Supply Chain Discovery Services in an Internet of Things Environment

By

Abdelmounaim Dahbi

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the degree of

Doctorate of Philosophy in Computer Science

Ottawa-Carleton Institute for Computer Science

School of Electrical Engineering and Computer Science

University of Ottawa

Ottawa, Ontario, Canada

To the memory of my father

To my beloved mother

To my beloved wife

To my dear daughter
To my dear son

Abstract

Electronic Product Code (EPC) refers to a numbering standard developed to uniquely identify physical objects, loads, locations, assets and other entities which are to be tracked or otherwise identified. The tracking technology consists of assigning Radio Frequency Identification (RFID) tags, holding universally unique EPC codes, to the entities to be identified. While the EPC-RFID technology is used to identify and capture data about the physical objects to be tracked in a supply chain, the EPCglobal Network ensures the exchange of the captured data between supply chain stakeholders. Such a real-time data exchange increases visibility and efficiency throughout the supply chain, and thus it increases both company profitability and customer satisfaction. The EPCglobal Network can be regarded as the backbone for the future Internet of Things (IoT). We focus our work in this thesis on Discovery Services (DS); a suite of network lookup services enabling users to retrieve all relevant information sources with regards to a given EPC. They can be viewed as search engines for the future business infrastructure deployed in the IoT. Motivated by the unprecedented and incessantly growing amount of EPC data, the expected epidemic growth in the solicitation frequency of the lookup service, and also the foreseen exceptionally large flow of highly sensitive EPC information, we focus on proposing solutions to problems pertaining to two main challenges; architecture design of Discovery Services and their security. On the architecture design level, we propose novel DS architectures with focus directed towards meeting four major requirements; network scalability, query responsiveness, service extensibility and acceptance. On the security level, we propose probabilistic security schemes aiming at securing even further Discovery Services in the IoT in general, and in the EPCglobal network in particular.

Acknowledgements

It is a great pleasure to thank the many people who made this thesis possible.

It is difficult to overstate my gratitude to my supervisor, Professor Hussein T. Mouftah, whose distinguished expertise and deep understanding, added considerably to my graduate experience. I am deeply grateful to him for his great guidance, continuous support and considerable patience. Throughout my thesis-writing period, Professor Hussein T. Mouftah incessantly provided instructive comments and helpful advice. My infinite thanks go to him.

Also, I wish to thank all members of my family, my friends and my colleagues for providing a loving environment for me. My parents have been particularly supportive. My beloved wife has been very considerate and encouraging. The unconditional love of my little treasures, Maryam and Adam, has been a source of inspiration for me.

Finally, I would like to thank the many people who have taught me throughout my life.

Contents

Abstract iii

Acknowledgements iv

List of Figures x

List of Tables xiii

List of Abbreviations xiv

List of Symbols and Notations xvii

1 Introduction 1
- 1.1 Lookup Services in the IoT . . . 1
- 1.2 Motivations and Objectives . . . 2
- 1.3 Thesis Contributions . . . 5
- 1.4 Thesis Outline . . . 6

2 Lookup Systems in the Internet of Things 8
- 2.1 Introduction . . . 8
- 2.2 Background . . . 9
  - 2.2.1 Radio Frequency IDentification (RFID) . . . 9
  - 2.2.2 Electronic Product Code (EPC) . . . 12
  - 2.2.3 EPCglobal Architecture Framework . . . 14
  - 2.2.4 EPCglobal Network . . . 15
- 2.3 EPC Information Services (EPCIS) . . . 17
- 2.4 Object Naming System (ONS) . . . 19
  - 2.4.1 ONS Architectures . . . 20
  - 2.4.2 ONS Limitations . . . 21
  - 2.4.3 ONS Security Issues . . . 22
- 2.5 Discovery Services . . . 23
  - 2.5.1 Discovery Services Implementations . . . 23
  - 2.5.2 Discovery Services Architectures . . . 26

2.5.2.1 ONS-based Discovery Services . . . 26
2.5.2.2 Non ONS-based Discovery Services . . . 28
2.5.3 Discovery Services in IoT . . . 30
2.5.4 Security in Discovery Services . . . 33
2.5.5 Privacy in Discovery Services . . . 34
2.6 Summary . . . 35

3 Secured Distributed Lookup Service in the EPCglobal Network 36
3.1 Introduction . . . 36
3.2 Currently Deployed Lookup System . . . 38
3.3 Proposed Lookup System Architecture . . . 40
3.3.1 Design Requirements . . . 40
3.3.2 Proposed Architecture . . . 41
3.3.3 Discussion . . . 43
3.4 Proposed Lookup System Security . . . 44
3.4.1 Security Requirements . . . 44
3.4.1.1 Authentication . . . 44
3.4.1.2 Data Integrity . . . 45
3.4.1.3 Confidentiality . . . 45
3.4.1.4 Availability . . . 45
3.4.2 Security of the Proposed Solution . . . 46
3.4.2.1 Key Distribution . . . 46
3.4.2.2 Data Publishing . . . 48
3.4.2.3 Data Retrieval . . . 49
3.4.3 Discussion . . . 50
3.5 Conclusion . . . 50

4 Query State Inference in Discovery Services 51
4.1 Introduction . . . 51
4.2 Threat Model . . . 53
4.3 Background and Notations . . . 53
4.4 Query Feature Extraction . . . 55
4.5 A Gaussian Security Scheme for Query State Inference . . . 56
4.5.1 Assumptions . . . 56
4.5.2 Gaussian Model . . . 57
4.5.3 Proposed Algorithm for Query State Inference . . . 58
4.5.4 Performance Evaluation . . . 59
4.5.4.1 Simulation Setup and Parameters . . . 59
4.5.4.2 Evaluation Approach . . . 59
4.5.4.3 Results . . . 61
4.5.4.3.1 Two Observed Features . . . 62
4.5.4.3.2 Three Observed Features . . . 62

4.6 A Hidden Markov Model Security Scheme for Query State Inference . . 65
4.6.1 Hidden Markov Model (HMM) 65
4.6.1.1 Elements of a Discrete HMM (DHMM) 66
4.6.1.2 Elements of a Continuous HMM (CHMM) 67
4.6.1.3 The Three Basic Problems Solved by an HMM 68
4.6.2 Assumptions 69
4.6.3 Proposed CHMM for Query State Inference 70
4.6.4 Performance Evaluation 71
4.6.4.1 Simulation Setup and Parameters 71
4.6.4.2 Evaluation Approach 72
4.6.4.3 Results 72
4.7 Conclusion 75

5 A Hierarchical Architecture for Distributed EPCglobal Discovery Services 76
5.1 Introduction 76
5.2 Assumptions 78
5.2.1 Geographical Binding 79
5.2.2 Hierarchical Identification 79
5.2.3 One Node Per Company 80
5.3 Hierarchical Naming 80
5.3.1 EPC Company Prefix 80
5.3.2 A More Comprehensive Scheme 80
5.4 Notations and Definitions 82
5.5 Flat Distributed Architecture (FDA) 83
5.5.1 Functionality of FDA 84
5.5.2 Merits and Limitations of FDA 85
5.6 Hierarchical Distributed Architecture (HDA) 85
5.6.1 Tree Model 87
5.6.2 Functionality of HDA 87
5.6.3 Inter-overlay vs Intra-overlay routing 88
5.6.4 Entry Nodes 89
5.6.5 Applicability to EPCglobal Discovery Services 90
5.6.6 Merits and Limitations of HDA 91
5.6.6.1 Scalability 92
5.6.6.2 Query Responsiveness 92
5.7 Horizontal vs Vertical Inter-overlay Routing 94
5.8 Conclusion 95

6 Performance Evaluation of the HDA Architecture 96
6.1 Introduction 96
6.2 Considered Performance Metrics 98
6.3 Performance Analysis 98

6.3.1 Theoretical Model . . . 98
6.3.2 Theoretical Results . . . 102
6.4 Simulation of HDA on Plametsim . . . 106
6.4.1 Simulation Setup . . . 106
6.4.2 Simulation Results . . . 107
6.5 Emulation of HDA on Planetlab . . . 109
6.5.1 Emulation Setup . . . 109
6.5.1.1 Constraints . . . 109
6.5.1.2 Emulation Steps and Parameters . . . 110
6.5.1.3 Emulation Tools . . . 110
6.5.2 Emulation Results . . . 111
6.6 Conclusion . . . 111

7 Threshold-based Distributed Discovery Services for EPCGlobal Network 114
7.1 Introduction . . . 114
7.2 Routing Protocols for EPCglobal Network . . . 116
7.2.1 Intra-Domain Routing protocol for EPCglobal Network . . . 116
7.2.2 Inter-Domain Routing protocol for EPCglobal Network . . . 117
7.3 Network Architecture . . . 118
7.3.1 EPC global network extension and operation . . . 118
7.3.2 EPC global network Architecture . . . 119
7.3.3 Intra and Inter-domain routing protocols interaction . . . 121
7.3.4 Signaling messages . . . 124
7.3.5 Advertisements thresholds . . . 126
7.3.5.1 Justified Update Blocking (JUB) . . . 127
7.3.5.2 Unjustified Update Acceptance (UUA) . . . 127
7.3.5.3 Unjustified Update Blocking (UUB) . . . 128
7.4 Performance Evaluation . . . 128
7.4.1 Performance analysis . . . 129
7.4.2 Simulation results . . . 131
7.4.2.1 Blocking probability . . . 133
7.4.2.2 Euclidean distance . . . 135
7.4.2.3 Number of Updates . . . 138
7.4.2.4 Intra-domain blocking effect . . . 141
7.5 Conclusion . . . 143

8 Inventory Management as a Service for Supply Chain Stakeholders 144
8.0.1 Outline . . . 147
8.1 Proposed Cloud-Centric Platform . . . 147
8.1.1 System Architecture . . . 147
8.1.2 Consumers' Transactions and Feedback . . . 148
8.2 Proposed Inventory Optimization Cloud Service . . . 149

8.2.1 Inventory Optimization Model 150
8.2.1.1 Assumptions 150
8.2.1.2 The Model 151
8.2.2 Inventory Optimization as a Cloud Service 153
8.2.2.1 Real-time Statistical Computation of Parameters . . . 153
8.2.2.2 Real-time Probabilistic Computation of the Inventory Threshold 155
8.3 Results 156
8.4 Conclusion 157

9 Conclusions and Future Work **160**
9.1 Conclusions 160
9.2 Future Work 163

References **164**

List of Publications **181**

List of Figures

Figure 2.1 RFID Reader-Tag Interaction 12
Figure 2.2 The SGTIN-96 EPC Format 14
Figure 2.3 The EPCglobal Architecture Framework Components 16

Figure 3.1 The ONS System in action 39
Figure 3.2 Our Proposed DHT-based Lookup System 43

Figure 4.1 An illustration of the EPCglobal Discovery Services 54
Figure 4.2 An illustration of the query feature extraction process 56
Figure 4.3 An illustration of two Gaussian distributions 57
Figure 4.4 The likelihood that a Gaussian generates a given value x 58
Figure 4.5 PSA and PSA Algorithms using 2 and 3 extracted features 60
Figure 4.6 Comparison of Detection Rate among Ref , PSA and PPA algorithms 63
Figure 4.7 Comparison of False Alarm Rate among Ref , PSA and PPA algorithms 63
Figure 4.8 Comparison of Detection Rate between PSA and PPA algorithms 64
Figure 4.9 Comparison of False Alarm Rate between PSA and PPA algorithms 64
Figure 4.10 An illustration of a discrete HMM $\lambda = (\pi, A, B)$ 68
Figure 4.11 The HMM model, illustrated in Figure 4.10, in action 69
Figure 4.12 A comparison of the Detection Rate between Ref algorithm and $CHMM$ model 73
Figure 4.13 A comparison of the False Alarm Rate between Ref algorithm and $CHMM$ model 73

Figure 5.1 An example of a three levels Hierarchical Geographically-based Company Prefix Assignment 81
Figure 5.2 An FDA-based overlay network for EPCglobal Discovery Services 84
Figure 5.3 A 1-level HDA-based overlay network for EPCglobal Discovery Services 86
Figure 5.4 An overview of HDA of a typical Overlay Network 88

Figure 6.1 Plot of the likelihood that two randomly selected nodes have the closest common parent . . . 103
Figure 6.2 Plot of the expected latency . . . 103
Figure 6.3 Theoretical performance results of HDA vs FDA (Nimber of hops) 105
Figure 6.4 Theoretical performance results of HDA vs FDA (time required per lookup) . . . 105
Figure 6.5 Performance simulation results of HDA vs FDA (Nimber of hops) 108
Figure 6.6 Performance simulation results of HDA vs FDA (time required per lookup) . . . 108
Figure 6.7 Comparison of the average number of hops in FDA vs HDA implemented on Planetlab . . . 112
Figure 6.8 Comparison of the average lookup time in FDA vs HDA implemented on Planetlab . . . 112

Figure 7.1 An instant of the routing table of node-1 . . . 117
Figure 7.2 EPCglobal network extension . . . 119
Figure 7.3 EPCglobal network architecture . . . 120
Figure 7.4 Inter and intra-domain interaction to perform update session of EPC1 between supply chain-1 and 4 . . . 121
Figure 7.5 Inter and intra-domain interaction to perform Update session of EPC2 between supply chain-4 and 1 . . . 122
Figure 7.6 Inter and intra-domain interaction to Update the inter-domain routing table of supply chain-1 . . . 123
Figure 7.7 Blocking probability for different thresholds value and 100 EPC 131
Figure 7.8 Blocking probability for different number of EPCs . . . 132
Figure 7.9 Justified Update Blocking for 8-node supply chains and 100 EPC shipped . . . 133
Figure 7.10 Unjustified update acceptance for an 8-node supply chains and 100 EPC shipped . . . 134
Figure 7.11 Unjustified Update Blocking for 8-node supply chains and 100 EPC shipped . . . 135
Figure 7.12 Total blocking for 8-node supply chains and 100 EPC shipped . . 136
Figure 7.13 Justified Update Blocking for 8-node supply chains and 250 EPC shipped . . . 136
Figure 7.14 Unjustified Update Acceptance for 8-node supply chains and 250 EPC shipped . . . 137
Figure 7.15 Unjustified Update Blocking for 8-node supply chains and 250 EPC shipped . . . 137
Figure 7.16 Total Blocking for 8-node supply chains and 250 EPC shipped . 138
Figure 7.17 Average Euclidean distance between routing tables of 100 EPCs 139
Figure 7.18 Average Euclidean distance between routing tables of 250 EPCs 139
Figure 7.19 Number of updates for 100 EPCs . . . 140

Figure 7.20 Number of updates for 250 EPCs 140
Figure 7.21 Justified Update Blocking due to intra-domain effect under various EPCs 141
Figure 7.22 Unjustified update acceptance due to intra-domain effect under various EPCs 142
Figure 7.23 Total blocking due to intra-domain effect under various EPCs . . 142

Figure 8.1 The proposed cloud platform vs ONS/Discovery Services 146
Figure 8.2 Overall architecture of the proposed cloud platform 148
Figure 8.3 Users' reviews/feedback and transactions reported to the platform 149
Figure 8.4 Stakeholder parameters and its relationship to its supplier and customer 150
Figure 8.5 Probability of stock-out for stakeholder S_i in terms of the threshold value, for $\mu_{i-1} = 0.8$ 158
Figure 8.6 Probability of stock-out for stakeholder S_i in terms of the threshold value, for $\mu_{i-1} = 0.8$ 158
Figure 8.7 Probability of stock-out for stakeholder S_i in terms of λ_i, for $\mu_{i-1} = 0.8$ 159
Figure 8.8 Probability of stock-out for stakeholder S_i in terms of λ_i, for $\mu_{i-1} = 1.6$ 159

List of Tables

Table 2.1 RFID tags classification based on the power source 10
Table 2.2 Comparison of RFID tags with bar codes 14
Table 2.3 A comparison of DS systems proposals 24

Table 6.1 Comparison of HDA vs FDA 106

Table 8.1 Decision variables used in Parameter Computation 154

List of Abbreviations

ADS	Aggregating Discovery Service.
BGP	Border Gateway Protocol.
BRIDGE	Building Radio frequency IDentification for the Global Environment.
CA	Certification Authority.
CHMM	Continuous Hidden Markov Model.
DDOS	Distributed Denial of Service.
DHT	Distributed Hash Table.
DNS	Domain Name System.
DNSSEC	Domain Name System Security Extensions.
DoR	Directory-of-Resources.
DS	Discovery Services.
EPC	Electronic Product Code.
EPC-BGP	Electronic Product Code Border Gateway Protocol.
EPC-OSPF	Electronic Product Code Open Shortest Path First.
EPCIS	EPC Information Services.
ESDS	Extensible Supply-chain Discovery Service.
FDA	Flat Distributed Architecture.

FQDN	Fully Qualified Domain Name.
GMN	General Manager Number.
GS1	Global System One.
HDA	Hierarchical Distributed Architecture.
HTML	Hypertext Markup Language.
IETF	Internet Engineering Task Force.
IoT	Internet of Things.
IP	Internet Protocol.
LPKI	Light Public Key Infrastructure.
MITM	Man in the Middle.
MONS	Multipolar Object Naming System.
NAPTR	Name Authority Pointer.
NoC	Notofication-of-Clients.
NoR	Notification-of-Resources.
OIDA	Object Information Distribution Architecture.
ONS	Object Naming System.
ONSSEC	Object Name System Security Extensions.
OSPF	Open Shortest Path First.
P2P	Peer-to-Peer.
PIR	Private Information Retrieval.
PKI	Public Key Infrastructure.
PML	Physical Makeup Language.
PTSP	Product Trace Service Platform.
QP	Query Propagation.

RDBMS	Relational Database Management System.
RFID	Radio Frequency IDentification.
RPC	Remote Procedure Call.
SGTIN	Serialized Global Trade Item Number.
SQL	Structured Query Language.
UDDI	Universal Description, Discovery, and Integration.
UPC	Universal Product Code.
UPnP	Universal Plug and Play.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
WS	Web Service.
WSDL	Web Service Description Language.
XML	eXtended Markup Language.

List of Symbols and Notations

$A_{i,j}$	The event: Two reservation requests, regarding the same EPC $e_j, 1 \leq j \leq M$, are received by $AS_i, 1 \leq i \leq N_A$ (either as an intermediate node or as a destination node) within a time frame less than μ. This event translates an error blocking, since AS_i receives the second reservation request before the first reservation request has been released (It is released in μ time units on average)
a_{u_p}	Registered account associated with user u_p
AS_i	Autonomous system with index $i, 1 \leq i \leq N_A$
$avgOrder_{BS_i}^{p,C}$	Notation representing the current average number of items, of category p, being processed by company C, per unit of time
$B_{i,j}$	The event: An advertisement, initiated by one of the other autonomous systems in the network, is received by $AS_i, 1 \leq i \leq N_A$, reserving the EPC $e_j, 1 \leq j \leq M$
B_i	Total number of reservation requests that are blocked at $AS_i, 1 \leq i \leq N_A$
BS_i	Notation referring to a given business step $i, 0 <= i < N_{BS}$, throughout the life-cycle of items of category p
C	Notation referring to a given stakeholder in a supply chain
$c_k^{u_p}$	Payment card k belonging to user u_p

$E[X]$	Expected value of a given random variable X
$E[Z]_a^b$	Expected value of random variable Z, while taking values in the interval [a,b]
e_j	EPC with index $j, 1 \leq j \leq N_E$
F	The probability of an EPC status being advertised as reserved
$fb(u, e)$	Most recent feedback score of user u regarding product item with EPC e
K	Number of levels used in Hierarchical Naming
L	A Random Variable representing latency between two randomly selected nodes in a flat overlay network
L_m^c	List of purchased items (transaction) m paid for using payment card c
lat_i	Average latency of a link for which the identifiers of the end nodes, d_i and d_j, are separated with logical distance $LD(d_i, d_j) = i$. Obviously $lat_0 < lat_1 < ... < lat_K$
$LD(d_i, d_j)$	The logical distance between the identifiers of two nodes d_i and d_j (two nodes storing data in the overlay), in terms of the number of levels separating the two nodes from their closest common parent in the tree-like hierarchical structure of the overlay network, illustrated in Figure 5.4, built based on the comprehensive hierarchical naming scheme described in Section 5.3.
$location(L_m^c)$	Location of transaction (list of items) L_m^c
M	Number of children subregions at $level\ i$ having the same parent subregion at $level\ i-1$
N	Number of nodes storing data in the overlay
N_L^c	Number of transactions (lists of items) paid for using payment card c
N^{paths}	Total number of paths

N_{avg}^{paths}	Average number of paths passing through an Autonomous System i
N_i^{paths}	Number of paths passing through an Autonomous System i
$N_c^{u_p}$	Number of payment cards belonging to user u_p
N_A	Number of autonomous systems in the network
N_E	Number of EPCs
N_L	Number of items in transaction (list of items) L
N_r	Number of retailers registered in the platform
N_u	Number of users having at least one account in the platform
N_{BS}	Number of possible business steps throughout the life-cycle of items of category p
p	Notation referring to a given product category in a supply chain
p_i	The stock-out probability threshold that stakeholder S_i is willing to accept
$PD(d_i, d_j)$	The physical distance separating the locations of the two physical nodes d_i and d_j (two nodes storing data in the overlay) in the globe.
r_v	Retailer identified by the index v
$S = \{d_1, d_2, ..., d_N\}$	The set of the N nodes storing data in the overlay
SO_i	the event that stakeholder S_i observes a stock-out right after the threshold th_i is reached, and consequently, an order of λ_i units has been directed to the immediate supplier S_{i-1}
T	Advertisement threshold for all autonomous systems $AS_i, 1 \leq i \leq N_A$
th_i	The inventory threshold that stakeholder S_i considers in its replenishment policy. In other words, the threshold number of items below which stakeholder S_i places an order to its immediate supplier S_{i-1}
$time(L_m^c)$	Timestamp of transaction (list of items) L_m^c

$timeDiffH_{BS_i}^{p,C}$ Notation representing the time that business step BS_i takes to complete, within the current query, compared to its last occurrence, within the immediate previous query

$timeDiffV_{BS_i}^{p,C}$ Notation representing the time that business step BS_i takes to complete compared to its predecessor business step BS_{i-1}, within the same current query

u_p User identified by the index p

ut Unit of time used to measure latency of an overlay link joining two (leaf) nodes, such as 1 millisecond or 1 microsecond. Hence, when we refer to the latency lat of a given overlay link with the value 100 for example, it actually means $lat = 100ut$ ($lat = 100ms$ or $lat = 100\mu s$, depending on what ut refers to)

X_{ad} A random variable representing the number of advertisements initiated by a given autonomous system $AS_i, 1 \leq i \leq N_A$

$X_{i,\delta}$ Random Variable representing the number of items sold by stakeholder S_i in δ units of time. $X_{i,\delta}$ follows Poisson distribution with rate $\delta \cdot \lambda_i$

Y_{i-1} Random Variable representing the inter-arrival of shipments from the supplier S_{i-1}, in answer to orders initiated by stakeholder S_i. Y_{i-1} follows exponential distribution with average inter-arrival time μ_{i-1}

δ arrival rate of **reservation** requests in the network (in seconds)

δ_i^j arrival rate of **reservation** requests in the $AS_i, 1 \leq i \leq N_A$, regarding the EPC $e_j, 1 \leq j \leq N_E$ (in seconds)

δ_i arrival rate of **reservation** requests in the $AS_i, 1 \leq i \leq N_A$ (in seconds)

δ_{avg} average arrival rate of **reservation** requests in the autonomous systems (in seconds). It is also the average arrival rate of **release** requests

Δ_{th_i} $\Delta_{th_i} = \frac{th_i}{\lambda_i}$: Average number of units of time required by stakeholder S_i to sell off th_i items, knowing that it sells an average of λ_i items in one unit of time

λ_i	Average number of items sold by stakeholder S_i to its customers, in one unit of time U
$\mathcal{P}$	Total Blocking Probability
μ	mean elapse time required for a given successful reservation request to be released (in seconds)
$\mu^{\boldsymbol{H},\boldsymbol{R}}_{p,C,BS_i}$	Standard deviation of the Gaussian distribution modeling the feature $timeDiffH^{p,C}_{BS_i}$ in "risky" queries.
$\mu^{\boldsymbol{H},\boldsymbol{S}}_{p,C,BS_i}$	Standard deviation of the Gaussian distribution modeling the feature $timeDiffH^{p,C}_{BS_i}$ in "safe" queries
$\mu^{\boldsymbol{O},\boldsymbol{R}}_{p,C,BS_i}$	Standard deviation of the Gaussian distribution modeling the feature $avgOrder^{p,C}_{BS_i}$ in "risky"
$\mu^{\boldsymbol{O},\boldsymbol{S}}_{p,C,BS_i}$	Standard deviation of the Gaussian distribution modeling the feature $avgOrder^{p,C}_{BS_i}$ in "safe" queries
$\mu^{\boldsymbol{V},\boldsymbol{R}}_{p,C,BS_i}$	Standard deviation of the Gaussian distribution modeling the feature $timeDiffV^{p,C}_{BS_i}$ in "risky" queries.
$\mu^{\boldsymbol{V},\boldsymbol{S}}_{p,C,BS_i}$	Standard deviation of the Gaussian distribution modeling the feature $timeDiffV^{p,C}_{BS_i}$ in "safe" queries
μ_i	Average number of units of time required by stakeholder S_i, to answer an order initiated by one of its customers
$\sigma^{\boldsymbol{H},\boldsymbol{R}}_{p,C,BS_i}$	Mean of the Gaussian distribution modeling the feature $timeDiffH^{p,C}_{BS_i}$ in "risky" queries
$\sigma^{\boldsymbol{H},\boldsymbol{S}}_{p,C,BS_i}$	Mean of the Gaussian distribution modeling the feature $timeDiffH^{p,C}_{BS_i}$ in "safe" queries
$\sigma^{\boldsymbol{O},\boldsymbol{R}}_{p,C,BS_i}$	Mean of the Gaussian distribution modeling the feature $avgOrder^{p,C}_{BS_i}$ in "risky" queries
$\sigma^{\boldsymbol{O},\boldsymbol{S}}_{p,C,BS_i}$	Mean of the Gaussian distribution modeling the feature $avgOrder^{p,C}_{BS_i}$ in "safe" queries

$\sigma^{\boldsymbol{V},\boldsymbol{R}}_{p,C,BS_i}$	Mean of the Gaussian distribution modeling the feature $timeDiffV^{p,C}_{BS_i}$ in "risky" queries
$\sigma^{\boldsymbol{V},\boldsymbol{S}}_{p,C,BS_i}$	Mean of the Gaussian distribution modeling the feature $timeDiffV^{p,C}_{BS_i}$ in "safe" queries
$P(\Lambda)$	Probability that event Λ occurs
$\binom{n}{k}$	n choose k

Chapter 1

Introduction

1.1 Lookup Services in the IoT

Although the concept of the IoT is still fuzzy and its boundaries are vague, it has recently gained extensive research interest. Such interest is partially justified by the wide horizons the IoT paradigm has opened, and by the new wide opportunities it is promising. IoT can be defined as a "world-wide network of uniquely addressable interconnected objects, based on standard communication protocols" [1]. Such a huge heterogeneous network will have to deal with tremendous amounts of bulk data. As a result, several challenges in terms of performance and security [2–5] have to be investigated.

The new IoT paradigm will have an important impact on almost each and every aspect of our lives. Business is one important aspect among others. The business infrastructure is one of the frameworks on which the impact is already apparent in the form of the EPCglobal Network. The EPCglobal Network [6] can be regarded as the backbone for the future IoT. Network scalability and lookup query responsiveness will, beyond all doubt, be among the major challenges in IoT in general, and in the EPCglobal Network specifically, in view of the expected epidemic growth. A well-established EPCglobal Network, especially from scalability and query responsiveness standpoints, would certainly lead to a more stable IoT.

Given the important advances already undertaken in the development of the EPCglobal framework, any future IoT-based business infrastructure will most probably build on

the current EPCglobal Network. Moreover, Discovery Services are expected to enable the users, both stakeholders and end customers, to reach all "relevant" information with regards to the inquired EPC.

The current design of the EPCglobal lookup service, based on the Object Naming System (ONS) [7], brings up numerous concerns. Particularly, the centralized architecture featuring the current design of the ONS has several worrying drawbacks as it has been elaborated in [8–11]. Moreover, it provides the information sources based on the company prefix of the EPC at hand. As a result, only the manufacturer information sources are provided in response to an ONS lookup query. Such limitation explains well enough the urgent need for distributed Discovery Services.

1.2 Motivations and Objectives

The main motivation behind the development of the EPCglobal Network is to provide trade partners (Suppliers, Manufacturers, Distributors and Retailers, etc.) with real time, and accurate data sharing network services regarding all the physical objects they are exchanging [6]. The EPC, designed to be unique across time, across space and over all existing physical objects, represents the key component to ensure both item's precise traceability of objects carrying RFID/EPC tags, and accurate information retrieval for those objects via the Internet. The EPCglobal Network real time data sharing between trade partners provide them with supply chain wide visibility resulting in a reduction of cost (e.g. efficient product recall and shelf replenishment [12]) and a reduction of loss (e.g. efficient anti-counterfeiting and anti-theft [13, 14]).

Although the real-time and accurate traceability gives companies a strong incentive to participate in the EPCglobal Network, companies would share their data only if the infrastructure offering the sharing capability, the EPCglobal Network in our case, is scalable, secure, easy to use, and more importantly collectively controlled and managed by all stakeholders.

Currently, the EPCglobal Network relies on the ONS for EPC data exchange. The ONS is a DNS-based lookup service ensuring a link between the trading partners in order to share EPC data. A number of lookup operations are performed in order to

localize information sources associated with the to-be-resolved EPC [7]. The ONS relies completely on a centralized ONS root managed exclusively by Verisign. The ONS root is similar to the Domain Name System (DNS) root except that, unlike the DNS root, the ONS root is not replicated in multiple geographical locations. The currently deployed ONS architecture has three major drawbacks which explain clearly the urgent need of an integrated Discovery Services system for EPC data lookup. They are the following:

- First, the unipolar design of the ONS root makes acceptance of the ONS system among supply chain stakeholders all around the world a major problem, since it is fully controlled and managed by a single entity (company, organization or country, etc.) [9, 10]. Furthermore, the unipolar architecture makes the ONS an attractive and easy target to certain security attacks. Different scenarios depicting possible attack models in such a unipolar architecture have been discussed in [10], while highlighting the political repercussions of such a design choice. Finally, the centralization and lack of redundancy of the ONS root create a single point of failure for the ONS system, and hence of the whole business infrastructure.

- Second, being based on the DNS system, the ONS inherits all the well-studied and well-documented weaknesses of the DNS including its poor performance, its complex configuration and its lack of security [8, 9]. The DNS system has always been an attractive target to Distributed Denial of Service (DDOS) attacks and also to Man In The Middle (MITM) attacks targeting the DNS resolution paths. Vulnerability to the Cache Poisoning attacks is another DNS weakness that may prove absolutely disastrous to systems relying on DNS for naming services.

- Third, the ONS has been designed to provide the information sources based on the company prefix of the EPC. As a result of this poor design, only the manufacturer information sources are provided in response to ONS lookup queries.

Although the final definition of Discovery Services requirements is not yet closed by EPCglobal, a number of proposals have enriched the literature ranging from architectures extending the currently deployed ONS system [10, 15–17] to complex solutions based on a clean-slate approach [9, 18–20]. Many of these works have focused on

the feasibility of P2P solutions and their high scalability, while others have focused on secure and/or privacy-preserving solutions.

In order to ensure and sustain the required quality of information flow among trading partners, focus has to be directed towards meeting three major requirements during the design of any Discovery Services architecture, besides the functionality requirement. They are the following:

- First, the control of the EPCglobal Network has to be equally shared between the subscribing companies, and hence between the countries hosting those companies. Such shared control would encourage companies/countries worldwide to accept the EPCglobal Network as the core engine of the future global business infrastructure.
- Second, given the exponentially-growing number of objects to be tracked in the EPCglobal Network, scalability of any proposed architecture for Discovery Services, and responsiveness of its underlying lookup queries, have to be regarded as a crucial design requirement.
- Third, given the sensitive nature of data exchanged among supply chain stakeholders, the lookup service requires special care from a security point of view. A number of proposals discussing the Discovery Services security have been published [11, 18, 19].

Our work aims at proposing Discovery Services architectures improving the scalability of the proposed P2P solutions, while improving the responsiveness of the lookup queries. Although a number of distributed architectures have been proposed in the literature, both their scalability and their responsiveness remain vulnerable, mainly because of their reliance on non-clustered or poorly-clustered flat distributed architectures. We apply hierarchical distributed architectures for Discovery Services in the EPCglobal network and we study their performance.

Our work aims also at securing those EPCglobal Discovery Services from a probabilistic risk assessment point of view, which will be able to detect risky queries in the case the traditional security measures fail, as in the case of a symmetric key disclosure. Focus has been put into building an additional security layer for Discovery Services on top

of the well-known security solutions. The proposed security schemes use probabilistic models in order to detect suspicious lookup queries in the EPCglobal network.

1.3 Thesis Contributions

In order to meet the objectives mentioned above, the thesis is divided into two major parts. In the first part we provide security schemes for IoT-based Discovery Services in general, and for the EPCglobal Network in particular. In the second part, we propose more efficient IoT-based Discovery Services architectures, with a focus on the EPCglobal Network. By "efficient architecture" we refer to a distributed architecture offering greater scalability and better query responsiveness. The contributions and accomplishments of this thesis are defined as follows:

- A secured distributed Discovery Services system has been proposed to solve some of the issues observed in the currently deployed ONS. The proposed system is based on the Distributed Hash Table (DHT) technique which has proved its robustness and efficiency in certain Peer-to-Peer (P2P) systems. The idea consists of upgrading the existing local ONS nodes belonging the EPCglobal Network subscribers into nodes of a DHT-based P2P network. EPC data is stored on and retrieved from these nodes. This upgrade aims at addressing the three major drawbacks, mentioned in Section 1.2, of the current ONS architecture.

- A security scheme aiming at predicting the state (safe or risky) of a lookup query has been proposed. This security scheme aims at providing an additional security layer to the EPCglobal Discovery Services. It is based on a simple statistical model based on the Gaussian model. The proposed security scheme consists of an inference algorithm deriving the most probable state for a given query.

- A security scheme, based on the Continuous Hidden Markov Model (CHMM), aiming at predicting the state (safe or risky) of a lookup query has been proposed. It is based on a probabilistic model which takes into consideration both the observations pertaining to the currently assessed query and the actual state of the previously assessed query. The proposed security scheme consists of a CHMM-based inference algorithm deriving the most probable state for a given query.

- A Hierarchical Distributed Architecture (HDA) for overlay networks offering Discovery Services in the future IoT-based business infrastructure, has been developed and its performance has been assessed in comparison with a Flat Distributed Architecture (FDA). A simulation of HDA and FDA has been developed for the EPCglobal Network on Planetsim. A more realistic emulation has also been developed on Planetlab. Both the analytical results and the experimental results have shown that HDA performs better than FDA, both in terms of network scalability and lookup query responsiveness.

- An extended architecture of the EPCglobal network, providing the flexibility to connect multiple supply chains, has been proposed. This architecture provides an interaction mechanism between intra-domain and inter-domain routing protocols in order to exchange information amongst supply chain stakeholders with regards to the exchanged physical objects. The proposed architecture enables implementation of distributed Discovery Services allowing traceability of items across multiple supply chains.

