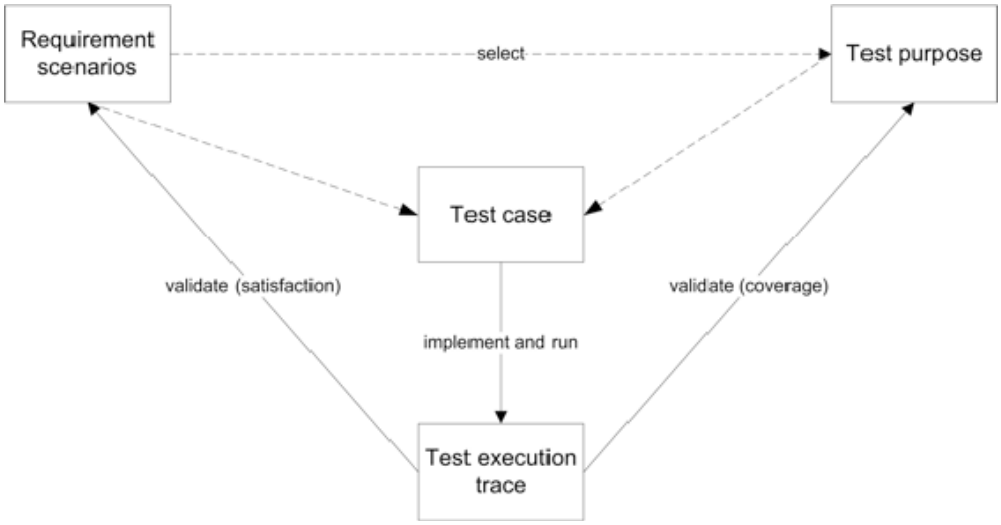


Figure 14: Overview of the testing artefacts

This testing framework does not require commitment to heavyweight formal methods. Hence, the transition from one test specification artefact to the other may be informal, as expressed by the dotted lines. For example, test purposes may be derived informally from the important requirements, and test cases may be proposed by the user to cover some intended purpose. Note that the framework does not preclude the use of more formal approaches. Would a complete specification of behaviour be available, then the framework could possibly be extended to support formal treatments such as: the verification that the behaviour model exhibits the requirement scenarios, or the automated generation of test cases from a model and a set of test purposes. However, such formal treatments were not investigated within HIDENETS.

Even if a complete specification of behaviour is not available, some automated treatments become possible by simply using scenario descriptions. They are indicated by solid lines in the figure and will be the focus of our work. The treatments include:

- Checking whether a test execution trace satisfies a requirement scenario.
- Checking whether a test execution trace covers a test purpose.
- Assisting in the implementation of test cases (and more specifically in the production of concrete contextual data).

6.2.3 Scenarios in mobile settings

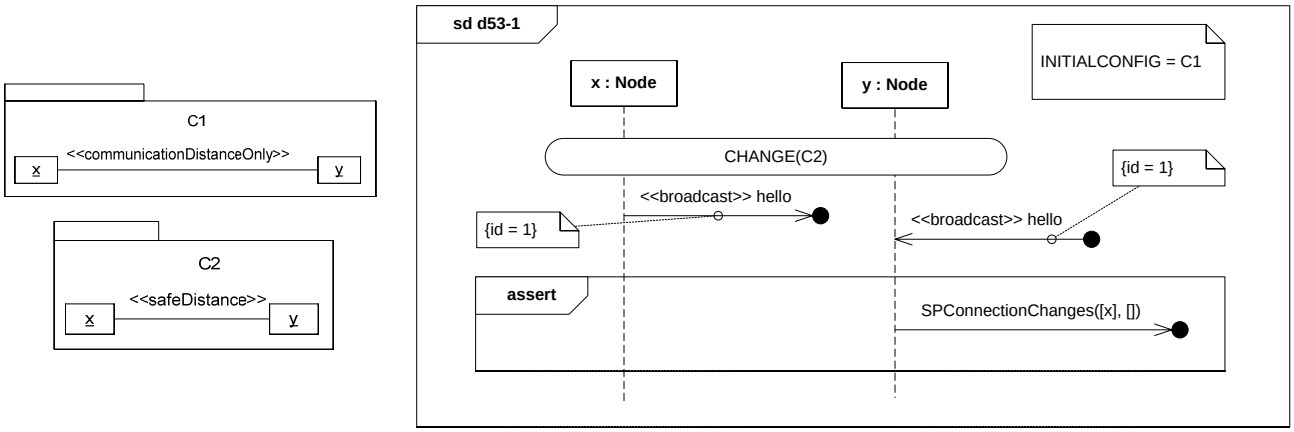
In order to better account for mobile settings, especially in the case of applications in the ad hoc domains, we propose the following extensions to existing graphical scenario languages:

- The spatial configurations of nodes have to be considered as first class concepts. They are introduced in labelled graph representations that form the spatial view of the scenario.
- The event view makes it explicit which communication event occurs in which spatial configuration, and configuration changes are introduced as global events.
- Broadcast communication in local vicinity is introduced by means of special symbols.

Figure 15 exemplifies how we defined such extensions in terms of UML elements. It represents a simple requirement scenario from a testing case study we investigated, a partitionable Group Membership Protocol (GMP) in the ad-hoc domain. In this protocol, groups split and merge according to location information carried by hello messages. Decision is based on the notion of safe distance, where the safe distance is strictly lower than the communication range. The requirement says that whenever a node – not being the leader (l2) of its current group – detects a new neighbour at a safe distance, it has to report the connection change. In the event view, note the global configuration change event, as well as the broadcast stereotype attached to the

hello message. In the spatial view, it is the responsibility of the designer to determine which relations conveniently abstract the concrete configurations for the target application. Here, the GMP behaviour is governed by two relations, being at communication range and being at a safe distance.

Other examples of scenarios for some HIDENETS use cases, namely the Platoon Driver Support Software and the Distributed Black Box, can be found in Deliverable D5.3 to illustrate the new language elements.



(a) Spatial view **(b) Event view**

Figure 15: Example of requirement scenario

6.2.4 Automated processing of scenario descriptions

In the proposed testing framework, scenario descriptions are not just for documentation. They are intended to be compiled into programs that automatically analyze execution traces. Requirements scenarios are used to check whether key properties are violated during testing. Test purposes are used to check whether desired fragments of behaviour are covered at least once during testing. Test cases need concrete contextual data (e.g., GPS coordinates) to implement the desired evolution of configurations, and the proposed solution is to extract matching data from preliminary runs of the context controller.

In each case, the identified treatments involve graph matching problems, at least for some part. This is due to the need to determine whether the physical nodes appearing in the trace can match abstract nodes appearing in the spatial views. Determining whether one graph G1 (here, coming from an abstract scenario) is matched by a subgraph of G2 (coming from a trace) can be solved by graph homomorphism building, which has been extensively studied in the literature. We use an existing facility that accommodates graphs with symbolic label variables and wildcards. The novelty of our tool consists in reasoning on sequences of graphs (i.e., sequences of spatial configurations). It complicates the matching problem, because of the need to retain a consistent valuation of variables across the sequence. Specifically, the accounting for abstract scenarios where nodes dynamically appear and disappear proved a tricky issue.

Once graph matching has determined which physical nodes can play the role of the nodes appearing in the scenario, trace analysis can proceed by comparing their communication events with the ones in the event view. This requires a well-defined semantics for the event view. As a general comment, the problems we encountered did not originated from the language extensions we proposed (broadcast communication, causal dependency on configuration change events). Rather, they came from the core UML constructs and concerned:

- The computation of a partial order of events for scenarios with non-determinism, and possibly non local choices (par, alt and opt constructs and their nesting).
- The interpretation of scenarios in terms of a partial description of behaviour, such that unrepresented events may interleave with the represented ones (ignore, consider constructs).

- The interpretation of modalities (e.g., assert construct) to determine events that may, must or must not occur, and of the nesting of modalities into other construct (e.g., meaning of an optional assert).

An overview of semantic problems can be found in [48]. The semantics we retained avoids some of these problems by syntactic restrictions (e.g., we do not allow the nesting of assert). We also made choices that depart from the usual (informal) interpretation of sequence diagrams, e.g., weak sequencing is no longer the default composition operators for language constructs. We argue that these restrictions and choices make it possible to assign a clear and unambiguous meaning to the diagrams.

6.3 Relevant publications

The testing work is presented in HIDENETS Deliverables D5.2 (Preliminary Testing Framework and Methodology) and D5.3 (Refined design and testing framework, methodology and application results).

The GMP testing case study has been published in [76], and some additional details are to be found in a research report [77]. [78] presents the justification for the mobility-related extensions to scenario languages, as well as preliminary work toward the implementation of the graph tool (see D5.3 for an up-to-date presentation).

[76] H. Waeselynck et al. “Mobile Systems from a Validation Perspective: a Case study”, Proc. of the 6th International Symposium on Parallel and Distributed Computing (ISPDC’07), IEEE CS Press, Austria, Jul. 2007.

[77] Z. Micskei, H. Waeselynck, M. D. Nguyen, and N. Riviere. “Analysis of a group membership protocol for Ad-hoc networks,” LAAS Technical Report no. 06797, November 2006.

[78] M.D. Nguyen, H. Waeselynck, N. Rivière, “Testing mobile computing applications : towards a scenario language and tools, 6th Workshop on Dynamic Analysis (WODA 2008), ACM Press, Washington D.C, USA, July 2008.

