INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team NETQUEST

Declarative Network Programming

Paris - Rocquencourt

Theme : Distributed Systems and Services

Activity Report

2009

Table of contents

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, Jan-Aug 2009]

PhD Student

Fang Wang [PhD, ISCAS]
Wenwu Qu [PhD, ISCAS]
Kun Suo [PhD, CASIA]

Post-Doctoral Fellow

Zhilin Wu [CASIA, Post doc 2007-2009]
Xiang Zhou [CASIA, Post doc 2009-2011]

Administrative Assistant

Lin Tang [Assistant, LIAMA]

Other

Xin Qi [Master, CASIA, 2007-2010]
Julian Boutillot [Intern, ENSPS, Strasbourg]
Patrick Daures [Intern, INSA Lyon]
Sandeep Manchem [Intern, IIT Guwahati, India]
Vincent Dubosclard [Intern, ENSM Saint-Etienne]

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. The Internet of things, in which potentially all objects, whether virtual or real, will become addressable and smart, attracts now a considerable attention.

One of the main barriers today to the development of such networks 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.

Our ambition is to show that the framework provided by the programming abstraction helps to write programs which are:

- easy to write in a concise way;
- efficient to execute, that is they can be compiled into efficient distributed algorithms, which can adapt to dynamic environment;
- verifiable thanks to a clear semantics;
- portable over heterogeneous devices and networks.

The objective of the project is to solve the **theoretical and practical problems** raised by the programming abstraction. We 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 (sensor, vehicles, M2M, etc.).

2.2. Highlights

The Netquest system has been ported over a network of iMote devices by Orange Labs, thus demonstrating the portability of a declarative system over a network of constraint terminals.

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

Logical formalisms, such as first-order logic (FO), fixed-point logic (FP), and monadic second-order logic (MSO) for instance, 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 [25] 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 for many interesting problems, it lacks recursion and the power of counting [27]. 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 [42]. The parallel complexity of FO has also been considered, it was shown [29] 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 on graphs has attracted only little attention. We consider 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 also 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 communication protocols [35], [33] , thus reviving the recursive languages developed in the 80's for deductive databases [21], [43], [22], well-suited to define routes in networks. 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

Several models have been proposed for distributed computation. The message passing model [20] 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 graphs, 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. Many applications rely on multihop networks which are dynamic, with failures and moving nodes, and moreover, nodes are generally 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 [32], [31], 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 [41], and MSO can be evaluated in linear-time over bounded tree-width graphs [26]. Can such results in a classical computational setting offer insights in the complexity of their distributed computations over graphs? We have shown that this is indeed the case for FO and MSO over restricted classes of graphs, which can be computed frugally, thus relating somehow the logical locality of FO with a weakened notion of locality of the computation.

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 [17]. 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 [33]. 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. Protocols

Network protocols are fundamental for networking. They provide basic services (e.g. routing) through constructing and maintaining distributed data structures, such as spanning tree, shortest path, dominating set, etc. Many networking protocols have been developed. Typical examples include various routing protocols, such as DSDV (Destination-Sequenced Distance Vector Routing) [38], OLSR (Optimized Link-State Routing) [30], AODV (Ad hoc On-demand Distance Vector Routing) [39], and VRR (Virtual Ring Routing) [23], as well as self-configuration and self-organization protocols, such as ASCNET [24] or FISCO [36].

Although network protocols are crucial for networking, their construction is a complex and error prone task, when more and more constraints (e.g. mobility, energy efficiency etc.) have to be taken into account. The challenges originate from the inherent complexity of developing correct program codes and from the distributed nature of networked systems.

Since protocols usually describe the behavior of nodes under events, such as messages received, they can be easily written in rule-based languages. Our objective is to show that the declarative specification of particular network protocols, can lead to efficient behavior. Thus through implementing protocols in declarative rule languages, we can reduce the inherent complexity of programming for network protocols.

To help cope with the distributed nature of networked systems, auxiliary tools need to be developed, such as protocol debugging tools, automatic protocol verification tools, etc. Actually many network protocols can be specified in declarative rule languages and the semantics of these languages can be strictly defined, so it seems promising to develop auxiliary tools based on declarative languages.

4. Application Domains

4.1. Spatial information

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 protocols in Netquest can be monitored using the network simulator WSNet, and visualized, by a visualization tools developed in the group, showing the network activity as well as the evolution of the databases in the nodes. Compared to the implementation of protocols in imperative programming languages, the declarative specification can be two order of magnitude shorter [34]. More generally, Netquest offers an environment which simplifies the design of protocols by relying on the DBMS for fundamental aspects such as 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 was introduced long ago in system R [19] for situations 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 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).

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, Xin Qi, Kun Suo.