- An integrated cloud-centric platform offering real-time and accurate inventory management as a service for supply chain stakeholders has been proposed. Using the parameters loaded by each stakeholder into the platform, and using the data collected by the platform, the optimization service computes, via a probabilistic model, the optimal "inventory threshold", to be considered by supply chain stakeholders in their replenishment policy. By optimal we mean that the proposed "inventory threshold" minimizes the stock disruption likelihood, while minimizing the allocated resources.

1.4 Thesis Outline

The rest of this thesis is organized as follows. In the next chapter, we present a detailed description of the EPCglobal Network and its main components. Then, we discuss the limitations of the currently deployed discovery system in the EPCglobal Network. Finally, we survey and categorize the proposed discovery services systems throughout the literatures. In Chapter 3, we first give an overview of the currently deployed lookup system in the EPCglobal Network. Then, we present our proposed secured

DHT-based distributed architecture and we compare it to other architectures that have been proposed in the literature. In Chapter 4, we investigate the applicability of the "Anomaly Detection" approach for more secured lookup services in the EPCglobal Network. We propose two security schemes aiming at detecting suspicious lookup queries. Our first proposed security scheme consists of two algorithms based on the Gaussian model. Our second proposed security scheme is based on a Continuous Hidden Markov Model (CHMM). We also provide the performance evaluation of the proposed security schemes. In Chapter 5, we present our proposed Hierarchical Distributed Architecture (HDA) and we show how it relates to the widely used Flat Distributed Architecture (FDA). In Chapter 6, we present the theoretical model of the proposed HDA Architecture, its simulation on Planetsim and its emulation on Planetlab. We also provide both its theoretical and its experimental performance evaluation. In Chapter 7, we first present the intra-domain and inter-domain routing protocols, namely Electronic Product Code Open Shortest Path First (EPC-OSPF) and Electronic Product Code Border Gateway Protocol (EPC-BGP) for EPCglobal network. Then, we present our proposed threshold-based DS mechanism along with the network architecture in detail. Finally, we present and discuss the queuing analysis and the numerical results. In Chapter 8, we first present the architecture of our proposed cloud-centric platform. We then present the inventory optimization service offered by this platform. A detailed description of the inherent probabilistic optimization model is also provided. We also show performance evaluation of our proposed inventory optimization service. In Chapter 9, we conclude the thesis and propose some future research work.

Chapter 2

Lookup Systems in the Internet of Things

2.1 Introduction

In this chapter we present a survey on Discovery Services systems in the Internet of Things in general, with a perspective on the EPCglobal Network; a global computer network built to link trading partners throughout the supply chain business steps. Its business value consists in enabling those trading partners, via lookup services, to trace and track, accurately and real-time, individual physical items in the supply chain. The incessantly exponentially-growing number of physical items exchanged between supply chain stakeholders requires scalability of the EPCglobal Network. The sensitive nature of the data being exchanged on the EPCglobal Network requires firm security measures. The possibility to read the RFID tags data without consent raises completely legitimate privacy concerns for the consumers. The security breaches observed in the Internet, along with the sensitive nature of the EPC data, makes the EPCglobal Network components an attractive target to both passive and active attacks.

The EPCglobal Network consists of three components interacting to ensure information flow between EPCglobal subscribers. They are Object Naming System (ONS), Discovery Services (DS) and EPC Information Services (EPCIS).

This chapter is structured as follows. Section 2.2 presents the overall architecture of the EPCglobal framework and the major components of the EPCglobal Network. In Sections 2.3 and 2.4 we survey the lookup system architectures, currently deployed in the EPCglobal Network and consisting of the ONS and the EPCIS. we also discuss designs of the proposed architectures, their limitations and their security drawbacks. In Section 2.5, we survey in detail and we categorize different implementations and architectures of Discovery Services proposed throughout the literature. We also present an overview of the important role Discovery Services are expected to play in a generic IoT environment, with a focus on EPCglobal Network. Finally, we conclude the chapter in Section 2.6.

2.2 Background

With the advent of RFID communications [21–23], identification and tracking of individual objects have become viable in several industrial applications such as manufacturing, logistics, ticketing, and anti-counterfeiting. The communication network for the corresponding purposes can be implemented based on the IoT concept which was initially introduced in the EPC global standards one and a half decade ago [6], [3, 24, 25]. IoT aims at broadening the existing Internet of computers paradigm by defining a large-scale network of objects where the EPCglobal associates each object with an RFID tag so that tracing an object throughout the supply chain is possible. The RFID tag of an object is read once the object is relocated throughout the supply chain and/or the information about corresponding object is altered. Once it is read, the RFID provides the EPC [26], [27], [28] which is a unique number associated with the corresponding object.

2.2.1 Radio Frequency IDentification (RFID)

RFID stands for Radio Frequency IDentification. It is an identification and tracking system that makes use of radio waves in order to transmit wireless the digital data encoded on an RFID tag and captured by an active RFID reader. Usually the transmitted digital data consists of the identity of an object, an asset, a location or a person, to

which the RFID tag is attached. A typical RFID tag consists of an integrated circuit chip attached to an antenna, all wrapped in some protective package such as a plastic card. The specifications of each of these components are determined by the application requirements. An RFID tag stores the data in the integrated circuit chip and transmits it using the antenna upon a request of an active RFID reader in its vicinity.

We can distinguish between RFID tags based on various criteria such as the power source, the data storage and the radio frequencies. In terms of the power source criterion, RFID tags are basically of two types; active and passive. An active RFID tag is self-powered by its own battery while a passive RFID tag comes with no battery. Although the presence of a battery consequently makes the active RFID tags larger, heavier and more expensive, it also provides longer transmission ranges and larger data storage. A third type, referred to in the literature as semi-passive or semi-active, is a combination of the two categories defined above. A semi-passive (semi-active) RFID tag holds an internal battery used only to power its circuit while data transmission relies on the reader for energy. This dependence on the RFID reader to supply energy for the neighboring RFID tags is called Energy Harvesting. Table 2.1 illustrates the power source based classes of the RFID tags.

Table 2.1 RFID tags classification based on the power source

Category	Battery Presence	Circuit Power Source	Transmission Power Source
Active	Yes	Battery	Battery
Passive	No	Energy Harvesting (Reader)	Energy Harvesting (Reader)
Semi-passive (Semi-active)	Yes	Battery	Energy Harvesting (Reader)

Regarding the classification of the RFID tags based on the data storage criterion, there are basically two categories; Read-only and Read/Write. This classification depend on the type of the chip of the RFID tag. RFID tags with Read-only chips are built to hold unique information that can not be altered while the ones with Read/Write chips allow

the RFID readers in their vicinity to add new data or to alter the existing data in their integrated circuit chips.

Another classification of the RFID tags can be based on the radio frequencies they operate at. Three basic frequency bands are of common use nowadays; Low Frequency (125/134KHz), High-Frequency (13.56 MHz) and Ultra High-Frequency (850 MHz to 950 MHz). These categories differ in terms of the read range and the data rate. The RFID applications varying requirements specify which frequency band to use.

An RFID reader consists of a microprocessor controlling a radio frequency transmitter/receiver. As illustrated in Figure 2.1, it sends information requests, in the form of energy fields, to "wake up" the close enough RFID tags which respond by communicating their stored data. Once the reader receives the tag's data, it passes it to a computer system through a communication interface (wired or wireless) to be processed according to the application's requirements.

The widespread use of the RFID technology is due particularly to three key factors. First, the outstanding and increasing reliability of the RFID-based identification systems represents the major appealing feature for all organizations having to manage asset identification and tracking. Furthermore, the continuously decreasing cost of the RFID tags and equipment has tremendously aided decision makers in different organizations in the approval process of the RFID technology. Moreover, the standardization efforts regarding the RFID technology have led to the establishment of international stable RFID standards. These standards have brought valuable assurances to the potential RFID technology acquirers with respect to the complicated interoperability problems. As a result, the RFID technology has enjoyed acceptance in a variety of domains such as commerce, education, sports, healthcare and transportation. Although the RFID technology has been successfully used for years in a variety of internal applications within organizations, the emerging standards has opened the door wide for new applications related to inter-organization asset tracking such as in supply chains.

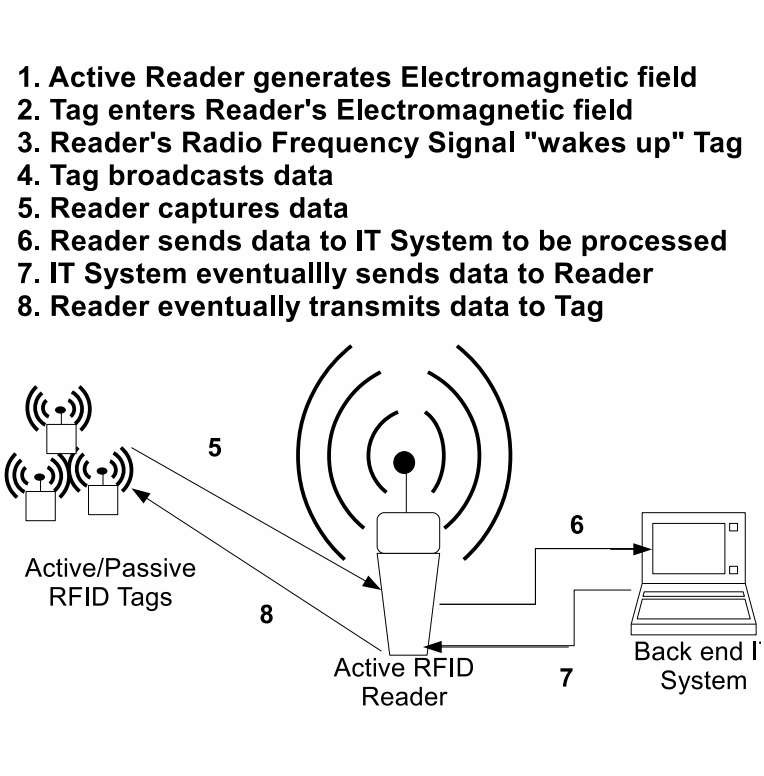


Figure 2.1 RFID Reader-Tag Interaction

2.2.2 Electronic Product Code (EPC)

Internet, complex as it is today, is emerging ceaselessly towards an even more complex system called the “Internet of Things” (IoT); a novel paradigm that will make of everyday things/objects potential Internet nodes, which can generate data and interact with each other to reach a common goal. This vision is strongly supported by the important advances made recently in several fields such as the wireless/wired communication, the sensing technologies, the software/middleware components and the hardware devices, especially in terms of the shrinking of the size, reduction of the weight, reduction of the cost and decline of the energy consumption for the radio devices [3]. The first use of the term “Internet of Things” has occurred as part of the EPC development. EPC refers to a framework for an extensible suite of unique identifiers designed to uniquely identify

physical objects, loads, locations, assets and other entities which are to be tracked, or otherwise identified [26].

EPC has been developed by the MIT Auto-ID Center, a consortium of over 120 organizations, as an industrial standard for global RFID usage to replace the ubiquitous Universal Product Code (UPC), also called the bar code. In October 2003, the MIT Auto-ID Center has been replaced by two separate entities; the Auto-ID Labs entity and the EPCglobal entity which represents a joint venture between the European Article Numbering (EAN) International (which changed its name into GS1 in 2005) and the Uniform Code Council (UCC) (the Numbering Organization in the USA which also changed its name into GS1 US in 2005). The Auto-ID Labs entity has been assigned as a task further development of the EPC/RFID technology, while the EPCglobal entity was responsible for the development and maintenance of the new EPCglobal Network. The purpose of this new network is to enable real-time and accurate data sharing between trading partners regarding physically exchanged items. The shared data is collectively stored and retrieved by each of the trading partners during each of the business events characterizing the life cycle of the exchanged items.

The specifications of the EPC identification scheme have been defined so that the needs of various industries are supported. Conversion procedures have been defined for the existing coding schemes whereas new coding schemes have also been designed where necessary [26, 29]. Although the EPC standards deal with a variety of encoding schemes and name spaces, the components of most EPCs include three main fields and three header fields, as illustrated in Figure 2.2. The three main fields are the company prefix, called also the manager number, the object class, called also the category code, and the serial number. The three header fields are the header, the filter and the partition. The company prefix is an identifier assigned by GS1 to a managing entity. The object class is a unique reference identifier assigned by the managing entity to a specific class of physical objects, locations or assets. The serial number is a unique identifier assigned by the managing entity to each and every single physical object, location or asset. The partition field indicates the number of bits in the GS1 Company Prefix field and the Object Class field. The type of EPC scheme at hand is determined by the value of the header field corresponding to the first eight bits of the sequence.

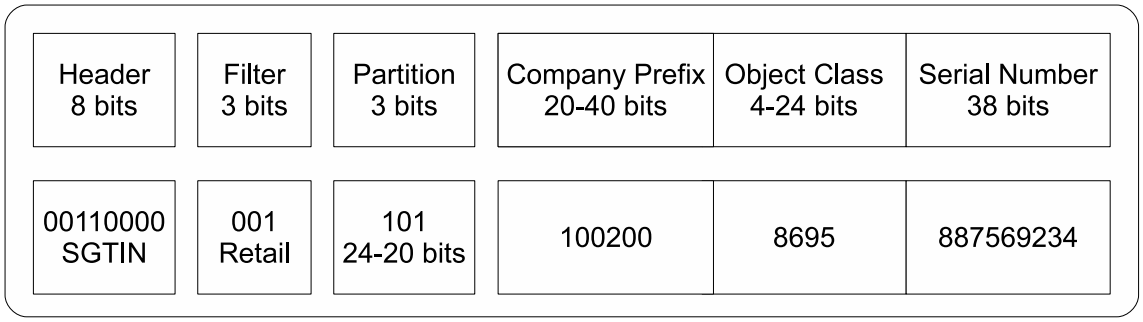


Figure 2.2 The SGTIN-96 EPC Format

The use of the RFID/EPC tags has multiple merits over the bar code. The main advantage characterizing RFID/EPC identification scheme appertains to its granularity; the fact that it identifies objects on the item level rather than on the product level. This feature enables targeted item-centric product recalls. It also has much less issues with scanned items orientation. Moreover, RFID scanning is much faster, much more accurate and provides real-time data access. Table 2.2 summarizes the merits of the RFID/EPC technology over the traditional bar codes technology.

Table 2.2 Comparison of RFID tags with bar codes

	Bar Code	RFID
Identifies objects on the item level	No	Yes
Scans multiple items simultaneously	No	Yes (Hundreds)
Is automated and much more accurate	No	Yes
Requires Line of Sight to scan items	Yes	No
Requires the scanned item to be very close to the scanning device	Yes	No

2.2.3 EPCglobal Architecture Framework

EPCglobal has set up a framework, called the EPCglobal Architecture Framework, for the future business infrastructure. The EPCglobal Architecture Framework is an open and vendor-neutral collection of hardware, software, data standards and core services, devoted to the common goal of item-level EPC data sharing regarding products that

move in the supply chain [6]. This framework defines a complete protocol stack aiming at full real-time traceability of items in the supply chains.

Figure 2.3 shows the three main modules of the EPCglobal Architecture Framework; the identification module, the data capture module and the data exchange module. The identification module consists of placing an RFID tag, which stores a unique EPC, on each physical object to be tracked in a supply chain.

The data capture module refers to the process of retrieving the data stored in the RFID tags, which may be the assigned EPC alone. The RFID tag is designed in a manner to disclose the stored data EPC code whenever it is in the vicinity of an active RFID reader. The disclosed EPC, along with other relevant information regarding the product and the current business event, are stored within information servers following specifications defined by EPCglobal.

The data exchange module ensures real-time information flow between trading partners. This information flow takes place thanks to querying services offered by the data exchange module. It enables supply chain stakeholders to localize and retrieve relevant information with regards to the exchanged physical objects. The retrieved data includes information regarding the movement of the exchanged physical objects throughout their life cycles. This data exchange is the major reason why the EPCglobal Network has been designed. Discovery of the information sources holding the targeted data represents the core service of EPCglobal Network.

2.2.4 EPCglobal Network

The EPCglobal Network is a worldwide network developed to ensure global interoperability between stakeholders in supply chains. It basically aims at increasing visibility and efficiency throughout the supply chain, and assuring higher quality information flow amongst trading partners [6]. It consists of three components interacting to ensure information flow between EPCglobal subscribers as illustrated in Figure 2.3. The three components are Object Naming System (ONS), Discovery Services (DS) and EPC Information Services (EPCIS). The ONS and Discovery Services are lookup systems

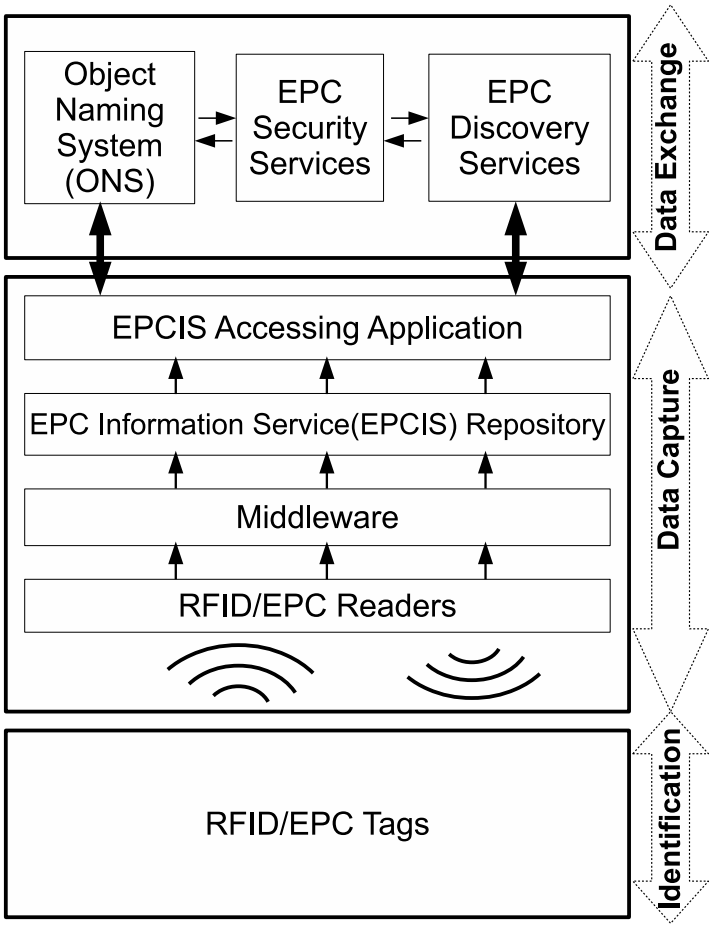


Figure 2.3 The EPCglobal Architecture Framework Components

ensuring linkage between trading partners within supply chains, while the EPCIS represents one of the information sources to which the ONS and Discovery Services may point.

ONS is a DNS-based lookup service ensuring a link between the trading partners in order to share and exchange EPC related data. It provides mapping information between EPC and Uniform Resource Identifier (URI), which is the server address hosting the EPC information related to the EPC in hand. The ONS maps EPCs to various forms of URI, such as EPCIS, Hypertext Markup Language (HTML), eXtended Markup Language Remote Procedure Call (XML-RPC) and Web Services (WS).

Discovery Services refer to a suite of services enabling any user, subject to authentication, to retrieve all relevant data appertaining to a given EPC. They are analogous to search engines of the Internet in that they represent tools to “discover” all “relevant”

information sources containing data about a given EPC in the EPCglobal Network [6]. Both the ONS and the Discovery Services require strong security mechanisms in order to authenticate the subscribers and to authorize their actions based on their access rights.

EPC Information Services (EPCIS) is defined by EPCglobal as a standard data sharing interface between applications capturing EPC related data (and eventually storing it into persistent repositories) and those querying such data [30]. As illustrated in Figure 2.3, EPCIS sits at the highest level of the data capture module of the EPCglobal Architecture Framework, above the middleware component which collects and filters raw tag reads, and then compiles the collected data with respect to the current business process step during which the EPC data capture has taken place [30]. EPCglobal does not specify how the EPCIS services and the EPCIS repositories should be implemented.

The EPCglobal Network promises, via the traceability services provided by its components, wide business visibility to supply chain stakeholders. This real-time, low-cost and accurate visibility, provided at the item-level, would allow companies to increase their profitability while increasing customers satisfaction. The core foundation of those traceability services is the lookup function, which consists of localizing sources pointing to information corresponding to the inquired EPC [7].

2.3 EPC Information Services (EPCIS)

The EPC Information Services (EPCIS) [30] refer to a network-based service that stores, hosts, and provides access to EPC information enabled by RFID-aided supply chains with respect to a specific item. The EPCIS is one of the information sources a discovery service, such as the ONS, may point to. It provides a repository (usually a relational database) for both event data and master data. Event data is data related to business processes taking place between trading partners, and master data refers to additional data providing the necessary context for interpretation of the event data [30]. This additional data includes historical data and semantic information related to the business process during which EPC data is being collected. The EPCIS provides a capture interface allowing EPC data storage in the EPCIS repository and a query interface providing access to the repository data.

Since EPCIS repositories represent the major source of information in DS systems, EPCIS has drawn much interest in the literature. Routing queries, specific to a given EPC, to the appropriate EPCIS is a crucial step in the EPC data retrieval process. The ONS [7] provides only the location of the EPCIS that has been originally assigned to the inquired EPC. Discovery Services [31] are supposed to provide pointers to the locations of all relevant EPCIS repositories holding data related to the inquired EPC.

Itsuki and Fujita [32] have proposed the application of the DHT technique [33], used in P2P network systems and called Chord [34], to the EPCIS system. They have claimed that this would reduce the EPCIS retrieval load on the ONS root. A simulation study has been carried out to compare the proposed system with the ONS. It has been claimed that the proposed system performs better than the ONS in terms of the number of accesses.

EPCglobal has defined generic specifications for the EPCIS with no specific implementation suggestions. In [35], an implementation method of the EPCIS resolution service, integrating EPCIS directory with Physical Makeup Language (PML) which has evolved from XML, has been proposed. Although the EPCIS usually uses a relational database as a service backend, the authors in [35] have chosen the directory service as the EPCIS storage carrier, and PML document as carrier of the returned results. A 2-level hierarchical cache structure has also been adopted.

Unlike the traditional ONS resolution service which returns URI addresses of the EPCIS servers where relevant information about the queried EPC can be found, the proposed resolution service in [35] provides the relevant information itself. A simulation study has shown that the proposed caching structure has improved the resolution service by 56.7% even under considerable pressure.

Huang et al. [36] have proposed a distributed ePedigree architecture, based on a set of EPCIS services, in order to overcome the problems of scalability and privacy of the current ePedigree centralized system. An ePedigree refers to the historical records of a given product throughout its product life cycle. The advent of the RFID/EPC technology has upgraded ePedigree creation and verification from the traditional cumbersome paper-based processing to the new efficient digital processing.

The EPC-based ePedigree digital processing has come with the item-level tracking possibility. Such fine granularity of object tracking would allow for a suitable solution

to the critical drug counterfeiting problem. Although the EPCglobal Network presents an appropriate platform for ePedigree digital processing, its centralized nature brings multiple drawbacks; its vulnerability to single point of failure, its lack of scalability, its additional overload in local databases and its privacy exposure. To address these drawbacks, a DHT-based distributed EPCIS architecture for ePedigree management has been suggested [36] to replace the traditional centralized ONS. Mutual authentication between trading partners is performed by way of certificates issued by a Certification Authority (CA).

A NoSQL big-data-oriented RFID-centric repository schema has been described in [37]. Another NoSQL-based approach for EPCIS repositories construction has been investigated in [38]. The proposed schema aims at offering EPC information exchange and traceability services. A Push Service Solution based on the existing Web Service standard and distributed storage has been proposed in [39], the goal being to realize the cooperative work between corporate partners and effectively solve the real-time data storage problem.

2.4 Object Naming System (ONS)

The ONS is a DNS-based lookup service ensuring a link between the trading partners in order to share and exchange data. The function of the ONS consists of a number of lookup operations taking the EPC as input and returning pointers (e.g. IP addresses) to the corresponding information services [7]. The ONS, as it is known today, relies completely on a centralized ONS root.

The function of the ONS consists of a number of lookup operations taking the EPC as input and returning pointers (e.g. IP addresses) to the corresponding information sources [7]. The use of the DNS infrastructure requires encoding of the EPC into a domain namespace syntactically correct. This domain namespace will then be used to query relevant information from the DNS system. The Name Authority Pointer (NAPTR) DNS record, detailed in [40], has been selected to map the ONS queries leveraging the DNS system.

Many works in the literature have looked into extending the existing ONS into Discovery Services. A lightweight EPC Discovery Service platform, called Product Trace Service Platform (PTSP), have been proposed in [15]. PTSP platform extends the existing ONS to efficient and non-expensive track-and-trace services between Enterprise Applications, using web services. Gyeongtaek et al. [41] have proposed a novel architecture for EPC data discovery. The discovery starts with a request to the ONS asking for a Uniform Resource Locator (URL), pointing to the list of all the corresponding information sources.

2.4.1 ONS Architectures

A number of research papers dealing with the architecture and design problems of the ONS system have been published. Both the scalability and the robustness requirements have been dealt with in [16] which has proposed a multi-root distributed architecture of the ONS based on the DHT technique. Although this architecture inherits all the well known merits of the DHT technique and also tries to conserve as much as possible the existing ONS root infrastructure, it still comes with two main drawbacks. First, the security issue has not been discussed at all and second it is based on the DNS system which may not be suitable for the future Internet.

A DHT-based architecture for the ONS, called the Object Information Distribution Architecture (OIDA), has been proposed in [9]. In addition to the robustness and the scalability inherited from the DHT, it has been claimed that the OIDA architecture could offer a data access control mechanism and also could enhance the privacy of its users. Although the OIDA architecture presents several advantages, it does not process all the necessary design and security requirements of a critical naming service such as the ONS; e.g., the availability requirement has not been discussed. It also does not specify who is responsible for the DHT nodes and how the DHT is built. In [42], the authors have experimented the open source solution Fosstrak in a business environment. They have designed and run typical pilot tasks on Fosstrack to test the ONS.

An ONS architecture called Multipolar ONS has been presented in [18]. This architecture solves the problem of unipolarity of the ONS root and also touches to the security

issues. A change to the unipolar design, allowing distribution of the ONS platform control amongst the involved parties, has been proposed. Moreover, the security aspect of the ONS has been examined via an outlook on multipolarity of the ONS Security Extensions (ONSSEC) protocol, an adaptation of the DNS Security Extensions (DNSSEC) protocol [43–46] for ONS data authentication. This architecture has the merits to introduce moderate changes to the existing ONS design (unlike [9]), although the only security requirement that has been dealt with is data authentication.

2.4.2 ONS Limitations

Currently, the ONS relies completely on a centralized ONS root managed exclusively by Verisign. It is set up on top of the DNS Top Level Domain ".com" and is called "onsspec.com". The ONS root is similar to the DNS root except that, unlike the DNS root, the ONS root is not replicated in multiple geographical locations. The currently deployed ONS architecture has three major drawbacks, they are the following:

- First, the ONS root unipolar design makes acceptance of the ONS among supply chain stakeholders all around the world a major problem, since it is fully controlled by a single entity (Verisign) [9, 10]. Moreover, the unipolar architecture makes the ONS an attractive and easy target to some security attacks. Different scenarios depicting possible attack models in such a unipolar architecture have been discussed in [10], while highlighting the political repercussions of such a design choice. Furthermore, the centralization and lack of redundancy of the ONS root create a single point of failure of the ONS, and hence of the whole business infrastructure.

- Second, being based on the DNS system, the ONS inherits all the well-studied and well-documented weaknesses of the DNS including its poor performance, its complex configuration and its lack of security [8, 9]. The DNS system has always been an attractive target to Distributed Denial of Service (DDOS) attacks and also to Man in the Middle (MITM) attacks targeting the DNS resolution paths. Vulnerability to the Cache Poisoning attacks is another DNS weakness that may prove absolutely disastrous to the critical systems, such as the business infrastructure, relying on DNS for naming services.

- Third, the ONS has been designed to provide the information sources based on the company prefix of the EPC. As a result, only the manufacturer information sources are provided in response to lookup queries addressed to the ONS.

2.4.3 ONS Security Issues

Because the ONS is a DNS-based lookup service, one obvious option to secure the ONS is to secure the DNS. In order to secure the DNS system, a set of extensions, called DNSSEC [43–46], have been proposed by Internet Engineering Task Force (IETF). The fact that DNSSEC protocol tries to add security to the DNS system while maintaining full backward compatibility makes it unsuitable for critical infrastructures such as the future business infrastructure. Moreover, DNSSEC offers only data integrity and origin authentication using digital signatures while the other security requirements have not been considered [47].

The ONS security has been investigated by Fabian et al. in [48]. Three requirements of a secure ONS system (namely confidentiality, data integrity and availability) have been highlighted and general mitigation techniques have been proposed. However, the availability requirement has not been considered.

Jing Sun et al. have proposed an adjusted a Public Key Infrastructure (PKI), called Lightweight PKI (LPKI), to build a trustworthy ONS system [49] in terms of exchanged data confidentiality, data integrity and origin authentication. The other security requirements (i.e., availability and privacy) have not been considered.

The privacy requirement has been discussed in [14]. A set of security prototypes have been proposed to protect the privacy of the querying entities. The proposed prototypes have been designed in such a way that the current architecture of the ONS is not radically changed. Although the privacy requirement has been studied thoroughly, the other security requirements have not been discussed.

2.5 Discovery Services

The EPC Discovery Services refer to a suite of network services that enables efficient object track-and-trace capabilities across multi-party supply chains in the EPCglobal Network [31]. The EPC Discovery Services can be regarded as a registry gathering locations of all EPCIS servers and other information sources holding information pertaining to a certain item identified by its unique EPC [6]. Conceptually, Discovery Services can be viewed as access-controlled "search engines" for the Internet of Things. The gigantic bulk of EPC related data, generated at the serial number level by supply chains, would place an unreasonable burden on the ONS and the EPCIS in the form they are designed today.

Unlike the ONS and the EPCIS for which EPCglobal standards have been defined, Discovery Services are not yet specified in official EPCglobal standards. Their concept has been recognized by IETF with the name of Extensible Supply-chain Discovery Service (ESDS) [50]. The ESDS provides applications with tools to identify all relevant information resources of a given item, and also to gain access to the information stored in those resources.

2.5.1 Discovery Services Implementations

Barchetti et al. [55] have developed and experimented a scalable Discovery Service as an extension of the framework FossTrak [56], an open architecture that implements most of the EPCglobal services and the entire EPCglobal protocol stack. To test the implemented traceability system, the authors in [55] have defined a use case simulating the main step of a typical pharmaceutical supply chain. The implementation consists of two phases: the configuration phase and the query phase. The configuration phase deals with configuration of the components of the Discovery Service, namely EPC data capture into EPCIS servers, EPCIS address indexation into the Discovery Service and ONS configuration specifying the corresponding Discovery Service link. The query phase consists of localizing and retrieving the EPC data associated with the inquired EPC.

Table 2.3 A comparison of DS systems proposals

	Features of the proposed DS solutions				
DS solution	**Approach**	**System Architecture**	**Control**	**Security**	**Privacy**
Multi-root ONS [16]	ONS-based	Multi-rooted, Centralized	Single Entity	N/A	N/A
OIDA [9, 18]	DHT-based	Decentralized	Multiple Entities, Not Equally shared	Data Integrity, Confidentiality, Authentication, Availability	Anonymity
MONS [10]	ONS-based	Centralized	Single Entity	Only Data Authentication	N/A
Name Service [51]	ONS-based, DHT-based	Decentralized	Multiple Entities	Data Access Control	N/A
Bridge Project [52, 53]	ONS-based, distributed via Web Services	Centralized	Multiple Entities, Not Equally shared	Data Integrity, Confidentiality, Authentication, Availability	Data protection
EPCDS [54]	DHT-based	Distributed	Multiple Entities	N/A	N/A
SHARDIS [11]	DHT-based	Distributed	N/A	Confidentiality	Protection against profiling
HDA and proposed security schemes (our work)	DHT-based	Hierarchical, Distributed, Cloud-based	Multiple Entities, Equally shared	Data Integrity, Confidentiality, Authentication, Access control, Protection against replay attacks	N/A

Although Gaj et al. [57] have surveyed a wide number of the latest trends in the industrial distributed systems from a communication point of view, the scalability issue in a large-scale inter-enterprise networks has not been discussed.

In [58], the authors have proposed an efficient framework for public data delivery in a smart city settings. The efficiency is devised in the form of a pricing function taking into consideration resource limitations, and quality and trust requirements. Simulation results have shown that the proposed framework performs better than other wireless sensor and mobile ad-hoc schemes, especially with respect to scalability. Li et al. [59] have proposed a web-scale service delivery for smart cities with a focus on addressing the scalability and the extensibility issues. The proposed service delivery system enables the stakeholders to collaboratively create novel services.

A Virtual Device (VD) can be defined as a collection of heterogeneous physical devices behaving as a single homogeneous logical device, in performing a given task. Composing a self-managing VD requires automation of the service discovery process, enabling the devices to discover, with little or no human interaction, the required services [60]. A distributed constraint satisfaction protocol has been devised for a more secure, more privacy-preserving and QoS-aware VD composition in MANETs, without a continuous broadcast service of advertisement messages [60].

BRIDGE, denoting Building Radio frequency IDentification for the Global Environment [52], [53], is an EU-funded project that aims to deploy the EPCglobal network in Europe through research and development activities. In BRIDGE, four DS approaches are proposed. The first approach is identical to directory lookup [61] which does not consider communication among discovery services and is called Directory-of-Resources (DoR). The second approach, namely Notification-of-Resources (NoR), inherits DoR and extends it by enabling a client to show interest about certain information by creating a subscription at the DS. The third approach is called Notification of Clients (NoC) where EPCIS servers provide any new information about a certain EPC to the DS. On the other hand, if a client shows interest in a specific EPC, the DS notifies all relevant EPCIS servers. An EPCIS sends notification to the clients informing them that there is new information about the EPC of interest while the client queries the respective related EPCIS servers. The fourth approach is called Query Propagation (QP) is a variant of the NoC while in QP the information is sent to the client by the EPCISs

immediately without the availability notification. DHT is reported by this EU project as the most promising approach, considering scalability as the predominant requirement for the discovery service design.

The authors in [62] have presented an integrated solution, combining Universal Plug and Play (UPnP) [63] and ONS [7], to the service discovery problem in the context of smart home applications. One of the major keys in smart home applications composition is real-time and accurate discovery of available devices in the home. Although UPnP is a standardized device discovery protocol, and is supposed to be sufficient for discovery of all types of devices in a constraint-free world, ONS comes in extremely handy for constrained devices.

2.5.2 Discovery Services Architectures

Numerous distributed architectures have been proposed in the literature to handle discovery services in the EPCglobal Network, either to overcome the limitations of the currently deployed ONS/EPCIS [10, 15, 17, 36, 41], or to improve previously suggested architectures [9, 10, 15–20]. Although, logarithmic scalability has been widely suggested in many proposals, only few works have proposed a solution to the expected super-exponential growth issue of the IoT, and especially of the future business infrastructure; the EPCglobal Network.

2.5.2.1 ONS-based Discovery Services

In [17], a distributed architecture for Discovery Services has been proposed. It has been claimed that the proposed architecture, based on extended ONS, offers better performance than the traditional data lookup system based on the traditional ONS. It improves the query forwarding mechanism. It also improves the routing performance in terms of the number of hops. Moreover, it is able to track-and-trace the items in the whole supply chain.