7. Proof-of-concept experimental set-up

The overall objectives of the experimental set-up during the whole HIDENETS project have been:

- To prove the feasibility and practical relevance of the HIDENETS approach, methodology and results by practical implementation of a relevant subset of the developed concepts,
- To provide an experimentation platform to test and validate the concepts developed in the projects,
- To provide practical evidence of the project results for dissemination among the standardization bodies, academic and industrial audiences. All of these issues have been addressed by means of proof-of-concept experimental set-ups. Their 'theoretic' background has been discussed elsewhere in this document; see the corresponding sections in Chapters 3.5.

One of the overall HIDENETS project goals involved the provision of architectural and design solutions concerning both network/protocol elements and technology components and their ensemble as 'middleware', required for the deployment of highly available and resilient mobility-aware services. The experimental research carried out has provided additional insight on this matter by identifying design consequences that are not present in the underlying models, allowing in this way the refinement of the initial solutions.

Another HIDENETS challenge has been to provide an implementation of the relevant parts of the design solutions to constitute a proof of concept prototype in the automotive application domain covering both ad-hoc car-to-car and car-to-infrastructure scenarios. This HIDENETS challenge has been the core objective for the prototype developments. Both the identification of which parts of the design solutions are relevant as well as the actual implementation of these components has been important for the work performed. This also involves the interfaces that have been defined in order to connect all individual hardware and software components.

Also, the prototypes have played a key role in the provision of the experimental laboratory set ups in order to perform an assessment of the dependability and QoS provided by the HIDENETS solutions. This is done through the evaluation of the selected scenarios both at model resolution level and within the experiments on the experimental laboratory set-up.

Proof-of-concept experimental laboratory set-ups have been targeted at the experimental validation of the concepts defined in the project. These laboratory implementations consist of the integration of the essential outcomes of the other HIDENETS work into several test-beds. The concepts defined in the project have been split over different test-beds and contribute to the dissemination of the aforementioned methods and tools. This will result in both awareness for high availability and resilience as foundational service qualities, as well as result in the dissemination of methods and tools for the development and deployment of highly available, resilient services.

In order to focus on essential functionalities and to show the benefits of a flexible and modular HIDENETS architecture, four focus areas have been identified, which have led to the definition of four specialised test-beds and an emulation tool:

- **Application Development test-bed**
  A test-bed showing a complex resilient application developed specifically on top of the HIDENETS solutions using model-based development methods. This application involves both the infrastructure and the ad-hoc domain.
- **Platooning test-bed**
  A platooning prototype that is used as a proof-of-concept for the ability to detect and react to timing faults, to assure safety and to handle certain malicious intrusions.

- **Distributed Black-Box test-bed**
  A distributed black-box application showing a car-to-car cooperative and secure backup scheme for critical data and a resilient store and retrieve system.
- **Resilient Communication test-bed**
  Optimised communication protocols for ad-hoc (c2c) networks and their impact on higher network layers.
- **Topology Emulation tool**
  Tool for performing reproducible experiments with dynamic topologies.

These four test-beds and the topology emulation tool will be described in the subsequent sections.

7.1 Application Development test-bed

The AD TB uses an example application in order to demonstrate (i) the benefits of applying the model driven software engineering methodology and tools developed in the context of HIDENETS "Design methodologies" and (ii) the possibilities for achieving high availability of some key infrastructural services by building them on a standards compliant SA Forum AIS (Service Availability Forum Application Interface Specification) implementation. The chosen application is a prototype Platoon Driver Support System (PDSS) that implements a simplified version of the platooning use-case, while strongly emphasising the development process itself.

7.1.1 Challenges and activities

The main challenges of the Application Development test-bed have been to exploit the benefits of the underlying HIDENETS middleware (the services developed for "Resilient Architecture and Middleware (WP2)" and "Resilient communication (WP3)") relying on the results of WP5, developing and using model based tools. We have chosen an application that involves nodes both in the ad-hoc and infrastructure domain, allowing to high light the future possibilities in the newly defined middleware services for the ad-hoc domain and in the SA Forum middleware that was selected as a solution for the infrastructure domain.

As a first step we designed the UML model of the application, going along the lines of the use-case driven nature of MDA resulting in a functionally decomposed system. As a second step, we scrutinised the HIDENETS middleware services to make use of them, highlighting their usefulness for application developers. Thirdly, we moved on to the design on the infrastructure side, using the Service Availability Forum specifications. Having the complete application model, we applied the profiles developed in HIDENETS (see chapter 5) from the underlying concepts and services. Finally, we developed and used model-based tools for the implementation, namely automatic code- and configuration generation for SA Forum compliant applications.

7.1.2 Lessons learned and guidelines

We learned how to derive UML design patterns from basic underlying dependability concepts and interfaces of HIDENETS services that are fundamental basic blocks for future application developers. Such results not only ease their burden when dealing with new, complex platforms or middle-ware. They also help in the improved understanding of the domain supported by the use-case driven nature of MDA.

Additionally, precious insight into the specifications of the Service Availability Forum could be gained, studying two different AIS implementations (OpenAIS, OpenSAF), learning their similarities and differences, and providing feedback to the standardization and implementation process.

7.1.3 Relevant publications

[102] Pintér G., Micskei Z., Kövi A., Égel Z., Kocsis I., Huszerl G. and Pataricza A.: Model-Based Approaches for Dependability in Ad-Hoc Mobile Networks and Services. In R. de Lemos, F. Di Giandomenico, C. Gacek, H. Muccini and M. Vieira (Eds.) Architecting Dependable Systems V (LNCS-5135) pp. 150-174. 2008, Springer

[117] Szatmári Z., Kövi A., and M. Reitenspiess: Applying MDA approach for the SA forum platform. In Proceedings of the 2nd Workshop on Middleware-Application interaction: Affiliated with the Discotec Federated Conferences 2008 (Oslo, Norway, June 03 - 03, 2008). MAI '08, vol. 306. ACM, New York, NY, 19-24. DOI= http://doi.acm.org/10.1145/1394272.1394278

7.2 Platooning test-bed

The platooning test-bed explores and validates the HIDENETS hybrid system architecture, along with the oracle services provided by the wormhole subsystem, in a realistic and complex car platooning scenario. It allows demonstrating: a) the feasibility of a hybrid system architecture, in particular the possibility of constructing a local wormhole with real-time properties that is connected with a general purpose payload part; b) the benefits of some of the oracle services, namely the Timely Timing Failure Detection service and the Reliable and Self-Aware Clock service; c) the benefits of some complex middleware services, namely the QoS Coverage service, the Intrusion Tolerant Agreement service and the Diagnostic and Reconfiguration Manager. In summary, the Platooning scenario allows to illustrate how solutions developed HIDENETS contribute to achieve dependable system behaviour, most notably in the presence of timing faults.

7.2.1 Challenges and activities

The initial challenge, although not specifically related to this particular test-bed, was concerned with the definition of the architecture and the several oracle and middleware services. Regarding the test-bed in particular, the main challenges in a first step concerned the concrete definition of the platooning scenario, namely its actors, how they should interact, the relevant physical details to be considered, the required properties and the considered fault model. In a second step it was necessary to define the concrete architecture of the platooning application, taking into account the required properties and the fault model, and selecting the services to be included in order to address all the requirements with improved dependability.

Regarding the implementation aspects of the platooning test-bed, it was necessary to: a) implement the several oracle and middleware services; b) implement the platooning application; c) integrate all the parts, solving interface issues among simulated and real components. Given that the challenges concerning the implementation of the several services considered in HIDENETS was addressed in deliverables D2.3 and D3.3, we just refer, in the following sections, to the remaining challenges.

7.2.1.1 Platooning application

The platooning application consists of software that runs embedded in a car, and which monitors and controls that car. As such, a major challenge of the platooning test-bed was to devise and design an algorithm which performed the central task of driving the car in a platoon. This is a difficult task because it requires reasoning about the timing of physical events, the timing of the application, and the physics of cars, and all of these factors depend on each other. The use of the duration measurement, timely timing failure detection

and reliable and self-aware clock oracle services eased this challenge, by enforcing well-defined, clear interfaces for the use of time.

Another difficulty found in the design of the algorithm was how to structure it according to the HIDENETS hybrid architecture. The design must comprise the division between the asynchronous payload and a synchronous wormhole, be able to maintain safety in case of payload timing failures and be able to correctly recover when the payload becomes timely again. Solving this challenge showed how the timely timing failure detection oracle service is a fundamental building block for designing algorithms using the HIDENETS hybrid architecture.

7.2.1.2 System interfaces

The platooning test-bed has a diverse number of components, with differing degrees of resources such as computational power, memory, floating-point hardware and with diverse support of programming languages. This creates a challenge of interoperability at the subsystem boundaries.

7.2.1.3 Physics simulation

The platooning application must include a model of the car physics, since its output depends on the way the real world can be expected to behave. This requires determining which physical behaviour is to be expected, which responses are acceptable and what formulas and constants model the evolution of the system best. This proved to be a challenge, but it also highlighted the real-world benefits of the hybrid system architecture and the oracle services, complementing more synthetic benchmarks.

7.2.2 Conclusions and lesson learned

One of the oracle services of the wormhole provides the ability to detect timing failures. To use this as part of an application proved to be tricky, since applying the service to an algorithm uncovered some pitfalls.

The algorithm used in the platooning test-bed is periodical, and each period starts a new round, where the algorithm collects information about the surrounding system and must produce a control decision before that round’s time slice expires. One lesson learned was that there must be two different kinds of rounds: the ones without a deadline and the ones including one. This should be taken into account when designing new algorithms, so that they easily fit to this duality.

