



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team NETQUEST

Declarative Network Programming

Liama - Beijing - Chine

THEME SYM

Activity
R *eport*

2008

Table of contents

1. Team	1
2. Overall Objectives	1
2.1. Overall Objectives	1
2.2. Highlights	2
3. Scientific Foundations	2
3.1. Introduction	2
3.2. Logic and query languages	2
3.3. Distributed algorithms and computation	3
3.4. Networking protocols	4
4. Application Domains	4
4.1. Introduction	4
4.2. Rapid prototyping of network protocols	4
4.3. Pervasive adaptation	4
4.4. Heterogeneous networks	5
5. Software	5
6. New Results	5
6.1. Distributed complexity of classical logics	5
6.2. A rule-based language for networking protocols	6
6.3. Declarative communication protocols	6
6.4. A problem related to data movement in networks	6
7. Contracts and Grants with Industry	6
8. Other Grants and Activities	6
8.1. Actions in France	6
8.1.1. CITI, INSA Lyon	6
8.1.2. LSR/IMAG Grenoble	7
8.2. Actions in China	7
9. Dissemination	7
10. Bibliography	7

NETQUEST is an Inria international project-team, located in the Sino-French IT Lab, LIAMA in Beijing, China, and attached to the Rocquencourt Unit.

1. Team

Research Scientist

Stéphane Grumbach [Leader of NETQUEST, Research Director, INRIA]
Michel Bauderon [DR CNRS, Sept-Dec 2008]

Technical Staff

Antoine Henry [Engineer, LIAMA, Jan-Oct 2008]

PhD Student

Fang Wang [PhD, ISCAS, 2003-2008]
Wenwu Qu [PhD, ISCAS, 2003-2008]
Kun Suo [PhD, CASIA, 2006-2011]

Post-Doctoral Fellow

Junhu Zhang [CASIA, Post doc 2006-2008]
Zhilin Wu [CASIA, Post doc 2007-2009]

Administrative Assistant

Lin Tang [Assistant, LIAMA]

Other

Tiantian Yin [Master, CASIA, 2005-2008]
Xin Qi [Master, CASIA, 2007-2010]
Jeremie Grolleau [Intern, CPE-Lyon]
Xavier How-Choong [Intern, INSA Lyon]
Alberto Martinez [Intern, INSA Lyon]
Vincent Dubosclard [Intern, Georgia Tech Lorraine]

2. Overall Objectives

2.1. Overall Objectives

Networks of independent entities, cooperating to handle global tasks, constitute a fascinating class of systems. They are widely found in nature, with cells exchanging information with their neighbors, or neurons through synapse connections, as well as in social organizations. Such organizations are making their ways in microelectronic systems and might in the near future become ubiquitous. They are made possible by the development of ever smaller and cheaper electronic devices with increased memory capacity and computational power, together with the standardization efforts for both wireless communication and data exchange format.

One of the main barriers today to the development of networks of cooperating devices is the **lack of programming abstraction**. Smart devices are usually dedicated systems based on ad hoc models, which are not generic enough to support the needs of future applications (flexibility, scalability, ease to maintain, etc.). The deployment of a sensor network for instance is a tedious task which requires an expertise in the underlying OS and hardware. Applications implemented today on top of TinyOS for instance impose to deal with low-level issues such as memory.

The objectives of the Netquest project are to develop solutions that allow to program networks in a declarative manner, by specifying the intended functionalities without having to deal with system aspects, much as in database systems. The separation of a **logical level**, accessible to users and applications, from the **physical layers** constitutes the basic principle of Database Management Systems. It is at the origin of their technological and commercial success. This fundamental contribution of Codd in the design of the relational model of data, has lead to the development of universal high level query languages, that all vendors recognize, as well as to query processing techniques that optimize the declarative queries into (close to) optimal execution plans.