Gyeongtaek et al. [41] have proposed a novel architecture for EPC data discovery extending the ONS and the EPCIS Discovery Services (EPCISDS) in the EPCglobal Architecture Framework. The EPC related data retrieval process starts with a request

routed to the ONS for the URL of the EPCISDS. Then a subsequent request is directed to the EPCISDS to get the list of the URLs of all the relevant EPCISs that may hold data relevant to the inquired EPC. Finally, queries are sent to those EPCISs to obtain the desired data.

A lightweight EPC Discovery Service platform, called Product Trace Service Platform (PTSP), have been proposed in [15]. The PTSP is based on the ONS and the Web Service technology and aims construction of efficient and non-expensive track-and-trace services between Enterprise Applications. It offers an effective environment that provides companies with the track-and-trace capability without having to integrate the EPCglobal Network.

An efficient and flexible distributed architecture for name lookup in large scale applications has been proposed in [51]. This architecture consists of both an upgraded version of the ONS, supporting customized EPC encoding, and a scalable DHT-based register/-query DS system, allowing nodes to point to other nodes in a chain model.

In [36], the authors have proposed a service to collect information about an item following the E-Pedigree model. All the EPCs of the objects being shipped are listed in the E-Pedigree document which is initially created and encrypted by the manufacturer. The E-Pedigree document is forwarded to the next entity in the supply chain where new data is appended and signed electronically, and the document is forwarded to the next entity. Furthermore, the authors have proposed a distributed architecture for forward/backward tracing through the supply chain. Tracing up the supply chain can be done by using the General Manager Number (GMN) stored in an EPC [26] along with ONS [64], [10] which is able to provide the GMN to the corresponding EPCIS. It is worthwhile to note that this approach uses distributed ONS architecture based on DHT. The retrieved EPCIS address through the distributed ONS architecture is used in traversing the entire supply chain to locate information about a specific EPC. This approach can ensure scalability in item-level product tagging due to distributed operations.

2.5.2.2 Non ONS-based Discovery Services

Several DS architecture proposals have been designed on top of a DHT. Load balancing in DHTs has been investigated in [65]. The authors have proposed a distributed symmetric algorithm enabling peers to balance their loads proportionally to their capacities, through migration/reallocation of virtual servers. In [66], the authors have examined the potential of implementing EPCglobal Discovery Services based on DHT-based distributed architectures. It has been shown that the proposed distributed architectures improves the network capacity compared to the centralized architectures.

Talevski et al. [67] have proposed a framework for inter-enterprise service integration and collaboration in a context where disparate systems are being used. In [68], the authors first evaluated the negative impacts of demand uncertainty on the performance of a distributed supply chain, and then suggested a coordination mechanism to minimize those impacts. They claimed that the proposed mechanism improves the performance of the system in terms of the cost and the retailer's fill rate.

The centralized index-oriented DS implementation proposed in [69] claims a better support of the big data concept and parallel processing. A new storage schema of DS data has been presented and claimed to be much more efficient than that based on a Relational Database Management System (RDBMS) both in terms of concurrent discoveries and concurrent publishes. A more scalable, more comprehensive DS system, in that it is compatible with different RFID standards, has been described in [70]. A novel distributed discovery mechanism based on randomized architecture has been studied in [71]. It brings up a solution to both the chain-breaking problem and the single point of failure problem, while offering a Chord-comparable performance.

An efficient and flexible distributed architecture for name lookup in large scale applications has been proposed in [51]. This architecture consists of both an upgraded version of the ONS, supporting customized EPC encoding, and a scalable DHT-based register/-query DS system, allowing nodes to point towards other nodes in a chain model.

An intelligent layer, built on top of the EPC Network using Bayesian reasoning, has been proposed in [72] to enable enhanced tracking capabilities. This can be regarded as an extended DS system offering to the supply chain stakeholders enhanced visibility.

A distributed Discovery Services architecture, built on top of the Electronic Product Code-Border Gateway Protocol (EPC-BGP) [73], has been illustrated in [74]. The proposed architecture can be used to collect information about mobile RFID-assigned objects, regardless of their location; within the EPCglobal Network or otherwise. Continuous real-time update of the EPC-BGP routing tables [73], following each and every operation with regards to each and every object, would certainly introduce significant computation and communication overhead. The proposed architecture suggests threshold-based deferred updates to the routing tables, rather than continuous real-time updates. Simulation results have shown that the proposed architecture ensures low blocking probability of EPC tracking requests while significantly reducing the communication and computing overhead.

In [66], the authors have examined the potential of implementing Discovery Services and ONS in the EPCglobal Network based on DHT-based distributed architectures. Unlike many previously proposed works, no assumption has been made on the item visibility along the supply chain. It has been shown that the proposed distributed architectures improves the network capacity compared to the centralized architectures.

The authors in [17] have suggested an enhanced P2P-based distributed Discovery Services system, with more focus on improving both load balancing and routing efficiency in terms of the number of hops. Load balancing in DHTs has also been investigated in [65]. The authors have proposed a distributed symmetric algorithm enabling peers to balance their loads proportionally to their capacities, through migration/reallocation of virtual servers.

In [54], the authors have analyzed the applicability of distinctive distribution schemes and presented an approach that allows product managers to decide in which discovery service their data is to be stored. The corresponding study has presented a prototypical implementation that is based on the open source P2P protocol Juxtapose (JXTA). The distribution of data among independent discovery services entities is inevitable in order to be able to cope with data volumes and request loads expected for RFID-enabled supply chains. This architecture has been designed to enable independent providers of DS functionality to form a federated network with a commitment to company overlapping collaboration. Clients and companies remain invisible in the network as they only interact with their trusted EPC Discovery Service (EPCDS) provider. It is argued that

control of mapping of a specific EPC to an EPCDS can reveal the company prefix in the EPC.

In [20], the authors have proposed an Aggregating Discovery Service (ADS). ADS is responsible for both forwarding the client queries to relevant EPCISs and aggregating the EPCIS responses into the client request after synchronizing the responses so that client complexity is reduced. Furthermore, ADS guarantees confidentiality of clients and delivers full and correct information for the requester and eliminates the need for fine-grained access control replicated at DS level. ADS is a centralized service that provides two interfaces, namely notify and query interfaces. Notify interface is used to inform the ADS about read events to be shared within the EPCglobal network. ADS receives the EPCIS URL of the submitting peer along with one or more EPCs that have been handled by the corresponding entity. The ADS maintains an association between submitting EPCISs and submitted EPCs so that ADS is able to determine all EPCISs that hold more information about an EPC. The query interface parses the query to extract relevant EPCs upon arrival of a client query request. ADS uses its internal database to look up the URL of EPCISs which are relevant for this query, and it forwards the original query to the relevant EPCISs. Each EPCIS replies back with the information about the EPCs of interest. Finally ADS aggregates and returns the results to the client.

2.5.3 Discovery Services in IoT

Discovery Services represent “search engines” of the future IoT. They refer to a suite of network services enabling IoT users to localize target devices and/or information sources. Several protocols offering Discovery Service, either in limited contexts or in large-scale applications, have been designed to facilitate the implementation of Discovery Services; i.e., Universal Description, Discovery, and Integration (UDDI) and EPC [29]. The ONS can be viewed as a primitive Discovery Service restricted by design to localizing the manufacturer’s data sources within the EPCglobal Network.

An overview of different approaches, established for implementing Discovery Services in the IoT, has been presented and assessed in [75]. Most of the proposed approaches in [75] would not scale in an IoT environment. Moreover, lookup query responsiveness has not been taken into consideration in the proposed comparative assessment.

Guinard et al. [76] have presented a comprehensive process allowing stakeholders to search efficiently for services in the context of a complex Internet of heterogeneous devices. Romer et al. [77] have surveyed the concepts searching real-world entities in the Web of Things and their implementations. In [78], the authors have surveyed the challenges of the integration of heterogeneous distributed enterprise architectures. Such integration is even more challenging in IoT-based industrial applications that use the new technologies such as the sensing technology in the wireless sensor networks (WSN) and the RFID technology.

Wu et al. [79] have introduced a distributed model for federated RFID data streams aiming at tracking and tracing items efficiently in the IoT. It is built on top a flat P2P overlay. They have also proposed and evaluated a set of algorithms aiming at improving the lookup efficiency and the scalability of the proposed system.

In [80], a security protocol enabling secure and scalable business operations in supply chains has been proposed. It has been claimed that the proposed protocol provides scalability, security and adaptability services for IoT deployment of large scale mobile RFID.

A hybrid P2P system, combining the advantages of both structured P2P networks and unstructured P2P networks, has been presented in [81]. The proposed system consists of a two-tier hierarchy P2P networks; the top tier of the hierarchy consists of a structured P2P network representing the core of the proposed system, while the bottom-tier consists of a set unstructured P2P networks, each of which is attached to a node in the core network. The impact of applying network coding to a P2P file sharing system has been investigated in [82], aiming at a higher throughput and a better resilience to link failure and churn.

Scalability of a distributed architecture, designed for Internet-scale information monitoring applications, has been discussed in [83], with the assumption that Continual Queries (CQs) are used as its primitives to express information-monitoring requests. A smart service partitioning scheme has also been introduced at the P2P protocol layer in order to achieve both good load balance and good system utilization.

Scalability of the IoT has been looked at from a data compression angle in [84] in which Li et al. have devised a framework for more accurate data sampling and acquisition in

wireless sensor networks based on the Compressed Sensing (CS) theory, while saving energy and communication resources.

Scalability and Self-configuration capability of service and resource discovery protocols have been discussed in [85]. A scalable distributed architecture has been proposed for service/resource discovery in large-scale IoT networks. A special node, referred to as "IoT Gateway", gathers and stores all information on resources of a constrained local wireless network. This gateway is also part of a P2P overlay, hence offering a Zeroconf-based, distributed and scalable data publish and retrieval system.

A novel gossip-based protocol, called CREW, for flash dissemination has been introduced in [86]. While scaling well and being robust with regards to various types of failures, the proposed protocol ensures rapid dissemination irrespective of network or content size.

The authors in [87] have proposed an approach for efficient deployment of tasks on top of computationally capable IoT devices. The constraint programming model has been used to minimize the overall cost, i.e., processing response time, of the deployment. It has been claimed that the proposed approach offers more scalability with regard to data processing in the IoT.

In a Cloud of Things environment, designed to provide a given service, it is crucial to provide the Cloud agents with the capability of remotely enabling the desired service through IoT devices (such as smart phones). Obviously, these IoT devices need to be localized (discovered) first [88–91]. A resource discovery distributed algorithm based on a gossip policy that selects IoT devices based on predefined sensing capabilities is discussed in [88]. This algorithm has been upgraded in [92] to enable efficient deployment of virtual sensor networks on top of a subset of given IoT devices. Service discovery in a Smart City context has been studied in [93]. The authors have proposed an architecture enabling users to discover, efficiently and in a transparent way, data-sources that are appropriate for their business context.

A generic search engine offering a resource discovery framework for the IoT has been proposed in [89]. The proposed search engine offers automatic discovery of resources, their properties and capabilities as well as the means to access them. In [94], the authors

have proposed a software framework composed of a Context Module and a Search Engine for semantics-based discovery of context-aware IoT devices and their interaction.

The authors in [95] have surveyed and categorized the current IoT resource discovery techniques, and compared their merits and limitations. In [91], an integrated IoT system architecture, driven by autonomous service/device discovery and integration, has been proposed; the main goal being to offer a framework for autonomous composition of IoT applications. This framework would enable users to dynamically select and orchestrate the published services in order to create customized IoT applications. A decentralized service discovery and selection model, based on the bio-inspired Response Threshold Model, has also been proposed in [96]. With proliferation of the number of IoT entities and its incessant growth, more research efforts have been put into scalable and efficient approaches for entity discovery such as in [97], which proposes an extension of the Distributed Context eXchange Protocol (DCXP) based on the publish/subscribe model. A P2P-based context-aware semantics-based service discovery mechanism has been proposed in [98].

2.5.4 Security in Discovery Services

Data and information stored in EPCIS repositories are commercially sensitive. This is the reason why supply chain stakeholders take the security requirement in DS very seriously. The security requirement in DS has been considerably studied in the literature. A centralized service aiming at DS user authentication has been proposed in [99], in which the focus has been put into data availability and data confidentiality. In [11], the authors have focused on privacy issues at the discovery phase of RFID information services. They present a DHT-based DS architecture to be deployed as a P2P network, offering high scalability and performance. Furthermore, it provides new mechanisms to protect client privacy from eavesdroppers.

An access control mechanism has been proposed in [41], in order for companies to assess their EPC data access rights depending on the profile of the inquirer. It is obvious that one of the core requirements in the design of a DS system is to enable supply chain stakeholder to fully control their own EPC data. Such control implies enforcement of both authentication and access control mechanisms allowing those stakeholders to

decide which information to make available for which organization. The Web Service Description Language (WSDL) [100, 101] has been used for the implementation of the Discovery Service.

In [80], a security protocol enabling secure and scalable business operations in supply chains has been proposed. It has been claimed that the proposed protocol provides security services for IoT deployment of large scale mobile RFID.

2.5.5 Privacy in Discovery Services

Client privacy in a DS system stands for the ability of its users, i.e., clients initiating queries towards the DS system, to retrieve data while the intension of the queries are kept hidden. Private Information Retrieval (PIR) protocol allows users to retrieve data from a database server without revealing which item is retrieved, thus the intention of the query originator are not known to the database server. PIR protocol has been used in [102] to implement client privacy in an EPCglobal DS system. Instead of sending the EPC of the inquired item/product as a keyword for the query, the authors have proposed to send a vector of keys derived from the inquired EPC and some secrets only known by the originator of the query. When the DS server receives the query originator's vector, it runs a suite of computations and then sends back its fuzzy response. The query originator can then reconstruct the targeted EPC records using the previously exchanged private secrets.

A privacy-enhanced P2P-based discovery service architecture has been introduced in [11]. The focus in that proposal has been to enhance the client privacy against profiling using a secret-sharing mechanism applied to the information of interest. The implementation has been carried out by insuring confidentiality of the client's query from eavesdroppers, even in the case no earlier key distribution amongst those clients has been arranged.

In [103], the authors have looked at discovery services from a privacy standpoint. They have first illustrated the complexity of the information exposure problem in pervasive computing environments. A probabilistic approach has been adopted to protect sensitive information exchanged between users and service providers. The approach is progressive in that users and service providers expose their information gradually, while

checking whether further exposure is necessary or not. This approach has been experimented and results have shown that the security requirements are met more efficiently. However, it has a major limitation with regards to its use within user-crowded areas.

Huang et al. [36] have proposed a distributed ePedigree architecture, based on a set of EPC Information Services, in order to overcome the problems of scalability and privacy of the current ePedigree centralized system.

2.6 Summary

In this chapter we first have presented a detailed description of the EPCglobal Network and its main components, we have surveyed proposals in the literature with regards to lookup system in the Internet of Things in general, with a perspective on the EPCglobal Network, and we have discussed the limitations of the currently deployed discovery system in the EPCglobal Network. We have also categorized the proposed works into two major categories; namely legacy-based architectures (ONS-based) versus clean-slate-approach-based architectures. A summary of the proposed discovery services systems and their features has been presented in Table 2.3.

Chapter 3

Secured Distributed Lookup Service in the EPCglobal Network

3.1 Introduction

The EPCglobal Network is a worldwide network developed to ensure global interoperability between stakeholders in supply chains. It basically aims at increasing visibility and efficiency throughout the supply chain, and assuring higher quality information flow amongst trading partners [6]. This can be achieved through accurate and real-time exchange of EPC data among companies involved in a supply chain. EPC data refers to all relevant data collected to describe the exchange process of physical objects amongst trading partners in a supply chain, as well as the context of such exchange.

Obviously, accurate and real-time exchange of the collected EPC data amongst distant trading partners in a supply chain requires secured network services. All these services are based on one fundamental service; the lookup service, also called discovery service. Throughout this thesis, unless otherwise stated, the expressions “Lookup services”, “Discovery Services” are used interchangeably.

This chapter focuses on the lookup service in the EPCglobal Network. Such lookup service aims at enabling users to locate and retrieve all relevant data related to a specific EPC. The EPCglobal Network consists of three components interacting to ensure information flow between EPCglobal subscribers; Object Naming System (ONS), Discovery

Services and the EPC Information Services (EPCIS). The ONS is a central lookup service whose main function is to localize information sources, such as Uniform Resource Locator (URL), of a given EPC. The Discovery Services is a suite of lookup services enabling any user, subject to authentication, to retrieve all relevant data regarding a specific EPC. The EPCIS is a set of EPCglobal standards for hosting, storing and sharing EPC related information between trading partners.

The EPCglobal Network has been developed in order to provide trade partners (Suppliers, Manufacturers, Distributors and Retailers, etc.) with real time, and accurate data sharing network services regarding all the exchanged physical objects [6]. Such real time data sharing would offer trading partners a wider visibility of the supply chain, resulting in a reduction of cost (e.g. efficient product recall and shelf replenishment [12]) and a reduction of loss (e.g. efficient anti-counterfeiting and anti-theft [13, 14]).

Although the real-time traceability and accurate visibility of supply chains, offered by the EPCglobal Network, gives companies a strong incentive to subscribe to the EPCglobal Network, companies would accept to share their data only if the infrastructure offering the sharing capability, the EPCglobal Network in our case, is salable, secure, easy to use, and more importantly not controlled by a single entity.

Currently, the lookup service in the EPCglobal Network relies completely on the ONS. Being a DNS-based centralized system, the ONS brings up numerous concerns, detailed in Section 1.2; namely the acceptance problem, the DNS legacy problem and the discovery problem. The acceptance problem refers to the unipolar control of the ONS root and its political repercussions [9, 10]. The DNS legacy problem is a result of building the ONS on top of the DNS system [104], which makes the ONS inherit all the well-studied and well-documented weaknesses of the DNS. The discovery problem refers to the critical limitation of the ONS regarding its ability to find all relevant data about a given EPC in the whole EPCglobal Network. Currently, only the manufacturer information sources are provided in response to ONS queries.

In order to assure the aimed quality of information flow amongst trading partners in a supply chain, two main challenges have to be taken care of while designing a lookup service. First given the enormous number of objects to be tracked in the EPCglobal Network and also because of its political sensitivity to a centralized control, a fully distributed, more scalable and equally-controlled architecture has to be adopted. Second,

considering both the number and the severity of the threats in the Internet today, security should be regarded as a vital design principle in the in the proposed architecture.

Given the drawbacks and limitation of the ONS system, mentioned above, we propose herein a fully distributed and secured EPC lookup system, which is based on the DHT technique, which has proved its robustness and efficiency in P2P file sharing systems. The idea is to upgrade the existing local ONS servers, belonging to the EPCglobal subscribing companies, into nodes of a DHT storing EPC related data. This upgrade aims at palliating the three major drawbacks, presented in Section 1.2, of the currently deployed ONS.

This chapter is structured as follows. In Section 3.2, we present the currently deployed lookup system in the EPCglobal Network and we discuss its limitations. In Section 3.3, we define the major design requirements any EPC lookup system has to fulfill and we describe a DHT-based distributed architecture that we compare to other architectures that have been proposed in the literature. Section 3.4 focuses on the security aspect of our distributed lookup system architecture. It first describes the security requirements that have to be fulfilled by an EPC lookup system, and then explains how those requirements are met in our proposed architecture. Finally, we conclude the chapter in Section 3.5.

3.2 Currently Deployed Lookup System

The EPCglobal Network currently relies on the Object Naming Service (ONS) for EPC related data exchange. The ONS is a DNS-based lookup service ensuring a link between the trading partners in order to share EPC data. It provides mapping information between EPC and Uniform Resource Identifier (URI) of the server hosting the EPC information related to the EPC in hand. The ONS maps EPCs to various forms of URI, such as EPCIS, HTML, XML-RPC and Web Services.

The function of the ONS consists of a number of lookup operations taking the EPC as input and returning pointers (e.g. IP addresses) to the corresponding information sources [7], as illustrated in Figure 3.1. The use of the DNS infrastructure requires encoding of the EPC into a domain namespace syntactically correct. This domain namespace will

then be used to query relevant information from the DNS system. The Name Authority Pointer (NAPTR) DNS record, detailed in [40], has been selected to map the ONS queries leveraging the DNS system. Currently, The ONS relies on a centralized ONS root managed exclusively by Verisign. It is set up on top of the DNS Top Level Domain ".com" and is called "onsspec.com".

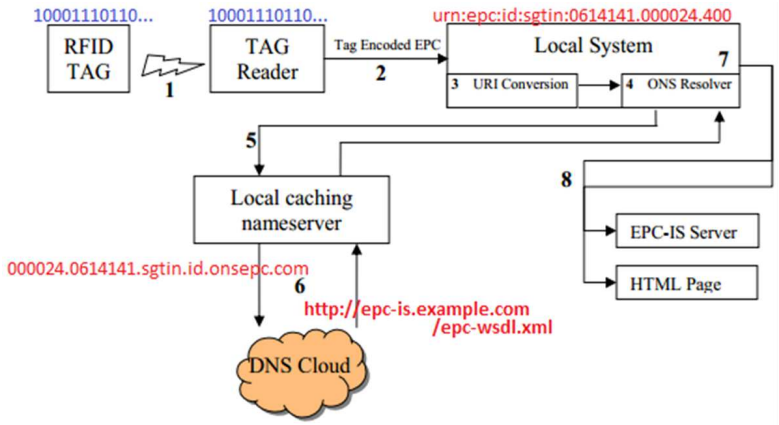


Figure 3.1 The ONS System in action; Under receipt of a query regarding an EPC e in the form of a Uniform Resource Identifier (URI) as shown in step 3, the local ONS client first resolves the URI into an appropriate Fully Qualified Domain Name (FQDN) (step 4), then it issues a DNS query for NAPTR records for that domain (step 5). Step 6 shows the DNS infrastructure returning a series of answers that contain service types and associated data (likely Uniform Resource Locators (URLs) that point to one or more services such as EPCIS). Finally, the ONS client extracts the services from the returned DNS NAPTR records and presents them to the corresponding information sources, such as the EPCIS server (steps 7 and 8)

The ONS relies completely on a centralized ONS root, similar to the Domain Name System (DNS) root, except that the ONS root is not replicated in multiple geographical locations. The ONS architecture has three major drawbacks, they are the following:

- First, the ONS root unipolarity aspect characterizing the ONS design makes its acceptance a major problem since root servers can be fully controlled by a single entity (company, organization or country, etc.) [9, 10]. Different scenarios depicting possible attack models in such a unipolar architecture have been discussed in [10], highlighting the political repercussions of such a design choice. Furthermore, the centralization and lack of redundancy of the ONS root create a single point of failure of the ONS, and thus the robustness of the whole business infrastructure is affected.

- Second, being based on the DNS system, the ONS inherits all the well-studied and well-documented weaknesses of the DNS including its poor performance, its complex configuration and its lack of security [8, 9]. The DNS system has always been an attractive target to Distributed Denial of Service (DDOS) attacks and also to Man in the Middle (MITM) attacks targeting the DNS resolution paths. Vulnerability to the Cache Poisoning attacks is another DNS weakness that may prove absolutely disastrous to systems relying on DNS for naming services, such as the business infrastructure.

- Third, the ONS has been designed to provide the information sources based on the company prefix of the EPC. As a result, only the manufacturer information sources are provided in response to ONS queries.

3.3 Proposed Lookup System Architecture

Although it seems wise to build the ONS lookup system on top of the DNS system, such decision would have critical impact on the future business infrastructure in terms of performance and security because of the DNS heritage. In this section, we first list the design requirements that should be fulfilled by any lookup service intended for the future business infrastructure, then we describe in detail our proposed architecture with a focus on how it fulfills the design requirements previously defined.

3.3.1 Design Requirements

The lookup service is expected to be vital in the future business infrastructure. Such a service will be used widely and in an unprecedented frequency given the trade volumes taking place nowadays, and their exponential growth rates. The following design requirements have to be fulfilled by any EPCglobal lookup system in order to avoid performance issues in the future (security requirements are discussed in Section 3.4.1):

- The lookup system must function as it is expected in all circumstances, i.e., it should return a list of pointers to information sources corresponding to the queried EPC [9].

- The lookup system must be robust and scalable, i.e., it should be able to serve a very large number of simultaneous queries, which will certainly keep increasing as the traditional Internet evolves ceaselessly to the "Internet of Things"; a paradigm that will make of everyday objects potential nodes of the Internet [3]. It should also react efficiently to the dynamics of the "Internet of Things", which is expected to be very challenging.

- The lookup system, being a key component of the future critical business infrastructure, should not, in any way, be controlled by a single entity. This is vital to its acceptance.

- The lookup system must ensure confidentiality and integrity of EPC data. It should also be resilient against different types of attacks, such as replay attacks.

3.3.2 Proposed Architecture

Our DHT-based architecture is illustrated in Figure 3.2. Each company, subscriber to the EPCglobal network, represents a node whose the identifier is the hash value of its manager number. A strong cryptographic hash function has to be selected in such a way that the probability of collision is small enough to ensure clean functionality of the system. Each node (replacing the local ONS server of the corresponding company) in the hash table knows its successor and a set of other nodes that can be determined in such a way to minimize the average and worst lookup costs, and at the same time minimize the complexity in terms of the amount of information each node will have to handle.

In practice, a manufacturer, with an item whose EPC is e, uses the same hash function, that it uses to compute its identifier in the DHT, to compute the hash value h, corresponding to the EPC e. It then constructs a pair consisting of a key and a value; the key being the hash value h and the value being the information corresponding to the item whose EPC is e. Once the pair is constructed, if the value h lays between the manufacturer's identifier and its successor's identifier, the manufacturer stores it in its own DHT node, otherwise it sends it to the neighbor whose the identifier is the closest to the hash value h. The process is then repeated at each receiving node until the pair reaches the appropriate storing node.

Our architecture ensures the three design requirements that a lookup system has to fulfill (the security requirements are dealt with in Section 3.4). First, it is a fully distributed architecture where no participating entity has more control than the others. Thus, the acceptance problem is overcome in our proposed architecture based on distributed and equally shared control of the system.

We can claim that our proposed architecture meets the functionality design requirement since the Distributed Hash Tables approach has proved its convenience in the well-known P2P file sharing systems where millions or even tens of millions of nodes collaborate to provide access to digitally stored information. Any robust P2P lookup protocol can be used in our proposed architecture as long as it is efficient in terms of node localization, such as Chord [34] or Koorde [105]. Chord [34] is a distributed peer-to-peer lookup protocol designed to locate efficiently the node storing data regarding the inquired EPC. Its simplicity, its scalability and its robustness, even in challenging networks, make it a good candidate to implement a DHT-based ONS pilot system. Koorde [105] improves Chord in terms of meeting interesting lower bounds for various operations such as the number of hops per lookup request.

To ensure the robustness of the system, a certain level of data replication has to be achieved in the system. This can be accomplished in our architecture through the construction of multiple DHT-based databases using either various hash functions or the same cryptographic hash function with a number of salts (keys to be concatenated with the EPC before applying the hash function). Such a replication mechanism has two main merits:

- First, information about the same EPC will be stored on various nodes, since each used hash function (or each used salt), builds a different hash table. Such a random replication of the EPC information in the system makes our proposed architecture robust against node failures and nodes unexpectedly leaving the hash table.
- Second, a querying node computes the hash values of the inquired EPC using all the used hash functions (or the used salts), and then decides, based on some distance metric, which of the computed hash values is the closest to its identifier in the hash table. In a load balanced hash table, the closest hash value to the

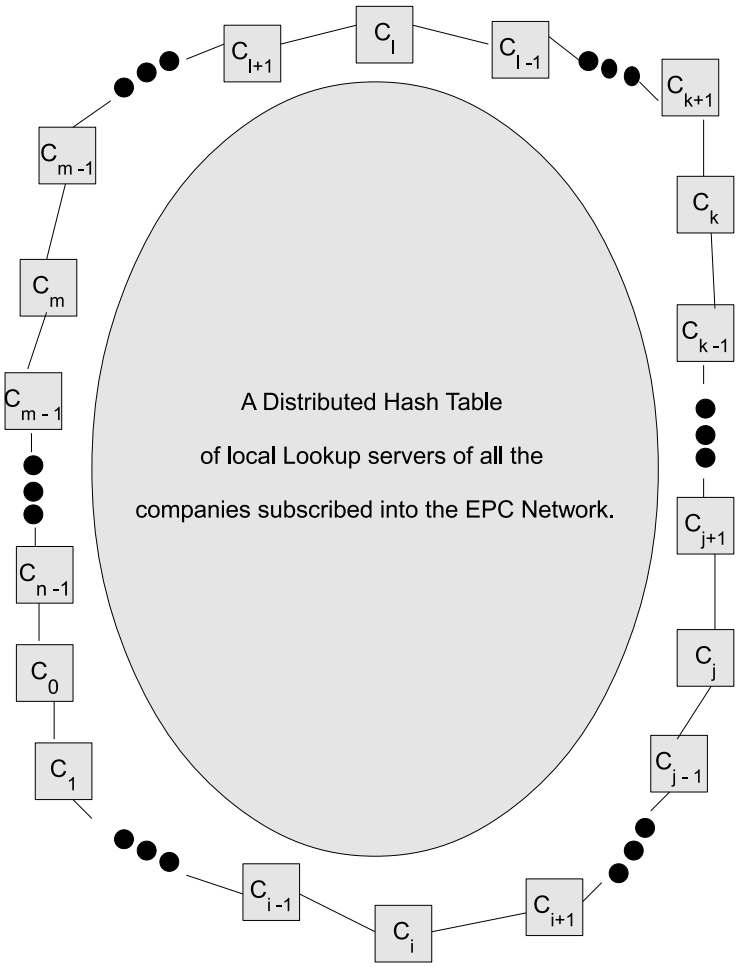


Figure 3.2 Our Proposed DHT-based Lookup System

querying node's identifier would, most probably, be the fastest to be located in the corresponding hash table. Thus, the performance design requirement is theoretically met in our proposed architecture. A deep experimental study will be taken care of in the future to confirm this claim in practice.

3.3.3 Discussion

There is no established and proved conclusion regarding the performance and the robustness of any DHT-based architecture proposed in the literature, including ours, so far. This is because of the absence of a real DHT-based lookup system on the same scale as the DNS. It is however important to note that these distributed architectures

theoretically promise better performance and also greater robustness than the classical DNS system.

3.4 Proposed Lookup System Security

Being based on the DNS system, the ONS inherits all the well-studied and well-documented security-related weaknesses of the DNS. The DNS system has always been an attractive target to Distributed Denial of Service (DDOS) attacks and also to Man in the Middle (MITM) attacks targeting the DNS resolution paths. Vulnerability to the Cache Poisoning attacks is another DNS security weakness. In this section we first present the security requirements that have to be met in any lookup system intended for the future business infrastructure, and then we describe our proposed security scheme.

3.4.1 Security Requirements

In order to deal with the potential security threats in the future business infrastructure, and because of the sensitivity of data exchanged between partners in the EPCglobal Network, a secure lookup system has to fulfill the following security requirements.

3.4.1.1 Authentication

Authentication of an entity refers to the process of validating its claimed identity against its real identity. In the context of an EPC lookup system, any entity querying information about a given EPC has to be authenticated before going further in the process of disclosing any data. Authentication is the answer to the spoofing threat, where attacking entities imitate trading partners in an attempt to gain access to sensitive data. Authentication is crucial in the lookup system because of the sensitivity of the EPC information exchanged between partners.

3.4.1.2 Data Integrity

Data integrity refers to the act of verifying the correctness and the completeness of the exchanged information between an expediter and a recipient. Because of the crucial role of the EPC lookup system in the future business infrastructure, checking the integrity of data exchanged between trading partners is an imperative requirement. Data integrity is the answer to both data tampering threat and betrayal of trusted servers. Data integrity is crucial in the lookup system because of the sensitivity to altering, of the EPC data, which is an important source of information in the decision making process within companies. A lookup system without data integrity would enable attackers to directly target the decision making process of companies.

3.4.1.3 Confidentiality

Confidentiality refers to the concealment of the exchanged information between an expediter and a recipient. Partners in the EPCglobal Network might be exchanging confidential EPC information about their business processes. Competitors, or other attackers, may be interested in this confidential information for different reasons. Confidentiality is the answer to the eavesdropping threat. Authentication is crucial in the lookup system because of the sensitivity to disclosure, of the EPC information exchanged between partners, especially by competitors.

3.4.1.4 Availability

Availability of a given system, from a security point of view, refers to its reliability and its robustness against attacks deliberately attempting to block it, such as denial of service attacks. Availability of the EPC lookup system implies attainability and usability of the lookup information it holds. The availability requirement has to be implemented in order to neutralize all types of denial of service attacks.

3.4.2 Security of the Proposed Solution

The DHT-based architecture which has been presented in Section 3.3 lacks security. In order to make it secure, mechanisms enforcing the security requirements described above have to be implemented. Those mechanisms will make the feasibility of all the threats hard enough in such a way that we can consider our system secure by the current standards. Adjustments may have to be made in the future, such as increasing the length of the used cryptographic keys.

3.4.2.1 Key Distribution

Because of the massive amount of data that is expected to be exchanged in the future business infrastructure, it is wise to use symmetric-key cryptosystems for encipherment and decipherment rather than public key cryptography. The reason is that public key cryptography, although deemed to be ideal since no secret sharing is needed, is much costly in terms of computation than symmetric cryptography.

A well known approach is to use asymmetric cryptography for authentication and symmetric cryptography for exchanged data integrity and confidentiality. Shared symmetric keys, to be used for data exchange, are generated and then exchanged using the asymmetric cryptography.

Our architecture assumes a public key infrastructure with a globally agreed-upon certification authority issuing a pair of public/private keys to each EPCglobal subscriber. The X.509 public key infrastructure standard has been recommended by EPCglobal to define a certificate profile for authentication purposes [106].

In our architecture, each node in the distributed hash table corresponds to an EPCglobal subscriber. The identifier of this node is the hash value of the EPC manager number, called also company prefix, assigned to the company it represents. Each company generates a set of symmetric keys to be shared with its partners in each and every supply chain depending on their security clearances. Inspired from Bell-LaPadula security model [107, 108], a given security clearance is represented by a specific symmetric key sent to all the partners, in a given supply chain, having access to the information labeled with a security level less or equal to this security clearance.

Each company generates a set of keys equal to the number of security clearances it has to manage in each supply chain it is a member of. Let us assume that a company C has classified the information it owns about its EPCs into n different security levels, $SC_i, 0 \leq i < n$, 0 referring to public data and $n-1$ to top secret data. Company C then generates a set S_{RK} of $n-1$ of random keys having same length m (m is a public parameter supposed to be known by all the subscribers, and is long enough to be considered secure in symmetric cryptography), $S_{RK} = \{RK_i, 0 < i < n\}$. Let us assume also that company C has q partners, and that each partner $P_j, 0 \leq j < q$ has been assigned a specific security clearance $SC_i, 0 \leq i < n$.

Company C constructs a set S_{SK} of n symmetric keys $S_{SK} = \{SK_i, 0 \leq i < n\}$ to be sent to its q partners $P_j, 0 \leq j < q$ as follows. For all partner $P_j, 0 \leq j < q$ having a security clearance $SC_i, 0 < i < n$, a symmetric key $SK_i, 0 < i < n$ is built out of a concatenation of the random keys $\{RK_1, ..., RK_i\}$ in the increasing order from left to right, $SK_i = RK_1...RK_i$. Company C labels publicly available data with security clearance SC_0 for which no symmetric key is defined since it is stored in clear in the DHT. Obviously, the symmetric key $SK_i, 0 \leq i < n$ has a length of $i \cdot m$ since it is composed of i random keys, each of which is of length m.