Another lesson learned was how to keep the periods and their respective deadlines in sync. Care must be taken because some of the obvious solutions are problematic in the studied scenarios, where the (asynchronous) payload sets deadlines for itself. We concluded that there are several possible correct solutions. The one adopted in the platooning test-bed was for the payload to request the wormhole the current time, compute from that value which iteration it should be processing and set a deadline for the time at which that iteration should finish.

7.2.3 Relevant publications

[91] Hugo Ortiz, António Casimiro and Paulo Veríssimo, Architecture and Implementation of an Embedded Wormhole, In Proceedings of the 2007 Symposium on Industrial Embedded Systems (SIES'07), Lisbon, Portugal, July 2007.

[92] António Casimiro, Odorico Mendizabal and Paulo Veríssimo, On the development of dependable embedded applications using specialised wormholes, 3rd International Workshop on Dependable Embedded Systems (WDES'06), Leeds, UK, October 2006.

[104] Luís Marques, António Casimiro and Paulo Veríssimo, Proof-of-concept Platooning Application Using the HIDENETS Architecture, The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'09), to be submitted.

7.3 Distributed Black-Box test-bed

7.3.1 Challenges and activities

The distributed black box test bed allowed us to show the feasibility and benefits of implementing dependable cooperative backup applications based on three HIDENETS middleware services: 1) Proximity Map, 2) cooperative backup and 3) trust & cooperation. We have mainly focused on the first two services, considering a lightweight implementation of the trust & cooperation services. One major challenge that has been faced during the development of the test bed was to investigate the possibility to validate these services in a laboratory set up that emulates at a reduced scale more complex car-to-car systems.

In our laboratory-scale set-up, cars run in a room of 14m* 7m. When compared to dimensions of the targeted system, i.e., a car accident that occurs in a zone of less than a kilometre, it represents a change of scale of about 50 in length. This led us to:

- Reduce the range of wireless communication, in order to have wireless communication only for nodes within 2 meters,
- Deploy a precise indoor localisation system, capable of attaining a centi-metric precision,
- Develop reduced-size cars, capable of carrying a laptop with its equipment at a maximum speed of about 1 meter per second.

7.3.2 Conclusions and lesson learned

Our results are promising. We successfully managed to build a full system, with a scale reduction of about 50, that runs 4 cars equipped with all devices to be found in next generation cars, everything being reduced in scale (positioning system, speed of vehicles, wireless communication, and, of course, the vehicles themselves). These results should be applicable to many other systems, since we put a particular care in being able to change the scale factor of every part of our emulator to fit almost any need of evaluation for a mobile system. Also, our test-bed allowed us to show how an application can use the many cars that compose the car-to-car system to build a very reliable storage at a relatively low cost. When compared to avionics black boxes, our system is not only cheaper, but is also able to provide different information, such as other cars' information, due to the distributed nature of the application.

Another lesson learned from our research was on the localization system. We started to use the cricket system, developed by at MIT, since it is an indoor version of a satellite-based GPS-like system: a constellation of geo-localised devices periodically send signals, and receivers compute their position by measuring the different signals received. However, this technology was not precise enough in the case of HIDENETS: the centi-metric precision is attained only when the receiver does not move.

In the case of moving vehicles, we investigated the use of the Evart system, initially developed for motion capture. Our experimentations showed that the Evart system, although technologically very different from a GPS system, is the best choice for our test-bed, since it has a centi-metric precision whatever the motion of the vehicle is, and it can also provide precise speed and direction measures.

7.3.3 Relevant publications

[105] M-O. Killijian, N. Rivière, M. Roy. Experimental evaluation of resilience for ubiquitous mobile systems. Workshop on Ubiquitous Systems Evaluation (USE), UbiComp 2007, Innsbruck (Autriche), Sept 16-19 2007, pp.283-287.

[106] M-O. Killijian, D. Powell, M. Roy, G. Séverac. Experimental Evaluation of Ubiquitous Systems. Why and how to reduce WiFi communication range. DEBS 2008 (2nd International Conference on Distributed Event-Based Systems). July 2008, Rome.

7.4 Resilient communication test-bed

7.4.1 Challenges and activities

The main challenges of the Resilient Communication test-bed are two-fold. First of all, a proof-of-concept has been provided that shows the major communication-related results of HIDENETS. Next challenge has been to provide a platform to allow the experimental research and evaluation of communication-related HIDENETS concepts.

More specifically, these challenges entail the following:

- Dependability improvements of using a multi-channel / multi-radio architecture in an ad-hoc mesh network,
- Testing of the communication protocols developed with respect to dependability,
- Testing of custom made distributed service replication middleware in the ad-hoc domain,
- Providing an emulation test-bed for wireless applications.

In order to address these challenges, the following activities have been implemented:

- Implementation and evaluation of
  - A multi-radio mesh node and a multi-radio ad-hoc mesh network,
  - Fast reroute algorithm,
  - Replication Manager.
- Development and implementation of the topology emulation tool. This tool is described in section 7.5.

7.4.2 Lessons learned and guidelines

- **Multi-channel multi-radio architecture**:
  The HIDENETS multi-radio implementation is done by inserting an intermediate layer between the radio interfaces and the network layer. The advantage of this implementation is a standard interface which can be combined with any upper layer protocols that support Ethernet-type MAC layers. The experimental setup has shown that this implementation is easy to implement and using multi-radio multi-channel increases network capacity. Although the capacity is increased, the gain is lower than expected due to node-internal interference between multiple radios. This interference should be taken into account both when designing hardware and software for multi-radio systems to minimise these effects. This could be handled by e.g. modifying the distance between the antennas as well as antenna characteristics, orientation, geometry, isolation between radios, as well as assigned frequencies. Considering the general scope of HIDENETS, the multi-channel multi-radio setting improves the dependability in multi-hop ad-hoc communication. This benefit is in particular present in more dense networks. The Car Accident use case has provided a setting for this investigation.

Besides multi-radio multi-channel, the experimental platform used for this also included a gateway to infrastructure that can be used by any mesh node.

- **Fast reroute**:
  This is implemented as an extension to the OLSR routing protocol and will be activated as long as OLSR is not able to discover a route to a destination. The Fast Reroute mechanism is triggered by a link-failure detection mechanism, and is able to set up a new route faster than OLSR with default parameters, thus providing HIDENETS with a failover routing mechanism. The performance of the Fast Reroute scheme depends on the link-failure detection accuracy and reactivity. Currently the link-failure detection is performed using beacon messages, and the frequency of these decides the reactivity of the detector. The beacon messages are vulnerable to collisions with traffic from hidden nodes, so the accuracy of the detector is a function of the topology and the traffic load. In order to improve the detector, the RSSI values can be used as an additional indicator of a failed link in conjunction with beacon messages. However, the RSSI value is card vendor specific, so using this is not an optimal solution.
- **Replication Manager:**
  We consider the use case of Assisted Transportation, and the applications involved in this scenario, namely the Traffic Flow Control application, to explain the impact of the component on the resilience of the overall system. The Replication Manager (RM) is replicating the accumulated traffic information to other information hubs. In a scenario where some cars in the network announce themselves as information providers to others, the others would depend on them. Therefore the information providers would receive information input, process it and replicate the digested information to other cars. The Replication Manager is selecting its replicas in such a way that all users of the service will receive the best possible information. The underlying dynamics of the network make it challenging and sometimes impossible to accommodate the requirements of all the users of the system. In case the users are too far away from the information source, they will have to change their preferred information source or the communication has to continue using cellular communication technologies. The threshold for the change in communication technology is depending on the freshness requirements from the application.

7.4.3 Relevant publications

[40] Manfred Reitenspieß et. al., “Experimental proof-of-concept set-up HIDENETS”, EU FP6 IST project HIDENETS, deliverable D6.3, June 2008.

[41] Z. Egel et. al., “Documentation and Evaluation of the experimental work”, EU FP6 IST project HIDENETS, deliverable D6.4, December 2008.

7.5 Topology Emulator tool

7.5.1 Challenges and activities

The main general challenge addressed by the Topology Emulator is to provide an evaluation environment for the applications and prototypes developed within HIDENETS. These applications all handle the challenges of running in unreliable environments, and in order to evaluate the performance of these applications, we need to create such unreliable environments, and even do it reliably during evaluation.

To do emulation, the Topology Emulator intercepts and forwards packets based on simulated network properties and these tasks must be carried out in due time and transparent to the sender and receiver of the packets.

Specifically, the design of the Topology Emulator addresses performance and scalability challenges itself. Hard real-time constraints govern the execution to emulate network properties in real-time and transparent to applications. Moreover, to use it in a real evaluation setting, the emulator must support connecting tens of nodes and emulating hundreds of links.

7.5.2 Conclusions and lesson learned

The Topology Emulator is a useful tool for a subset of applications in HIDENETS. Simulating the dynamic network properties from node mobility is handled in the emulator. Affecting the node mobility from the network properties is not currently handled by the emulator. This means that while applications such as access to medical data and a distributed black-box are suitable for evaluation, platooning and assisted driving are not, as the latter applications affect the movement of a node based on other nodes' movements and the network properties.

Thus, to successfully use the emulator in its current state, the application must not expect to affect the movement of the node it resides in. Integration of such a feature is currently under investigation in the derived open-source project called "Air-in-a-box" [30].

7.5.3 Relevant publications

The original work on the Topology Emulator is described in [28]. This work is published in its original form in [29]. Moreover, the project continues as an open-source project [30], where the developers are addressing some of the limitations of the original work.