The abstraction we propose relies on a very simple idea: **model the network as a database**. The network is thus essentially hidden and perceived by each node as a database, with which it interacts through declarative query languages. The communication between devices thus consists of queries and data. Each node should be equipped with a **distributed query engine**, evaluating all queries whether posed by the node itself or received from other nodes. This approach blurs the traditional distinction between communication and application layers. Both are handled in a uniform fashion using queries evaluated by the query engine.

We consider two levels of abstraction: (i) the global level, where the network can be programmed as a whole, and (ii) the local level, where programs specify the nodes behavior. At the global level, a node can for instance fire a query asking for a route satisfying some properties, without any algorithmic specification. The network will rely on the distributed query engine on each node to evaluate that query, thus resulting in a distributed algorithm. At the local level of abstraction on the other hand, the behavior of the nodes can be specified. Routing protocols for instance are programmed at the local level by specifying the exchange of messages necessary to achieve a task.

"Modeling the network as a database" provides the programming abstraction which is lacking to multihop networks, but it also raises difficult **theoretical and practical problems**, to which our efforts are devoted. In fact we will concentrate on the following three research directions.

- Establish **theoretical foundations** for network query languages: design, distributed complexity, and expressive power.
- Implement **distributed query engines** to execute queries of the network query languages, with distributed optimization.
- Validate the declarative approach through **real network problems**, such as networking protocols, and distributed network applications.

2.2. Highlights

The year 2008 was devoted to the implementation of the Netquest system, which is being tested with real networking protocols.

3. Scientific Foundations

3.1. Introduction

The scientific foundations rely firstly on the field of databases, from its theoretical foundations to system issues such as query processing. Distributed algorithms and models of distributed computations are also of fundamental importance. Finally, we also rely on networking protocols such as those used in ad hoc networks as well as in sensor networks.

3.2. Logic and query languages

Keywords: *complexity, declarative rule languages, expressive power, logic.*

Logical formalisms, such as first-order logic (FO), fixed-point logic (FP), monadic second-order logic (MSO) etc., allow to express problems in a declarative way. Instead of describing how to compute problems step by step, only the desired results of the computation are specified by logical expressions. The use of declarative query languages based on logical formalisms for data management was largely exploited by Codd in the 1970's [22] in the relational model in which there is a separation between the logical and the physical levels. Since then, the investigation of the theoretical foundations of query languages has been a strong focus of the database theory community.

Two important measures characterize query languages: their expressive power and their complexity. Given a query language, deciding which problem can be expressed in this language characterizes its expressive power. How complex it is to compute the queries in a given query language, characterizes its computational complexity.

The expressive power and the complexity of classical logics have been intensively studied. The expressive power of FO, for instance, has been shown to be rather limited: it can only express local properties, and it lacks the power of counting and recursion [24]. The complexity of FO has also been shown to be quite low: FO formulas can be evaluated in space logarithmic in the size of their input [39]. The parallel complexity of FO has also been considered, it was shown [26] that FO formulas can be evaluated in constant time, independently of the size of their inputs, on Boolean circuits with arbitrary fan-in gates, the well-known AC^0 class.

Although classical query languages have been intensively studied in the context of centralized and parallel computation, their distributed computation has attracted only little attention. We first investigate the expressive power of classical query languages for describing distributed data structures, such as spanning trees, routing tables, and dominating sets, etc., and study their distributed complexity. We then propose to introduce new primitives into classical query languages to design proper logical formalisms for multihop networking (global level abstraction), while achieving a nice balance between expressive power and distributed computational complexity.

For the local level of abstraction, declarative rule languages (variants of Datalog) have been used to describe the communication protocols [32], [30]. They revived the recursive languages developed in the 80's for deductive databases [18], [40], [19], well-suited to define routes in networks, and showed that query languages allow the expression of protocols, one or two orders of magnitude simpler than classical imperative programming languages. We continue this trend to demonstrate the potential of declarative rule languages for the local abstraction level, clarifying their semantics in asynchronous distributed computation, investigating further their expressive power and the complexity of their distributed evaluation.