We have developed a system, **Netquest** [14] to be embedded on each node of a network, which evaluates in a distributed manner programs 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 (e.g. 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 Engine, which evaluates locally the Netlog programs, and generate messages to send, in addition to updates in the local database. Currently, Netlog in push mode has been implemented. Other versions of the engine will be developed in the near future for other types of programs and queries. Finally, the last component is a router, which handles the communication, in cooperation with the Engine. It accesses the database to find routes. It can also handle implicit destinations, which are to be evaluated by the 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 protocol design in the Netlog language, protocols are developed through an interface which follows the Netquest protocol format, and performs some syntactic verification. Then the declarative protocol is transformed into SQL queries, and stored into the embedded DMBS.

The Netquest system has been installed as a library on the network simulator WSNet. WSNet is 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.

Netquest has also been ported on smart phones (with SQL Server) as well as on a network of iMote devices (with SQLite) in cooperation with Orange Labs in Beijing.

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), Fixpoint (FP) and monadic second order (MSO) logic on various classes of graphs. In [11], we show that first-order properties can be frugally evaluated, that is, with only a bounded number of messages, of size logarithmic in the number of nodes, sent over each link, over bounded degree networks as well as planar networks. Moreover, we show that the result carries over for the extension of first-order logic with unary counting. These results relate the locality of the logic, in the sense of Gaifman [28], 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 [32].

In [10], we considered fixpoint logic, and showed that it can be evaluated with a polynomial number of messages of logarithmic size. We then showed that the (global) logical formulas can be translated into rule programs describing the local behavior of the nodes of the distributed system, which compute equivalent results. We also introduced local fragments of the logic which have a nice expressive power and admit tighter upper-bounds with bounded number of messages of bounded size.

In [15], we considered monadic second order logic, and showed that MSO can be evaluated in distributed linear time with only a constant number of messages sent over each link for planar networks with bounded diameter, as well as for networks with bounded degree and bounded tree-length. The distributed algorithms rely on the translation of MSO sentences into finite automata over trees, and on nontrivial transformations of linear time sequential algorithms for the tree decomposition of bounded tree-width graphs.

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 rule-based language, *Netlog*, which extends SQL with recursion, non-determinism and communication primitives, based on rules à la Datalog [40]. 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. This language admits two procedural semantics, in push mode corresponding to proactive protocols, and in pull mode corresponding to reactive protocols. The distributed fixpoint semantics is introduced in [9]. 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 [18] 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 have developed a library of protocols written in Netlog, such as routing protocols for ad hoc networks, DSDV (Destination Sequence Distance Vector) [38], OLSR (Optimized Link-State Routing) [30], and VRR (Virtual Ring Routing) [23]. Some of these protocols were experimented in the framework of the WSNet platform, or in the iMote testbed. In [12], adaptive protocols for MANET are considered.

6.4. Data motion in networks

Participants: Stéphane Grumbach, Zhilin Wu.

We considered *motion planning on directed graphs*, a problem related to data motion 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], [16], thus extending results by Papadimitriou et al. on undirected graphs [37].

7. Contracts and Grants with Industry

7.1. Orange R&D Lab

Close links have been developed with the Orange R&D Beijing Lab. The Netquest system has been ported on a network of iMote devices, thus demonstrating the feasibility of the approach for constrained terminals. This concludes a cooperation agreement, CRC Bamboo, between INRIA and FT, associating the LIAMA with the Beijing Lab of France Telecom R&D active 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, on declarative networking, started in the framework of a Sino-French PRA project for the period 2007-2008. Now supported by an ANR project, Ubiquest, for the period september 2009 - august 2012, our work will focus on 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

The 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, in the framework of the ANR project, Ubiquest, for the period september 2009 - august 2012.

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 APWeb'09, DASFAA'09, DEXA'09, DS2ME@ICDE'09, ICE-B 2009, IDEAS'09, OSSC'09, SITIS'09, WAIM'09, WISM'09, APWeb'10, DASFAA'10, DEXA'10, IDEAS'10, MDM'10.

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] S. Grumbach, Z. Lacroix, P. Rigaux, L. Segoufin. *Optimization Techniques*, Springer, 2000, p. 319-334.

[3] S. Grumbach, T. Milo. *Towards Tractable Algebras for Bags.*, in "J. Comput. Syst. Sci.", vol. 52, n^{o} 3, 1996, p. 570-588.

[4] 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, http://dx.doi.org/10.1007/s00236-004-0101-y.

[5] S. Grumbach, P. Rigaux, L. Segoufin. *Handling Interpolated Data.*, in "Comput. J.", vol. 46, n^{o} 6, 2003, p. 664-679, http://comjnl.oxfordjournals.org/content/vol46/issue6/index.dtl.

[6] 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, http://portal.acm.org/citation.cfm?doid=276304.276324.

[7] S. Grumbach, J. Su. *Finitely Representable Databases.*, in "J. Comput. Syst. Sci.", vol. 55, n^{o} 2, 1997, p. 273-298.