[28] N. Jensen, Morten ; Nickelsen, Anders, "Evaluation of Routing Dependability in MANETs using a Topology Emulator ", Elektronik og IT, Kandidatuddannelsen (Spec. Distribuerede Systemer), 2007

[29] Nickelsen, Anders ; Jensen, Morten N.; Matthiesen, Erling Vestergaard ; Schwefel, Hans-Peter, "Scalable emulation of dynamic multi-hop topologies.", Proceedings of ICWMC 2008.

[30] http://air-in-a-box.sourceforge.net

8. Generalization aspects

8.1 HIDENETS contributions to standards

Over the period of the project HIDENETS results have been presented, contributed and synchronised with other research projects like COOPERS [66] and standardization bodies like the SA Forum [64] and the Car 2 Car Communication Consortium [65].

8.1.1 Service Availability Forum

The SA Forum is a consortium of industry-leading communications and computing companies working together to develop and publish high availability and management software interface specifications. The SA Forum then promotes and facilitates specification adoption by the industry.

The SA Forum is unifying functionality to deliver a consistent set of interfaces, thus enabling consistency for application developers and network architects alike. This means significantly greater reuse and a much quicker turn around for new product introduction.

The SA Forum's mission is to foster an ecosystem that enables the use of commercial off-the-shelf building blocks in the creation of high availability network infrastructure products, systems and services.

From the outset, interaction with the SA Forum was seen as key for the success of the HIDENETS project for a number of reasons. The most important are

- SA Forum requirements were input for establishing HIDENETS research and work topics.
- SA Forum programming interfaces were referenced in the HIDENETS architecture to leverage existing industrial state of the art technology.
- At the same time, it was assured that the HIDENETS architecture and project results were in synch with ongoing industrial projects and could fit to industrial standards (w.r.t. architecture or programming interfaces) or enhanced for industrial use.
- A number of HIDENETS results were achieved with the clear vision in mind that they respond to important requirements as established by SA Forum member companies or which have evolved in the course of the SA Forum interface specification work.
- Last but not least, HIDENETS expertise was made available to enhance SA Forum specification work.

To achieve the challenging goals mentioned above, Fujitsu Siemens Computers (FSC) and Budapest University of Technology and Economics (BUTE) were members of the SA Forum. Both HIDENETS partners participated regularly in SA Forum face-to-face membership meetings and leadership meetings. FSC and BUTE hosted SA Forum leadership meetings as well as membership meetings in the course of the project. HIDENETS research results were regularly presented during the f2f meetings.

8.1.2 Car 2 Car Communication Consortium

The Car-2-Car Communication Consortium (C2C-CC) is the central standardization organisation for car-to-car and car-to-infrastructure communication. It is mainly organised by the automotive industry, but also

ETSI and national regulatory authorities as well as a set of research institutions are contributing to the efforts. Carmeq is a subsidiary of Volkswagen, one of the nine C2C-CC partners. In the course of the HIDENETS project, Carmeq has provided input to Volkswagen in regular meetings with Volkswagen working group chairs of the consortium. With Gerard Segarra, Renault, a major contributor to the C2C-CC is a member of the HIDENETS Advisory Board.

A huge amount of stakeholder requirements needs to be taken into account due to the large number of C2C-CC partners and associated members. The establishment of a unified European frequency band took top priority at C2C-CC. As a result, the standardization process is slow-going and main research activities of the consortium were related to physical and link layer challenges. Therefore it was a key task for HIDENETS to attract and inform the consortium on ongoing research topics like enhanced middleware services or the development of sophisticated node architecture and on other suitable dependability means, which have been investigated within HIDENETS.

The HIDENETS node architecture has been presented to the C2C-CC architecture working group in February 2008. A cooperative link has been established to the COMeSafety Project [67] in spring 2008. COMeSafety is responsible for evaluating project results of projects not integrated in the C2CCC and for making recommendations on their importance for the consortium. The link to COMeSafety is maintained on a continuous basis. A key result of the cooperative link is the publication of a HIDENETS article in the COMeSafety newsletter [71] which addresses the wider C2C-CC audience.

The HIDENETS node architecture, being one of the most sophisticated architectures in the car to car communication domain, will be input to a later version of the COMeSafety architecture document [73]. This architecture document serves as a basis for decision-making in the C2C-CC as well as for other new projects derived from the consortium like PRE-DRIVE C2X[72].

As a result of the regular meetings with Volkswagen experts, HIDENETS was able to support the physical layer working group of the C2C-CC. HIDENETS results influenced the decision on the specification of the on-board car equipment. Results on a development of a multi-radio multi-channel solution were submitted to the working group as input and driver of the ongoing discussion.

In the second half of 2008, HIDENETS results were presented to the security working group of Volkswagen Research responsible for car-to-car communication. The HIDENETS business impact analysis has been attracted by a business team of Carmeq. Represented by Carmeq, HIDENETS participates the Car-2-Car Communication Consortium forum as well as the demonstration of the first car-to-car prototypes at the OPEL test centre. Furthermore Carmeq generated a condensed internal report about HIDENETS to Volkswagen to underline the opportunities of the results in the car to x domain.

8.1.3 Relation to other research projects

A number of other dissemination activities have been undertaken such as participation at industry events, workshops, panels, poster sessions, and standards groups. In particular, HIDENETS members participated in the following activities:

- **MOSAIC: Mobile System Availability Integrity and Confidentiality** (French National project, partners LAAS-CNRS (coordinator) - IRISA and Institute Eurecom: http://www.laas.fr/mosaic)

This project investigated different approaches for designing secure and dependable cooperative backup services based on mobile ad-hoc systems. The solutions developed in HIDENETS, in the context of the distributed black-box application, result from the extension and generalisation of the preliminary approaches investigated in the context of MOSAIC. IRISA focused on new paradigms for spontaneous communications in the context of ubiquitous computing and ad hoc systems. Institute Eurecom mainly

investigated trust and cooperation mechanisms, whereas LAAS addressed fault tolerance approaches based on fragmentation, redundancy and scattering.

- **German National Project SIM-TD** (Secure Intelligent Mobility - Testfield Deutschland)

The SIM-TD project is targeted on traffic safety and traffic efficiency by using both WLAN-based ad hoc Car2Car and Car2Infrastructure technologies (RSU, GSM, UMTS). SIM-TD will progress large scale field tests with up to 300 test vehicles to validate the feasibility of the used communication technologies and the impact on realistic traffic scenarios. Already existing prototype solutions will be combined to compile a reliable and secure overall system architecture. The evaluation of relevant business cases is a major topic beside technical aspects. HIDENETS results have been presented to Volkswagen Research at the early stage of SIM-TD to support architectural design decisions.

- **FP7 PRE-DRIVE C2X**: (PREparation for DRIVing implementation and Evaluation of C-2-X communication technology)

PRE-DRIVE C2X is focused on development and system specification of communication architectures on C2C and C2I. It is motivated to prepare simulations and European field tests. An aggregated HIDENETS document attracted interest by Volkswagen and has been provided to relevant PRE-DRIVE partners. The document describes HIDENETS' hybrid node architecture and its features and supports future design decision for the PRE-DRIVE C2X architecture.

- **Coopers** (FP6 Integrated Project Cooperative Systems for Intelligent Road Safety)

Coopers focuses on the definition and realization of car to infrastructure (C2I) applications, the on-board unit development, and experimental field-tests to study the feasibility of C2I applications. Consideration of dependability aspects are not an explicit part of COOPERS, hence the HIDENETS C2I solutions are complementing the COOPERS technical scope. Members of the HIDENETS team have had intensive contact to the COOPERS consortium, where in several individual talks and in a joint workshop combined interests and project interaction options were identified.

- **SeVeCom** (FP6 Integrated Project Secure Vehicle Communication)

While a large part of the HIDENETS solutions is motivated by prevention of and tolerance to accidental faults, the explicit analysis and development of schemes based on cryptographic methods for privacy and malicious intrusions is focus of the SeVeCom project. HIDENETS and SeVeCom exchanged their initial solution approaches in a joint workshop in 2006 and subsequently had interactions on individual researcher level. The actual integration of the dependability solutions of HIDENETS and the security/privacy approaches of SeVeCom is part of future work beyond the scope of HIDENETS.

- **SAFESPOT** (FP6 Integrated Project Cooperative Systems for Road Safety)

The objective of SAFESPOT is to design cooperative systems based on vehicle-to-vehicle and vehicle-to -infrastructure communication to improve road safety. The solutions investigated to prevent road accidents are based on the development of a “safety margin assistant” to detect in advance potentially dangerous situations. HIDENETS have a more general scope than SAFESPOT in terms of: i) the applications considered, and ii) the system layers providing dependability related services. In particular, SAFESPOT focuses on the application level and on communication and sensing technologies, and does not address middleware layer dependability related services. Moreover, this project does not really focus on fault tolerance approaches, including mechanisms for the detection and recovery of errors.

8.2 Dependability process

From a more abstract perspective the HIDENETS project combines two ambitious challenges: the development of car-to-car communication solutions and the development of quantitative evaluation techniques to predict dependability properties. The first one has to cope with new requirements derived from high dynamicity and uncertainties caused by environmental influences but also involves runtime support, application development and prototyping, while the second has to handle the very high complexity of the system by means of analytical, simulative and experimental evaluation techniques to improve the prediction of a system's dependability properties and to identify failures as early as possible.

The overall rationale of HIDENETS is to contribute solutions to the scientific community and industry. Especially for industrial use it is recommended to derive generalised knowledge from the project's experiences into a process description. Therefore the following section describes generalised HIDENETS knowledge as a dependability process description.