3.3. Distributed algorithms and computation

Keywords: *bottom-up, decentralized computation, top-down evaluation.*

We consider a distributed computation model based on the message passing model [17]. It relies on a communication graph, in which each node only knows the local neighboring topology, and can only communicate with its one-hop neighbors. Distributed algorithms for graph problems, such as spanning tree, coloring, dominating sets, etc. have been widely investigated and constitute still an active area of research.

In multihop networks, the networking information is fully distributed over the entire network. Moreover, multihop networks are usually dynamic and nodes are usually constrained, with limited CPU, memory, energy, etc. The decentralized evaluation of logical formalisms over multihop networks thus requires high efficiency, scalability, and fault tolerance.

The locality of distributed algorithms is of special interest in this context. An algorithm is local [29], [28], if it solves a global problem in distributed time which is either a constant, that is independent of the size of the network, or at least smaller than the diameter of the network. We consider generalization of these definitions to communications bounded but not necessarily local, resulting in the class of frugal computations.

The centralized evaluation of classical logical formalisms for graphs has been intensively investigated. It is well-known that FO can be evaluated in linear-time over bounded degree graphs [38], and MSO can be evaluated in linear-time over bounded tree-width graphs [23]. These results on the centralized evaluation lead to efficient distributed evaluation algorithms for the query languages used at the global level of abstraction.

The distributed evaluation of declarative rule languages, used at the local level of abstraction, has been considered by several researchers. Abiteboul et al. introduced distributed Datalog (dDatalog), by adding locations to the relations and rules (but not to the tuples), and devised distributed query-by-subquery technique (dQSQ) to evaluate dDatalog [14]. On the other hand, Loo et al. adapted the bottom-up evaluation techniques of Datalog, e.g. semi-naive evaluation and magic set rewriting, to NDlog, another version of distributed

Datalog with locations for tuples [30]. Our purpose is to adapt both the bottom-up and top-down evaluation techniques of Datalog queries to the distributed setting, and combine them in a fully decentralized way.

3.4. Networking protocols

Keywords: *distributed data structures, performance comparison, routing, self-configuration, self-organization.*

Networking usually implies the construction and maintenance of distributed data structures, such as spanning tree, shortest path, dominating set, etc. In multihop networking, some additional constraints (e.g. mobility, energy efficiency, etc.) make the construction and maintenance of these distributed data structures challenging.

Typical multihop networking protocols include self-configuration and self-organization protocols, such as ASCNET [21], FISCO [33], and routing protocols, such as DSDV (Destination-Sequenced Distance Vector Routing) [35], OLSR (Optimized Link-State Routing) [27], AODV (Ad hoc On-Demand Distance Vector Routing) [36], and VRR (Virtual Ring Routing) [20]. The declarative specification of these classical protocols will be compared with their implementation in imperative languages. With similar efficiency, the declarative expressions of the protocols offer considerable advantages for the facility to develop, maintain, and adapt their codes.

4. Application Domains

4.1. Introduction

Applications of ubiquitous networks are emerging in many areas such as intelligent transportation, games, social networking, sensor networks, ambient intelligence, etc. We have considered widely spatial information systems in the past. Their interaction with networks is of great interest to support queries relating to the ambient space and positioning issues. Distributed in spatial environment of different scale (e.g. building, landscape) sensor networks constitute a promising application to validate the Netquest approach.

4.2. Rapid prototyping of network protocols

The Netquest approach provides a (global-level or local-level) programming abstraction that allows network protocol designers to program their protocols in a declarative way. The Netquest system is responsible for transforming these protocols into low-level code and executing them. The computation of these protocols in Netquest can be simulated and visualized using WSNET, which provides simulations of network environments. Compared to the implementation of protocols in imperative programming languages, the declarative specification can be two order of magnitude shorter [31]. More generally, Netquest offers an environment which simplifies the design of protocols by relying on the DBMS for fundamental aspects such as the transactions.