Company C encrypts then all the constructed symmetric keys twice. The first encryption is carried out using its own private key in order to ensure origin authentication and data integrity. The second encryption, takes the output of the first one and encrypts it using the public key of the intended recipient. This is to ensure confidentiality. Company C then uses the resulting data as the value in the (key, value) pairs to be disseminated into the constructed distributed databases. The keys in those (key, value) pairs represent the hash values of the manager number of each trading partner for which the symmetric key is addressed.

Once a node receives a (key, value) pair, it compares the key with its own identifier. If they match, the data in the pair is processed, otherwise it is forwarded to the next closest neighbor. In order to get the symmetric key, the intended partner first decrypts the received value in the (key, value) pairs twice using its own private key, and then decrypts the result using the public key of the originating company.

After the key distribution process is completed, each company ends up with two types of symmetric keys to manage, in addition to its pair of public and private key issued by

the certification authority. The first type corresponds to the symmetric keys it has sent to its partners (outgoing symmetric keys), and the second to the symmetric keys it has received from its partners (incoming symmetric keys).

All these keys have to be stored in a local secure database. The outgoing symmetric keys have to be indexed both by security clearance and per supply chain. A matching of each partner, represented by its company prefix, with its corresponding security clearance and the supply chain it belongs to, has also to be stored in the local database. The incoming symmetric keys have also to be stored in that local database, indexed with the company prefix of the corresponding sender and also per supply chain. All these symmetric keys are stored by each company after they are encrypted using its own public key. This is to ensure only this company can decrypt them using its own private key.

3.4.2.2 Data Publishing

After completing the key distribution process with all its partners, a company can then publish its lookup data into the DHT in the form of (key, data) pairs, of course after encrypting it using the symmetric keys, which has been shared with its partners. The pair (key, data) regarding some to-be-published EPC is built as follows.

- A company C having some data D to publish about a specific EPC e first classifies this data into n (or less) classes $D_i, 0 \leq i < n$, each of which is assigned the corresponding security level SC_i. The n security levels (also called security clearances) are supposed to have been defined (per supply chain) beforehand depending on the policies of the company and the type of the information it handles (an example of four security levels would be Top Secret, Secret, Confidential and Public). The company C has also to define w the number of different hash functions (or salts) that will be used to duplicate data in the Lookup system to meet the availability requirement.
- Once the data classes of EPC e are defined, company C constructs n (or less depending on the number of classes) pairs $(h_{\{i,k\}}, I_i), 0 \leq i < n, 0 \leq k < w$ where k refers to the identifier of the used hash function (or the used salt), $h_{\{i,k\}}$ refers to the EPC e encrypted using a symmetric key $SKey_i$ of length m computed out

of the XOR operation of the i random keys making up the exchanged symmetric key $SK_i = RK_1...RK_i$ and then hashed using the hash function (or the salt) identified by k, $h_{\{i,k\}} = hash_k(SKey_i(e))$, and I_i refers to the information that can be accessed by all partners having security clearances great or equal to SC_i, encrypted using the symmetric key SK_i, $I_i = SK_i(D_i)$. Obviously, the use of symmetric keys of length m for all the security levels would certainly speed up the encryption/decryption processes. The XOR operation allows the compression of all the symmetric keys exchanged between trading partners from lengths of $k \cdot m, 1 < k < n$ into length of m, m being the publicly known length of the random keys and n the number of security clearances defined for the company in hand.

At the end of this process, company C will end up with $n \cdot w$ (or less) pairs $(h_{\{i,k\}}, I_i), 0 \leq i < n, 0 \leq k < w$ referring to the same EPC e, but whose access is determined by the shared symmetric keys.

- Finally, company C sends the generated pairs into the DHT to be stored in the appropriate node.

3.4.2.3 Data Retrieval

A company inquiring about a specific EPC has to issue a query to locate the targeted EPC in the DHT. Obviously, the targeted EPC is used to compute the corresponding hash value which represents the key used to locate the node storing the targeted information.

All partners having security clearance $SC_i, 0 \leq i < n$, would be allowed to retrieve all information of sensitivity levels less or equal to SC_i. They only have to extract the symmetric key corresponding to the targeted security level $j, 0 \leq j \leq i$ out of the symmetric key they received from the owner of this information. To do that, they simply have to extract the first $j \cdot m$ bits to the left of the symmetric key SK_i, which corresponds to the first j random keys, on which the XOR operation is carried out to form the same symmetric key that has been used by the EPC data provider to encrypt the published EPC data.

3.4.3 Discussion

Our proposed security model, based on a flexible and efficient key distribution mechanism using a public key infrastructure, implements the confidentiality requirement not only for the data but also for the EPC codes themselves. It also ensures both data integrity and origin authentication. Moreover, the use of different hash functions (or salts) helps in ensuring the availability of the system thanks to data replication and the smooth self-configuration of the DHT.

The fact that the keys corresponding to the EPC codes are computed after the encryption of those EPCs, makes our architecture resilient against EPC dictionary attacks. Furthermore, it would boost the confidence of reluctant companies in our architecture since, unless the exchanged symmetric keys have been let out, nobody can determine on which node of the DHT a specific EPC is being stored.

3.5 Conclusion

In this chapter, we have presented a DHT-based, scalable and more secure architecture for data lookup in the EPCglobal Network. The proposed architecture aims at replacing the currently deployed ONS system with a full discovery services system. In addition to the design and security requirements discussed above, our proposed distributed and secured architecture implements also access control on the EPC information using a military-style classification inspired from Bell-LaPadula security model.

It is worth mentioning that the proposed key distribution mechanism is only as secure as the public key infrastructure. This illustrates the criticality of the PKI impact on the security of the whole business infrastructure.

Chapter 4

Query State Inference in Discovery Services

4.1 Introduction

Anomaly detection refers to approaches and techniques looking into identification of cases that are unusual within a dataset that is seemingly homogeneous. Anomaly detection is an important research topic that has been investigated from different standpoints. In network traffic management, several proposals [109–112] have targeted detecting anomalous traffic or nodes with abnormal traffic. In medical informatics, anomaly detection has been used to detect disease outbreaks in a specific area [113, 114]. Early detection of faulty industrial units, such as motors and turbines, is crucial to the industry actors, as it prevent further escalation of the components' wear and tear, and hence worse financial losses [115, 116].

Various anomaly detection techniques have been studied in the literature; including but not limited to Neural Networks [117, 118], Bayesian Networks [114] and Rule-based Systems [119]. One of the well-known techniques is based on parametric statistical modeling, such as HMM [117]. For improved anomaly detection accuracy, several works [120] have proposed combining multiple HMM models. Others have suggested combining the HMM model with other models, such as Neural Networks [121, 122].

The core foundation of the EPCglobal Discovery Services is the lookup service, which consists of localizing all the information sources pointing to all relevant data with regards to the EPC at hand. Conceptually, Discovery Services can be viewed as search engines, specific to the global business infrastructure in the future IoT, in that they represent tools to "discover" all "relevant" information related to specific "keywords"; the EPCs in the case of the EPCglobal Network [6]. This explains the imperative need for robust mechanisms to secure this vital and extremely sensitive service.

Although several proposals have discussed security of the lookup service in Discovery Services, none, to the best of our knowledge, has looked at securing Discovery Services from an "Anomaly Detection" point of view. This approach consists of identifying lookup queries which do not conform to some pattern learned based on a given probabilistic model.

We propose two different security schemes aiming at predicting the state (safe or risky) of the lookup queries. The first security scheme consists of two algorithms based on the Gaussian model illustrated in Section 4.5.2, while the second is based on a Hidden Markov Model. Both security schemes first analyze the EPCglobal transaction pattern for each combination $(product\ category, company, business\ step)$. Upon receiving an EPC query, a Feature Extraction Process converts it into a vector of observed real values. These real values are assumed to follow a Gaussian distribution per state, i.e., a Gaussian distribution for safe queries and another one for risky queries.

The rest of this chapter is organized as follows. In Section 4.2, we discuss the threat model considered by this work. In Section 4.3, we present some background and we define the notations used in this chapter. In Section 4.4, we describe the Query Feature Extraction process, which consists of converting lookup queries into vectors of values. These values are used as input to the proposed probabilistic models. In each of the next two sections, we present our proposed security schemes aiming at detecting suspicious queries and we evaluate their performance. Our first proposed security scheme, consisting of two algorithms based on the Gaussian model, is presented and assessed in Section 4.5. In Section 4.6, we present and evaluate our second security scheme, based on a Continuous Hidden Markov Model (CHMM). Finally, Section 4.7 concludes this chapter.

4.2 Threat Model

One of the well-known threat models that almost any Internet-based system must take into account is Denial of Service Attacks (DoS). A DoS attack is an attack in which one or many compromised systems are used to target a single system, the goal being to flood its resources in an attempt to disrupt the services it is offering. Although DoS attacks sound less alarming than other attacks where information is the target (such as Man-in-the-Middle attacks, IP spoofing attacks and Replay attacks), they have to be taken care of vigilantly for the following reasons. First, in many cases, DoS attacks have had a damaging impact on the reputation of targeted companies. Second, unlike other apparently-more-dangerous attacks, DoS attacks require much less resources and are much more straightforward to setup.

4.3 Background and Notations

We propose in this chapter two security schemes, designed to detect risky queries directed to an EPCglobal Discovery Services system considering both the business habits of the stakeholders and the habits of the attackers. Figure 4.1 depicts a supply chain whose the actors exchange physical items through a sequence of Shipping and Receiving operations.

Figure 4.1 illustrates also that each of the involved stakeholders carries out a subset of business steps from the set $S_{BS} = \{BS_i, 0 <= i < N_{BS}\}$, where BS_i denotes business step i and N_{BS} denotes the number of possible business steps throughout the life-cycle of items of category p. The chronological order of the business steps is defined within the indexes; For any i, j such that $0 <= i < j < N_{BS}$, BS_i takes place before BS_j.

At the end of each business step $BS_i, 0 <= i < N_{BS}$, the stakeholder triggers a query through which it either updates the Discovery Services repository with the new state of the exchanged physical item (PUT operation), or it retrieves, out of the Discovery Services repository, all the relevant data with regards to the inquired EPC (GET operation). We denote $Query_{BS_i}^{p,C}$ the query triggered by a company C during a business step $BS_i, 0 <= i < N_{BS}$ regarding the product category p.

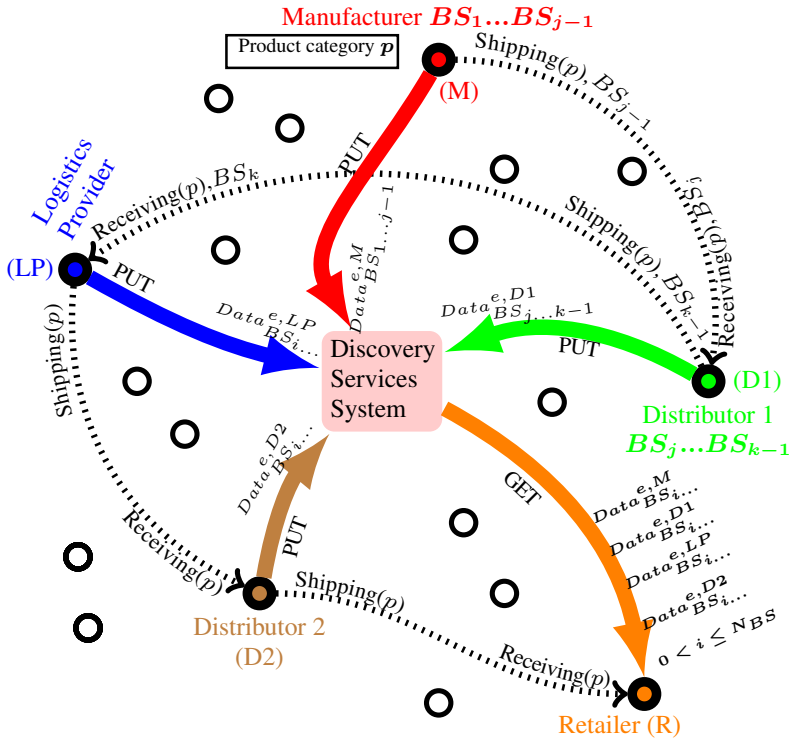


Figure 4.1 An illustration of the EPCglobal Discovery Services to be offered for the actors of a given supply chain regarding the exchange of items of a given product category p. Each stakeholder notifies, via a PUT query, the Discovery Services System with data related to a given shipment of physical items of category p during a given business step $BS_i, 0 <= i < N_{BS}$, where BS_i denotes business step i and N_{BS} denotes the number of possible business steps throughout the life-cycle of items of category p. A GET query regarding a given shipment of items of category p should return all data relevant to the inquired shipment. Data exchanged between a stakeholder and the Discovery Services System is denoted $Data_{BS_i}^{p,stakeholder}, 0 <= i < N_{BS}$.

The query $Query_{BS_i}^{p,C}$ holds following information:

- $Query_{BS_i}^{p,C}[id]$ denotes the identifier of the "shipment" being inquired
- $Query_{BS_i}^{p,C}[timestamp]$ denotes the date and time of the current business step BS_i
- $Query_{BS_i}^{p,C}[data]$ denotes the data to be stored (PUT operation) or being retrieved (GET operation) during the current business step BS_i. It is also denoted as $Data_{BS_i}^{p,C}$ (as illustrated in Figure 4.1)
- $Query_{BS_i}^{p,C}[nature]$ denotes the nature of the operation (PUT/GET)
- $Query_{BS_i}^{p,C}[businesstep]$ denotes the current business step

- $Query_{BS_i}^{p,C}[nbrItems]$ denotes the number of items in the currently processed shipment

4.4 Query Feature Extraction

Upon receipt of a given EPC query $currentQuery_{BS_i}^{p,C}$, originating from given company C in the context of a given business step $BS_i, 0 \leq i < N_{BS}$ with regards to a given shipment of items having product category p, the Discovery Services system first converts the received query into a vector v of real values. In our case, given a company C processing items of category p , we compute the following values, as illustrated in Figure 4.2:

- $timeDiffH_{BS_i}^{p,C}$ representing how much time it took business step BS_i to take place within the current query, compared to its last occurrence within the immediate previous query.
- $timeDiffV_{BS_i}^{p,C}$ representing how much time it took business step BS_i to take place compared to its predecessor business step BS_{i-1}, within the same current query.
- $avgOrder_{BS_i}^{p,C}$ representing the current average number of items, of category p, being processed by company C, per unit of time.

$$v=(timeDiffV_{BS_i}^{p,C}, timeDiffH_{BS_i}^{p,C}, avgOrder_{BS_i}^{p,C}) \tag{4.1}$$

$$\begin{aligned} timeDiffV_{BS_i}^{p,C} &= currentQuery_{BS_i}^{p,C}[timestamp] - currentQuery_{BS_{i-1}}^{p,C}[timestamp] \\ timeDiffH_{BS_i}^{p,C} &= currentQuery_{BS_i}^{p,C}[timestamp] - previousQuery_{BS_i}^{p,C}[timestamp] \\ avgOrder_{BS_i}^{p,C} &= \frac{previousQuery_{BS_i}^{p,C}[nbrItems]}{timeDiffH_{BS_i}^{p,C}} \end{aligned} \tag{4.2}$$

$previousQuery_{BS_i}^{p,C}$ denotes the previous query having processed a similar shipment with respect to the current query $currentQuery_{BS_i}^{p,C}$. Both of the queries have been carried out by the same company C during the same business step BS_i, processing items

of the same product category p. The main difference lay in the identifier of the inquired "shipment" and also in the date and time business step BS_i, regarding the inquired "shipment", took place; i.e., $previousQuery^{p,C}_{BS_i}[timestamp] < currentQuery^{p,C}_{BS_i}[timestamp]$ and $previousQuery^{p,C}_{BS_i}[id] \neq currentQuery^{p,C}_{BS_i}[id]$.

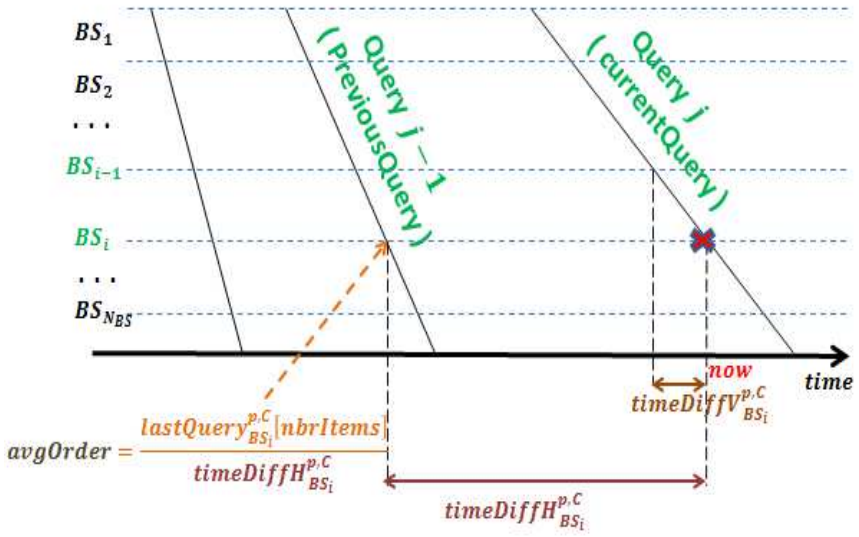


Figure 4.2 An illustration of the query feature extraction process

4.5 A Gaussian Security Scheme for Query State Inference

4.5.1 Assumptions

Our proposed Gaussian security scheme makes the following assumptions:

- Each and every query, directed to the Discovery Services System, inquires about a "shipment" of product items rather than one item. A "shipment" contains only items of the same product category;
- Each and every query, performed by a company C regarding a certain "shipment", corresponds to one and only one business step $BS_i, 0 \leq i < N_{BS}$. Therefore, a company C might issue one or more queries for the same "shipment" depending on the number of business steps the "shipment" goes through within its premises.

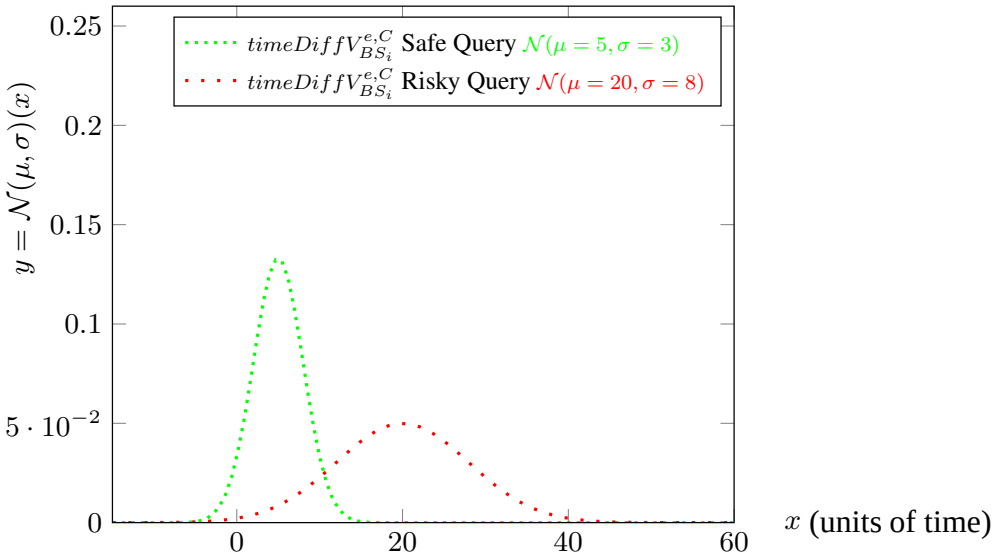


Figure 4.3 Two Gaussian distributions describing the pattern for safe queries and the pattern for risky queries, with regards to the observed feature $timeDiffV$ (defined in Section 4.4)

- Each of the extracted features $timeDiffV^{p,C}_{BS_i}$, $timeDiffH^{p,C}_{BS_i}$ and $avgOrder^{p,C}_{BS_i}$ follows two different Gaussian distributions depending on the state of the corresponding query. Hence, $timeDiffV^{p,C}_{BS_i}$ follows a Gaussian distribution with a mean $\mu^{V,R}_{p,C,BS_i}$ and a standard deviation $\sigma^{V,R}_{p,C,BS_i}$ for risky queries, and another Gaussian distribution with a mean $\mu^{V,S}_{p,C,BS_i}$ and a standard deviation $\sigma^{V,S}_{p,C,BS_i}$ for safe queries. idem for $timeDiffH^{p,C}_{BS_i}$ and $avgOrder^{p,C}_{BS_i}$.

4.5.2 Gaussian Model

In our proposed security scheme, we assume, as stated in Section 4.5.1, two Gaussian distributions for each of the observed features $timeDiffV^{p,C}_{BS_i}$, $timeDiffH^{p,C}_{BS_i}$ and $avgOrder^{p,C}_{BS_i}$. Figure 4.3 illustrates an example of a Gaussian distribution for the observed feature $timeDiffV^{p,C}_{BS_i}$ in the case of safe queries, and another one in the case of risky queries.

The score computation in all the algorithms, defined in Section 4.5, is based on the probability that a certain Gaussian distribution generates a given real value. This is shown in detail in Figure 4.4.

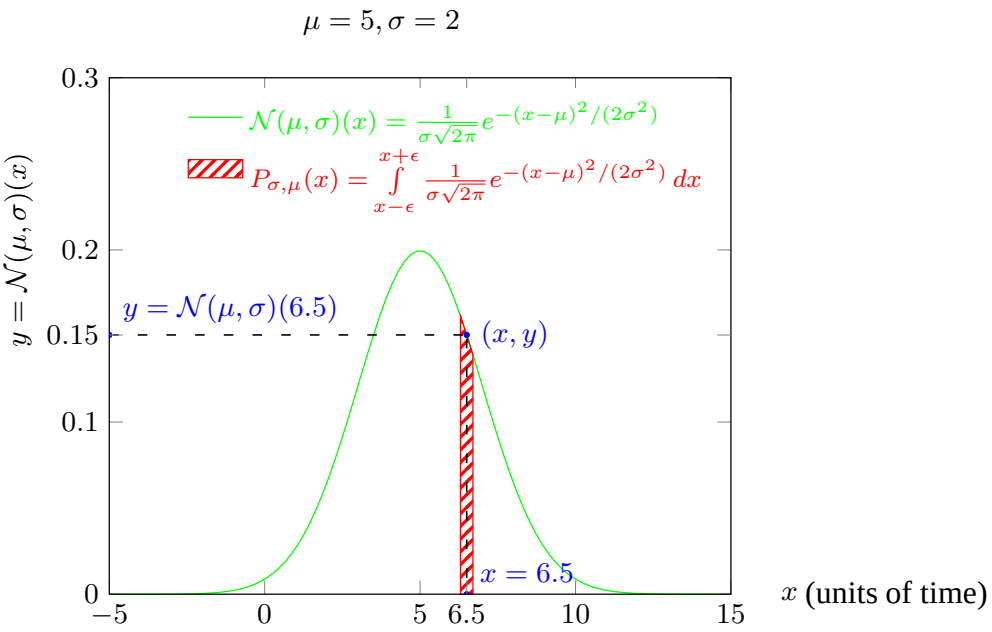


Figure 4.4 Illustration of the probability that a given value x be generated using Gaussian distribution $\mathcal{N}(\mu = 5, \sigma = 2)(x)$, which is the small area, shaded in red, delimited by the values $x - \epsilon$ and $x + \epsilon$ under the bell curve. ϵ denotes a very small value. In our simulations we used $\epsilon = 0.05$.

As a reference, we define a simple algorithm called Ref, inferring the most probable state of a given query, by reversing the Gaussian random number generation process detailed in Section 4.5.4.1. Our proposed algorithms, described in Section 4.5.3, are compared against this reference algorithm. Ref algorithm, illustrated in 4.5, computes the probabilities that each of the two Gaussian distributions (the one for safe queries and the other for risky queries) generates the observed values. These probabilities are then compared to predict the most probable state.

All the algorithms detailed in Figure 4.5 assume that the Feature Query Extraction procedure has been carried out beforehand as described in Section 4.4. Therefore the vector v, defined in Equation 4.1, is assumed to be passed as input to the algorithms.

4.5.3 Proposed Algorithm for Query State Inference

Upon receipt of a given EPC query $currentQuery_{BS_i}^{p,C}$, originating from given company C in the context of a given business step $BS_i, 0 < leqi < N_{BS}$, with regards to a given

shipment of product category p, the Discovery Services system calls the Probability Product Algorithm (PPA) illustrated in Figure 4.5 in order to assess the risk of the query, and hence infer its most probable state. The PPA algorithm first computes two scores, $score^{Prd}_{\boldsymbol{Safe}}$ for the query being "safe" and $score^{Prd}_{\boldsymbol{Risky}}$ for the query being "risky". Finally, the two scores are compared to predict the most probable state.

For evaluation reasons, we also present the Probability Sum Algorithm (PSA), defined in Figure 4.5, which differs from the PPA algorithm only in the way the scores are computed.

The PSA and PPA algorithms, detailed in Figure 4.5, use the notations defined in the List of Symbols and Notations at the beginning of this thesis and in Section 4.4.

4.5.4 Performance Evaluation

4.5.4.1 Simulation Setup and Parameters

Because there is no available real data that we can test our Gaussian security scheme with, we generated data using the Gaussian distributions assumed in Section 4.5.1. The parameters of these Gaussian distributions (i.e., the mean and the standard deviation) are defined in the List of Symbols and Notations at the beginning of this thesis. The data consists of sequences of 2000 vectors of the form $(trueState,\ timeDiffV^{p,C}_{BS_i},\ timeDiffH^{p,C}_{BS_i},\ avgOrder^{p,C}_{BS_i})$. The value of the element $trueState$, either "safe" or "risky", is randomly selected using a uniform distribution. Depending on the selected state, we generate a vector of three random numbers $(timeDiffV^{p,C}_{BS_i}, timeDiffH^{p,C}_{BS_i},\ avgOrder^{p,C}_{BS_i})$, using the corresponding Gaussian distributions.

The PPA algorithm, detailed in Figure 4.5, is then called to predict the state of the queries, represented by their supposedly observed values $(timeDiffV^{p,C}_{BS_i}, timeDiffH^{p,C}_{BS_i},\ avgOrder^{p,C}_{BS_i})$.

4.5.4.2 Evaluation Approach

Let us denote $trueStateVEC$ the vector of the true states of the queries, and $predictedStateVEC$ the vector of the queries' states predicted by one of the algorithms defined in Section

Initialization used by the algorithms below (For notations' definitions, refer to Figure 4.4 and the List of Symbols and Notations at the beginning of this thesis):

$P_{\boldsymbol{S}}^{\boldsymbol{V}} \leftarrow P_{\sigma_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{S}},\mu_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{S}}}(timeDiffV_{BS_i}^{p,C})$; // Probability that $timeDiffV_{BS_i}^{p,C}$ is generated using $\mathcal{N}(\mu_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{S}}, \sigma_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{S}})$

$P_{\boldsymbol{R}}^{\boldsymbol{V}} \leftarrow P_{\sigma_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{R}},\mu_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{R}}}(timeDiffV_{BS_i}^{p,C})$; // Probability that $timeDiffV_{BS_i}^{p,C}$ is generated using $\mathcal{N}(\mu_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{R}}, \sigma_{p,C,BS_i}^{\boldsymbol{V},\boldsymbol{R}})$

$P_{\boldsymbol{S}}^{\boldsymbol{H}} \leftarrow P_{\sigma_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{S}},\mu_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{S}}}(timeDiffH_{BS_i}^{p,C})$; // Probability that $timeDiffV_{BS_i}^{p,C}$ is generated using $\mathcal{N}(\mu_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{S}}, \sigma_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{S}})$

$P_{\boldsymbol{R}}^{\boldsymbol{H}} \leftarrow P_{\sigma_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{R}},\mu_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{R}}}(timeDiffH_{BS_i}^{p,C})$; // Probability that $timeDiffV_{BS_i}^{p,C}$ is generated using $\mathcal{N}(\mu_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{R}}, \sigma_{p,C,BS_i}^{\boldsymbol{H},\boldsymbol{R}})$

$P_{\boldsymbol{S}}^{\boldsymbol{O}} \leftarrow P_{\sigma_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{S}},\mu_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{S}}}(avgOrder_{BS_i}^{p,C})$; // Probability that $timeDiffV_{BS_i}^{p,C}$ is generated using $\mathcal{N}(\mu_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{S}}, \sigma_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{S}})$

$P_{\boldsymbol{R}}^{\boldsymbol{O}} \leftarrow P_{\sigma_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{R}},\mu_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{R}}}(avgOrder_{BS_i}^{p,C})$; // Probability that $timeDiffV_{BS_i}^{p,C}$ is generated using $\mathcal{N}(\mu_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{R}}, \sigma_{p,C,BS_i}^{\boldsymbol{O},\boldsymbol{R}})$

$PPA(P(timeDiffV), P(timeDiffH), P(avgOrder))$***: PPA Algorithm using 3 features; compute and then compare the scores***

$score_{\boldsymbol{Safe}}^{Prd} \leftarrow P_{\boldsymbol{S}}^{\boldsymbol{V}} \cdot P_{\boldsymbol{S}}^{\boldsymbol{H}} \cdot P_{\boldsymbol{S}}^{\boldsymbol{O}}$; $score_{\boldsymbol{Risky}}^{Prd} \leftarrow P_{\boldsymbol{R}}^{\boldsymbol{V}} \cdot P_{\boldsymbol{R}}^{\boldsymbol{H}} \cdot P_{\boldsymbol{R}}^{\boldsymbol{O}}$; // The score is the product of three probabilities

if $score_{\boldsymbol{Safe}}^{Prd} > score_{\boldsymbol{Risky}}^{Prd}$ **then** $RISKY$ **else** $SAFE$ **end if**

$PSA(P(timeDiffV), P(timeDiffH), P(avgOrder))$***: PSA Algorithm using 3 features; compute and then compare the scores***

$score_{\boldsymbol{Safe}}^{Sum} \leftarrow P_{\boldsymbol{S}}^{\boldsymbol{V}} + P_{\boldsymbol{S}}^{\boldsymbol{H}} + P_{\boldsymbol{S}}^{\boldsymbol{O}}$; $score_{\boldsymbol{Risky}}^{Sum} \leftarrow P_{\boldsymbol{R}}^{\boldsymbol{V}} + P_{\boldsymbol{R}}^{\boldsymbol{H}} + P_{\boldsymbol{R}}^{\boldsymbol{O}}$ // The score is the sum of three probabilities

if $score_{\boldsymbol{Safe}}^{Sum} > score_{\boldsymbol{Risky}}^{Sum}$ **then** $RISKY$ **else** $SAFE$ **end if**

$PPA(P(timeDiffV), P(timeDiffH))$***: PPA Algorithm using 2 features; compute and then compare the scores***

$score_{\boldsymbol{Safe}}^{Prd} \leftarrow P_{\boldsymbol{S}}^{\boldsymbol{V}} \cdot P_{\boldsymbol{S}}^{\boldsymbol{H}}$; $score_{\boldsymbol{Risky}}^{Prd} \leftarrow P_{\boldsymbol{R}}^{\boldsymbol{V}} \cdot P_{\boldsymbol{R}}^{\boldsymbol{H}}$ // The score is the product of two probabilities

if $score_{\boldsymbol{Safe}}^{Prd} > score_{\boldsymbol{Risky}}^{Prd}$ **then** $RISKY$ **else** $SAFE$ **end if**

$PSA(P(timeDiffV), P(timeDiffH))$***: PSA Algorithm using 2 features; compute and then compare the scores***

$score_{\boldsymbol{Safe}}^{Sum} \leftarrow P_{\boldsymbol{S}}^{\boldsymbol{V}} + P_{\boldsymbol{S}}^{\boldsymbol{H}}$; $score_{\boldsymbol{Risky}}^{Sum} \leftarrow P_{\boldsymbol{R}}^{\boldsymbol{V}} + P_{\boldsymbol{R}}^{\boldsymbol{H}}$ // The score is the sum of two probabilities

if $score_{\boldsymbol{Safe}}^{Sum} > score_{\boldsymbol{Risky}}^{Sum}$ **then** $RISKY$ **else** $SAFE$ **end if**

$Ref(P(timeDiffV))$***: Reference Algorithm using the feature*** $timeDiffV$***; compare the probablilities***

if $P_{\boldsymbol{S}}^{\boldsymbol{V}} > P_{\boldsymbol{R}}^{\boldsymbol{V}}$ **then** $RISKY$ **else** $SAFE$ **end if**

$Ref(P(timeDiffH))$***: Reference Algorithm using the feature*** $timeDiffH$***; compare the probablilities***

if $P_{\boldsymbol{S}}^{\boldsymbol{H}} > P_{\boldsymbol{R}}^{\boldsymbol{H}}$ **then** $RISKY$ **else** $SAFE$ **end if**

$Ref(P(avgOrder))$***: Reference Algorithm using the feature*** $avgOrder$***; compare the probablilities***

if $P_{\boldsymbol{S}}^{\boldsymbol{O}} > P_{\boldsymbol{R}}^{\boldsymbol{O}}$ **then** $RISKY$ **else** $SAFE$ **end if**

Figure 4.5 Probability Sum Algorithm (PSA) using 2 and 3 extracted features, Probability Product Algorithm (PPA) using 2 and 3 extracted features, and the reference algorithms for Query State Inference taking as input each of the extracted features $(timeDiffV_{BS_i}^{p,C}, timeDiffH_{BS_i}^{p,C}, avgOrder_{BS_i}^{p,C})$

4.5. Our evaluation approach consists of computing and comparing the detection rate and the false alarm rate of each of the algorithms.

We first compute the following:

- TP (True Positive) number of queries correctly inferred as risky
- FP (False Positive) number of queries incorrectly inferred as risky
- TN (True Negative) number of queries correctly inferred as safe
- FN (False Negative) number of queries incorrectly inferred as safe

Then we compute the detection rate and the false alarm rate using Equation 4.3 and Equation 4.4.

$$DetectionRate = \frac{TP}{TP + FN} \tag{4.3}$$

$$FalseAlarmRate = \frac{FP}{FP + TN} \tag{4.4}$$

4.5.4.3 Results

We conducted extensive experiments to evaluate the performance of the PPA Algorithm, compared to the PSA algorithm and to the reference algorithms, all defined in Figure 4.5. Each experiment starts by generating simulation data out of a set of three couples of Gaussian distributions, each of which corresponds to an observed feature; i.e., $timeDiffV_{BS_i}^{p,C}$, $timeDiffH_{BS_i}^{p,C}$ or $avgOrder_{BS_i}^{p,C}$). Each couple consists of a Gaussian distribution for safe queries, and another one for risky queries, as shown in Figure 4.5.

we run 10 experiments per scenario. A scenario defines the gap between the mean of the "safe query" Gaussian distribution and "risky query" Gaussian distribution. Then we plot the average detection rate and the average false alarm rate, defined in Section 4.5.4.2, with a 95% confidence interval. From one scenario to another we changed the gap between the mean of the "safe query" Gaussian distribution and "risky query" Gaussian distribution.

4.5.4.3.1 Two Observed Features

Figure 4.6 illustrates the detection rate of the three reference algorithms; identified in Figure 4.5 as $Ref(P(timeDiffV))$, $Ref(P(timeDiffH))$ and $Ref(P(avgOrder))$; in comparison with the detection rate of both the PSA algorithm and the PPA algorithm using two of the observed features, identified in Figure 4.5 as $PSA(P(timeDiffV), P(timeDiffH))$ and $PPA(P(timeDiffV), P(timeDiffH))$ respectively.