In industry, the application of specific processes and engineering standards has been established a long time ago and a wide range of standards and guidelines have been approved. The use of standardised processes improves product properties and reduces costs caused during a product's life cycle. In relation to the field of functional safety IEC 61508 can be mentioned as a generic process description for functional safety which again is adapted in the ISO WD 26262 standard for automotive safety engineering [69]. Much work has been done in CoBiT which is a framework for IT governance. CoBiT [68] integrates different standards like ISO 9000 in quality management, BS7799 in information security, and ITIL[70] in services management. Due to the specialty focus of the mentioned standards they do not meet directly the requirements in context of dependable, highly dynamic mobile ad hoc networks. These standards are neither formed to meet a holistic dependability oriented development nor a continuous dependability control. A second adverse aspect of the mentioned standards is reasoned by focusing on products with a relatively high degree of its technical maturity. Especially the ISO WD 26262 assumes comprehensive, proven and tested specifications about the system in question. Therefore, it does not cover a product's life cycle during the research phase with its feasibility studies and does not support an efficient transfer in between a company's organisational units like research/predevelopment and (serial) development.

Figure 16 illustrates a specific dependability process derived from the HIDENETS activities which can be reused for similar projects in research and development. This process description is focused on the early stages of a product's life cycle which is the most crucial phase of a development process in terms of dependability and subsequent economic success. Therefore, the process description does not focus on a holistic process description of a product's life cycle. Later process stages like serial development, after SOP (Start Of Production) and maintenance are not covered. Especially, the development of very complex systems need to focus on dependability means at an early stage. Being a research project is an advantage of HIDENETS as it concentrates on this crucial stage and also deals with car-to-car communication solutions as a very complex system. Therefore this process description is not focused on project management guidelines, already established in other standards, but can be seen as a helpful, concrete technical process description, directly applicable to industrial research and predevelopment. This process is targeted to support the development of dependable systems being inherently complex. The term dependable systems means in this context software as application but also embedded software. For HIDENETS it can be named road traffic related WLAN-based communication.

As a starting point of the process, naturally, there is an initial event (e.g. a new idea, arising technologies and related needs, relaxed regulatory requirements), which facilitates new possibilities and provokes ideas where and how to apply them. The initial event within HIDENETS is the basic idea that cars communicate with each another as well as to infrastructure. In a first step a bundle of different new services need to be identified. The aim of the services should be clearly mentioned as well as its basic functionalities and their assumed top level dependability requirements. Afterwards possible use cases need to be described involving a specific one or a set of services. A use case description lists all services, actors and their roles, and other

interacting systems involved in, plus a common explanation of the dependability challenges assumed from this early stage of the process. Within HIDENETS 17 services/applications and 6 use cases have been depicted.

Thereafter a user oriented dependability requirement analysis need to be processed to verify the dependability requirements mentioned in the service and use case descriptions. A user oriented dependability requirement analysis is required to identify a user's expectations towards the service in question. The use cases and their interacting services need to be evaluated by means of demonstrators and prototypes. The development of such demonstrators can be supported by UML use case models derived from the use case descriptions. The analysis can be used to identify hidden problems in terms of usability and unforeseen failures. It needs to be clarified which aspects of a service are useful or not, is there a situation which has not been covered by the use case, does a service disturb the driver or passengers, what kind of degraded service would be acceptable. Concrete measurements can help to refine the functional specifications by determining the required accuracy of service in terms of value and time. Further on, such an analysis should identify a user's perceived degree of rejections in case of incorrect service. In addition, the user oriented requirement analysis aims to determine and describe validation criteria from the user expectations. There is no unique measure which fits always best as a dependability criterion. This decision needs to be met individually for each system. In general, measures will be given most likely in terms of probabilities of a certain event per time like for instance the SILs (Safety Integrity Levels) of the IEC 61508 standard which defines the occurrence of a fatality per hour associated by defined ranges of probabilities to each level (SIL 2 is associated by a probability of $\geq 10^{-7}$ to $<10^{-6}$). Each SIL is recommended by a set of specific engineering means. However, probabilities can also be linked to demands, driven kilometres, or specific environmental/traffic situations.

This third step is highly recommended and marks a prominent difference to safety related processes. The identification and specification of safety requirements is requiring engineers and safety specialists familiar with the intended system. There is no need of a strong involvement of end customers.

The results of this user oriented analysis can lead to changes, extensions and refinements of the use case and service descriptions towards functional specifications and therefore demand further iteration loops of these 3 steps as often as needed.

As illustrated in Figure 16, a system description will be performed in parallel to these 3 steps. The system description integrates non-functional and functional specifications. As complete as required all relevant non-functional requirements need to be collected that can affect the (new) system. Scope and depth of this requirement analysis depend on the stage of maturity of the project. A deepened focus to these non-functional requirements is out of scope for HIDENETS but some can be named like economic project limitations likely on budget and project time, technical prerequisites like the use of COTS, and regulatory requirements but also costs and technical properties of necessary hardware equipment and interface constraints to other connected systems. These requirements flow into the system description. Again, scope and depth depend on the stage of maturity of the project. The objective of this description is to explain the intended system to reach an adequate understanding about its functionalities, interfaces, environmental conditions, legal requirements, and potential hazards and failures. Both, system description and service/use case description influence one another. Some aspects of the system description imply constraints for the services in mind and vice versa some services can extend the system description if detected as necessary in the user oriented requirement analysis. So, the conceptual phase of the process consists of 2 iteration loops which influence each other.

Refined system specifications and user oriented top level dependability requirements are outcomes of the conceptual process phase. The system specifications will flow into the subsequent development phase as input for the system modelling while the top level dependability requirements will be used as validation criteria at the end of the process.

The development phase starts by modelling the system at a high level of abstractions. Depending on the description of the future system and the already processed architecture decisions, different means of analytical formalisms fit best for e.g. fault and event tree, Markov chain and process, time and Petri net. Different dependability models of the system can be constructed to cover fault tolerance and redundancy schemes. This step is a starting point in constructing an overall conceptual model of the system which will be

decomposed into conceptual sub-models. Within HIDENETS a top-down abstraction based decomposition approach has been performed to cope with the complexity of the system. Figure 16 depicts that this decomposition is based on four different abstraction levels and the interplay between these levels need to be modelled to guarantee a realistic simulation of the real system. The modelling technique can be chosen individually as it fits best for each specific level.

Afterwards, modelling at each level requires the creation of conceptual sub-models for all included components. In other words, this approach divides a problem into sub-problems. This can be done by means of analytical formalisms and simulation techniques. In a next step, some of the sub-models will be implemented. These implementations can be used for experimental and simulative evaluations. This way conceptual modelling will be supported by experimental measurements and vice versa simulation can be used to generate input data including faults for experimentations. Especially the generation of fault data by means of fault models can strongly improve the quality of experimental measurements in terms of dependability prediction. In general, the measures of interest achieved by evaluation can be used as input for other evaluations of other components at the same abstraction level. Conceptual models and implemented code will be refined by each iteration loop. As mentioned, the interplay of conceptual models and implemented code can be chosen individually for each abstraction level and component. Therefore, if of benefit, a bottom-up approach can be processed in parallel to take advantage of implemented code which was tested properly or proven in use. In HIDENETS, implemented code was reused at communication level to perform experimental measurements and evaluations. That means an implementation does not need to be based on previous conceptual modelling but can influence the modelling based on the experimental results.

The entire development phase leads to nearly faultless code and realistic models so that functionality and behaviour of the specific abstraction level can be well predicted in terms of fault tolerance and failure occurrence. The achieved measures of interest will serve as input for neighbouring levels of abstraction to support the same design concept at this level towards a holistic system model which can be evaluated based on conceptual models and implemented code.

As illustrated, the measurements of the holistic system evaluation will be compared with the top level dependability requirements derived from the user expectations. If the results of the system evaluation do not satisfy the dependability requirements, a next iteration loop needs to be performed which again starts by the step of modelling the system. Depending on the holistic system evaluation the system needs to be slightly modified or requires adding new components. Typically new components are supposed to perform needed dependability measures like fault prevention and tolerance or fault removal. In case of a successful matching of the system evaluation a system has been developed which is predicted to act properly in terms of the expected dependability requirements.

This dependability process has been generated from the HIDENETS experiences and is focused on an efficient use of a combination of engineering languages and analysis languages.