4.3. Pervasive adaptation

In the absence of sufficient or sufficiently accurate knowledge, adaptive methods have been developed for query execution, that allow to alternate query processing with query execution phases. This trend, started long ago in the case of system R [16] for cases where the statistics would be misleading or incomplete. This is now a topic of increasing interest with the development of applications running over data distributed over networks. Adaptation is the key challenge for ubiquitous networks. More generally, the capacity to self-assemble, grow, repair, organize, evolve over long period of time while maintaining essential functionalities is of fundamental importance for networks of cooperating objects. The declarative combination of networking and application layers, jointly processed by distributed query engines, offers a huge potential for pervasive adaptation, because the query engine can adapt the queries to the network (adaptive evaluation) and the network to the queries (Quality of service and content based routing). We will try to give evidence of this claim on real applications.

4.4. Heterogeneous networks

Heterogeneous networks, in which the node architecture, operating system, data format, etc. might vary significantly, pose additional challenges to network management and applications. Netquest offers a high level abstraction which allows to specify an application or a protocol independently of the underlying architecture. The Netquest system can run on any type of devices assuming nodes are not too constrained and are equipped with a (local) DBMS. The Netquest approach will be used to test network management and data-centric applications in heterogeneous networks.

5. Software

5.1. Netquest system

Participants: Michel Bauderon, Stéphane Grumbach, Antoine Henry, Xin Qi, Kun Suo.

We have implemented a system, also called **Netquest**, to be embedded on each node of a network, which evaluates in a distributed manner queries expressed in the *Netlog* language. The architecture of the system differs drastically from classical embedded systems such as TinyDB for instance. It relies on an embedded DBMS (MySQL), which handles all the data whether related to applications or to the network. The database also stores the protocols expressed in a declarative manner. The DBMS plays a fundamental role in the system which relies in particular on its transaction management. The main component is the Distributed Query Engine, which evaluates locally the queries and generate subqueries for other nodes. Currently, Netlog in push mode has been implemented for network queries. Other versions of the distributed query engine will be developed in the near future. Finally, the last component is a hybrid router, which handles the routing functions, but in a rather original way. In particular the hybrid router can handle implicit destinations, which are to be evaluated by the query engine. The use of an embedded linux together with a DBMS changes the way distributed systems can be programmed, and results in simplifications of the nodes development and maintenance, for its use of standard software components.

To facilitate the protocol design in the Netlog language, we have designed the *Netquest protocol format*, and developed a *graphical interface* to allow the protocol designers to write the declarative program for protocols conveniently. Then the declarative protocol is transformed into SQL queries, and stored into the embedded DBMS.

The Netquest system has been installed as a library on the network simulator WSNET. WSNET will be used to simulate wireless networks, test the Netquest system, as well as applications running on Netquest. To visualize the simulation result of Netquest on WSNET, we have developed a *visualization tool* to illustrate the message passing between network nodes, and the computation of the distributed query engine of the Netquest system in each node.

6. New Results

6.1. Distributed complexity of classical logics

Participants: Stéphane Grumbach, Zhilin Wu.

We have obtained results on the distributed complexity of first-order (FO) and monadic second order (MSO) logic on graphs of bounded degree [12]. These results relate the locality of the logic, in the sense of Gaifman [25], to the complexity of the distributed computation in terms of the number of messages handled on each node, which can be shown to be constant, a property weaker but which resembles the locality of distributed computations [29]. We have also obtained results on the distributed complexity of fixpoint logic (FP) on general graphs [11]. We plan to pursue the investigation of the distributed computation of classical logics for graphs, and propose a new logic which includes a non-deterministic construct to be able to express classical distributed computing problems.

6.2. A rule-based language for networking protocols

Participants: Stéphane Grumbach, Wenwu Qu, Kun Suo, Fang WANG, Zhilin Wu.