Figure 4.7 illustrates the false alarm rate of the three reference algorithms; identified in Figure 4.5 as $Ref(P(timeDiffV))$, $Ref(P(timeDiffH))$ and $Ref(P(avgOrder))$; in comparison with the false alarm rate of both the PSA algorithm and the PPA algorithm using the probabilities of two features, identified in Figure 4.5 as $PSA($ $P(timeDiffV), P(timeDiffH))$ and $PPA(P(timeDiffV), P(timeDiffH))$ respectively.

Figure 4.6 and Figure 4.7 show that the PPA algorithm performs better than the other algorithms both in terms of detection rate and the false alarm rate. Hence, the PPA algorithm improves much better the inference accuracy of risky queries in EPCglobal Discovery Services under the assumptions stated in Section 4.5.1.

4.5.4.3.2 Three Observed Features

Figure 4.8 illustrates the detection rate of the PSA algorithm using two and three observed features, identified in Figure 4.5 as $PSA(P(timeDiffV), P(timeDiffH))$ and $PSA(P(timeDiffV), P(timeDiffH), P(avgOrder))$ respectively, in comparison with the detection rate of the PPA algorithm using two and three observed features, identified in Figure 4.5 as $PPA(P(timeDiffV), P(timeDiffH))$ and $PPA($ $P(timeDiffV), P(timeDiffH), P(avgOrder))$ respectively.

Figure 4.9 illustrates the false alarm rate of the PSA algorithm using two and three observed features, identified in Figure 4.5 as $PSA(P(timeDiffV), P(timeDiffH))$ and $PSA(P(timeDiffV), P(timeDiffH), P(avgOrder))$ respectively, in comparison with the false alarm rate of the PPA algorithm using two and three types of observations, identified in Figure 4.5 as $PPA(P(timeDiffV), P(timeDiffH))$ and $PPA($ $P(timeDiffV), P(timeDiffH), P(avgOrder))$ respectively.

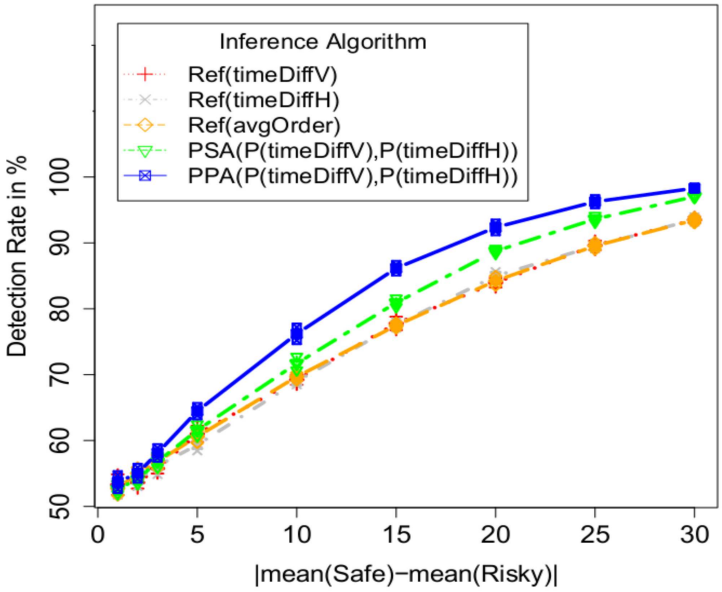


Figure 4.6 A comparison of the Detection Rate of Ref algorithms, PSA algorithm and PPA algorithm using 2 observed features (refer to Figure 4.5 for the definition of the algorithms)

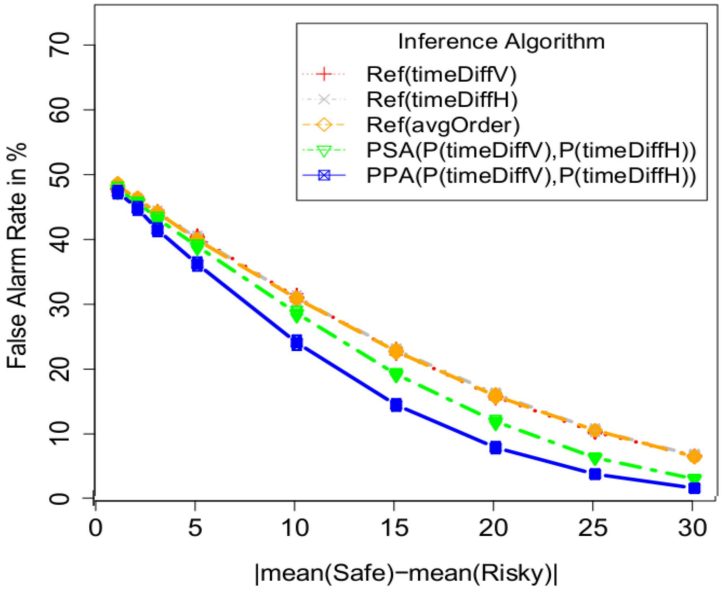


Figure 4.7 A comparison of the False Alarm Rate of the Ref algorithms, PSA algorithm and PPA algorithm using 2 observed features (refer to Figure 4.5 for the definition of the algorithms)

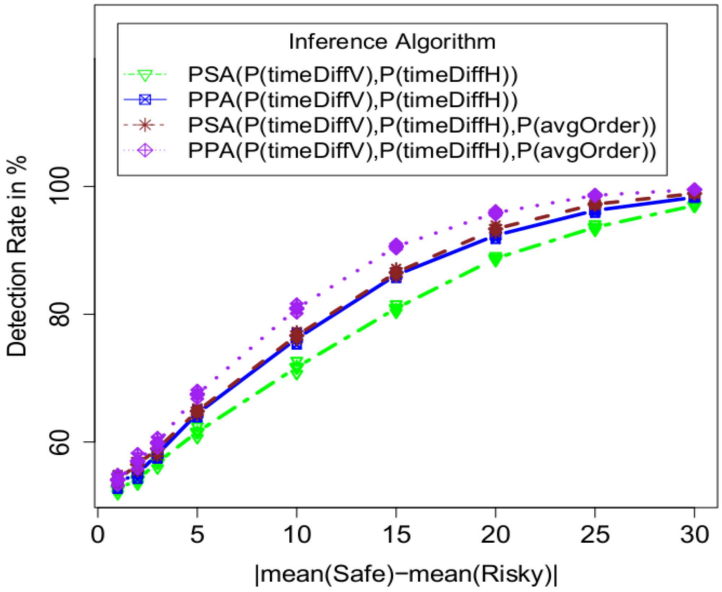


Figure 4.8 A comparison of the Detection Rate of the PSA algorithm and the PPA algorithm using 2 and 3 observed features (refer to Figure 4.5 for the definition of the algorithms)

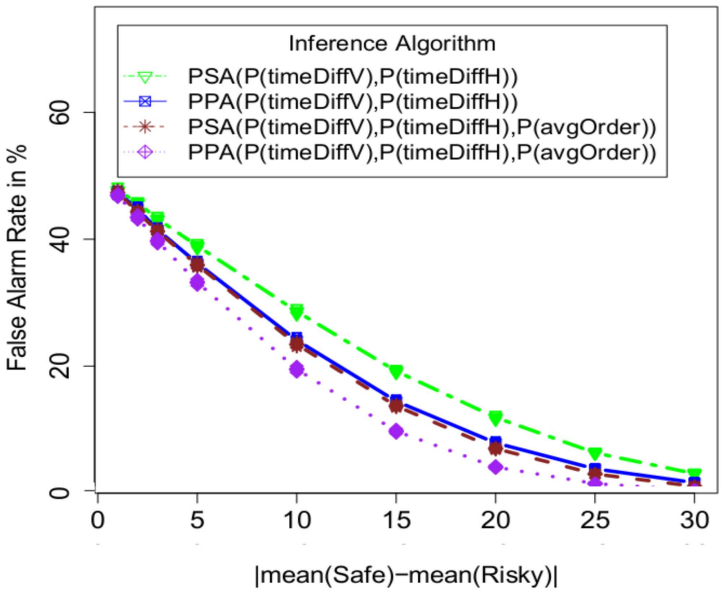


Figure 4.9 A comparison of the False Alarm Rate of the PSA algorithm and the PPA algorithm using 2 and 3 observed features (refer to Figure 4.5 for the definition of the algorithms)

Figure 4.8 and Figure 4.9 indicate that the PPA algorithm performs better than the PSA algorithm, both in terms of the detection rate and the false alarm rate. They also show that the accuracy of the two algorithms increases with the number of the observed features involved. Hence, the PPA algorithm improves much better the inference accuracy of risky queries in EPCglobal Discovery Services under the assumptions stated in Section 4.5.1.

4.6 A Hidden Markov Model Security Scheme for Query State Inference

4.6.1 Hidden Markov Model (HMM)

A Hidden Markov Model (HMM) is a statistical model based on a fully observable Markov Model. Although a Markov Model and an Hidden Markov Model resemble in being stochastic models modeling and analyzing processes where the state at a certain time t depends, in a non-deterministic way, on the previous states, they differ in how observable the system state is. While Markov Models are restricted to processes where the states are fully observable by the means of observable physical events, Hidden Markov Models extend the model to controlled processes where the states are hidden and the observable physical events are related to the state of the system without directly inferring that state.

Hidden Markov Models are statistical models suitable for systems which consist of two inter-related stochastic processes; i.e., the states stochastic process and the observations stochastic process. The states stochastic process is hidden and can only be observed through the observations stochastic process. It has been proposed by Rabiner in [123] to extend the traditional Markov Models to cases where the observations are a probabilistic function of the states.

Let us consider a system undergoing a change of states within a set of N possible states $\{S_1, S_2, ..., S_N\}$ at a time instant $t = 1, 2, ...$, and let us denote the actual state of the system at time t as q_t. Let us also denote the set of M possible observations (M

being the number of distinct observation symbols per state, i.e., the alphabet size) as $V = \{V_1, V_2, ..., V_M\}$, and let the actual observation symbol emitted at time t as O_t.

4.6.1.1 Elements of a Discrete HMM (DHMM)

A discrete HMM (DHMM), denoted by λ and modeling the considered system, is illustrated in Figure 4.10. It can be written as

$$\lambda = (\pi, A, B) \tag{4.5}$$

where π is the initial state probabilities vector whose the element $\pi_i, 1 \leq i \leq N$ describes the probability that the system's initial state (actual state q_1 at time $t = 1$) is S_i.

$$\pi = \{\pi_i, 1 \leq i \leq N\}, \sum_{i=1}^{N} \pi_i = 1 \tag{4.6}$$

$$\pi_i = P(q_1 = S_i), \pi_i \geq 0, 1 \leq i \leq N \tag{4.7}$$

A is the state transition probabilities stochastic matrix whose the element $a_{ij}, 1 \leq i, j \leq N\}$ describes the probability that the system transits to state S_i at any time t, given that it has been in state S_j at time $t - 1$. This definition applies only to first order Markov Chains, which stipulates the Markov Property; the state q_t at time $t = 2, 3, ...$ depends only on the immediate predecessor state q_{t-1} at time $t - 1$.

$$A = \{a_{ij}, 1 \leq i, j \leq N\}, \sum_{i=1}^{N} a_{ij} = 1 \tag{4.8}$$

$$a_{ij} = P(q_t = S_i | q_{t-1} = S_j), a_{ij} \geq 0, 1 \leq i, j \leq N \tag{4.9}$$

B is the observation stochastic matrix whose the element $b_{ij}, 1 \leq i \leq M, 1 \leq j \leq N\}$ describes the probability that the system emits observation V_i given that it is in state S_j, at any time $t = 1, 2, ...$

$$B = \{b_{ij}, 1 \leq i \leq M, 1 \leq j \leq N\}, \sum_{i=1}^{M} b_{ij} = 1 \tag{4.10}$$

$$b_{ij} = P(O_t = V_i | q_t = S_j), b_{ij} \geq 0, 1 \leq i \leq M, 1 \leq j \leq N \tag{4.11}$$

Figure 4.11 illustrates the DHMM Model, defined in Figure 4.10, in action, with $O = \{O_1, O_2, ..., O_T\}$ a sequence of T observations where each observation O_t (observed at time t) is one of the symbols from the set $V = \{V_1, V_2, ..., V_M\}$.

4.6.1.2 Elements of a Continuous HMM (CHMM)

Similar to a DHMM, a Continuous HMM (CHMM) λ can be written $\lambda = (\pi, A, B)$, where the definitions of the initial state probabilities vector π (Equation 4.6 and Equation 4.7) and the state transition probabilities matrix A (Equation 4.8 and Equation 4.9) are exactly the same as described for a DHMM in Section 4.6.1.1.

The only difference between a DHMM and a CHMM lays in the definition of the emission probabilities, which are defined for continuous observations x belonging continuous infinite domain $\mathcal{X}$ rather than a discrete finite domain, e.g., $x \in \mathbb{R}_D$, D being the dimension of the observed vector.

Hence for a CHMM, B denotes the vector of functions whose the element $b_i, 1 \leq i \leq N$ describes the probability density function on the continuous domain $\mathcal{X}$, given the system is in state S_i, at any time $t = 1, 2, ...$

$$B = \{b_i, 1 \leq i \leq N\} \tag{4.12}$$

$$\forall x \in \mathcal{X}, b_i(x) = P(O_t = x | q_t = S_i), b_i(x) \geq 0, \int_{\mathcal{X}} b_i(x)\, dx = 1 \tag{4.13}$$

A typical probability density function that is widely used in modeling CHMM is the Gaussian distribution. B is then defined as mixture of Gaussians, identified by a vector $\mathcal{G}$ of M Gaussian distributions $\mathcal{G} = \{\mathcal{N}(\mu_k, \sigma_k), 1 \leq k \leq M\}$, and a vector $\mathcal{C}_i$ of M mixture components per state $i, 1 \leq i \leq N$, $\mathcal{C}_i = \{c_{ik}, 1 \leq k \leq M\}$, $\sum_{k=1}^{M} c_{ik} = 1$. The Gaussian mixture $b_i(x)$ is then written as follows:

$$b_i(x) = \sum_{k=1}^{M} c_{ik} \mathcal{N}(x | \mu_k, \sigma_k) \tag{4.14}$$

4.6.1.3 The Three Basic Problems Solved by an HMM

A Hidden Markov Model represents the foundation for solving three basic problems [123]:

- The evaluation problem: This problem looks for the probability that a particular sequence of observations $O = \{O_1, O_2, ..., O_T\}$ is generated by a given HMM model $\lambda = (\pi, A, B)$. i.e., $P(O|\lambda)$;
- The inference problem (called also the decoding problem): This problem looks for the most probable sequence of hidden states $X = \{X_1, X_2, ..., X_T\}$ that "best" explains a given sequence of observations $O = \{O_1, O_2, ..., O_T\}$ under a given HMM model $\lambda = (\pi, A, B)$. i.e., $\underset{X}{\text{argmax}} P(X|O, \lambda)$;
- The learning problem: This problem looks for the HMM model parameters $\lambda = (\pi, A, B)$ that "best" fit a given sequence of observations $O = \{O_1, O_2, ..., O_T\}$. In other words, adjust the model parameters to maximize $P(O|\lambda)$.

Solutions to the three problems above have been detailed in [123].

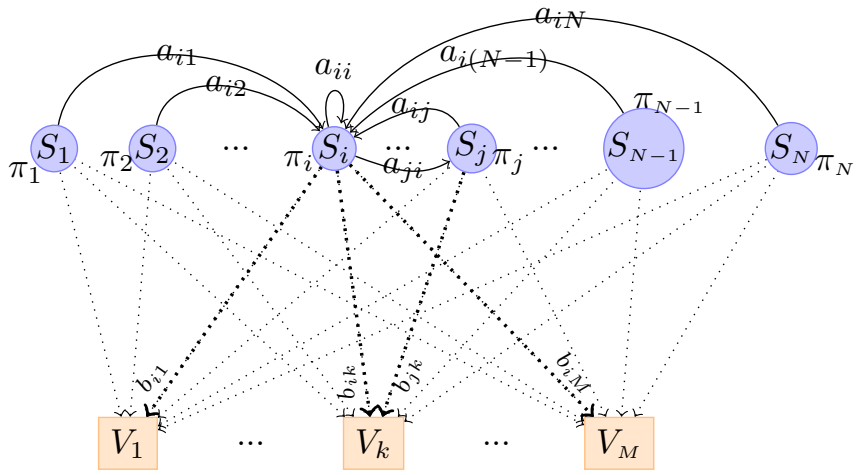


Figure 4.10 An illustration of a discrete HMM $\lambda = (\pi, A, B)$ modeling a system with N states, each of which can emit M discrete symbols $V_1, V_2, ..., V_M$. $a_{ij}, 1 \leq i, j \leq N$ is the probability that the modeled system transits from state S_i to state S_j. $b_{ik}, 1 \leq i \leq N, 1 \leq k \leq M$ is the probability that the system emits symbol V_k while in state S_i. $\pi_i, 1 \leq i \leq N$ is the probability that the system's initial state is S_i.

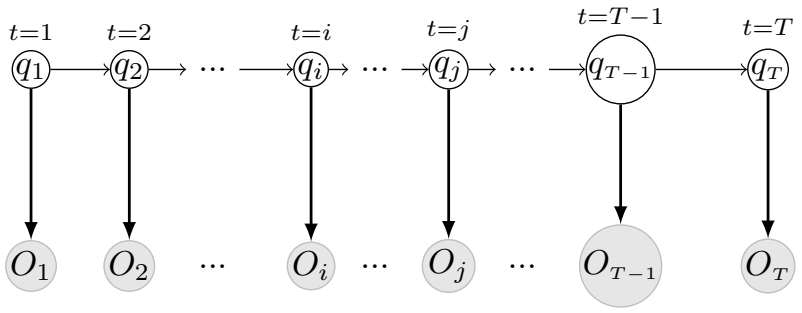


Figure 4.11 The HMM model, illustrated in Fig 4.10, in action.

4.6.2 Assumptions

Our proposed CHMM security scheme makes the following assumptions:

- Each and every query, directed to the Discovery Services System, inquires about a "shipment" of product items rather than one item. A "shipment" contains only items of the same product category. This assumption is very reasonable since inter-company transactions take place in terms of shipments of items.
- Each and every query, performed by a company C regarding a certain "shipment", corresponds to one and only one business step $BS_i, 0 <= i < N_{BS}$. Therefore, a company C might issue one or more queries for the same "shipment" depending on the number of business steps that "shipment" goes through within its premises. This assumption holds simply by defining logical business steps where needed. A company can define its own set of internal business operations detailing a business step $BS_i, 0 <= i < N_{BS}$.
- The number of items per query follows a Poisson distribution with an average δ_S for safe queries, and a different average δ_R for risky queries. Such average can be computed statistically using past data. This assumption holds because usually when an attacker decides to intrude, his/her intrusion attempts are successive until either success or decision to quit. Since this number of attempts is not known, a Poisson distribution with average δ_R is chosen to model it. Moreover, each company builds a certain pattern in its business operations over time. One aspect of this pattern is the number of (safe) queries initiated by that company before an intrusion attempt. Since this number of (safe) queries is not known, a Poisson distribution with average δ_S is chosen to model it.

- The Markov Property: the current state of a given query, performed by a company C at time t, depends only on the state at time $t-1$ of a similar query, that is the query performed right before the current one, by the same company C inquiring about the same product category during the same business step. This assumption has been made to simplify the CHMM model.

- The extracted feature ${}_{avgOrder}{}^{p,C}_{BS_i}$ follows two different Gaussian distributions depending on the state of the corresponding query. Hence, ${}_{avgOrder}{}^{p,C}_{BS_i}$ follows $\mathcal{N}(\mu^R_{p,C,BS_i}, \sigma^R_{p,C,BS_i})$ for risky queries, and $\mathcal{N}(\mu^S_{p,C,BS_i}, \sigma^S_{p,C,BS_i})$ for safe queries.

4.6.3 Proposed CHMM for Query State Inference

We propose a query state inference scheme based on a Continuous HMM $\lambda_{CHMM} = (\pi, A, B)$ as defined in Section 4.6.1.1 and Section 4.6.1.2. B being mixtures of Gaussians identified by a vector of 2 Gaussian distributions $\mathcal{G}$, a vector $\mathcal{C}_S$ of 2 mixture components for safe queries and a vector $\mathcal{C}_R$ of 2 mixture components for risky queries.

$$\mathcal{G} = \{\mathcal{N}(\mu^S_{p,C,BS_i}, \sigma^S_{p,C,BS_i}), \mathcal{N}(\mu^R_{p,C,BS_i}, \sigma^R_{p,C,BS_i})\} \tag{4.15}$$

$$\mathcal{C}_S = \{1, 0\} \tag{4.16}$$

$$\mathcal{C}_R = \{0, 1\} \tag{4.17}$$

Where $\mathcal{N}(\mu^S_{p,C,BS_i}, \sigma^S_{p,C,BS_i})$, defines the pattern of the feature $avgOrder$ for safe queries, and $\mathcal{N}(\mu^R_{p,C,BS_i}, \sigma^R_{p,C,BS_i})$, defines the pattern of the feature $avgOrder$ for risky queries.

Upon receipt of a given EPC query ${currentQuery}^{p,C}_{BS_i}$, originating from given company C in the context of a given business step $BS_i, 0 <= i < N_{BS}$ with regards to a given shipment of items having product category p, the Discovery Services system calls the CHMM λ_{CHMM}, defined above, in order to assess the risk of the query, and hence infer its most probable state. The proposed CHMM is called to solve the inference problem (defined in Section 4.6.1.3). As a result, the most probable sequence of hidden states

$\{q_1, q_2, ..., q_T\}, q_i \in Safe, Risky, 1 \leq i \leq T$ that "best" explains a given sequence of T observations $O = \{O_1, O_2, ..., O_T\}$, is inferred.

The proposed CHMM-based security scheme assumes that the Feature Query Extraction procedure has been carried out beforehand as described in Section 4.4. As a result, the values of the extracted feature $_{avgOrder}{}^{p,C}_{BS_i}$ can be passed as input to the proposed CHMM.

$$\{q_1, q_2, ..., q_T\} = \underset{X_1, X_2, ..., X_T}{\operatorname{argmax}} P(X_1, X_2, ..., X_T | O_1, O_2, ..., O_T, \lambda) \tag{4.18}$$

4.6.4 Performance Evaluation

4.6.4.1 Simulation Setup and Parameters

Because there is no available real data that we can test our CHMM-based security scheme with, we generated data using the Gaussian distributions assumed in Section 4.6.2. The parameters of these Gaussian distributions (i.e., the mean and the standard deviation) are defined in the List of Symbols and Notations at the beginning of this thesis. The data consist of two large sequences of vectors of the form $(trueState, {}_{avgOrder}{}^{p,C}_{BS_i})$. $trueState$, whose the value represents either "safe" or "risky" query, is randomly selected using a uniform distribution. Depending on the selected state, we generate N Gaussian random numbers representing the feature $_{avgOrder}{}^{p,C}_{BS_i}$), using the corresponding Gaussian distributions. N being the number of queries of the same state that are performed sequentially, before a query with a different state takes place. N is assumed to follow a Poisson distribution with an average δ_S for safe queries, and a different average δ_R for risky queries.

As mentioned above, the data consist of two large sequences of vectors of the form $(trueState, {}_{avgOrder}{}^{p,C}_{BS_i})$. The first sequence, consisting of 5000 vectors, is used to train the CHMM λ_{CHMM} and the second, consisting of 2000 vectors, is used to evaluate the performance of the CHMM λ_{CHMM} in predicting the state of the queries, each of which is represented by the values of the feature $_{avgOrder}{}^{p,C}_{BS_i}$. The training process focuses on finding the combination of initial state probabilities, state transition probabilities and

mixture of Gaussians representing the emission probabilities which best explains the training data (the first sequence of vectors).

4.6.4.2 Evaluation Approach

Let us denote $trueStateVEC$ the vector of the true states of the queries, and $predictedStateVEC$ the vector of the queries' states predicted by the proposed CHMM λ_{CHMM}, described in Section 4.6.3, and by the reference algorithm $Ref(P(avgOrder))$, defined in Figure 4.5. Our evaluation approach consists of computing and comparing the detection rate and the false alarm rate for these two algorithms.

We first compute the following:

- TP (True Positive) number of queries correctly inferred as risky
- FP (False Positive) number of queries incorrectly inferred as risky
- TN (True Negative) number of queries correctly inferred as safe
- FN (False Negative) number of queries incorrectly inferred as safe

Then we compute the detection rate and the false alarm rate using Equation 4.19 and Equation 4.20.

$$DetectionRate = \frac{TP}{TP + FN} \tag{4.19}$$

$$FalseAlarmRate = \frac{FP}{FP + TN} \tag{4.20}$$

4.6.4.3 Results

We conducted extensive experiments to evaluate the performance of the proposed continuous HMM λ_{CHMM}, compared to the reference algorithm $Ref(P(avgOrder))$, defined in Figure 4.5. Each experiment starts by generating simulation data as described

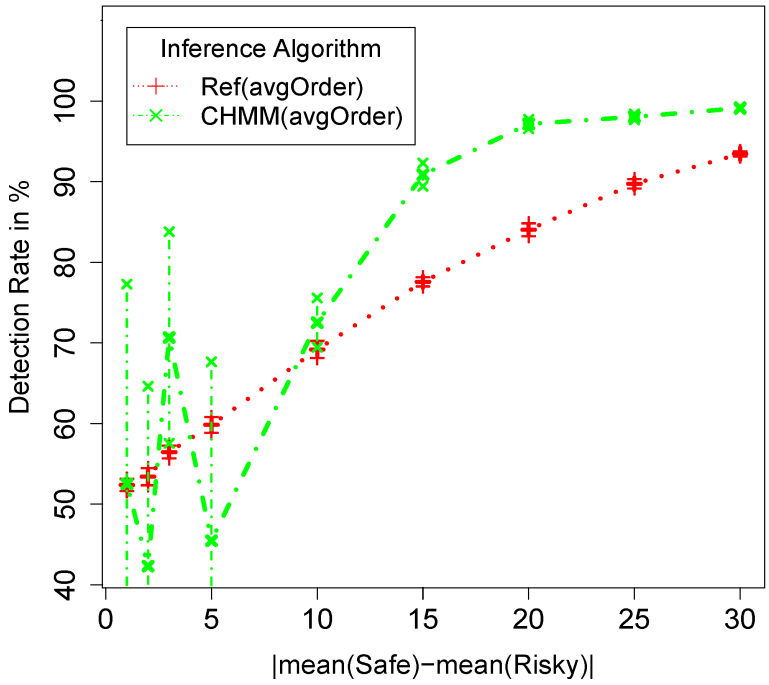


Figure 4.12 A comparison of the Detection Rate between the reference algorithm $Ref(P(avgOrder))$ (defined in Figure 4.5) and the proposed CHMM-based Model (described in Section 4.6.3), with a 95% confidence interval

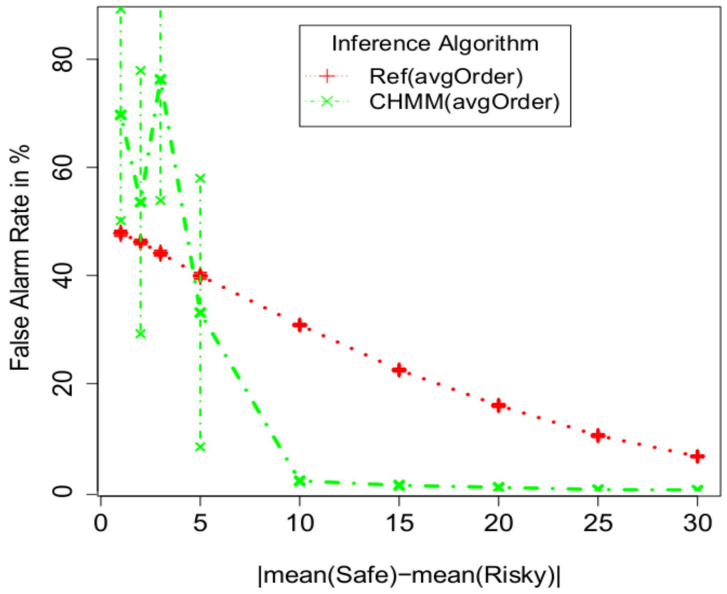


Figure 4.13 A comparison of the False Alarm Rate between the reference algorithm $Ref(P(avgOrder))$ (defined in Figure 4.5) and the proposed CHMM-based Model (described in Section 4.6.3), with a 95% confidence interval

in Section 4.6.4.1. Then, we run 10 experiments per scenario. A scenario defines the gap between the mean of the "safe query" Gaussian distribution and that of the "risky query" Gaussian distribution. The smallest the gap is, the most challenging the inference is. This is because a small gap suggests the data corresponding to the two Gaussian distributions is not easily distinguishable. The results show that the CHMM makes great use of its learning capabilities of data sequentiality to perform better than the reference algorithm.

We plotted the average detection rate and the average false alarm rate, defined in Section 4.6.4.2, with a 95% confidence interval.

Figure 4.12 illustrates the average detection rate of the reference algorithm $Ref(P(avgOrder))$, defined in Figure 4.5, and that of the proposed CHMM model, described in Section 4.6.3.

Figure 4.13 shows the average false alarm rate of the reference algorithm $Ref(P(avgOrder))$, defined in Figure 4.5, and that of the proposed CHMM inference model, described in Section 4.6.3..

Figure 4.12 and Figure 4.13 indicate that the CHMM inference model performs better than the reference algorithm both in terms of the average detection rate and the average false alarm rate, when the patterns of the Gaussian distributions featuring the two different states are distinct enough; in other words, when the expression $|mean(Safe) - mean(Risky)| > 5$ holds. On the other hand, the reference algorithm $Ref(P(avgOrder))$ offers a better performance when $|mean(Safe) - mean(Risky)| \leq 5$. These observations hold because when the Gaussian distributions, modeling safe and risky queries, are distinguishable enough, the CHMM parameters, learned during the training phase, are more accurate. Furthermore, the CHMM makes a decision regarding the state of a given query based on both the current observations and the previous states, while the reference algorithm makes its decision based only on the current observation, without taking into consideration the previous state.

4.7 Conclusion

In this chapter, we have presented two security schemes securing even further the EPCglobal Discovery Services. Both security schemes aim at detecting suspicious Discovery Services lookup queries in the EPCglobal network.

The first proposed security scheme is based on the Gaussian model presented in Section 4.5.2. A lookup query is first converted into a vector of observed real values. These observed values are assumed to follow a Gaussian distribution both for safe and risky queries. Then, a classification algorithm computes a score for each state; i.e., safe and risky. This score is then used to infer the state of the query. We conducted extensive experiments. The results show that, compared to a simple Gaussian model, our proposed scheme improves both the detection rate and the false alarm rate.

The second proposed security scheme is based on a Continuous Hidden Markov Model (CHMM). A lookup query is first converted into a vector of observed real values. These observed values, which are assumed to follow a Gaussian distribution both for safe and risky queries, are used to train the CHMM. Once trained, the CHMM is then used to infer the state of the query at hand. We conducted extensive experiments. The results show that our CHMM-based security scheme enhances even better both the detection rate and the false alarm rate.

Chapter 5

A Hierarchical Architecture for Distributed EPCglobal Discovery Services

5.1 Introduction

Given the important advances already undertaken in the development of the EPCglobal framework, any future IoT-based business infrastructure will most probably build on the current EPCglobal Network. Moreover, Discovery Services are expected to enable the users, both stakeholders and end customers, to reach all "relevant" information with regard to the inquired Electronic Product Code (EPC). EPC is a universal identifier designed to uniquely identify physical objects, loads, locations, assets and other entities which are to be tracked, or otherwise identified [26]. Its structure of and that of company prefix will require upgrade in order to be much more flexible with respect to the variety of distributed architectures spanning the IoT.

The EPCglobal Network promises, via the traceability services provided by its components, wide business visibility to supply chain actors. This real-time, low-cost and accurate visibility, provided at the item-level, would allow companies to increase their

profitability while increasing customers satisfaction. The core foundation of those traceability services is the lookup function, which consists of localizing sources pointing to information corresponding to the inquired EPC [7].

The current design of the EPCglobal lookup service, based on the ONS, brings up various concerns. Particularly, the centralized architecture featuring the current design of the ONS has several worrying drawbacks as it has been elaborated in [8–11]. Moreover, it provides the information sources based on the company prefix of the EPC at hand. As a result, only the manufacturer information sources are provided in response to an ONS lookup query. Such limitation explains well enough the urgent need for distributed Discovery Services as a lookup service, rather than ONS.

Although the final definition of Discovery Services requirements is not yet closed by EPCglobal. A number of proposals have enriched the literature ranging from architectures extending the currently deployed ONS system such as in [10, 15–17] to complex solutions based on a clean-slate approach such as in [9, 18–20]. Many of these works have focused on the feasibility of P2P solutions and their high scalability. However, to the best of our knowledge, none has looked at enhancing both the scalability of the proposed P2P solutions, while improving the responsiveness of the lookup queries. Such challenge is of crucial importance given its impact on the survivability of large-scale applications in the future IoT, such as the EPCglobal network.

In order to ensure and sustain the required quality of information flow between trading partners, focus has to be put towards meeting two major requirements in the design of any Discovery Services architecture, coupled with the functionality requirement. First, the control of the EPCglobal Network has to be equally shared between the subscribing companies, and hence between the countries hosting those companies. Such shared control would encourage companies/countries worldwide to accept the EPCglobal Network as the core engine of the future global business infrastructure. Second, given the super-exponentially-growing number of objects to be tracked in the EPCglobal Network, scalability of any proposed architecture for Discovery Services, and responsiveness of its underlying lookup queries, have to be regarded as a pivotal design requirement. Obviously the security requirement is likewise important, but it is beyond the scope of this work.

This chapter introduces two architectures for distributed and scalable Discovery Services; a Flat Distributed Architecture (FDA), which speaks for the majority of the distributed architectures that have been proposed in the literature so far, and a Hierarchical Distributed Architecture (HDA) representing the core contribution of this work. FDA is introduced solely for performance assessment (Chapter 6) reasons. It plays the role of a reference against which HDA is compared. The comparison aims at proving that HDA is more scalable and its lookup responsiveness is much better.

We describe in detail our proposed Hierarchical Distributed Architecture (HDA) for overlay networks, intended for the future IoT-based business infrastructure. HDA refers, as detailed in Section 5.6, to a set of proximity-based structured Peer-to-Peer (P2P) overlay networks with inter-routing capabilities (routing of queries amongst nodes belonging to different overlay networks), while FDA refers, as described in Section 5.5, to one large structured P2P overlay network, involving all the nodes in one P2P system.

The remainder of this chapter is structured as follows. In Section 5.2, we present the assumptions made during design of the proposed HDA architecture. Section 5.3 describes in detail the assumption of hierarchical identification. In Section 5.4, we present the notations and the definitions used to describe both the proposed HDA architecture and the reference FDA architecture. In Section 5.5, we describe functionality, merits and limitations of the reference FDA architecture. In Section 5.6, we first introduce the tree model adopted in the design of the proposed HDA architecture. Then , we describe its functionality, merits and limitations. We also discuss its applicability to the EPCglobal Network, its scalability and its query responsiveness. Before concluding this chapter, we briefly discuss in Section 5.7 the impact of the how inter-overlay routing is carried out (vertically vs horizontally) on the performance of the proposed HDA architecture.