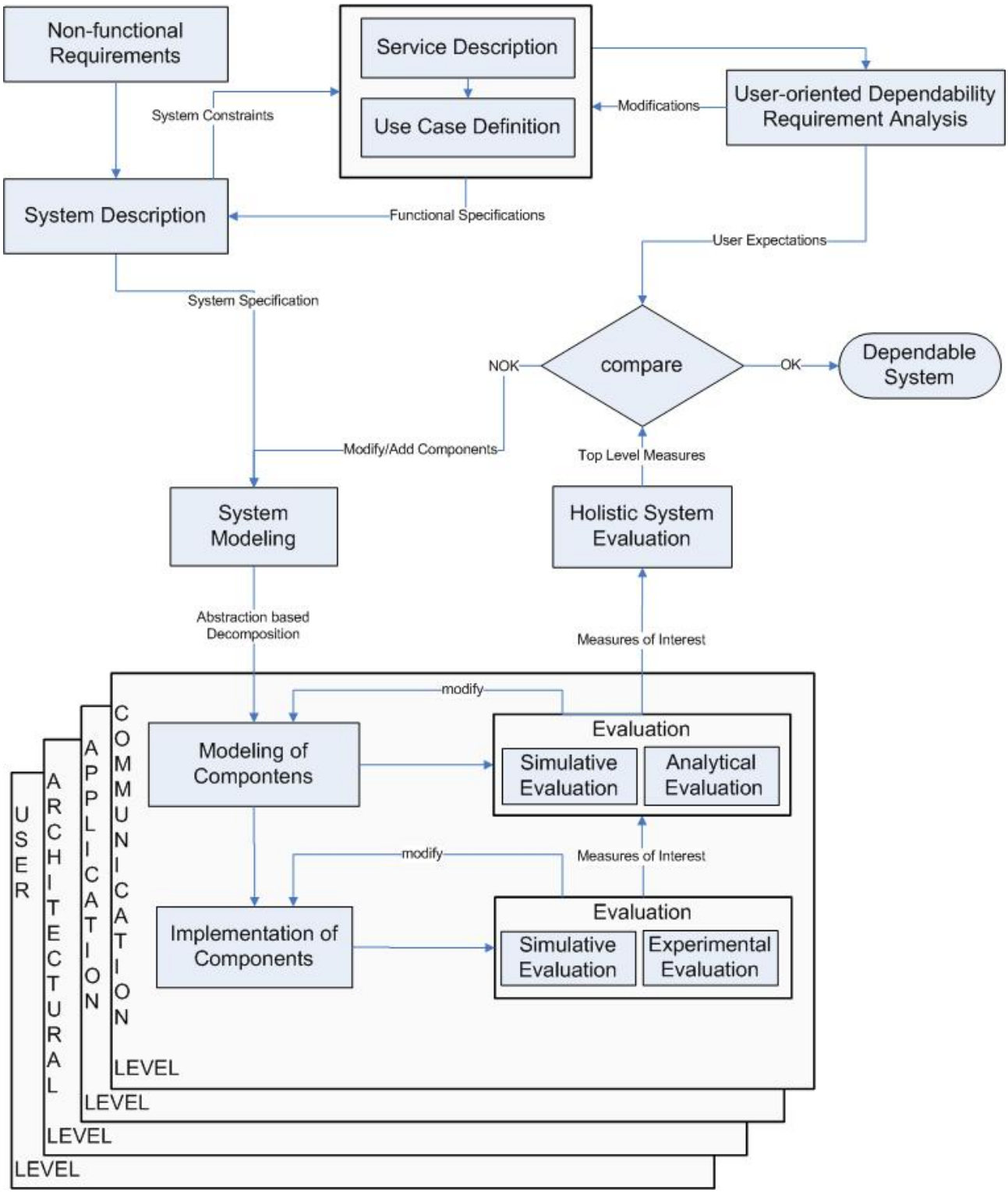


Figure 16: HIDENETS dependability process

9. Outlook

After introducing the results and lessons learned of HIDENETS activities and presenting generalised aspects of the HIDENETS knowledge in a condensed way, this chapter gives an overview of ideas where and how to adapt HIDENETS solutions and which new research issues we recommend to further investigation.

9.1 Relevance for other application fields and networking scenarios

The need for dependability solutions will continuously advance due to the overall pervasive introduction of distributed computing systems in almost all areas of life. Therefore HIDENETS solutions can and will be adapted to various domains and application fields in our all-day life. Typical domains are e.g. remote health monitoring public transportation systems, the financial sector, embedded system like train control systems and public safety and disaster relief. This section describes some examples of where to apply these solutions.

9.1.1 Public safety and disaster relief

The results achieved in HIDENETS may be of relevance for public safety and crisis management applications. Dependability of services and reliable and efficient communications are crucial for crisis management in man-made or natural disasters.

Traditional systems have shown major limitations in disasters as 9/11, hurricane Katrina, and the bombing in the London metro. One of the major problems had to do with the strong reliance on services offered by the infrastructure, and in particular on terrestrial communication facilities. The operation of the first responders was severely hampered when the infrastructure was damaged or overloaded.

These problems have been recognised and large initiates have been launched to address the shortcomings of current public safety and disaster recovery systems. Examples are the US SAFECOM project and the North American/European MESA programme that, amongst many other results, have proposed an architecture and a set of requirements for next generation systems. The architecture reflects the hierarchy of the command and control structure and contains infrastructure as well as ad-hoc components. Requirements such as high availability, security, mobility support, robustness, and quality of service are considered essential. The architecture as well as the requirements have a lot in common with HIDENETS and it is very likely that this field could largely benefit from the results of the project.

In particular, the results of HIDENETS on multi-homing, multi-radio communication, distributed back box, and replication servers are very relevant for this application area. It is worth mentioning, that domain specific issues, as the use of dedicated radio technologies, different mobility models, connectivity patterns that reflect the command hierarchy, strong security requirements, prioritizing of traffic, and different applications require modifications and extensions of the HIDENTS results before being directly applicable to this area.

9.1.2 Car-to-home and car-to-mobile device

HIDENETS solutions are developed for car-to-car and car-to-infrastructure scenarios and can be adapted to car-to-home and car-to-mobile device scenarios which are scenario subsets of the overall car-to-x domain.

Similar to HIDENETS scenarios car-to-home scenarios deal with up and downloads of documents, multimedia streaming and data synchronization of navigation and remote control information. In an exemplary situation a user prepares a longer trip and uploads helpful documents from its home computer to its car like navigation maps, trip routes or audio books.

Other car-to-home use cases discussed in the automotive domain overlap with car-to-infrastructure use cases and are related to applications and needs which are mainly in usage during a trip like instant messaging.

More abstract, from a communication perspective car-to-home is focused on connecting a driver's home infrastructure with his/her car. This can be realised through a connection between the car and a standard WLAN router placed at home, via UMTS, or car-to-RSU communication. The selection of the most appropriate technology depends on the use case scenario in question, the on-board equipment, the resulting communication costs, and the distance between home and car. These conditions can be integrated to the always-best-connected principle of HIDENETS. Most of the use cases can be processed over a one or multi hop connection while the car is parking nearby in the neighbourhood. Hereby, HIDENETS solutions like resilience routing and fast IP rerouting are suitable to be adapted to this particular situation in which passing vehicles can be used for packet forwarding. In conclusion, car-to-home scenarios demand the dependability solutions developed in HIDENETS and additionally focus more on privacy and security requirements.

The second mentioned scenario subset is called car-to-mobile device in which e.g. laptops or mobile phones placed in the car are connected to the car. These devices can be connected via Bluetooth, USB, or Firewire to e.g. take advantage of the on-board communication technologies. For instance, passengers could use applications on their laptop which need to be online and therefore utilise fully the HIDENETS ABC principle integrated in the car. This way the car is a mobile router for external applications and devices. Moreover, these external applications can use the HIDENETS middleware and wormholes services (e.g. via SA Forum interfaces).

9.1.3 Trustworthy network infrastructures

HIDENETS solutions will be relevant when developing solutions for Future Internet[11] with focus on trustworthy network infrastructures. Methods have been studied within HIDENETS for resilience when connecting to infrastructure networks with the aim of being Always Best Connected. Further work is required for smart access point selection and mobility management.

Solutions that support communication resilience based on wireless network diversity will be of profound importance for operators in the future. Terminals should be enabled to access the Internet anytime and everywhere, using a variety of applications, and should have the possibility for swapping between different networks and using different technologies depending on application demands, cost etc. Due to the inherent unreliability and continuously changing conditions, resilient solutions are sought for. Resilience and QoS differentiation is necessary to prioritise critical applications when parts of the network are exposed to failure.

Also cross-layer design to handle the heterogeneity and dynamics associated with such network scenarios will be of importance. In this context protocol design (new protocols or modification of existing protocols)

[11] Future Internet is a general term for work addressing solutions for a next generation Internet

from the link level to the transport level are of special interest, and also design of application specific architectures and middleware solutions.

9.2 Discussion on adaptation opportunities of HIDENETS oracles

Given that with current existing technologies for wireless communication it is not possible, or not adequate, to assume synchronous properties for general purpose communication channels, we have not considered a distributed implementation of the TTFD service. In fact there exist algorithmic solutions for distributed versions of this service that could be used in a practical implementation. However, the effectiveness of the solutions would be tied to the coverage of the synchronicity properties which, as mentioned, is not as good as it would be required, for instance, for the implementation of safety-critical applications.

We obviously consider this an open research issue and as far as we know, despite considerable research work in the last few years concerned with the development of predictable solutions for Medium Access Protocols (MAC), effective solutions do not yet exist.

Given that it is not possible to implement a distributed TTFD service with the required coverage (timeliness of detection would probably fail too often), some specific kinds of applications, namely those that have strict requirements on the duration of distributed actions, can not use this service that would be useful to timely detect when one of those actions would incur a timing failure.

One question that could be asked is “why is it necessary to have synchronous communication channels in order to timely detect distributed timed actions?” The problem is that a general purpose service must allow arbitrary timed actions to be specified (at any time and for any deadline), and must allow each of these actions to be “monitored” in run-time. However, without a synchronous network, timely dissemination of information about on-going timed actions, cannot be ensured and hence it would also be impossible to ensure that a timing failure affecting one of those timed actions would always be timely detected.

At this point it may be interesting to mention that in some particular cases the problem could be overcome. In particular, in situations where the involved participants know a priori the exact points in time in which timed actions take place (e.g., if these actions are executed periodically starting from some known initial point in time) and their specified expected duration, then, given the existence of a global time frame (which in the case of HIDENETS could be provided by the R&SAClock service) it is possible to detect in a timely manner when one of those timed actions has not been timely. In fact, this is the case covered by freshness detection. It is however not covered in terms of implementation, as it is integrated in the TTFD service (for which we focused on a non-distributed implementation). Use cases found in domains outside of HIDENETS (e.g. Railways), can indeed make use of this specific case of timely timing failure detection. For more details on how the problem has been addressed in these contexts the reader might refer to XX.