For the local level of abstraction, expressing the nodes behavior, we have designed a new language, *Netlog* [10], which extends SQL with recursion, non-determinism and communication primitives, based on rules à la Datalog [37]. This language has been shown to be suitable to express a large collection of classical networking protocols. It thus allows to give a declarative specification of both networking protocols as well as network applications, much like the language proposed by the P2 group in Berkeley for similar purposes. This language admits two procedural semantics, in push mode corresponding to proactive protocols, and in pull mode corresponding to reactive protocols. Netlog is currently mainly used to express protocols which are stored in a declarative manner in the database of the nodes. We have adapted the semi-naive bottom-up algorithm [15] for Datalog to evaluate Netlog and implemented it in the Netquest system.

6.3. Declarative communication protocols

Participants: Stéphane Grumbach, Wenwu Qu, Kun Suo.

We designed the *Netquest protocol format* (NPF) to express networking protocols in Netlog. Then we used NPF to express routing protocols for ad hoc networks, such as DSDV (Destination Sequence Distance Vector) [35], OLSR (Optimized Link-State Routing) [27], and VRR (Virtual Ring Routing) [20] and test, and simulate them in the Netquest system, integrated on WSNET.

6.4. A problem related to data movement in networks

Participants: Stéphane Grumbach, Zhilin Wu.

We consider *motion planning on directed graphs*, a problem related to data movement in a network with constrained node capacity and unidirectional links, which is an abstraction of the structure of wireless sensor networks. We proposed two algorithms for solving feasibility of motion planning on acyclic and strongly connected directed graphs respectively [13], thus extending results by Papadimitriou et al. on undirected graphs [34].

7. Contracts and Grants with Industry

7.1. France Telecom

Close links exist with the France Telecom RD Beijing Lab, which started a new action on sensor networks. Our objective is to test our query engine on a testbed developed by FT. A CRC contract between INRIA and FT, associating the LIAMA with the Beijing Lab of France Telecom RD has been signed for the period July 2007 - June 2009.

8. Other Grants and Activities

8.1. Actions in France

8.1.1. CITI, INSA Lyon

Cooperation with Stéphane Ubeda and Fabrice Valois from the INRIA Ares project of CITI, INSA Lyon, on declarative networking in the framework of a Sino-French PRA project 2007-2008. Our aim is to investigate formally various networking protocols, such as flooding, self-configuration, self-organisation, routing, medium access, in the context of multihop networks, by using their declarative modeling.

8.1.2. LSR/IMAG Grenoble

Cooperation with Christine Collet and Christophe Bobineau from the Laboratory LSR/IMAG in Grenoble on the development of query optimization techniques in the context of networks.

8.2. Actions in China

8.2.1. ISCAS, Institute of Software

Close links exists with the Institute of Software of the Chinese academy of Sciences, ISCAS. The professor Huimin LIN, academician, is the supervisor of the two students Fang WANG and Wenwu QU.

9. Dissemination

9.1. Dissemination

Stéphane Grumbach is a PC member of APWeb08, WAIM'08, SITIS'08, DEXA'09, DASFAA'09, and DS2ME@ICDE'09. He is co-PC-Chair of the 9th International Conference on Mobile Data Management (MDM'08) Beijing, China, April 27-30, 2008

Michel Bauderon is a PC member of ICGT'08, and a member of the steering committee of ICGT.