5.2 Assumptions

Throughout this chapter, we assume the following:

5.2.1 Geographical Binding

In order to build an HDA-based data lookup system, in the complex IoT, it is important to check an important assumption related to the structure of the keys indexing the handled data. Those keys should hold some geographical information, unambiguous enough to exclusively sort the nodes out into different geographical groups.

This assumption is in fact verified in the EPC structure, since the first three digits in the company prefix correspond to the country (geographical location) which has issued that company prefix [6, 124].

Although Globalization has left no meaning to geographical diversity, since companies can do business anywhere in the world regardless of their location, the geographical information, related to the EPCglobal company prefix issuing country, is used in the proposed HDA architecture for inter-country routing of lookup queries. An example of inter-country data storage/retrieval is illustrated in Figure 5.3, in which a Canadian company, under receipt of a lookup query regarding an EPC belonging to an American company, routes it immediately to the American P2P overlay network.

5.2.2 Hierarchical Identification

We assume hierarchical naming for both the keys' and the nodes' identifiers. By hierarchical naming, we mean that the nodes identifiers are tightly bound to their geographical location. As a result, the nodes physical proximity is reflected in the similarity of the leftmost part of their identifiers. Moreover, the size of the similar leftmost part of a given pair of identifiers indicates the physical distance separating the two corresponding physical nodes; the smaller the size of the similar leftmost part of a given pair of identifiers is, the closer the two corresponding nodes are.

This assumption is very reasonable, given that it is already partially satisfied in the company prefix assigned to each and every company all over the world. A company prefix, illustrated in Figure 2.2, is a number which uniquely identifies companies all over the globe. It is assigned by GS1 organization. Each company applies for its own company prefix locally in the country where it is located. Once obtained, this company

prefix is then used to create any subsequent EPC number, to be linked to the company, and hence to the country having issued the company prefix.

5.2.3 One Node Per Company

For simplicity reasons, we assume that each company is represented by a one and only one node in the overlay network. This assumption is reasonable because any company wishing to have more than one node in the overlay, can simply deploy a gateway to handle its nodes locally, and register the gateway as its representative node.

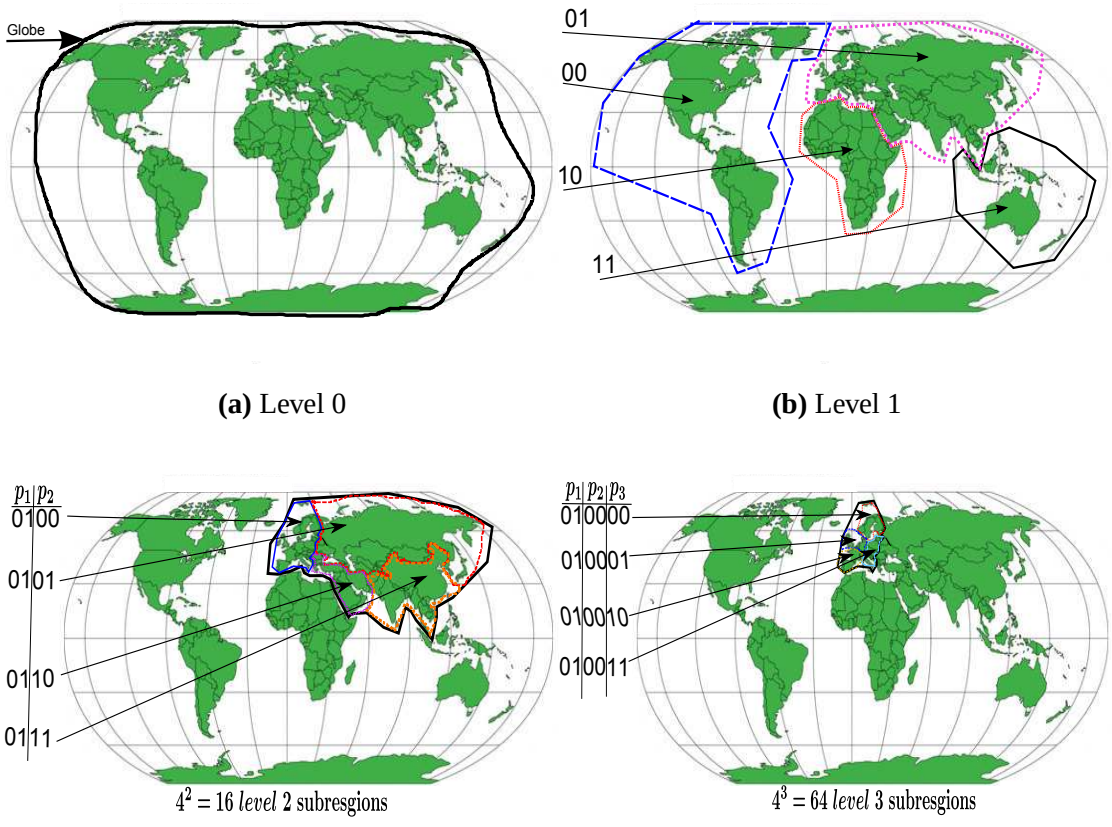
5.3 Hierarchical Naming

5.3.1 EPC Company Prefix

A company prefix, illustrated in Figure 2.2, is a number which uniquely identifies companies all over the globe. It is assigned by GS1 organization. Each company applies for its own company prefix locally in the country where it is located. Once obtained, this company prefix is then used to create any subsequent EPC number, to be linked to the company, and hence to the country having issued the company prefix.

5.3.2 A More Comprehensive Scheme

As shown in Figure 2.2, the length of the company prefix can be up to 40 bits. In such scheme, we can use up to 24 bits to hierarchically name different geographical regions, instead of identifying only the issuing country. With 24 bits, we can uniquely identify up to 2^24 different regions over the globe. The hierarchical naming procedure is illustrated in Figure 5.1. Instead of assigning "random" country codes without taking geographical correlations into consideration, the available 24 bits can be used in smartly in such a way that geographical proximity is reflected in the identifiers. As illustrated in Figure 5.1a identification starts at $level - 0$ using 2 bits identifiers (00, 01, 10 and 11), splitting the globe into four large regions. Each $level - 0$ region, with identifier



(a) Level 0 **(b)** Level 1

(c) Level 2 **(d)** Level 3

Figure 5.1 An example of a three levels Hierarchical Geographically-based Company Prefix Assignment

$I, I \in \{00, 01, 10, 11\}$, is then split into 4 subregions whose the identifiers are $I00$, $I01$, $I10$ and $I11$. Figure 5.1b shows the subregions of $level - 0$ region 01.

In general, each $level - i$ region, with identifier I, is then split into 4 subregions whose the identifiers are $I00$, $I01$, $I10$ and $I11$. Figure 5.1c shows the subregions of $level - 1$ region 0100.

Such comprehensive scheme would allow us to go up to 12 levels with no required changes to the current structure of the EPC. This is an extremely interesting advantage of our proposal, in the sense that the upgrade of the current EPCglobal framework towards implementing the comprehensive hierarchical naming scheme described in Figure 5.1, would be feasible and cost-effective.

Unless otherwise stated, the expressions "Overlay Network", "Flat overlay network" and "Flat Distributed Architecture" refer to the same thing in the remainder of this chapter; a flat structured P2P overlay network, for which the nodes are identified using Hierarchical Naming as described above in this section. By "flat", we mean that all the nodes belong to one large same overlay network. As a result, given the randomness brought by hashing in the identification scheme of structured P2P overlays, any pair of nodes, no matter how far away they are from each other, could be logically linked as "close" neighbors in the overlay.

Moreover, the expressions "Hierchical Overlay Network" and "Hierchical Distributed Architecture" refer to the same thing in the remainder of this chapter; a set of flat structured P2P overlay networks identified using Hierarchical Naming as described above in this section, each of which involves a subset of the N nodes. Each node, belonging to a given overlay network, has a unique identifier binding the overlay network unique identifier with a unique number distinguishing it from the other nodes belonging to the same network.

5.4 Notations and Definitions

Let us consider a set of N nodes $S = \{d_1, d_2, ..., d_N\}$, with N being a very large number. Each node $d_i, 1 \leq i \leq N$ belongs to one and only one $level - K$ subregion, as can be visualized in an extension of Figure 5.1 with K levels. Let us also assume that each subregion in $level\ j, 0 \leq j \leq K-1$ can be split into M smaller subregions, each of which belongs to $level\ j+1$. For simplicity reasons, we assume that M is an exponent number of 2, i.e., $M = 2^m$. In Figure 5.1, $M = 4$ and $m = 2$.

Since each $level\ v, 0 \leq v \leq K$ contains M^v subregions (Level 0 refers to the whole globe as shown in Figure 5.1), let us we denote by $r_u^v, 0 \leq v \leq K, 1 \leq u \leq M^v$ subregions belonging to $level\ v$.

We denote by $id(r_u^v) = p_1...p_v$, where $1 \leq v \leq K, 1 \leq u \leq M^v, 1 \leq p_i \leq M, 1 \leq i \leq v$, the unique identifier of each subregion r_u^v belonging to $level\ v$. Each subregion identifier $p_1...p_v$ is made of two parts; $p_1p_2...p_{v-1}$ identifies the parent subregion, while p_v distinguishes the M subregions belonging to that parent subregion. This notation

is used in Figure 5.1c and Figure 5.1d to distinguish different subregions belonging to $level\ 2$ and $level\ 3$ respectively.

As mentioned above, each node $d_i, 1 \leq i \leq N$ belongs to one and only one $level - K$ subregion $r_u^K, 1 \leq\ u \leq M^K$, uniquely identified by $p_1^i p_2^i ... p_K^i, 1 \leq p_i \leq M$. Since each $level - K$ subregion contains $\frac{N}{M^K}$ nodes, node d_i is identified as follows; $id(d_i) = id(r_u^K)q$ where $1 \leq q \leq \frac{N}{M^K}$ and $id(r_u^K)$ refers to the identifier of the $level - K$ subregion where node d_i is physically located.

Definition 5.1. Let d_i and d_j, two nodes in an overlay network with identifiers $p_1^i p_2^i ... p_K^i q^i, 1 \leq p_i \leq M, 1 \leq q^i \leq \frac{N}{M^K}$ and $p_1^j p_2^j ... p_K^j q^j, 1 \leq p_j \leq M, 1 \leq q^j \leq \frac{N}{M^K}$ respectively. We define the logical distance $LD(d_i, d_j)$, between the identifiers of two nodes d_i and d_j, as follows:

$$LD(d_i, d_j) = \begin{cases} K, & \text{if } p_1^i \neq p_1^j \\ K-1, & \text{if } p_1^i = p_1^j, p_2^i \neq p_2^j \\ . & \\ . & \\ . & \\ 2, & \text{if } p_1^i = p_1^j, p_2^i = p_2^j, ..., p_{K-2}^i = p_{K-2}^j, p_{K-1}^i \neq p_{K-1}^j \\ 1, & \text{if } p_1^i = p_1^j, p_2^i = p_2^j, ..., p_{K-1}^i = p_{K-1}^j, p_K^i \neq p_K^j \\ 0, & \text{if } p_1^i = p_1^j, p_2^i = p_2^j, ..., p_K^i = p_K^j \end{cases} \tag{5.1}$$

5.5 Flat Distributed Architecture (FDA)

FDA refers to a structured P2P architecture, in which each subscribing company is represented by a single node. The identifier of this node is the hash value of its identifier. FDA is called "Flat" because all the nodes belong to one same large structured P2P overlay network, regardless of their geographical locations. Figure 5.2 illustrates a FDA-based P2P overlay including nodes from different countries. FDA is introduced in this chapter to serve as a reference to which HDA, described in Section 5.6, is compared.

In this section, we explain how Discovery Services work in FDA, and we present the merits and the limitations of FDA.

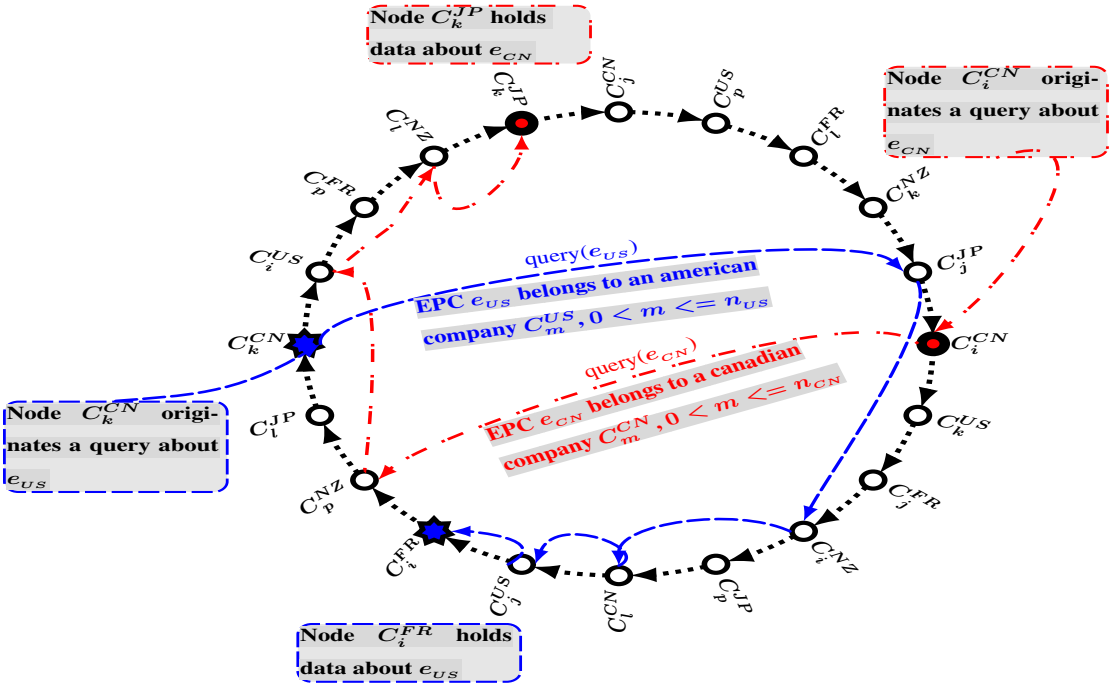


Figure 5.2 FDA-based overlay network for EPCglobal Discovery Services illustrating two lookup queries for EPC data retrieval

5.5.1 Functionality of FDA

In a flat structured P2P overlay architecture, nodes are usually identified through hashes of their unique identifiers. The hashing is what allows each and every key to be precisely assigned to one and only one node in the overlay. As a result, the lookup function is much more efficient than it is in unstructured overlay P2P topologies. Structured P2P overlay systems use the Distributed Hash Table (DHT) as a substrate, in which given data (i.e., value), identified with a given identifier (i.e., key), is deterministically stored and sought at a certain peer whose the identifier corresponds to the key at hand.

In FDA, each node in the overlay records routing information regarding a set of nodes, including its immediate successor. Those nodes are determined in such a way to minimize the number of hops required per lookup query while minimizing the amount of routing information each node has to hold. This ensures efficient data lookup while balancing the loads among the overlay nodes. Chord [34] is one of the well-known

distributed algorithms in which the number of hops per lookup query scales logarithmically with the number of nodes. Chord [34] will be used to simulate FDA and HDA on Planetsim [125]. The objective of the simulation is to evaluate the performance, and then compare the two architectures.

In terms of functionality, a FDA-based overlay Network is simply a flat P2P Network whose the nodes belong to the subscribing companies. The data lookup in a FDA-based overlay Network can based on any well-known structured P2P technique such as Distributed Hash Table (DHT). All the nodes in this architecture belong to the same P2P network, no matter what their geographical locations are, as shown in Figure 5.2.

5.5.2 Merits and Limitations of FDA

Although FDA architecture may not seem attractive to companies because of its cost and its risks related to the possible exposure of their own data, it comes with a major merit related to its distributed nature; it grants the subscribing companies, and hence the hosting countries, equal shares of the business infrastructure control. This would solve the acceptance problem related to the excessive concentration of power into the hands of a single organization/country. Such inter-company equally shared control overbalances by far the drawbacks of FDA architecture since well rounded security mechanisms, such as public/symmetric cryptography and access control, can be deployed to dissipate all the fears pertaining to the potentially dangerous exposure of the EPC data.

Moreover, the randomness of hashing, widely used in DHT-based structured P2P overlay systems, makes of distant nodes, logical neighbors in the overlay network. As a result, the average latency of the hops within a flat P2P overlay network is rather high.

5.6 Hierarchical Distributed Architecture (HDA)

HDA refers to a set of structured P2P overlay networks, each of which is attached to one $level-K$ subregion and involves only nodes representing companies that are physically located in this subregion. The identifier of each node is the hash value of its identifier. HDA is called “Hierarchical” because it is based on a multi-level tree structure where

each node at a given level, except the root, points to one parent node at the immediate upper level, representing the subregion in which the child node is physically located. Moreover, the nodes, representing the subscribing companies, belong to different P2P overlay networks depending on their geographical locations. Figure 5.3 illustrates the HDA-based overlay network corresponding the flat FDA-based overlay network shown in Figure 5.2. In Figure 5.3, each country owns its own P2P overlay network including only nodes hosted in that country, while in Figure 5.2, one same P2P overlay network involves all the nodes regardless of their geographical locations.

In this section we first describe the tree model of HDA splitting the one large P2P overlay network into several smaller P2P overlay networks, then we explain how Discovery Services work in HDA, and finally, we present the merits of HDA, as well as its drawbacks with regard to its overhead compared to FDA.

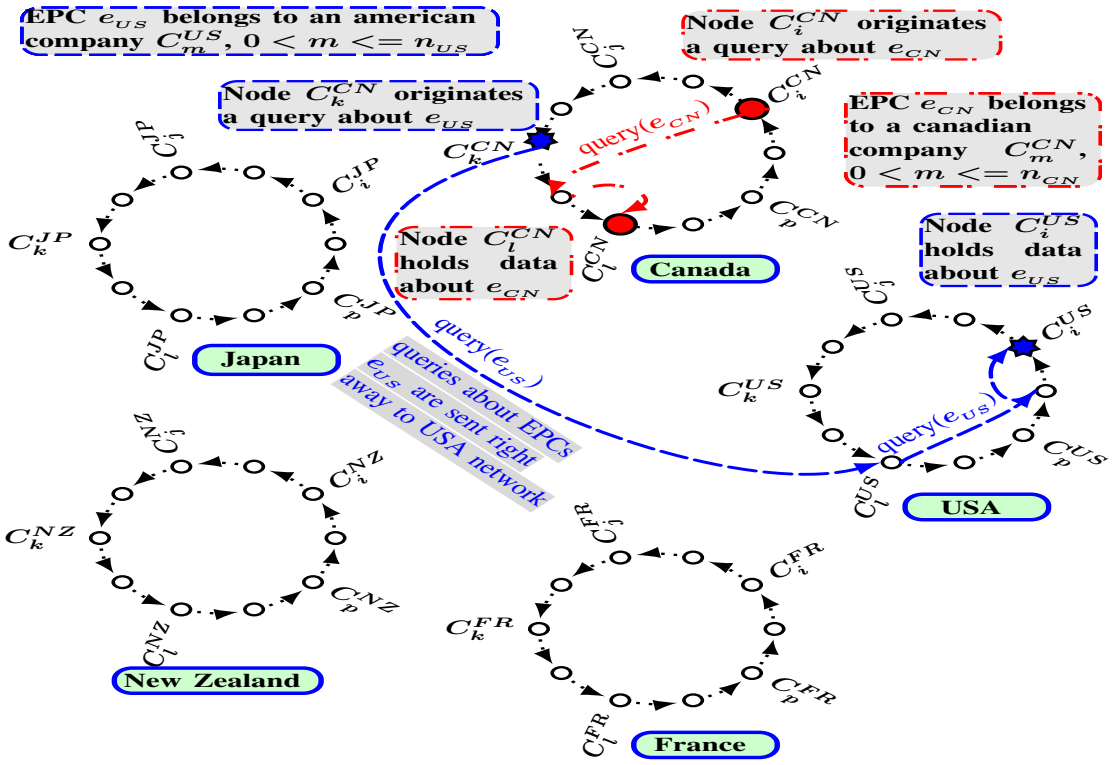


Figure 5.3 A 1-level ($K = 1$) HDA-based overlay network for EPCglobal Discovery Services illustrating two Lookup queries for EPC data retrieval. This figure illustrates an example of the hierchical architecture described in Figure 5.4 with number of levels $K = 1$ and number of regions per level $M = 5$

5.6.1 Tree Model

Figure 5.4 illustrates the HDA tree model corresponding to hierarchical naming shown in Figure 5.1. This model indicates that HDA comes with an overhead with regard to the number of nodes involved in the routing process among the N nodes belonging to the overlay networks composing HDA. Each and every data-storing node, that belongs to a given subscribing company, appears at $level\ K+1$ in the model, as a leaf node. All intermediate nodes in the model shown in Figure 5.4 are overhead nodes in HDA in comparison to FDA. In the remainder of the chapter, we call those intermediate nodes "HDA nodes", in order to distinguish them from the data storing nodes that makes the overlay networks. Moreover, whenever the word "node" is used with no specification, it should refer to the data storing nodes making the P2P overlays. Furthermore, unless otherwise stated, the expression "leaf node" and "node" are used interchangeably in HDA.

All HDA nodes, belonging to a given level $i, 1 \leq i \leq K$ subregion, point to the same parent HDA node at the immediate prior level $i-1$ (The root node at level 0 has no parent). Similarly, leaf nodes making a given P2P overlay network at level $K+1$, point to the same parent HDA node at level K, and hence are located geographically in one same $level-K$ subregion. As a result, since latency of a link depends on the physical distance separating its two ends, and since the diameters of the P2P overlay networks making HDA is much smaller than that of the FDA-based large P2P overlay network, the average latency of hops within any pair of the HDA-based overlay networks would certainly be much smaller than that of hops within the FDA-based overlay network. This claim will be proved analytically in Section 5.6.6.1.

5.6.2 Functionality of HDA

As mentioned in Section 5.6.1 above, only nodes (leaf nodes) belonging to level $K+1$ make the HDA overlay networks, as illustrated in Figure 5.4. On the one hand, each overlay network in HDA functions exactly as described in Section 5.5.1. The only differences being that the diameters of the HDA overlay networks are much smaller than that of the FDA overlay network both in terms of size (number of nodes involved) and diameter, since they physically cover a much narrower region; on the other hand, each

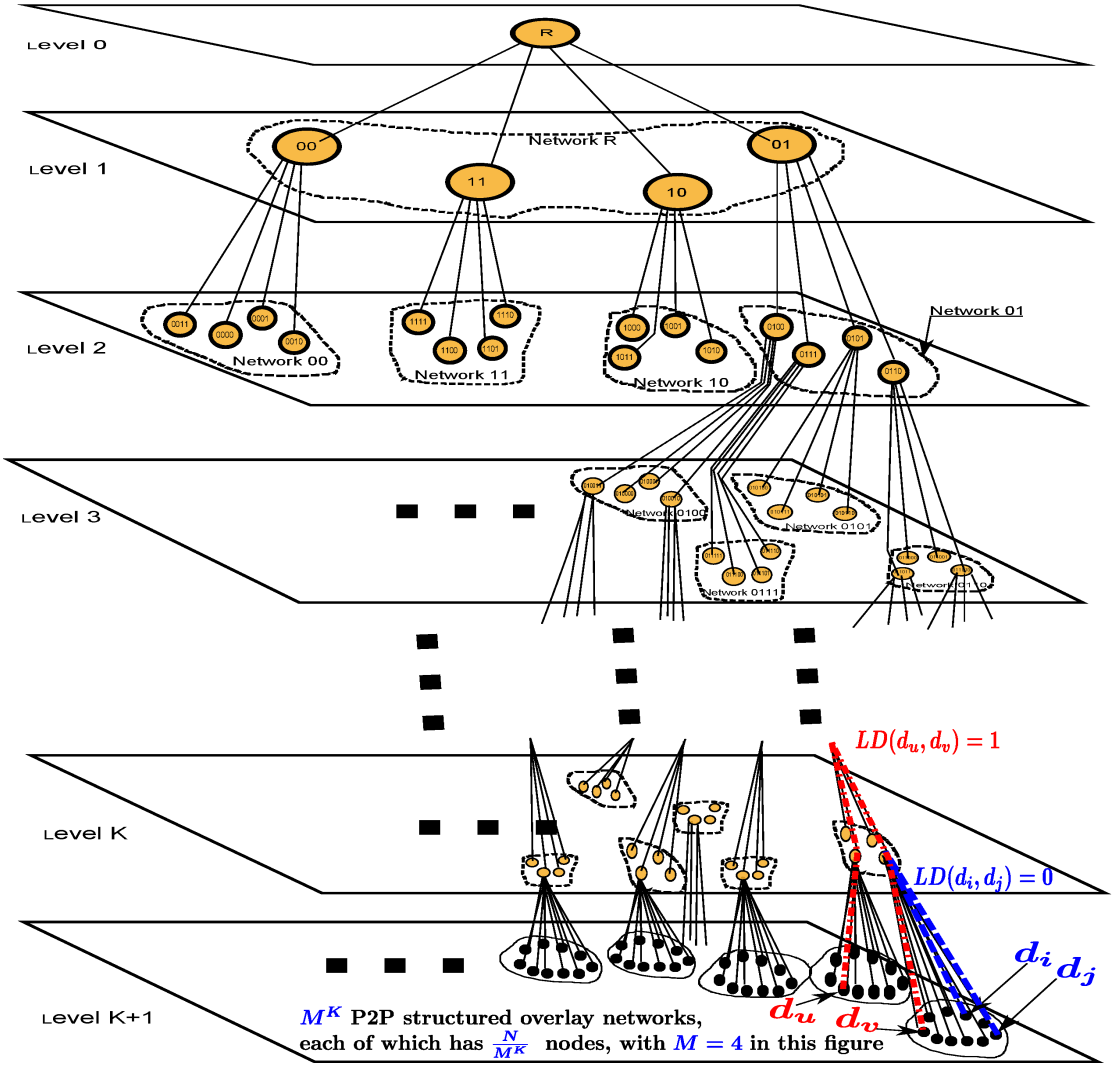


Figure 5.4 An overview of a Hierarchical Architecture of an Overlay Network with N nodes, K levels and $M = 4$ regions per level. Level K consists of M^K networks, each of which contains $\frac{N}{M^K}$ nodes. Figure 5.3 illustrates an example of this hierchical architecture with number of levels $K = 1$ and number of regions per level $M = 5$

node holds two types of routing information, namely intra-overlay routing information like it is the case in FDA, and inter-overlay routing information.

5.6.3 Inter-overlay vs Intra-overlay routing

While within a given HDA overlay, intra-overlay routing turns out to be the same as any conventional structured P2P protocol, although with much lighter routing tables and much more latency-effective links. The inter-overlay routing is performed based

on the inherent geographical information that can be retrieved from the key of the data being routed. This stipulates that all the (leaf) nodes have, in addition to the intra-overlay information, inter-overlay routing information, simply mapping each and every $level - K + 1$ subregion with an entry node (i.e., IP address and port). Such entry node is approached whenever a lookup query, either to publish data or to retrieve it, regarding a key (EPC) belonging to a different $level - K + 1$ subregion is received.

The receiving node computes the logical distance between the company prefix of the received key and its own identifier, as described in 5.4. This distance conveys the number of levels separating the node representing the company owning the key at hand, and the node receiving the lookup query inquiring about that key, from their closest common parent, as illustrated in Figure 5.4.

Intra-overlay routing information is used to route lookup queries, for which the inquired key and the receiving node belong to companies which are located in the same $level - K + 1$ subregion, and hence their nodes belong to the same P2P overlay network. Inter-overlay routing information is used to route lookup queries, for which the inquired key and the receiving node belong to companies which are located in two distinct $level - K + 1$ subregions, and hence their nodes belong to different P2P overlay networks. Any communication protocol for both Inter-overlay connectivity and Intra-overlay connectivity can be used, as long as it operates in the application layer so that it can retrieve and process the routing information from the inquired key at hand.

5.6.4 Entry Nodes

Each overlay network in HDA elects at least one of its nodes as its "entry node". An entry node plays the role of a gateway which allows a node, located in a given subregion, to route lookup queries whose the key belong to a company located in a different subregion. Intermediate $level - K$ nodes represent the underlying overlay networks at level $K + 1$.

Each $level - K$ node first elects a list of entry nodes for its underlying overlay network. Then it sends the list to its immediate parent node at level $K - 1$. Upon receipt of a list of entry nodes from its immediate children, an intermediate node routes it to its immediate parent. As a result, the root node ends up with all the entry nodes lists of all

the overlay networks at level $K+1$. This is called the collection process. It requires $\sum_{i=1}^{K} M^i = \frac{M^{K+1}-1}{M-1} - 1$ messages.

Upon receipt of a list of entry nodes, the root node routes it to its immediate children at level 1. Consequent to receiving an entry nodes list from its immediate parent, an intermediate node routes it to its immediate children. As a result, all the nodes involved in the overlay networks at level $K+1$, end up with all the entry nodes lists of all the overlay networks. This is called the dissemination process. It requires $\sum_{i=1}^{K} M^i = \frac{M^{K+1}-1}{M-1} - 1$ messages.

Therefore, neglecting the election process which can be straightforward, the entry nodes collection/dissemination requires $2.\sum_{i=1}^{K} M^i = 2.\frac{M^{K+1}-1}{M-1} - 2$ messages to be exchanged amongst the nodes.

5.6.5 Applicability to EPCglobal Discovery Services

HDA with $K = 1$ levels and $M = 256$ (approximate number of countries in the world), is applicable to the EPCglobal Network with absolutely no change to the structure of the EPC, and a little upgrade of the nodes with regards to inter-overlay routing capabilities and entry nodes election and exchange.

Each EPCglobal subscribing company can be associated with the country that has issued its company prefix [6]. Although a company is obviously not restricted to do business just within the borders of the country that has issued its company prefix, such association can be used to define the issuing country as the subregion where that company is located. Therefore, the node owned by a company belongs to the P2P overlay network of the country which has issued its company prefix. An example of two EPC lookup queries depicting two different scenarios is illustrated in Figure 5.3. In the first scenario, a lookup query regarding an EPC $e_{_{CN}}$ belonging to a Canadian company has been initiated within the Canadian network by node C_i^{CN}. The query is therefore processed until it reaches the node C_l^{CN} holding data about $e_{_{CN}}$. In average, the lookup query requires $\frac{log(n_{_{CN}})}{2}$ hops within the Canadian network, where $n_{_{CN}}$ denotes the number of nodes in the Canadian network.

In the second scenario, a lookup query regarding an EPC e_{US} belonging to an American company has been initiated within the Canadian network by node C_k^{CN}. The query is routed right away to the American network via the entry node C_l^{US}. It is then processed within the American network until it reaches the node C_i^{US} holding data about e_{US}. Therefore, in average, only $1 + \frac{log(n_{US})}{2}$ hops (1 inter-country hop and $\frac{log(n_{US})}{2}$) hops within the American network, where n_{CN} denotes the number of nodes in the American network. As opposed to the same scenarios presented in Figure 5.2, the nodes in other countries (New Zealand, Japan, France..) are exempted from participating in routing the lookup queries for the scenarios presented in Figure 5.3.

Obviously, this matching mechanism does not provide identification of the country of origin of a given physical product. Such identification is in fact not even possible, since Globalization has left no meaning to geographical diversity and companies can do business anywhere in the world regardless of their geographical location. Nevertheless, the issuing country identification will be used to route all the data related to the EPCs owned by a given company to the P2P overlay network of the country which has issued its company prefix. Likewise, any lookup request regarding data of a given EPC is automatically routed to the country which has issued the inherent company prefix. This information of the issuing country can be retrieved from the first three digits of the company prefix, which correspond in fact to the Global Standards One (GS1) prefix, or one of the GS1 prefixes, assigned by the GS1 organization to that country [124].

5.6.6 Merits and Limitations of HDA

HDA outperforms FDA both in terms of the lookup time and the lookup number of hops performance metrics, instead of focusing uniquely on the number of hops. This improvement is possible thanks to the use P2P overlay networks with much smaller diameters and much less nodes involved, as well as to the simple inter-overlay routing mechanism adding only one inter-overlay hop to the required number of hops within a P2P overlay network.

5.6.6.1 Scalability

The best P2P lookup protocols in the literature scale logarithmically, with the number of nodes, in their complexity with regards to the average number of hops required per lookup query, and to the size of the routing table each node has to handle. Chord algorithm [34] is one very well-known example of such protocols.

Hence with N number of nodes, if FDA performs a lookup in an average of $log(N)$ hops, and each node should hold an average of $log(N)$ entries in its routing table. In HDA with K levels and M children subregions per parent subregion, each node belonging to an overlay network would handle an intra-overlay routing table of $log(\frac{N}{M^K}) = log(N) - K.log(M)$ entries. Moreover, a lookup query within any of the HDA overlay networks is carried out in an average of $log(\frac{N}{M^K}) = log(N) - K.log(M)$. a lookup query amongst two distinct HDA overlay networks is carried out in an average of $1 + log(\frac{N}{M^K}) = 1 + log(N) - K.log(M)$; 1 is the inter-overlay hop routing a lookup query to the overlay network having the same identifier as its key. Tis hold because the number of nodes per overlay network in HDA is $\frac{N}{M^K}$, rather than N.

Obviously, HDA comes with the overhead of the inter-overlay routing table, whose the complexity of the number of entries is in the order K^M. This holds because each node needs at least one entry node to each and every overlay network.

Another overhead that HDA observes is the number of exchanged messages required in the collection process and the dissemination process of entry nodes, as detailed in Section 5.6.4. Neglecting the election process which can be straightforward, the entry nodes collection/dissemination processes require $2.\sum_{i=1}^{K} M^i = 2.\frac{M^{K+1}-1}{M-1} - 2$ messages to be exchanged amongst the nodes. This is an extensive overhead. However, since the collection and dissemination are not recurrent operations that can be carried out off line once in a while, this overhead is not relevant enough in our assessment.

5.6.6.2 Query Responsiveness

HDA improves FDA, in terms of both the average lookup time and the average number of hops required for a lookup query. The following three facts support the claimed improvement:

- First, the average number of hops required per lookup request decreases considerably in HDA, thanks to dividing the "flat" P2P overlay network into many much smaller HDA P2P overlay networks in terms of size, as shown in Figure 5.4. As mentioned above, the current EPCglobal Network can be upgraded by means of HDA into a set of linked overlay networks, each of which is associated to one and only one country; i.e., HDA EPCglobal Network with $K = 1$ levels, and $M = 190$ subregions per intermediate node, M being the number of countries. In such upgraded EPCglobal Network, illustrated in Figure 5.3, if the total number of the involved countries in the EPCglobal Network is between 2^7 and 2^8 countries (which is a reasonable assumption), then the average number of hops per lookup request is decreased by at least 6 hops; $6 = K.log(128) - 1$, as detailed above in Section 5.6.6.1.

- Second, the decrease in the size of the HDA P2P overlay networks will consequently lead to a decrease of the size of the intra-overlay routing tables, which will, in turn, have a positive impact on the average time required to perform a lookup.

- Third, the observed proximity in the geographical arrangement of the HDA P2P overlay networks will consequently lead to a significant decrease of their diameters, which will in turn have a great positive impact on the average time required to perform a lookup.

Although Globalization enables companies to do business worldwide, physical proximity of their premises makes business opportunities more likely to be taken. In reality, we can easily observe that companies tend to do business more locally than globally. As a result, in a real IoT-based business infrastructure, a given company would see much more lookup queries, regarding its own EPCs, initiated by neighboring companies. The closer two companies are (in terms of physical distance), the more probable an exchange of goods is. In other words, the closer two HDA overlay networks are, the more probable an exchange of lookup queries is. This would certainly improve, even better, the responsiveness of the lookup queries in HDA, since the latency of the inter-overlay routing hop in HDA would be, most probably, pretty much similar to the latency of intra-overlay routing hops.