Finally, we must highlight that despite the fact that the distributed version of the TTFD has not been implemented, it is nevertheless possible to implement applications with safety-critical requirements that depend on the timeliness of the execution. As we showed in the platooning test-bed, it is possible to design the application in such a way that timeliness requirements have to be met for local tasks only. Then, provided that interactions between the payload and oracles is done in a way that timely reaction to timing failures can be performed within the oracles, it is possible to secure the desired safety properties.

9.3 Open research issues

Naturally, responses of questions create new questions. This section mentions some of the most demanding open research issues which are important from a HIDENETS perspective.

Use case extensions

A driver's perception is crucial for successful market introductions. This perception is influenced by individual environmental and situational influences. Monitoring the driver behaviour during a trip can produce input about current needs which can help to optimise the entire system from application to communication level to increase the dependability perception.

Car-to-car equipped vehicles can be seen as a wide net of sensors which may output useful information. The question arises how to control and use such a sensor net to get this information in an environment with rarely spread infrastructure and privacy requirements.

Resilient routing

Within HIDENETS a set of the best IP fast reroute schemes for scenarios with proactive link-state routing have been developed and analysed. These schemes rely on full topology knowledge to calculate the alternative next hops. As a future task it is required to optimise their performance in cases where only limited neighbour information is available or in cases where different nodes have inconsistent topology views.

Also, the work on resilience in ABC networks could be extended by adding smartness to the schemes for network detection and selection. The Access Points (AP) could provide congestion data in beacons to avoid congested APs.

Architecture development

Considering the developing of architectures which allow the provision of dependable and secure pervasive services, further research is required on new abstractions/building blocks specific to spatial computing. This includes geo-localised versions of resilient storage of information in ad-hoc mode. Also, the definition, implementation and evaluation of applications take advantage of these new building blocks (e.g. traffic jam avoidance by computing every road segment's occupation).

Evaluation approaches

The development of a holistic and more trustable assessment process is one of the HIDENETS topics which are highly demanded in almost all engineering disciplines outside the car-to-x/communication domain. Further research is needed in combining experimental measurements and stochastic modelling, assessment of the impact of the introduced approximations on the final indicators and the integration of multiple techniques (like FMEA, Hazard analysis).

Test-bed and testing

The experimental work is still containing a lot of open research issues that have not been addressed. These mainly involve enhancements in the power control, interference, and multi-hop behaviour. Test-beds have been created in an indoor laboratory setting. A focus on real car-2-car scenarios would also involve the use of special antennas, and other standards like 802.11p. Furthermore, there is need for the development of realistic platforms, at a laboratory scale, for the validation and evaluation of fault-tolerance algorithms targeting systems which comprise a large set of communicating mobile devices. It is well-known that two different network simulators may yield experimental performance results that strongly diverge. The impact on verification results is still to be investigated. Hence it would be interesting to assess simulation environments from a testing perspective to check the convenience of their facilities in terms of conformance and robustness of test scenarios.

In addition, it is required to develop a scenario language for describing interactions in mobile settings which respect to extension on time, industrial case studies, test purpose and case as well as on consolidations tools.

Industrial adoption and economics of dependability related work

HIDENETS has put high focus on the interworking and interaction with dependability related standards fora such as the Service Availability Forum or the Car to Car Communication Consortium. It has been shown that there is potential for excellent cross-fertilization of the research and the industrial world. Further enforcement of such cooperation will be an important topic for future dependability related projects. It is this cooperation, which is needed for industry to make progress in the development of high-dependability and highly dependable solutions and at the same time help prioritise research activities.

All aspects of dependability related work are influenced by economic considerations in one way or the other. In HIDENETS, a first, rudimentary step in understanding these considerations was taken with the Business Impacts Analysis which can be extended to a) measure the cost of major losses to society caused by undependable technologies and b) support a clear understanding of the implications of dependability qualities in their day-to-day use and its positive impact on the dependability industry. The evolution of dependability technologies is heavily driven by technological aspects. A somewhat wider perspective must be taken to understand the full potential, but also the full implications of dependability on society, environment, cost of living, use of technology etc.

Annex I The dependability and resilience conceptual framework

This section introduces some basic concepts and terminology related to dependability and resilience issues that are used in this document to characterise the HIDENETS reference model. The related concepts will be useful to define the properties, the threats, and the resilience and fault tolerance related requirements.

1. Basic concepts and terminology

The definitions presented in the following are based on the dependability concepts that have been developed and updated since the mid-seventies by the Fault-Tolerant Computing community, and especially the IFIP Working Group 10.4 [3-7][31], [32], [33], [34], [35] . It is noteworthy that other concepts similar to dependability exist, such as survivability, trustworthiness and resilience (e.g., see [32] for a definition of some of these concepts and a comparison with dependability). Among these, the concept of resilience extends the classical notion of fault tolerance usually applied to recover system functions in spite of operational faults, to some level of adaptability, so as to be able to cope with system evolution and unanticipated conditions12. Throughout this report, however, in most cases dependability, resilience and trustworthiness will be used interchangeably to refer to the ability to deliver a service that can justifiably be trusted.

The service delivered by a system (in its role as a service provider) is its behaviour as perceived by its user(s). The function of a system is what the system is intended to do and is described by the functional specification in terms of functionality and performance. Correct service is delivered when the service implements the system function. A service failure occurs when the delivered service deviates from correct service. A failure is thus a transition from correct service to incorrect service. The period of delivery of incorrect service is a service outage. The transition from incorrect service to correct service is a service restoration. Based on the definition of failure, an updated definition of dependability, which complements the initial definition in providing a criterion for deciding if the service is dependable, is as follows: the ability of a system to avoid service failures that are more frequent and more severe than is acceptable.

A systematic exposition of dependability consists of three main parts: the threats to, the attributes of and the means by which dependability is attained. The dependability threats correspond to faults, errors and failures that might affect the service(s) delivered by the system. The dependability attributes define the main facets of dependability that are relevant for the target system and applications. The dependability means correspond to the methods and techniques used to support the production of a dependable system. These means can be classified into four major categories:

Fault prevention: to prevent the occurrence or introduction of faults,

Fault tolerance: to avoid service failures in the presence of faults,

Fault removal: to reduce the number and severity of faults,

Fault forecasting: to estimate the present number, the future incidence, and the likely consequences of faults.

Fault prevention and fault tolerance aim to provide the ability to deliver a service that can be trusted, while fault removal and fault forecasting aim to reach confidence in this ability by justifying that the functional and the dependability and security specifications are adequate and that the system is likely to meet them.

Fault prevention is part of general engineering and can be attained through the use of rigorous development techniques, high-level specification and design methodologies, structured programming, information hiding, modularization, etc.

Fault tolerance which is aimed at failure avoidance is generally implemented by error detection and subsequent system recovery. More details about these techniques are provided in Section 4.

[12] This interpretation is actually in line with the related on-going terminology work being carried out within the ReSIST project (www.resist-noe.org): Resilience is the ability to deliver, maintain and improve service when facing threats (accidental or malicious) and evolutionary changes. Such evolutionary changes could be of various types: functional, environmental or technological (hardware and software), or might occur in short term (related to dynamicity or mobility of the system components of its environments), in medium term (related to the introduction of new versions or reconfigurations) or in long term (e.g., as a result of reorganizations).

Fault removal is performed both during the development phase and the operational life of a system. During the development, it consists of three steps: verification, diagnosis, and correction. Verification is the process of checking whether the system adheres to given properties, termed the verification conditions. If it does not, the other two steps are applied. Verification activities are generally implemented using a combination of static analysis, model checking, theorem proving, testing, etc.

Finally, fault forecasting is conducted by performing an evaluation of the system behaviour with respect to fault occurrence or activation. Evaluation has two aspects: a) qualitative, or ordinal evaluation which aims to identify, classify and rank the failure modes or the combinations of events that would lead to system failures, and b) quantitative, or probabilistic, evaluation, which aims to evaluate in terms of probabilities the extent to which some of the attributes of dependability are satisfied; those attributes are then viewed as measures of dependability. Various methods can be used to support these evaluations, including analytical modelling, simulation, experimental measurements as well as judgements.

The solutions investigated in the HIDENETS project cover various dimensions of dependability taking into account the four classes of dependability means (fault prevention, fault tolerance, fault removal, and fault forecasting). The development of these solutions is based on the analysis of the specific requirements and challenges characterizing various applications and use case scenarios in particular from car-to-car and automotive domains.

In the following, we present more detailed concepts related to: 1) the dependability properties, 2) the threats to be addressed to satisfy these properties, and 3) the fault tolerance mechanisms that can be used to cope with the threats.

2. Dependability related properties

Depending on the applications considered, different facets of dependability may be important, i.e., different emphasis may be put on different attributes of dependability. Basic dependability attributes are defined as follows:

Availability: readiness for correct service,

Reliability: continuity for correct service,

Safety: absence of catastrophic consequences on the user(s) and the environment,

Confidentiality: absence of unauthorised disclosure of information,

Integrity: absence of improper system alterations,

Maintainability: ability to undergo modifications and repairs,

Several other dependability attributes can be obtained as combinations or specialization of the primary attributes listed above. In particular, security is defined as the concurrent existence of a) availability for authorised users only, b) confidentiality and c) integrity where ‘improper’ means ‘unauthorised’.