10. Bibliography

Major publications by the team in recent years

- [1] M. KOUBARAKIS, T. K. SELLIS, A. U. FRANK, S. GRUMBACH, R. H. GÜTING, C. S. JENSEN, N. A. LORENTZOS, Y. MANOLOPOULOS, E. NARDELLI, B. PERNICI, H.-J. SCHEK, M. SCHOLL, B. THEODOULIDIS, N. TRYFONA (editors). *Spatio-Temporal Databases: The CHOROCHRONOS Approach*, Lecture Notes in Computer Science, vol. 2520, Springer, 2003.
- [2] M. BAUDERON, M. MOSBAH. *A Unified Framework for Designing, Implementing and Visualizing Distributed Algorithms.*, in "Electr. Notes Theor. Comput. Sci.", vol. 72, n^o 3, 2003, p. 570-588.
- [3] S. GRUMBACH, Z. LACROIX, P. RIGAUX, L. SEGOUFIN. *Optimization Techniques*, Springer, 2000, p. 319-334.
- [4] S. GRUMBACH, T. MILO. *Towards Tractable Algebras for Bags.*, in "J. Comput. Syst. Sci.", vol. 52, n^o 3, 1996, p. 570-588.
- [5] S. GRUMBACH, M. RAFANELLI, L. TINININI. *On the equivalence and rewriting of aggregate queries.*, in "Acta Inf.", vol. 40, n^o 8, 2004, p. 529-584.
- [6] S. GRUMBACH, P. RIGAUX, L. SEGOUFIN. *Handling Interpolated Data.*, in "Comput. J.", vol. 46, n^o 6, 2003, p. 664-679.
- [7] S. GRUMBACH, P. RIGAUX, L. SEGOUFIN. *The DEDALE System for Complex Spatial Queries.*, in "Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA", 1998, p. 213-224.

- [8] S. GRUMBACH, J. SU. *Finitely Representable Databases.*, in "J. Comput. Syst. Sci.", vol. 55, n^o 2, 1997, p. 273-298.
- [9] S. GRUMBACH, L. TINININI. *On the content of materialized aggregate views.*, in "J. Comput. Syst. Sci.", vol. 66, n^o 1, 2003, p. 133-168.

Year Publications

Other Publications

- [10] S. GRUMBACH, F. WANG, Z. WU. *Netlog, a logical language for networks*, technical report, 2008.
- [11] S. GRUMBACH, F. WANG, Z. WU. *On the distributed evaluation of recursive queries on graphs*, technical report, 2008.
- [12] S. GRUMBACH, Z. WU. *On the distributed evaluation of MSO on graphs*, technical report, 2008.
- [13] Z. WU, S. GRUMBACH. *Feasibility of motion planning on acyclic and strongly connected directed graphs*, Discrete Applied Mathematics (submitted), 2008.

References in notes

- [14] S. ABITEBOUL, Z. ABRAMS, S. HAAR, T. MILO. *Diagnosis of asynchronous discrete event systems: datalog to the rescue!*, in "Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD- SIGART Symposium on Principles of Database Systems, Baltimore, Maryland, USA", 2005.
- [15] S. ABITEBOUL, R. HULL, V. VIANU. *Foundations of Databases.*, Addison-Wesley, 1995.
- [16] M. M. ASTRAHAN, M. W. BLASGEN, D. D. CHAMBERLIN, K. P. ESWARAN, J. GRAY, P. P. GRIFFITHS, W. F. KING III, R. A. LORIE, P. R. MCJONES, J. W. MEHL, G. R. PUTZOLU, I. L. TRAIGER, B. W. WADE, V. WATSON. *System R: Relational Approach to Database Management*, in "ACM Trans. Database Syst.", vol. 1, n^o 2, 1976, p. 97-137.
- [17] H. ATTIYA, J. WELCH. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, Wiley-Interscience, 2004.
- [18] F. BANCILHON. *Naive Evaluation of Recursively Defined Relations*, in "On Knowledge Base Management Systems", 1985, p. 165-178.
- [19] F. BANCILHON, D. MAIER, Y. SAGIV, J. D. ULLMAN. *Magic sets and other strange ways to implement logic programs (extended abstract)*, in "PODS '86: Proceedings of the fifth ACM SIGACT-SIGMOD symposium on Principles of database systems, New York, NY, USA", ACM, 1986, p. 1-15.
- [20] M. CAESAR, M. CASTRO, E. B. NIGHTINGALE, G. O'SHEA, A. ROWSTRON. *Virtual ring routing: network routing inspired by DHTs*, in "SIGCOMM Comput. Commun. Rev.", vol. 36, n^o 4, 2006, p. 351-362.
- [21] A. CERPA, D. ESTRIN. *ASCENT: Adaptive Self-Configuring sSensor Network Topologies*, in "SIGCOMM Comput. Commun. Rev.", vol. 32, n^o 1, 2002, p. 62-62.