[8] 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, http://dx.doi.org/10.1016/S0022-0000(02)00033-8.

Year Publications

International Peer-Reviewed Conference/Proceedings

[9] S. Grumbach, F. Wang. *Netlog, a Rule-based Language for Distributed Programming*, in "PADL'10, Twelfth International Symposium on Practical Aspects of Declarative Languages, Madrid, Spain, january", 2010 CN .

[10] S. Grumbach, F. Wang, Z. Wu. *On the distributed evaluation of recursive queries on graphs*, in "NOTERE 2009, the 9th annual international conference on New Technologies of Distributed Systems, Montreal, Canada, july", 2009 CN .

[11] S. Grumbach, Z. Wu. *Logical locality entails frugal distributed computation over graphs*, in "WG 2009, the 35th International Workshop on Graph-Theoretic Concepts in Computer Science, Montpellier, june", 2009 CN .

[12] W. QU. *Declarative Composition Routing Protocols Adaptation in MANET Using Rough Set Theory*, in "WiCOM 2009, the 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, September", 2009 CN .

[13] Z. WU, S. GRUMBACH. *Feasibility of Motion Planning on Directed Graphs*, in "Theory and Applications of Models of Computation, 6th Annual Conference, TAMC 2009, Changsha, China, May 18-22", 2009, p. 430-439 CN .

Workshops without Proceedings

[14] M. BAUDERON, C. BOBINEAU, S. GRUMBACH, A. HENRY, X. QI, W. QU, K. SUO, F. WANG, Z. WU. *Netquest: An Abstract Model for Pervasive Applications*, in "Pervasive 2009, the 7th International Conference on Pervasive Computing, Nara, Japan, may", 2009 CN .

Other Publications

[15] S. GRUMBACH, Z. WU. *On Distributed Model Checking of MSO on Graphs*, 2009, CoRR abs/0904.1902 CN .

[16] Z. WU, S. GRUMBACH. *Feasibility of motion planning on acyclic and strongly connected directed graphs*, 2009, CoRR abs/0904.1920 CN .

References in notes

[17] 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.

[18] S. ABITEBOUL, R. HULL, V. VIANU. *Foundations of Databases.*, Addison-Wesley, 1995.

[19] 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.

[20] H. ATTIYA, J. WELCH. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, Wiley-Interscience, 2004.

[21] F. BANCILHON. *Naive evaluation of recursively defined relations*, Springer-Verlag New York, Inc., New York, NY, USA, 1986.

[22] 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, http://doi.acm.org/10.1145/6012.15399.

[23] 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.

[24] A. CERPA, D. ESTRIN. *ASCENT: Adaptive Self-Configuring sEnsor Network Topologies*, in "SIGCOMM Comput. Commun. Rev.", vol. 32, n^{o} 1, 2002, p. 62–62.

[25] 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, http://doi.acm.org/10.1145/362384.362685.

[26] B. COURCELLE. *Graph Rewriting: An Algebraic and Logic Approach*, in "Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)", Elsevier and MIT Press, 1990, p. 193-242.

[27] H. EBBINGHAUS, J. FLUM. *Finite model theory*, Springer-Verlag, Berlin, 1999.

[28] H. GAIFMAN. *On local and non-local properties*, in "Proceedings of the Herbrand Symposium, Logic Colloquium'81, North Holland", 1982.

[29] N. IMMERMAN. *Expressibility and Parallel Complexity*, in "SIAM J. Comput.", vol. 18, n^{o} 3, 1989, p. 625-638.

[30] 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.

[31] F. KUHN, R. WATTENHOFER. *Constant-time distributed dominating set approximation*, in "Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing, PODC", 2003.

[32] N. LINIAL. *Locality in Distributed Graph Algorithms.*, in "SIAM J. Comput.", vol. 21, n^{o} 1, 1992, p. 193-201.

[33] 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.

[34] 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, http://doi.acm.org/10.1145/1095809.1095818.

[35] 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, http://doi.acm.org/10.1145/1080091.1080126.

[36] 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", 2007.

[37] C. H. PAPADIMITRIOU, P. RAGHAVAN, M. SUDAN, H. TAMAKI. *Motion planning on a graph*, in "FOCS'94", 1994, p. 511-520.

[38] C. E. 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.

[39] C. E. 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.

[40] R. RAMAKRISHNAN, J. D. ULLMAN. *A survey of research on deductive database systems*, in "Journal of Logic Programming", vol. 23, 1993, p. 125–149.

[41] D. SEESE. *Linear time computable problems and logical descriptions*, in "Electr. Notes Theor. Comput. Sci.", vol. 2, 1995.

[42] 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, http://doi.acm.org/10.1145/800070.802186.

[43] L. VIEILLE. *Recursive Axioms in Deductive Databases: The Query/Subquery Approach*, in "Expert Database Conf.", 1986, p. 253-267.