The attributes of dependability may be emphasised to a greater or a lesser extent depending on the application: availability, integrity and maintainability are generally required, although to a varying degree depending on the application, whereas reliability, safety and confidentiality may or may not be required. The extent to which a system possesses the attributes of dependability should be considered in a relative, probabilistic sense, and not in an absolute, deterministic sense. Due to the unavoidable presence or occurrence of faults, systems are never totally available, reliable, safe or secure.

Integrity is a prerequisite for availability, reliability and safety, but may not be so for confidentiality (for instance, attacks via covert channels or passive listening can lead to a loss of confidentiality, without impairing integrity). The definition given above for integrity — absence of improper system alterations extends the usual definition as follows: (a) when a system implements an authorization policy, ‘improper’ encompasses ‘unauthorised’; (b) ‘improper alterations’ encompass actions that prevent (correct) upgrades of information; (c) ‘system state’ encompasses hardware modifications or damages.

Besides the attributes listed above, other secondary attributes can be considered to refine the primary attributes. An example of such a secondary attribute is robustness, i.e., dependability with respect to external faults, which characterises a system’s reaction to a specific class of faults.

The notion of secondary attributes is especially relevant for security, when we distinguish among various types of information. Examples of such secondary attributes are:

Accountability: availability and integrity of the identity of the person who performed an operation

Authenticity: integrity of a message content and origin, and possibly of some other information, such as the time of emission.

Non-reputability: availability and integrity of the identity of the sender of a message (non-repudiation of the origin), or the receiver (non-repudiation of reception)

Variations in the emphasis on the different attributes of dependability directly affect the appropriate balance of the techniques (fault prevention, tolerance, removal, forecasting) to be employed in order to make the resulting systems dependable. This problem is all the more difficult as some attributes conflict (e.g., availability and safety, availability and security), necessitating design trade-offs.

3. Threats

The dependability threats mainly correspond to the faults, errors, and failures that should be covered by the target applications to satisfy the desired dependability properties.

A service may fail either because it does not comply with the functional specification, or because this specification did not adequately describe the system function. A service failure occurs when at least one or more external state(s) of the system deviate from the correct service state. The deviation is called an error. The adjudged or hypothesised cause of an error is called a fault.

A system may not, and generally does not, always fail in the same way. The ways a system can fail are its failure modes, which may be characterised according to four viewpoints: 1) the failure domain, 2) the detectability of failures, 3) the consistency of failures, and 4) the consequences of failures on the environment.

The failure domain viewpoint leads to the distinction of content failures (e.g., incorrect values) and timing failures. Value failures are a particular case of content failures. Timing failures may be of two types: early or late depending on whether the service was delivered too early or too late. Failures when both content and timing are incorrect fall into two classes:

halt failure, or simply halt, when the service is halted (the external state becomes constant, i.e., system activity, if there is any, is no longer perceptible to the users); a special case of halt is silent failure, or simply silence, when no service at all is delivered at the service interface (e.g., no messages are sent in a distributed system).

erratic failures otherwise, i.e., when a service is delivered (not halted), but is erratic (e.g., babbling).

The detectability of failures viewpoint addresses the signalling of the service failures to the users. Signalling at the service interface originates from detection mechanisms in the system that check the correctness of the delivered service. When the losses are detected and signalled by a warning signal, then signalled failures occur. Otherwise, they are unsignalled failures. The detection mechanisms themselves have two failure modes: 1) signalling a loss of function when no failure has actually occurred, that is a false alarm, 2) not signalling a function loss, that is an unsignalled failure. When the occurrence of service failures result in reduced modes of service, the system signals a degraded mode of service to the user(s). Degraded modes may range from minor reductions to emergency service and safe shutdown.

The consistency of failures viewpoint when two or more service users are involved leads to the distinction of consistent failures (when the incorrect service is perceived identically by all the users) from inconsistent failures, also called Byzantine failures, (when some or all users perceive an incorrect service differently),

The consequences of failures on the environment viewpoint lead to the grading of failure modes according to different *failure severities*. The failure modes are ordered into severity levels, to which are generally associated maximum acceptable probabilities of occurrence. The number, the labelling, and the definition of the severity levels, as well as the acceptable probabilities of occurrence, are application-related, and involve the dependability and security attributes for the considered application(s).

When designing a dependable system, it is very important to identify which fault classes are to be taken into account because different means are to be used to deal with different fault classes. Thus, fault assumptions influence directly the design choices, and also the level of dependability that can be achieved.

Faults and their sources are very diverse. They can be classified according to different criteria: the phase of creation (development *vs.* operational faults), the system boundaries (internal *vs.* external faults), their phenomenological cause (natural *vs.* human-made faults), the dimension (hardware *vs.* software faults), the persistence (permanent *vs.* transient faults), the objective of the developer or the humans interacting with the system (malicious *vs.* non malicious faults), their intent (deliberate *vs.* non-deliberate faults), or their capability (accidental *vs.* incompetence faults).

Malicious faults are human-made faults that are generally introduced with the malicious objective to alter the functioning of the system during use. The goals of such faults are: 1) to disrupt or halt service, causing denials of service; 2) to access confidential information; or 3) to improperly modify the system. They can be grouped into two classes: 1) malicious logic faults that encompass faults introduced during the development phase such as Trojan horses, logic or timing bombs, and trapdoors, as well as operational faults such as viruses, worms or zombies (see e.g., [32] for a precise definition of these terms); and 2) intrusion attempts that are operational external faults. The external character of intrusion attempts does not exclude the possibility that they may be performed by system operators or administrators who are surpassing their rights.

The list of failures and faults assumptions to be addressed in the development process should be completed by the specification of the acceptable degraded operation modes as well as of the constraints imposed on each mode, i.e., the maximal tolerable service interruption duration and the number of consecutive and simultaneous failures to be tolerated, before moving to the next degraded operation mode. The analysis of the impact of the simultaneous loss or degradation of multiple functions and services requires particular attention. Depending on the dependability needs and the system failure consequences on the environment, the need to handle more than one nearly concurrent failure modes could be vital. Such an analysis is particularly useful for the specification of the minimal level of fault tolerance that must be provided by the system to satisfy the dependability objectives. It also provides preliminary information for the minimal separation between critical functions that is needed to limit their interactions and prevent common mode failures.

4. Fault tolerance

Fault tolerance is aimed at failure avoidance. It is generally implemented by error detection and subsequent system recovery (or simply recovery).

There exist two classes of error detection techniques:

Concurrent error detection which takes place during service delivery

Pre-emptive error detection which takes place while service delivery is suspended; it checks the system for latent errors (i.e., that are not yet detected) and dormant faults (i.e., that are not yet activated).

Recovery transforms a system state that contains one or more errors (and possibly faults) into a state without detected errors and faults that can be activated again. Recovery consists of error handling and fault handling.

Error handling eliminates errors from the system state. It may take three forms:

Rollback, where the state transformation consists of returning the system back to a saved state that existed prior to error detection; that saved state is a checkpoint,

Compensation, where the erroneous state contains enough redundancy to enable error elimination,

Roll forward, where the state without detected errors is a new state.

Fault handling prevents faults from being activated again. It involves four steps:

Fault diagnosis, which identifies and records the cause(s) of error(s) in terms of both location and type,

Fault isolation, which performs physical or logical exclusion of the faulty components from further participation in service delivery,

System reconfiguration, which either switches in spare components or reassigns tasks among non-failed components,

System re-initialization, which checks, updates and records the new configuration and updates system tables and records,

Usually, fault handling is followed by corrective maintenance that removes faults isolated by fault handling.

Systematic usage of compensation may allow recovery without error detection. This form of recovery is called fault masking. However, such simple masking will conceal a possibly progressive and eventually fatal loss of protective redundancy; thus practical implementations of masking generally involve error detection (and possibly fault handling), leading to masking and recovery.

The choice of error detection, error handling and fault handling techniques, and of their implementation is directly related to and strongly dependent upon the fault assumptions. The classes of faults that can actually be tolerated depend on the fault assumptions considered in the development process. Various techniques for achieving fault tolerance can be used such as performing multiple computations in multiple channels, either sequentially or concurrently, where the channels may be of identical design (if the objective is to tolerate independent physical faults or elusive design faults) or may implement the same function via separate designs and implementations, i.e., through design diversity (if the objective is to tolerate solid design faults). Other techniques include the use of self-checking components which provide the ability to define error confinement areas.

Fault tolerance is a recursive concept: it is essential that the mechanisms that implement fault tolerance should be protected against the faults that might affect them. Examples of such protection are voter replication, self-checking checkers, stable memory for recovery programs and data.

The notion of coverage, in particular attached to the efficiency of the fault tolerance techniques and mechanisms especially with respect to the failure assumptions they rely upon, is essential to ensure the overall ability to actually achieve the targeted dependability and security levels.

Systematic introduction of fault tolerance is often facilitated by the addition of support systems specialised for fault tolerance (e.g., software monitors, service processors, and dedicated communication links).

Fault tolerance is not restricted to accidental faults. Some mechanisms of error detection are directed towards both malicious and non-malicious faults (e.g., memory access protection techniques) and schemes have been proposed for the tolerance of both intrusions and physical faults, via information fragmentation and dispersal, as well as for tolerance of malicious logic, and more specifically of viruses, either via control flow checking, or via design diversity. It is noteworthy that the extension and adaptation to security of traditional techniques for tolerating accidental faults, led to the emergence of the intrusion tolerance concept. The focus of intrusion tolerance is on ensuring that systems will remain operational (possibly in a degraded mode) and will continue to provide core services despite faults due to intrusions.