- [22] E. F. CODD. *A Relational Model of Data for Large Shared Data Banks*, in "Commun. ACM", vol. 13, n^o 6, 1970, p. 377-387.
- [23] B. COURCELLE. *Graph Rewriting: An Algebraic and Logic Approach*, in "Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)", Elsevier and MIT Press, 1990, p. 193-242.
- [24] H. EBBINGHAUS, J. FLUM. *Finite model theory*, Springer-Verlag, Berlin, 1999.
- [25] H. GAIFMAN. *On local and non-local properties*, in "Proceedings of the Herbrand Symposium, Logic Colloquium'81, North Holland", 1982.
- [26] N. IMMERMAN. *Expressibility and Parallel Complexity*, in "SIAM J. Comput.", vol. 18, n^o 3, 1989, p. 625-638.
- [27] P. JACQUET, P. MUHLETHALER, T. CLAUSEN, A. LAOUITI, A. QAYYUM, L. VIENNOT. *Optimized link state routing protocol for ad hoc networks*, in "Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International", 2001, p. 62-68.
- [28] F. KUHN, R. WATTENHOFER. *Constant-time distributed dominating set approximation*, in "Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing, PODC", 2003.
- [29] N. LINIAL. *Locality in Distributed Graph Algorithms.*, in "SIAM J. Comput.", vol. 21, n^o 1, 1992, p. 193-201.
- [30] B. T. LOO, T. CONDIE, M. N. GAROFALAKIS, D. E. GAY, J. M. HELLERSTEIN, P. MANIATIS, R. RAMAKRISHNAN, T. ROSCOE, I. STOICA. *Declarative networking: language, execution and optimization.*, in "Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, USA, June 27-29", 2006.
- [31] B. T. LOO, T. CONDIE, J. M. HELLERSTEIN, P. MANIATIS, T. ROSCOE, I. STOICA. *Implementing declarative overlays*, in "SIGOPS Oper. Syst. Rev.", vol. 39, n^o 5, 2005, p. 75-90.
- [32] B. T. LOO, J. M. HELLERSTEIN, I. STOICA, R. RAMAKRISHNAN. *Declarative routing: extensible routing with declarative queries*, in "SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA", ACM, 2005, p. 289-300.
- [33] J.-L. LU, F. VALOIS, D. BARTHEL, M. DOHLER. *FISCO: A Fully Integrated Scheme of Self-Configuration and Self-Organization for WSN.*, in "IEEE/WCNC", 12007.
- [34] C. H. PAPADIMITRIOU, P. RAGHAVAN, M. SUDAN, H. TAMAKI. *Motion planning on a graph*, in "FOCS'94", 1994, p. 511-520.
- [35] C. PERKINS, P. BHAGWAT. *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*, in "ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications", 1994, p. 234-244.
- [36] C. PERKINS. *Ad-hoc on-demand distance vector routing*, in "In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications", 1999, p. 90-100.

- [37] R. RAMAKRISHNAN, J. D. ULLMAN. *A survey of research on deductive database systems*, in "Journal of Logic Programming", vol. 23, 1993, p. 125–149.
- [38] D. SEESE. *Linear time computable problems and logical descriptions*, in "Electr. Notes Theor. Comput. Sci.", vol. 2, 1995.
- [39] M. Y. VARDI. *The complexity of relational query languages (Extended Abstract)*, in "STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing, New York, NY, USA", ACM, 1982, p. 137–146.
- [40] L. VIEILLE. *Recursive Axioms in Deductive Databases: The Query/Subquery Approach*, in "Expert Database Conf.", 1986, p. 253-267.