In our performance assessment of HDA vs FDA, we assume that lookup queries, addressed to a given (leaf) node d_i, are equally likely to be initiated by any other (leaf) node d_j, no matter how close/far away they are to/from each other.

5.7 Horizontal vs Vertical Inter-overlay Routing

As mentioned in Section 5.6.6.1, HDA comes with an overhead with regards to the additional inter-overlay routing table each leaf node has to deal with. This table is of exponential complexity with respect to the number of levels K, which represents a major limitation of HDA.

To address this issue, we suggest a vertical inter-overlay routing, rather than a horizontal one, as described in Section 5.6.3. Vertical inter-overlay routing requires that each node in HDA knows just its immediate parent node, rather than a large table of M^K entry nodes in horizontal inter-overlay routing. When a leaf node, in a given overlay network O_i, receives a lookup request, it first checks whether the inquired key belongs to a company in the same overlay networks O_i. Then, in the case the inquired key is associated with the same overlay O_i, the receiving node routes the query locally, within O_i, following the adopted structured P2P protocol. On the contrary, if the inquired key is associated with a distinct overlay network $O_j \neq O_i$, the receiving node routes the query to its immediate parent node.

Once an intermediate node receives a lookup query, it first checks whether it is a common parent of the node representing that key, and the initiating leaf node. Then, in the case this is true, it routes the query down the tree to its immediate child node having the smallest logical distance with the key; i.e., the closest in terms of identifiers. On the contrary, if the receiving intermediate node is not a common parent of the node representing the key at hand, and the initiating leaf node, the receiving intermediate node routes the query up the tree to its immediate parent node.

Vertical inter-overlay routing cancels the overhead of the inter-overlay table. However the number of hops per lookup query is larger than that of inter-overlay routing. In vertical inter-overlay routing, the worst number of hops required per lookup query is

rather $log(N) - K.log(M) + 2.K = log(N) - K.(log(M) - 2)$, $2.K$ pertaining to the hops up and down the tree.

It is worth mentioning that vertical inter-overlay routing would be interesting only in the case M is large enough to make the term $K - log(M) - 2$ significant enough in the expression. With $M = 4$ for example, we can see that the number of hops in HDA with vertical inter-overlay routing is $log(N)$, same as with FDA. Therefore M has to be at least 5 for vertical inter-overlay routing to be worth considering at all. Obviously, the larger M, the better.

5.8 Conclusion

In this chapter, we have described in detail a Hierarchical Distributed Architecture, called HDA, and a Flat Distributed Architecture, called FDA, for overlay networks offering Discovery Services in the future IoT-based business infrastructure. The EPCglobal Network is an example of such IoT-based business infrastructure. The main objective of this proposal is to overcome the worrying drawbacks that the existing EPCglobal ONS-based lookup system. We have looked into enhancing both the network scalability and the responsiveness of the lookup queries. Such two-fold improvement is of crucial importance given its impact on the survivability of large-scale applications in the future IoT, such as the EPCglobal network.

In the next chapter, we present a simulation of the two architectures (FDA and HDA) as a platform for EPCglobal Discovery Services, via an implementation on Planetsim. We also evaluate performance of the two architectures both analytically and experimentally.

Chapter 6

Performance Evaluation of the HDA Architecture

6.1 Introduction

In this chapter, we present an assessment of the two architectures (HDA and FDA) described in the previous chapter. Both architectures are designed for distributed and scalable Discovery Services. FDA architecture speaks for the majority of the distributed architectures that have been proposed in the literature so far, and HDA architecture represents the core contribution of our work in the previous chapter. FDA has been introduced solely for performance assessment reasons. It plays the role of a reference against which HDA is compared. The comparison aims at proving that HDA is more scalable and its lookup responsiveness is much better.

The claimed improvement is first proved theoretically in Section 6.3, then confirmed through a simulation of HDA and FDA on Planetsim [125]. Chord algorithm [34], which is one of the well-known distributed algorithms in which the number of hops per lookup query scales logarithmically with the number of nodes, has been adopted in the simulation.

To assess our proposed discovery services architecture HDA, we provide two major sections, the first one relates to theoretical performance analysis and performance results

of the proposed hierarchically-linked overlay networks model. Whereas the second one provides the performance results derived out of simulation on Planetsim [125].

In our performance assessment of HDA and FDA, we assume that lookup queries, addressed to a given (leaf) node d_i, are equally likely to be initiated by any other (leaf) node d_j, no matter how close/far they are to/from each other.

For all the results presented in this chapter, we assume an average maximal latency lat_K of $100ut$ (ut is defined as one unit of time as explained in the List of Symbols and Notations) characterizing the most costly overlay links, joining (leaf) nodes whose the closest common parent is the root. We also assume that the average latency lat_i of an overlay link, joining (leaf) nodes whose the closest common parent is at level i, is 5% less than the average latency lat_{i+1} of an overlay link, joining (leaf) nodes whose the closest common parent is at level $i+1$. Hence $lat_K = 100ut, lat_i = (1-5\%).lat_{i+1}, 0 \leq i < K$.

Since the objective of this work is to evaluate and compare the performance of the hierarchical HDA architecture and that of the flat FDA architecture, rather than to asses the various algorithms that may be selected for the implementation, we believe that the conclusions drawn from the analysis and the simulation would still hold, had any other distributed algorithm such as Pastry or Koorde, been used instead of Chord [34]. The robust implementation of Chord offered in Planetsim [125], is the major reason behind our choice of Chord.

The remainder of this chapter is structured as follows. Section 6.2 presents the metrics used to evaluate the performance of the HDA Architecture presented in Chapter 5. In Section 6.3, we present the theoretical model of the proposed HDA Architecture and its performance analysis. In Section 6.4, we present a simulation of the proposed HDA architecture on Planetsim and the corresponding results. In Section 6.5, we present the emulation of HDA architecture on Planetlab and the corresponding results. Finally, Section 6.6 concludes this chapter.

6.2 Considered Performance Metrics

The assessment of the two architectures, namely HDA and FDA, is based on the two following performance metrics:

- The average number of hops required for a lookup query to reach its destination. Analysis of this performance metric aims at showing how lookup queries require less number of hops in HDA architecture than in FDA architecture.
- The average time required for a lookup query to reach its destination. Analysis of this performance metric aims at showing how lookup queries require less time in HDA architecture than in FDA architecture.

6.3 Performance Analysis

6.3.1 Theoretical Model

First, we compute the probability that two randomly selected (leaf) nodes, illustrated in Fig 5.4, have the closest common parent at exactly level $l, 0 \leq l \leq K$. Lemma 6.2 derives the equations representing this probability. These equations make use of Lemma 6.1 which derive the probability that two randomly selected (leaf) nodes have the closest common parent at "at least level l", $0 \leq l \leq K$ (i.e., they have the closest common parent at level $l, l+1, ..., K-1$ or K).

Lemma 6.1. *For notations and definitions refer to Section 5.4 in the previous chapter and to List of Symbols and Notations at the beginning of this thesis. Let d_i and d_j, two randomly selected nodes in the set S. The probability that d_i and d_j have the closest common parent at "at least level l", $0 \leq l \leq K$ (i.e., d_i and d_j have the closest common parent at level $l, l+1, ..., K-1$ or K) is:*

$$P(LD(d_i, d_j) \geq l) = \frac{\frac{N}{M^l} - 1}{N - 1} \tag{6.1}$$

Proof. First note that there are M^l different subregions at level l, each of which is represented by an intermediate node as illustrated in Figure 5.4. Moreover, each $level-l$ intermediate node is a common parent to $\frac{N}{M^l}$ leaf nodes. Let us denote S_l the set of those $\frac{N}{M^l}$ leaf nodes, having the closest common parent at level l.

On the one hand, there are $\binom{N}{2}$ ways of random selection of two leaf nodes out of the N nodes, all equally likely; on the other hand, for each and every $level-l$ intermediate node, there are $\binom{\frac{N}{M^l}}{2}$ ways of random selection of two leaf nodes out of S_l, all equally likely. Since there are M^l distinct intermediate nodes at level l, then the number of ways of random selection of two leaf nodes having the same closest common parent at "at least level l" is $M^l.\binom{\frac{N}{M^l}}{2}$, all equally likely.

As a result, the probability that two randomly selected leaf nodes have the closest common parent at "at least level l", $0 \leq l \leq K$ (i.e., they have the closest common parent at level $l, l+1, ..., K-1$ or K) is:

$$
\begin{aligned}
P(LD(d_i,d_j) \geq l) &= \frac{M^l.\binom{\frac{N}{M^l}}{2}}{\binom{N}{2}} \\
&= \frac{M^l.\frac{(\frac{N}{M^l})!}{2.(\frac{N}{M^i}-2)!}}{\frac{N!}{2*(N-2)!}} \\
&= \frac{M^l.\frac{N}{M^l}.(\frac{N}{M^i}-1)}{N.(N-1)} \\
&= \frac{\frac{N}{M^l}-1}{N-1}
\end{aligned}
$$

□

Lemma 6.2. *For notations definitions refer to Section 5.4 in the previous chapter and to List of Symbols and Notations at the beginning of this thesis. Let d_i and d_j, two randomly selected nodes in the set S. The probability that d_i and d_j have the closest common parent at exactly level $l, 0 \leq l \leq K$ is:*

$$P(LD(d_i, d_j) = l)_{0 \leq l < K} = \frac{N.(M-1)}{(N-1).M^{l+1}} \tag{6.2}$$

$$P(LD(d_i, d_j) = K) = \frac{(\frac{N}{M^K} - 1)}{N-1} \tag{6.3}$$

Proof. For the case $l = K$. Given that the logical distance separating the identifiers of any two nodes has the upper bound of K, we can derive Equation 6.3, using lemma 6.1, as follows:

$$\begin{aligned} P(LD(d_i, d_j) = K) &= P(LD(d_i, d_j) \geq K) \\ &= \frac{(\frac{N}{M^K} - 1)}{N-1} \end{aligned}$$

For the case $l \neq K$. The probability that two randomly selected leaf nodes have the closest common parent at exactly level $l, 0 \leq l \leq K$ is the probability that they have the closest common parent at level $l, l+1, ..., K-1$ or K (lemma 6.1 applied to level l), minus the probability that they have the closest common parent at level $l+1, ..., K-1$ or K (lemma 6.1 applied to level $l+1$). We derive Equation 6.2 as follows:

$$\begin{aligned} P(LD(d_i,d_j)=l) &= P(LD(d_i,d_j)\geq l) - P(LD(d_i,d_j)\geq l+1) \\ &= \frac{\frac{N}{M^l}-1}{N-1} - \frac{\frac{N}{M^{l+1}}-1}{N-1} \\ &= \frac{\frac{N}{M^l} - \frac{N}{M^{l+1}}}{N-1} \\ &= \frac{N.(M-1)}{(N-1).M^{l+1}} \end{aligned}$$

The probability property $\sum_{l=0}^{K} P(D(d_i,d_j)=l)=1$ has been verified.

□

Lemma 6.2 can then be used to compute the expected value of the random variable L, denoting the latency of links joining a randomly selected pair of nodes amongst the (leaf) nodes, as illustrated in Figure 5.4. We denote this expected value $E[L]$. Equation 6.4 illustrates the formula to derive $E[L]$. For notations definitions refer to Section 5.4 in the previous chapter and to List of Symbols and Notations at the beginning of this thesis.

$$E[L] = \sum_{l=0}^{K} (P(D(d_i,d_j)=l).lat_l) \tag{6.4}$$

Let us denote $time_{FDA}$ and $time_{HDA}$ the time required for a lookup query to be answered in a FDA-based Discovery Services overlay network, and in a HDA-based Discovery Services overlay network respectively. The two metrics are derived in Equation 6.5 and Equation 6.6, using Equation 6.4.

$$\begin{aligned} time_{FDA} &= log(N).E[L] \\ &= log(N).\sum_{l=0}^{K}(P(D(d_i,d_j)=l).lat_l) \end{aligned} \tag{6.5}$$

$$\begin{aligned} time_{HDA} &= (log(N) - K.log(M)) + E[L] \\ &= (log(N) - K.log(M)).lat_0 + \sum_{l=0}^{K}(P(D(d_i, d_j) = l).lat_l) \end{aligned} \tag{6.6}$$

6.3.2 Theoretical Results

Given two randomly selected (leaf) nodes d_i and d_j out of the set S of N nodes, Figure 6.1 plots the probability that the logical distance separating the identifiers of d_i and d_j, denoted $LD(d_i, d_j)$, is equal to a certain value $l, 0 \leq l \leq K$. It compares plots of the function $y = P(LD(d_i, d_j) = x)$, for different values of K (Number of levels) and M (Number of children subregions at $level\ i$ having the same parent subregion at $level\ i-1$). It is worth mentioning that $P(LD(d_i, d_j) = x)$ implies that the two randomly selected nodes d_i and d_j have the closest common parent at a level $l = K - x$.

Figure 6.1, shows that the probability $P(LD(d_i, d_j) = x)$ exponentially decreases with x. In other words, given the randomness brought by DHT in structured P2P overlay networks setup, it is much more probable that physically distant nodes would become neighbors in the underlying overlay network. As a result, the hops in FDA are much more costly in terms of latency. Therefore, although the number of hops required per lookup query scales logarithmically with the number of nodes in FDA, query responsiveness, in terms of time required per lookup query, is to be improved.

Figure 6.2 illustrates the claim above. It shows that the expected latency of a link, whose ends are randomly selected, is rather high close to the maximum latency $lat_K = 100$, characterizing the most costly links in the network; i.e., the links with the maximum logical distance K. This holds no matter what the values of K and M are. For example, it can be seen in Figure 6.2 that with $K = 10$ and $M = 3$, the expected latency of a randomly selected link is greater than 97.5, in the presence of links with latencies in the set $100, 95, 90, 85, 80, 75, 70, 65, 60, 55, 50$ ($lat_K = 100, lat_i = (1 - 5\%).lat_{i+1}$).

Figure 6.3 shows the time required per lookup query, formulated in Equation 6.5 for FDA and in Equation 6.6 for HDA, in terms of the number of levels K, for different values of the number of children subregions per parent region M. The graph shows that the higher the number of levels K is, the lower the time required per lookup query

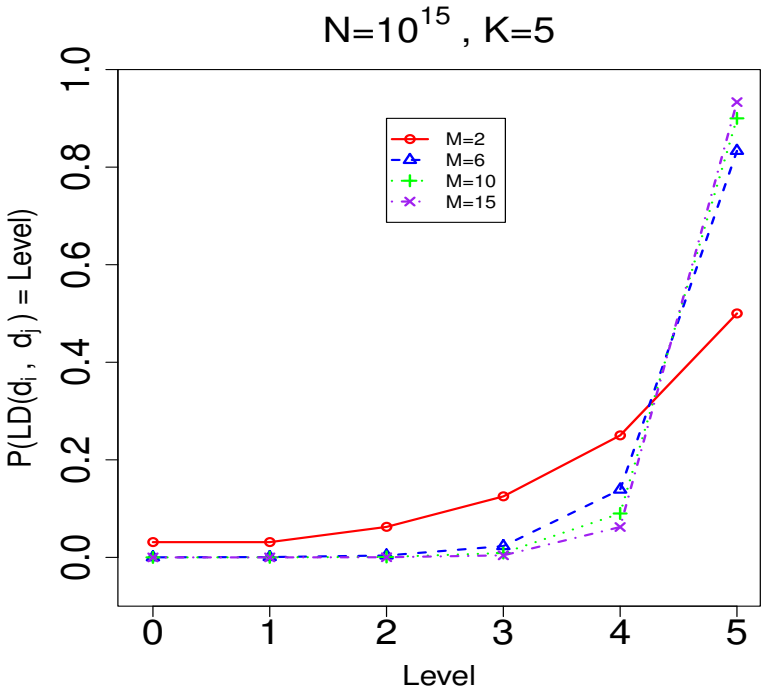


Figure 6.1 Plot of the likelihood that two randomly selected (leaf) nodes have the closest common parent at a certain level (refer to lemma 6.2)

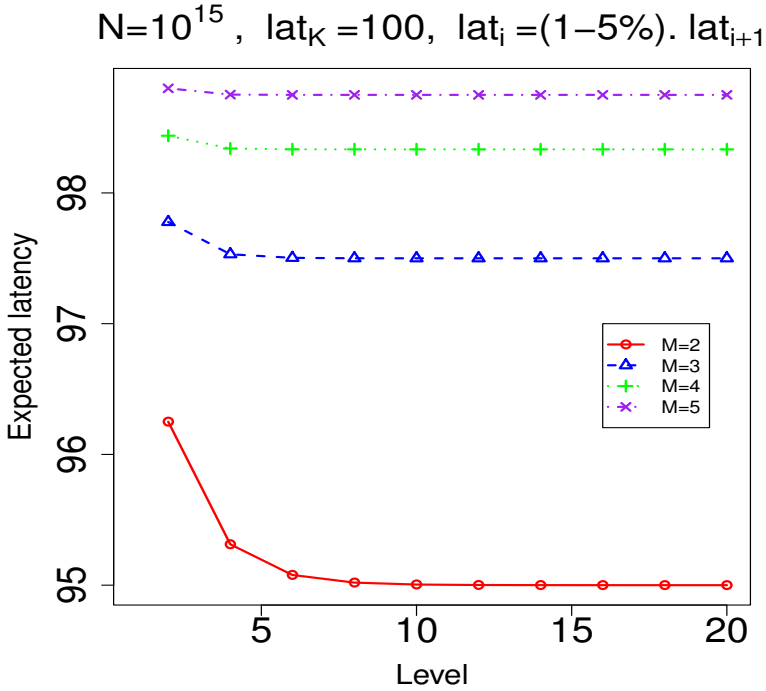


Figure 6.2 Plot of the expected latency of a randomly selected link whose ends are (leaf) nodes (refer to Equation 6.4)

gets. The observed trend in query responsiveness is better than linear. As an example, with $K = 12$ and $M = 2$, query responsiveness decreases from about 4800 (FDA) to about 2700 (HDA). The improvement is of over 43%. With $K = 12$ and $M = 10$, the improvement is over 83%.

Figure 6.4 shows the time required per lookup query, formulated in Equation 6.5 for FDA and in Equation 6.6 for HDA, in terms of M; the number of children subregions per parent region, , for different values of the number of levels K. The graph shows that the higher the number of children subregions per parent region M is, the lower the time required per lookup query gets. The observed trend in query responsiveness is logarithmic. As an example, with $K = 4$ and $M = 10$, query responsiveness decreases from about 4800 (FDA) to less than 3200 (HDA). The improvement is of over 33%. With $K = 6$ and $M = 10$, the improvement is over 56%.

From results shown in Figure 6.3 and Figure 6.4, we can claim that it is better to design HDA with maximum number possible of levels K, rather than splitting parent regions into larger number of children subregions; i.e., larger M. This is because the effect of increasing K is much more important than that of increasing M. This confirms our claim that HDA remarkably improves query responsiveness, compared to FDA.

Table 6.1 provides a recapitulation of the differences between HDA and FDA. As seen in the table, HDA performs better in terms of scalability with regard to the required number of hops per lookup query or node join/leave. The larger K and M are, the better the performance is. Nevertheless, HDA has a significant overhead related to entry nodes exchange and inter-overlay routing. The smaller K and M are, the better the performance is.

Since the collection and dissemination operations, carried out during entry nodes exchange, are not recurrent operations that can be carried out off line once in a while, entry nodes exchange overhead is acceptable. However, inter-overlay routing overhead is to be taken into account, because it affect each and very lookup query. In this case, we are in front of contradicting objectives, for which optimal values of K and M, observing a trade-off amongst the different metrics, have to be identified.

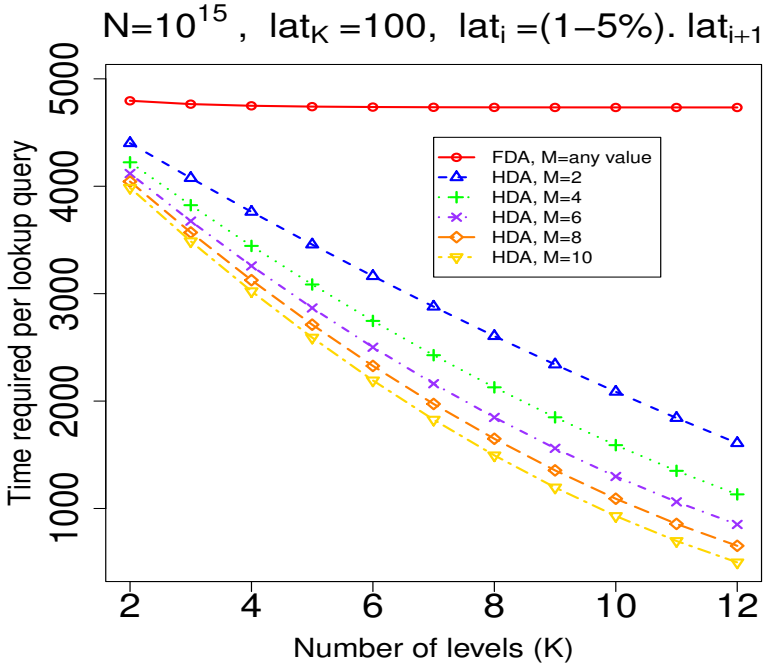


Figure 6.3 Theoretical performance results of HDA vs FDA. Time required per lookup query for FDA and HDA, in terms of the number of levels K, for different values of the number of children subregions per parent region M

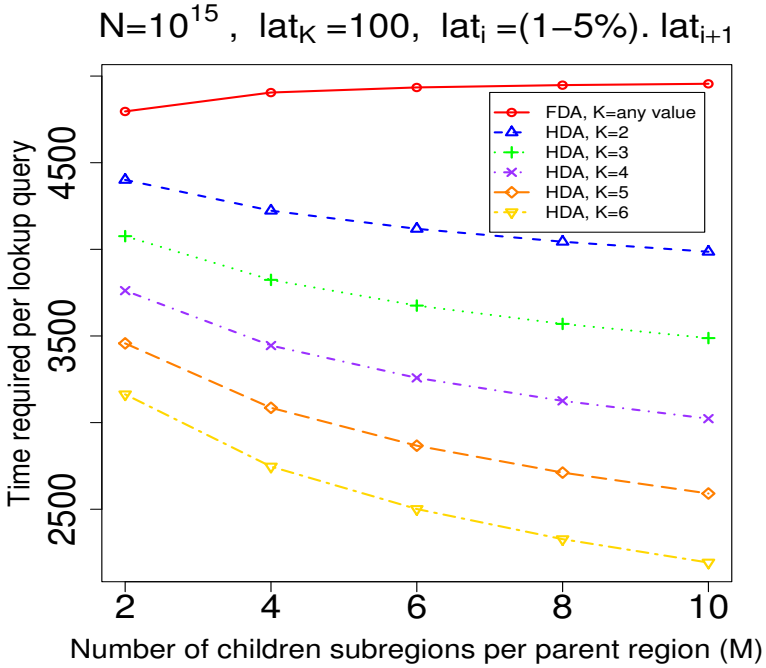


Figure 6.4 Theoretical performance results of HDA vs FDA. Time required per lookup query for FDA and HDA, in terms of the number of children subregions per parent region M, for different values of the number of levels K.

Table 6.1 Comparison of HDA vs FDA

Algorithm	Chord-based FDA	Chord-based HDA
Required number of hops per lookup query	$log(N)$	$1 + log(N) - K.log(M)$
Required number of hops per node Join/Leave	$log(N)$	$log(N) - K.log(M)$
Inter-overlay routing table size	0	M^K
Required number of message for entry nodes exchange	0	$2.\frac{M^{K+1}-1}{M-1} - 2$

6.4 Simulation of HDA on Plametsim

6.4.1 Simulation Setup

In our simulation, we have used the algorithm Chord [34] under the well-known simulation/experimentation framework for large scale overlay services; Planetsim [125]. Since the objective of this work is to evaluate and compare the performance of the hierarchical HDA architecture and that of the flat FDA architecture, rather than to asses the various algorithms that may be selected for the implementation, we believe that the conclusions drawn from the simulation would still hold, had any other distributed algorithm such as Pastry and Koorde, been used instead of Chord [34]. The robust implementation of Chord offered in Planetsim [125], is the major reason behind our choice of Chord, besides its widely recognized good reputation and its appealing simplicity.

We have used Chord original implementation in Planetsim as a representative for FDA. For HDA, we have undertaken a major upgrade of Planetsim framework. As a result, we have been able to parametrize Planetsim in such a way that it can be run either in a FDA mode or in a HDA mode. The parametrization has enabled us to specify the number of (leaf) nodes involved N, the number of levels K, the number of children subregions per parent region M, as parameters to be provided to the simulator.

We have also updated Planetsim framework to make it capable of tracking each and every hop, along with the distance separating its ends, during a lookup query. Such update has enabled us to evaluate both the average number of hops and the average simulated time, required per lookup query. By “simulated time”, we mean time computed based on the number of hops, which takes into account the cost of each hop with regards to the distance separating its two ends. As mentioned in Section 5.4 in the previous chapter,

this distance refers to the number of levels separating the two (leaf) nodes; ends of the hop at hand; from their closest common parent in a tree-like hierarchical structure of the overlay network, as illustrated in Figure 5.4. Its formula is derived in Section 5.4 in the previous chapter (refer to definition 5.1 in the previous chapter).

We conducted a series of experiments, each of which is characterized by the number of (leaf) nodes involved N, the number of levels K, the number of children subregions per parent region M. Each experiment consists of processing 10000 lookup queries, for the same tuple of parameters (N, K, M), in both FDA mode and HDA mode, $N \in \{5000, 7500, 10000, 12500, 15000, 17500, 20000\}$, $K \in \{2, 3, 4, 5, 6, 7\}$, $M \in \{2, 3, 4, 5, 6, 7\}$.

For each experiment, The average of the considered performance metrics are computed over the 10000 lookup queries. The initiating (leaf) nodes of the lookup queries are randomly selected, all the nodes being equally likely. Moreover, the inquired keys of the lookup queries are randomly built.

Given that running Planetsim with large number of nodes is extremely demanding in terms of memory and computing power, we opted to run the experiments on a cluster of $X86$ based multicore machines running Linux, called the SW (Linux) Cluster [126], belonging to High Performance Computing Virtual Laboratory (HPCVL) [127]. The nodes, used in simulation, are Dell PowerEdge $R410$ Servers with a 6-core Intel Xeon processor (Intel $x5670$ / $x5675$) running at 3.07 GHz, and with 64 Gbyte of physical memory [126].

6.4.2 Simulation Results

Figure 6.5 shows the average number of hops required per lookup query, for both FDA and HDA, in terms of the number of nodes N, for different values of the number of levels K and the number of children subregions per parent region M. Figure 6.6 illustrates the time required per lookup query for both FDA and HDA, in terms of the number of nodes N, for different values of the number of levels K and the number of children subregions per parent region M. The two graphs show that the average number of hops per lookup query scales logarithmically for both FDA and HDA. However, HDA is more

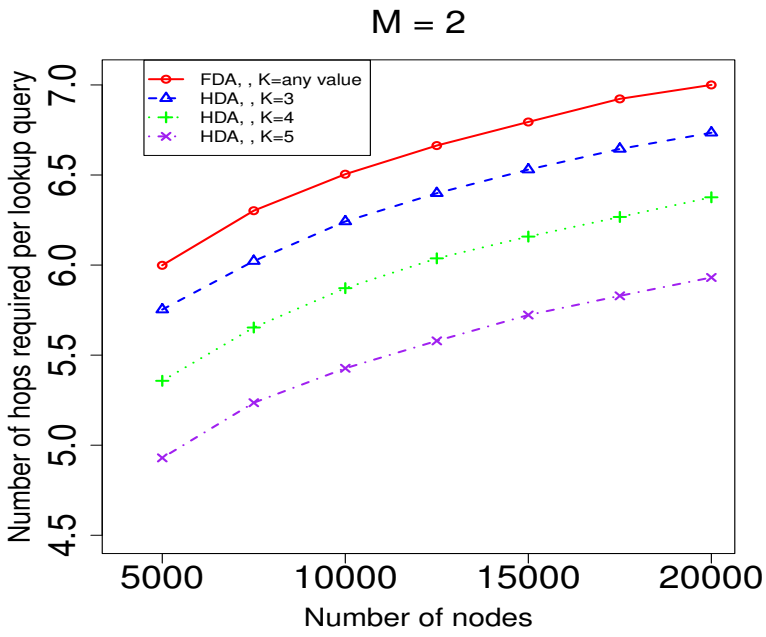


Figure 6.5 Performance simulation results of HDA vs FDA implemented on Planetsim using Chord. Number of hops required per lookup query for FDA and HDA, in terms of the number of nodes N, for different values of the number of levels K and the number of children subregions per parent region M

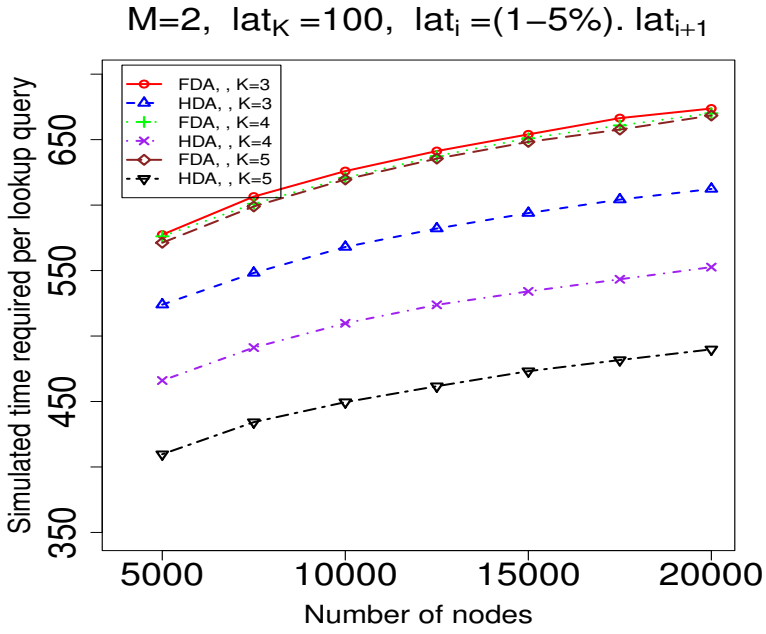


Figure 6.6 Performance simulation results of HDA vs FDA implemented on Planetsim using Chord. Time required per lookup query for FDA and HDA, in terms of the number of nodes N, for different values of the number of levels K and the number of children subregions per parent region M.

advantageous in the sense that, the larger K is, the better the number of hops and the time required per lookup query.

In Figure 6.6, we can also see that the gap separating the HDA plot lines increases with the number of nodes N. For example the gap separating the two plot lines $HDA, K = 3$ and $HDA, K = 4$, is larger at the point $N = 20000$, than it is at the point $N = 5000$. The increase of the gap separating the plot lines, with the number of nodes N, implies a larger decrease in the lookup time when N gets larger. This confirms our claim that HDA is more scalable than FDA.

6.5 Emulation of HDA on Planetlab

6.5.1 Emulation Setup

An implementation of the two architectures HDA and FDA, presented in Section 5.6 and Section 5.5 respectively in the previous chapter, has been performed on Planetlab using the well known algorithm Chord [34]. Planetlab is a global research open platform consisting of more than 1000 nodes scattered all around the world.

6.5.1.1 Constraints

Although Planetlab consists of more than 1000 nodes, the following constraints have forced us to stop at the number of 64 as the maximum number of nodes that can be used for the experiments:

- First, Planetlab platform consists of two different networks. Planetlab Europe comprising the majority of the nodes located in Europe and Planetlab Central comprising nodes from the rest of the world. The choice of Planetlab Central has shrunk the number of available nodes to less than 500.
- Second, no guaranty regarding the stability of the nodes on Planetlab is given. Any node can go down or update its public key anytime without prior notice, which makes the number of useful nodes decrease to about 350.

- Third, restricted access has been observed on some nodes. These nodes have to be eliminated through a tedious and time-consuming process as the experiments go along.

Although the number 64 may seem insignificant compared to the expected scale of P2P networks in the future IoT, the trend of the measured performance metrics can still be captured with no ambiguity.

6.5.1.2 Emulation Steps and Parameters

Since HDA architecture aims at improving FDA architecture, the two architectures have to be tested with the same nodes sets. In FDA architecture implementation, all the nodes belong to the same P2P network, while in HDA architecture implementation, a P2P network is created per geographical location (per country in the case of the EPCglobal business infrastructure). The average number of hops and the average lookup time is measured in the two architectures to be compared.

For simplicity, the experiments comparing FDA to HDA have been run on the nodes belonging to two countries, namely Canada and Japan. Nodes are selected so that half of the nodes set belong to each country. The choice of these two countries was driven by the stability of their nodes and also because of the significant distance separating them. The number of hops and lookup time have been recorded for 100 lookup requests, then the average is plotted along with a 95% confidence interval, assuming the measured metrics are normally distributed. Because the maximum number of nodes available on the two countries is only 12 each, we selected more available nodes from neighboring countries. Particularly, we assumed some nodes located in north of USA as being Canadian nodes, and some nodes located in north east China as being Japanese nodes. HDA experiments have been run with 8, 12, 16, 20, 24, 28 and 32 nodes to evaluate the trend of the two performance metrics for the two architecture.

6.5.1.3 Emulation Tools

“g++” and “make” programs have been installed on each node in order to compile the C++ client/server code implementing the peers. A Postgres database has also been

installed on each node, in order to simulate storage and retrieval of the $(key, value)$ pairs.

The algorithm Chord [34] has been adopted in the emulation of the two architectures as an implementation algorithm. Since the objective of this work is to show that HDA architecture performs better than flat P2P architectures, focus has been put on comparing the two architectures rather than comparing the algorithms that may be used to implement those architectures. We believe that the conclusions drawn from the emulation would still hold, had any other distributed algorithm, such as Pastry or Koorde, been used in the implementation as an alternative to Chord [34]. We chose Chord [34] because it is the simplest to implement.

6.5.2 Emulation Results

In this section, an assessment of HDA and FDA architectures, presented in Section 5.6 and Section 5.5 respectively in the previous chapter, is presented. This assessment aims at analyzing and comparing the scalability of the two architectures.

Figure 6.7 illustrates the average number of hops per lookup for each nodes set for the two architectures. It shows that the number of hops is almost the same for the two architectures. This is because the experiments have been run with only 2 countries. As claimed above, the number of hops in HDA architecture can be decreased by $m - 1$ if 2^m countries have been used in a large HDA-based system.

In terms of the lookup time, Figure 6.8 illustrating the average lookup time per request in the two architectures, shows that HDA architecture lookup time is almost half FDA architecture lookup time. Obviously, this would be even better if more than two countries have been used in the experiments.

6.6 Conclusion

In this chapter, we have provided both theoretical and experimental assessment of the HDA Architecture presented presented in Chapter 5. First, we have provided a detailed theoretical performance analysis of the proposed architecture. Then, we have assessed

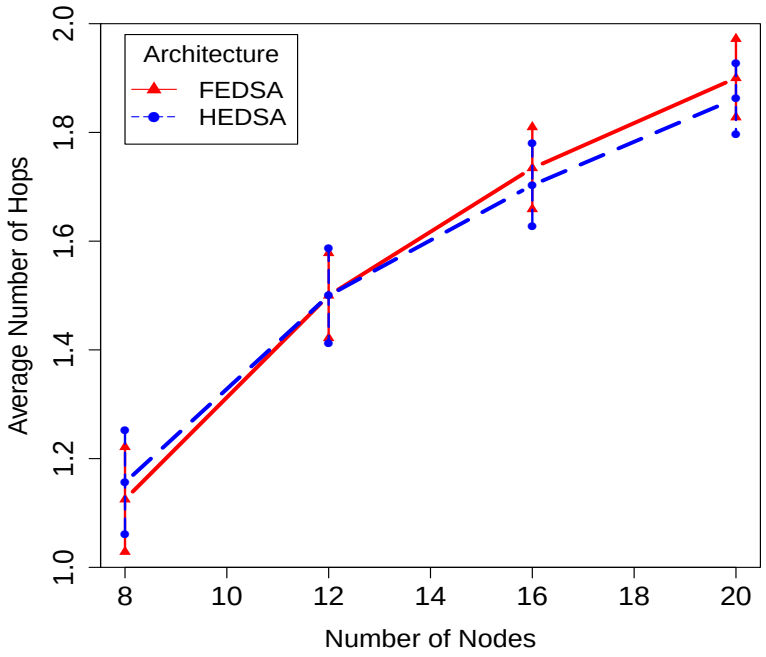


Figure 6.7 Comparison of the average number of hops, required per lookup query in FDA/HDA architectures, implemented on Planetlab using Chord

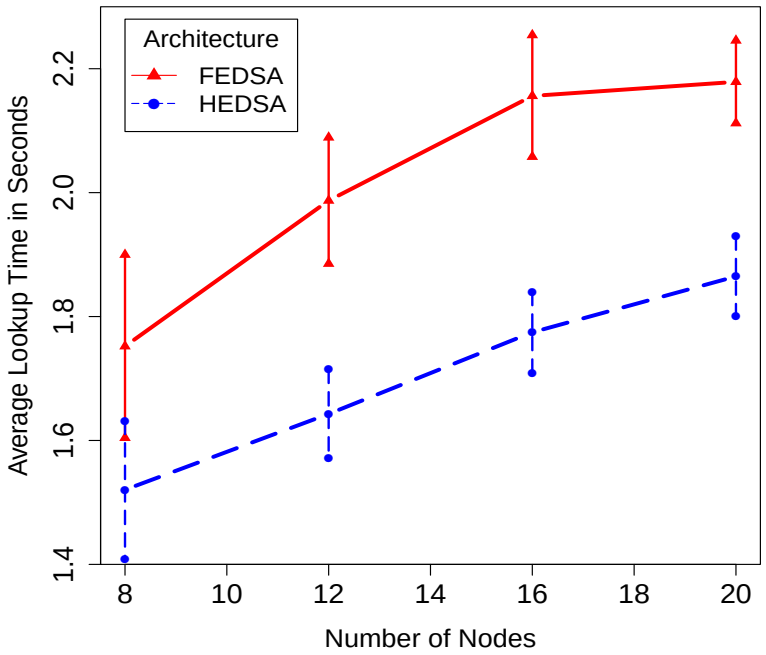


Figure 6.8 Comparison of the average lookup time, required per lookup query in FDA/HDA architectures, implemented on Planetlab using Chord

the proposed architecture experimentally both via a simulation on Planetsim and an emulation on Planetlab. Both theoretical and experimental results have proved the claim

brought forward in the beginning of this chapter, stating that the HDA Architecture presented in Chapter 5 is more scalable and more efficient than flat (non-hierarchical) distributed architectures. Efficiency has been measured in terms of lookup time and number of hops required per lookup query.

Chapter 7

Threshold-based Distributed Discovery Services for EPCGlobal Network

7.1 Introduction

With the advent of RFID communications [21–23], identification and tracking of individual objects have become viable in several industrial applications such as manufacturing, logistics, ticketing, and anti-counterfeiting. The communication network for the corresponding purposes can be implemented based on the Internet of Things (IoT) concept which was initially introduced in the Electronic Product Code (EPC) Global standards one and a half decade ago [6], [3, 24, 25]. IoT aims at broadening the existing Internet of computers paradigm by defining a large-scale network of objects where the EPCglobal associates each object with an RFID tag so that tracing an object throughout the supply chain is possible. The RFID tag of an object is read once the object is relocated throughout the supply chain and/or the information about corresponding object is altered. Once it is read, the RFID provides the EPC [26], [27], [28] which is a unique number associated with the corresponding object.

EPCglobal aims at a worldwide standard for RFID, and the EPCglobal network architecture consists of three major components: i) The client application, ii) Object Naming System (ONS) root [7, 128, 129], iii) EPC Information Services (EPCIS) [30, 130]. The client application acts as middleware and receives the Electronic Product Code (EPC)

from the RFID reader. Upon converting the EPC to a Uniform Resource Identifier (URI) which is recognizable by the ONS, the client application transfers the EPC to the ONS. The ONS searches the EPC in given repositories in order to localize its corresponding information sources. The information sources are usually retrieved as Uniform Resource Locators (URL) that can be fetched by the client application. The EPCIS data server represents the main information source for EPC information. It provides information about the corresponding EPC, and it is accessed by the client application through the URL previously retrieved by the ONS.

The solution above is viable for a single company but in the concept of EPCglobal network [131, 132], the object has to be traced across the supply chain amongst different organizations. Hence, EPC discovery service emerges as a core component in data retrieval for a specific object. The standardization activities for EPC discovery services are still ongoing by the EPCglobal Data Discovery Working Group (DS WG) while several studies have recently addressed the design of discovery services [11, 20, 36, 52–55, 61]. Most of the existing research studies have presented the Discovery Services (DS) for the EPCglobal network as distributed systems serving the following fundamental lookup function: Given an EPC identifies a real-world object, the DS return a list of Internet addresses of EPCIS that offer all the information about the object across the entire supply chain. Furthermore, most of these approaches propose discovery services based on P2P technology relying on a Distributed Hash Table (DHT) [34, 133, 134]. The drawback of this assumption is that each distributed entity of the DS allows other entity to store information about an object that is not within its geo-restriction. Thus, each DS entity on a P2P network sets access privileges to a subset of its resources - such as processing power, disk storage, or network bandwidth - for another network participant.

As a remedy to this drawback, we propose hierarchical distributed discovery architecture that is based on EPC Border Gateway Protocol (EPC-BGP) to allow different supply chains to exchange EPC status and use EPC Open Shortest Path First (EPC-OSPF) to exchange EPC status within the members of each supply chain. This architecture inherits the agility of the previous architecture in which each supply chain can share EPC status information without revealing the details of its network to other competitor. It only shares the status of an EPC to other DS entities notifying them about the existence of new information about the object holding the corresponding EPC. Furthermore, the

proposed scheme defines a threshold denoting the number of changes in the EPC status of a node, and triggers EPC status advertisements once the threshold is exceeded. We evaluate the performance of our proposal through simulations, and our results confirm that advertising the update messages of the EPC-BGP upon a certain amount of change (i.e.,, at the level of, 0.05∼0.10) in the routing table can ensure low blocking probability of EPC tracking request while significantly reducing the communication and computing overhead.

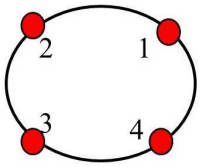
This chapter is structured as follows. Section 7.2 presents the intra and inter-domain routing protocols, namely EPC-OSPF and EPC-BGP for EPCglobal network. In Section 7.3, we present our proposed threshold-based DS mechanism along with the network architecture in detail. In Section 7.4, we present and discuss the queuing analysis and the numerical results. Section 7.5 concludes this chapter.

7.2 Routing Protocols for EPCglobal Network

7.2.1 Intra-Domain Routing protocol for EPCglobal Network

We adopt Open Shortest Path First (OSPF) as the intra-domain routing protocol for EPCglobal network. The reason behind this selection is as follows. The Internet Engineering Task Force (IETF) proposes OSPF as an intra-domain routing protocol for IP networks [135] and has been shown to be bandwidth-efficient. One drawback of OSPF is that it may have larger Link-State Advertisement (LSA) packet sizes compared to other intra-domain routing protocols such as Intermediate System to Intermediate System (IS-IS) which has a fixed routing packet size.

In this study, we define a new LSA for the EPCglobal network, namely EPC-LSA that is used to convey EPC information in the local supply chain. This new LSA has a 10-bit LS type field that has not been used yet by the IETF [136]. This new LSA will pass information about EPCs. Once new information about a specific EPC is available, the corresponding status of this EPC will be changed from True to False indicating that there is new information about this specific EPC. OSPF initially exchanges routing information among the supply chain nodes. Once this information is exchanged, each



Dest	Path	Cost	Status of EPC		
1	****	C <1,1> =0	F	T	T
2	2	C <1,2> =1	F	T	T
3	2, 3	C <1,2> + C <2,3> =2	F	T	T
4	4	C <1,4> =1	F	T	T
Node 1 Intra-domain Routing table					

Figure 7.1 An instant of the routing table of node-1

node constructs a routing table to reach other nodes in the supply chain by running Dijkstra's shortest path algorithm. Upon constructing the routing tables, information about objects can be exchanged using EPC-LSA messages. Figure 7.1 illustrates a snapshot of the routing table for node-1 where the first EPC has new information that should be shared with the rest of the supply chain. We define a supply chain head such that once new information is available about a specific EPC this node establishes connection with the corresponding supply chain head (i.e.,, node-1), and then, the supply chain head exchanges this information with the other nodes in the supply chain.

7.2.2 Inter-Domain Routing protocol for EPCglobal Network

Electronic Product Code Border Gateway Protocol (EPC-BGP) is built on top of BGP [137], [138], while research towards improving the operation and scalability of BGP is still active [139], [140]. EPC-BGP adopts the features of BGP, and further enables the EPC information to be exchanged among edge routers to maintain the up-to-date status of the EPC. All EPC information is exchanged in a BGP update message, e.g., the AS-Path attribute that list all the EPCglobal networks that are traversed by these EPCs. Thus, upon the receipt of a request to obtain the information about a given EPC,

the gateway of the corresponding EPCglobal network uses the AS-Path attribute to retrieve information. Each gateway maintains an EPC table where each row stores the information of an EPC such as its status. The status of an EPC is set to false if upon product is recalled, i.e., there is new information about the corresponding EPC. The EPC status is set as true once its information has been shared with other EPCglobal networks. Furthermore, EPC-BGP introduces a new route advertising scheme which is triggered by the number of changes that have taken place in the status of the EPCs in the EPCglobal network. A new routing protocol is emergent due to the basic requirement of EPCglobal for having routing information at the edge routers so that most up-to-date information about any EPC is guaranteed. Therefore, EPC-BGP aims at offering a reliable mechanism to exchange EPCs among different EPCglobal networks to guarantee the most up-to-date information about each EPC. For the details of EPC-BGP, the reader is referred to [73].

7.3 Network Architecture

7.3.1 EPC global network extension and operation

As shown in Figure 7.2, the gray part denotes the changes to the EPCglobal network which is basically a gateway connecting the EPCglobal network to the Internet and removing the ONS root. The operation of the EPCglobal network is as follows. The normal process starts with reading the tag via an RFID reader in order to get the EPC and present it to the client application. In order to illustrate the EPCglobal network operation, we consider an example of reading a Serialized Global Trade Item Number (SGTIN-96) key. The combination of a GTIN and a unique serial number, does correspond to an EPC. The SGTIN-96 is presented to the client application as a bit sequence and the client application converts the sequence of bits as shown in this example: urn:epc:id:sgtin:200452.5742.5508265. The client application server presents this bit stream to the local ONS as shown in step-3 in Figure 7.2. The ONS converts the EPC to the following format 5742.200452.sgtin.id.onsepc.com prior to sending it to the ONS root (step-4). The ONS root returns a series of responses that contain Uniform Resource Locator (URLs) pointing to one or more services, e.g.,, an EPCIS Server (step-5). Next, depending on the service requirements of the ONS Client, the ONS uses one or more of

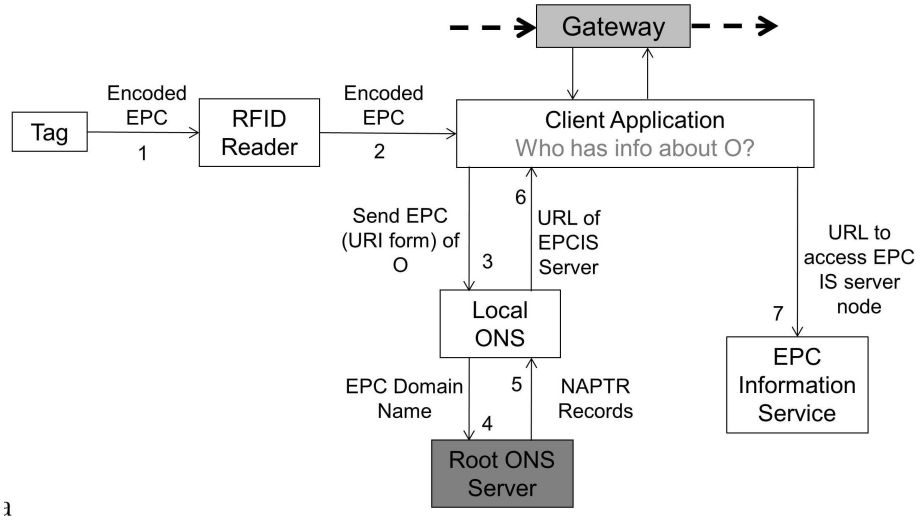


Figure 7.2 EPCglobal network extension

the Name Authority Pointer (NAPTR) records returned to locate an appropriate service. Finally, the local ONS returns the selected URL back to the client application where it will be presented to the EPC-IS data server to obtain information about the desired object. In the new EPCglobal network architecture, the local ONS holds the URL for all the local EPCs besides any roaming EPC that has been hosted in this EPCglobal network.

7.3.2 EPC global network Architecture

Normally, a supply chain can have up to 15 members however, for the sake of simplicity, we illustrate supply chains consisting of four members in Figure 7.3. In each supply chain ring, members are interconnected via a gateways. Furthermore, each supply chain has a supply chain head that connects the entire supply chain to the Internet through an inter-domain gateway. As mentioned above, in each supply chain, information about objects that are shipped locally within the same supply chain can be exchanged via EPC-OSPF protocol. On the other hand, if an object has to be shipped outside a local supply chain, this information will be exchanged using EPC-BGP. In both intra-domain or inter-domain shipments, the routing information about reaching all destinations is exchanged to construct a map showing routes to all destinations in the same network. Once the map is constructed, the second phase calls the EPC-OSPF to exchange the EPCs of objects that is being shipped to other nodes in the same supply chain. When

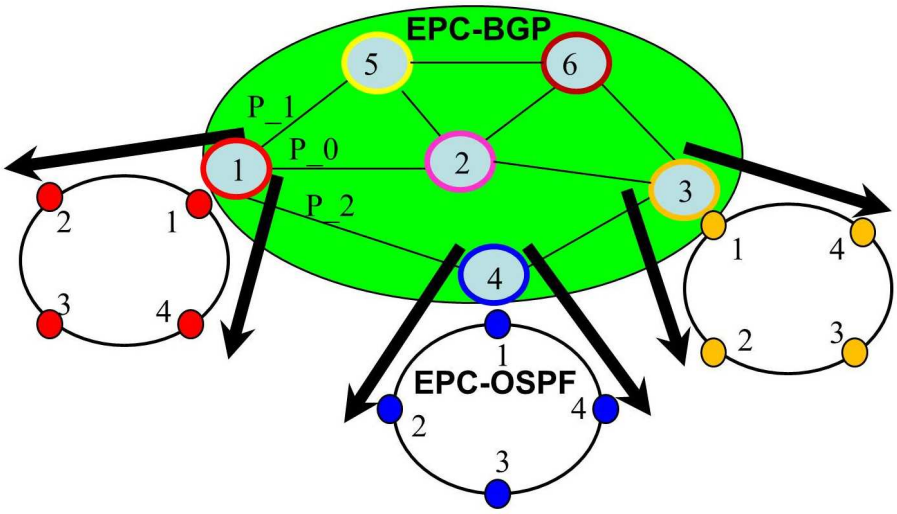


Figure 7.3 EPCglobal network architecture

a group of objects are to be sent to a destination located in another domain, the supply chain head runs the EPC-BGP to find a route towards the destination. At the time of production, an object is attached an RFID tag that holds an EPC to identify this object. This RFID will be read at multiple locations across the supply chain, the first time the RFID is being read, the client application gets the EPC from the reader and converts the EPC to the format that can be realized by the local ONS. Once the ONS receives this EPC, it realizes that this EPC has been read for the first time, so it register it in the ONS and link the EPC to a URL that can be used by the client application to gain information about this EPC through the EPC-IS. Once the object is being shipped to other nodes within the same supply chain, it registers itself in the local ONS of neighbour nodes. If the object is being shipped to another supply chain, due to unrecognized destination, the host node sets a path with the supply chain head while all nodes along the path forward the routing message towards the supply chain head as they do not recognize the destination of the shipment. Once the supply chain head receives the message, it calls EPC-BGP to send the EPC to the desired destination. At each inter-domain node along the path, the shipped EPCs will register themselves in the local ONSs of the inter-domain nodes.

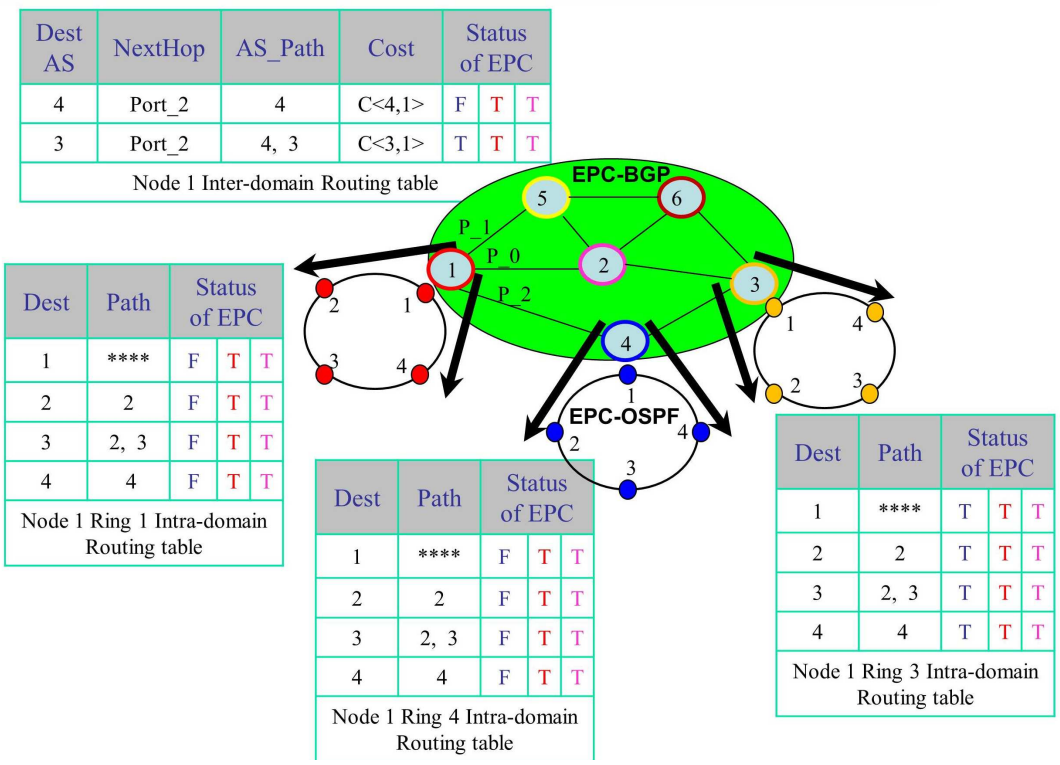
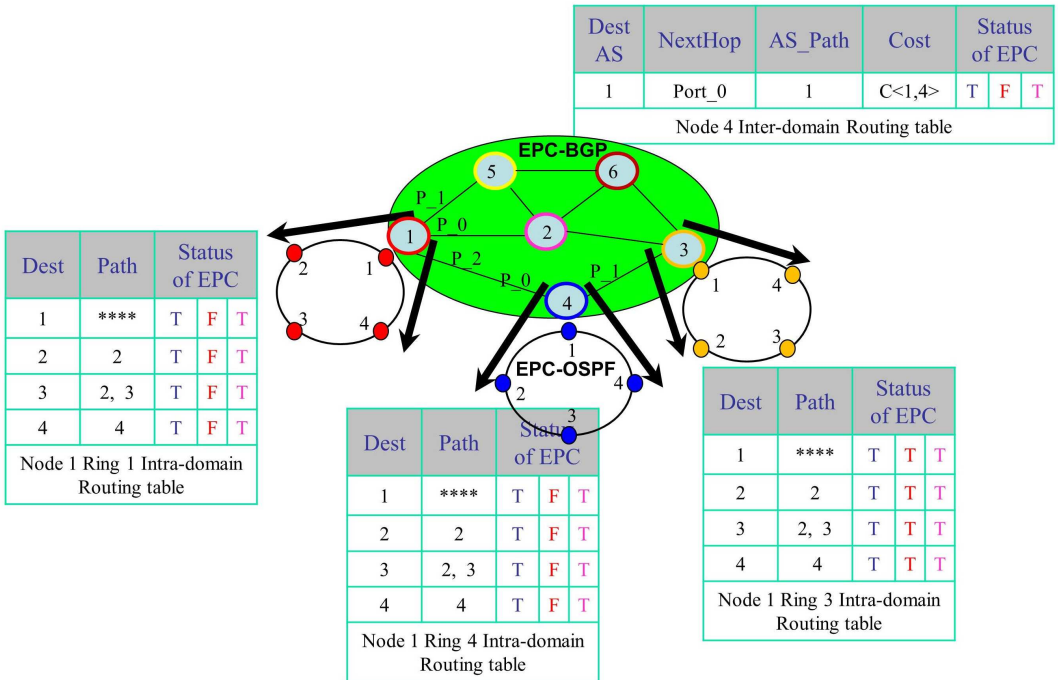


Figure 7.4 Inter and intra-domain interaction to perform update session of EPC1 between supply chain-1 and 4

7.3.3 Intra and Inter-domain routing protocols interaction

EPC-OSPF and EPC-BGP need to exchange information about an object that traverses more than one supply chain. The exchanged information should not reveal any detailed information about the interconnection network nor the supply chain networks. Figure 7.4 illustrates a scenario of an update session to obtain information about object-1 that has been shipped from supply chain-1 to supply chain-4. In the first row of the inter domain routing table of node-1, the status of EPC of object-1 has been changed from TRUE to FALSE across the entire path. Furthermore, this status update has also been reflected to the following entities: *i*) The intra-domain routing table of the supply chain head-1, *ii)* intra-domain routing table of supply chain -4, and *iii)* the inter-domain routing table of supply chain head-1. Following the end of the update session, the status of the EPC along the path will be switched back to TRUE.

In Figure 7.5, it is also seen that new information exists about object-2 in supply chain-4. This is indicated by changing the corresponding EPC status from TRUE to FALSE in



Dest AS	NextHop	AS_Path	Cost	Status of EPC		
1	Port_0	1	C<1,4>	T	F	T

Node 4 Inter-domain Routing table

Dest	Path	Status of EPC		
1	****	T	F	T
2	2	T	F	T
3	2, 3	T	F	T
4	4	T	F	T

Node 1 Ring 1 Intra-domain Routing table

Dest	Path	Status of EPC		
1	****	T	F	T
2	2	T	F	T
3	2, 3	T	F	T
4	4	T	F	T

Node 1 Ring 4 Intra-domain Routing table

Dest	Path	Status of EPC		
1	****	T	T	T
2	2	T	T	T
3	2, 3	T	T	T
4	4	T	T	T

Node 1 Ring 3 Intra-domain Routing table

Figure 7.5 Inter and intra-domain interaction to perform Update session of EPC2 between supply chain-4 and 1

supply chain head-4. It is worthwhile noting that this new information about object-2 in supply chain-4 could have been triggered by any node within the corresponding supply chain. The node that has product recall information contacts the supply chain head, and the supply chain head starts the update sessions on all nodes that this specific object has traversed. The supply chain head first updates its intra-domain routing table by setting the status of all destinations that hosted this EPC to FALSE as shown in Figure 7.5. If the supply chain head realizes that object-2 has new information and has been shipped to other supply chains, it sets the status of the object in the inter-domain routing table as FALSE. Figure 7.5 shows the update session between supply chain-4 and supply chain-1 of object 2 that is located in supply chain-4.

To clarify the interaction of inter and intra-domain routing protocols, we assume continuous update scenario in which supply chain heads advertise every single change in their routing table. In our scenario, both supply chain-1 (Figure 7.4)and supply chain-4 (Figure 7.5) have a single change in their inter-domain routing tables after the information from intra-domain has been reflected in these inter-domain routing tables. Both supply chain-4 and 1 advertise the status of the shipped EPCs to other supply chains using EPC-BGP. This new status of the EPC will be exchanged through an EPC-BGP update

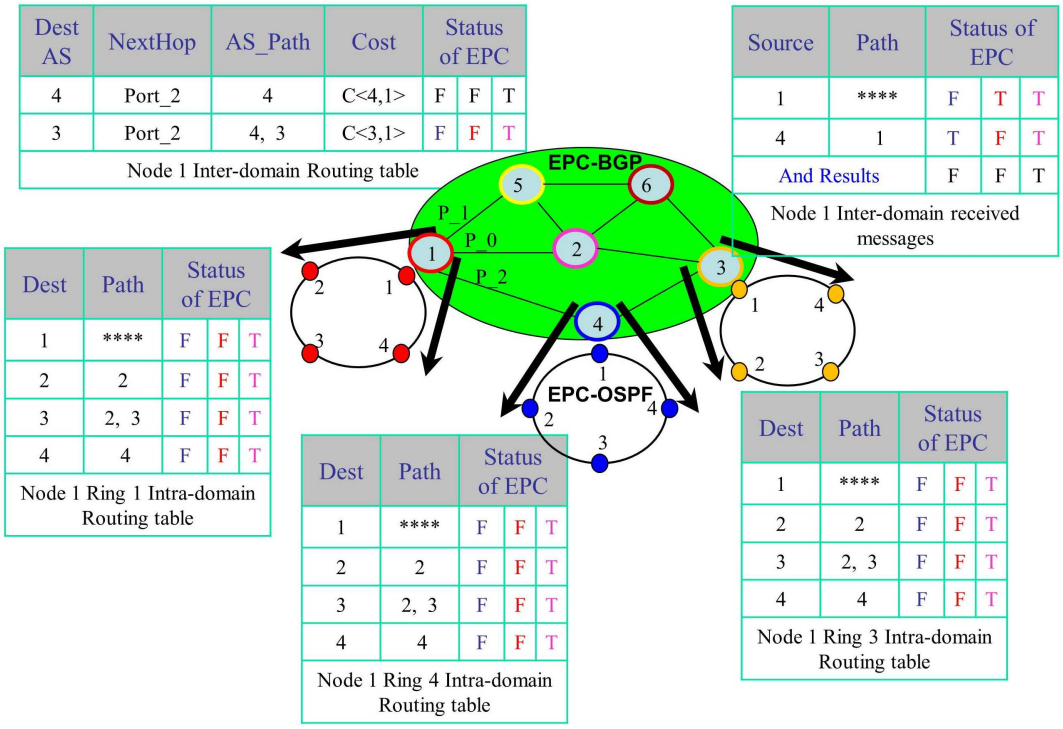


Figure 7.6 Inter and intra-domain interaction to Update the inter-domain routing table of supply chain-1

message as explained in [73]. Thus, supply chain-1 advertises the status of the shipped EPCs to all other supply chains in the network, and at the same time supply chain-4 advertises the status of the shipped EPCs to all other supply chains. Figure 7.6 shows the received advertised information from supply chain-4 and the local inter-domain information of supply chain-1. Each supply chain head makes use of the received information to update its inter-domain routing table. Figure 7.6 shows the information that has been received from supply chain-4 to supply chain-1. The top right section of the figure shows the AND operation on the received EPC status at supply chain-1. Basically supply chain-1 performs an AND operation on the EPCs status across the entire path. As we can see in Figure 7.6 the routing table of supply chain-1 is updated denoting that both EPC1 and EPC2 are under an update session that requires sometime to exchange information about these objects among different supply chains. It also shows that EPC1 and EPC2 status that has been shipped to supply chain-3 is also reserved indicating that some information are available about these two EPCs.

7.3.4 Signaling messages

To track objects and/or to provide control on the EPCglobal network architecture, a signaling framework is emergent where signaling messages have the ability to interact between intra and inter-domain routing information to track or control the information about any object.

Signaling requests start by changing the status of an object from TRUE to FALSE at each inter or intra-domain routing table. The procedure of setting an update path is presented here. Once the reader scans the object, it gets the EPC from the object and forwards the EPC to the client application. Upon receiving the URL from ONS, the client application applies the URL on the local EPCIS to get all the information about this object while at the same time, contacting the supply chain head which connects the supply chain to the backbone network. The supply chain head sets a local intra-domain message to all the nodes in the supply chain. Each node receiving the EPC from the supply chain head will change the status of the corresponding EPC to FALSE in its intra-domain routing table, lookup for the information about the received EPC in the local ONS and send the information back to the supply chain head after retrieving it from the local EPCIS. The supply chain head changes the status of EPC in the intra-domain routing table from TRUE to FALSE, and at the same time, it checks if the EPC has been shipped to any supply chain residing in another domain. If the EPC has been shipped to any other supply chain, the supply chain head initiates an Inter-domain signaling message using the routing information at the inter-domain level. The supply chain head makes use of the AS-path to gain the information at all the supply chains that host this object. At each supply chain head, the same procedure takes place, the supply chain head changes the inter-domain status from TRUE to FALSE, and sends a message to all the nodes in the supply chain after setting the intra-domain status as FALSE. Each node quires queries the local ONS for local URL to retrieve information about this roaming EPC from the EPC-IS and send back the results to the supply chain head. Once all the AS of the AS-path attribute has been traversed, all the information at each AS is sent back to the supply chain head where the request has been initiated. Finally, the supply chain sends the information to the local node that submitted the request. We have defined the following messages in the signaling scheme:

$Intra-domainUpdateQueryRequest(INUQR)$: is initiated by a supply chain head based on a request by any node in the supply chain to query all the information about an object. This message makes use of the routing information available at supply chain head to find all the paths that this object of interest has traversed. At each node across the path, this message enforces each node to change the status of the EPC of interest from TRUE to FALSE indicating that this EPC is under an update session.

$Intra-domainHoldUpmessage(INHU)$: keeps the session alive until the inter-domain routing information about the EPC of interest has been collected. Once the supply chain head receives the INUQR, it checks the EPC of interest, and if the supply chain head realizes that this object has been shipped to other supply chains, it sends INHU to the node who initiates the information query about the EPC of interest.

$Inter-domainUpdateQueryRequest(UQR)$: is needed to retrieve information about an object across multiple supply chains, and it is initiated by the supply chain head. The message makes use of the inter-domain routing table to find out the path that this object traverses and queries all the information about this object from each supply chain on the path. UQR enforces each supply chain head along the path to change the status of the EPC of interest from TRUE to FALSE indicating that the corresponding EPC is under an update session. It is worthwhile to mention that across the path, supply chain heads are responsible to retrieve the information about the EPC of interest via INUQR within its supply chain.

$Intra-domainTear-Downmessage(INTD)$: is initiated by the supply chain head that started the EPC update session, and it tears down the intra-domain connection of an update session for a given EPC. At each node across the path, this message enforces each node that hosted this object to change the status of the EPC of interest from FALSE back to TRUE indicating that the EPC is available free again.

$Inter-domainTear-Downmessage(TD)$: is initiated by the supply chain head where the update session request has been initiated from. At each supply chain head across the path, this message enforces each supply chain head to change the status of the EPC of interest from FALSE back to TRUE indicating that this EPC is available free again and the inter-domain connection of an update session has been released.

$Intra - domainClearmessage(INCLR)$: clears all data about a specific object that no longer exists in the supply chain. INCLR results clearing up the local ONS, the EPCIS and the routing table across the path that the corresponding object has traversed. INCLR is sent out by any node realizing the expiry of an EPC within a supply chain.

$Inter-domainClearmessage(CLR)$: is initiated by the supply chain head and sent to all supply chain heads across the path in which the corresponding object has traversed. CLR results in clearing the ONS of all supply chain heads that hosted the corresponding object as well as all the supply chains that have been traversed by the object. Furthermore, CLR removes the EPC from the inter-domain routing table.

$Intra - domainConfirmation(INCONF)$: is a reply back message from all nodes within the supply chain that store information about the EPC of interest.

$Inter - domainconfirmation(CONF)$: a reply back message from all supply chain heads that store information about the EPC of interest.

7.3.5 Advertisements thresholds

An update request takes place when there is a request to gain all the information about an object across all the supply chains that an object traverses. Figure 7.4 illustrates an update request scenario. In the figure, the inter-domain routing table of node-1 has two entries where the former denotes the routing information about objects that has been shipped to ring-4 or AS-4. The routing information allows update requests from AS-1 to reach the objects in AS-4 to retrieve information about any object located in AS-4. In the figure, both supply chain-1 and supply chain-4 have the status of object-1 as FALSE indicating that the object is under an update session. If the status of object-1 has not been exchanged with other supply chains, any update request for object-1 initiated by any other AS leads to blocking of the request. In AS-4, we observe the same phenomenon where new information (e.g.,, product recall) exists about object-2 requiring a local update session in AS4. If the second entry of inter-domain routing table of node-1 is being used to retrieve information about object-2, the request will be blocked. Thus, the status of object-2 in AS-4 should be exchanged with other domains to reflect the current status of this object in order to prevent blocking. Hence, the status of each EPC must be updated however deciding on the frequency of updates is a challenging issue as

more frequent updates lead to increased communication overhead whereas infrequent updates are expected to introduce increased blocking probability. In order to cope with this issue, we propose a threshold-based update where the threshold denotes the number of changes that the inter-domain routing table has experienced. Thus, once the number of changes in the inter-domain routing table exceeds the pre-defined threshold, the corresponding supply chain head advertises its routing table so that all supply chain heads can be aware of the status information in the advertised table. It is worthwhile to note that threshold-based updates can introduce three types of blocking as defined below:

7.3.5.1 Justified Update Blocking (JUB)

occurs when the requested EPC is being updated in real-time and as provided by the routing table of the initiating node. Probability of JUB is formulated in Equation 7.1 where α is the number of justified update blocking events and N is the total number of requests.

$$PJUB = \frac{\alpha}{N} \qquad \text{where} \quad 0 \leq \alpha \leq N \tag{7.1}$$

7.3.5.2 Unjustified Update Acceptance (UUA)

denotes the situation where the EPC is not available in real time whereas the routing table of the gateway shows that the requested EPC is available. The reason of this difference is that each gateway may wait for a certain period of time to report about new updated EPCs to other nodes. Consequently, if the corresponding EPC at other gateways is being used, it will lead to blocking somewhere along the path. Probability of UUA is formulated in Equation 7.2. Where β is the number of Unjustified Update Acceptance

$$PUUA = \frac{\beta}{N} \qquad \text{where} \quad 0 \leq \beta \leq N \tag{7.2}$$

7.3.5.3 Unjustified Update Blocking (UUB)

occurs when the requested EPC is available for update in real time whereas the routing table of the edge node shows that the requested EPC is not available. Similar to UUA, the reason for the difference between the global table and the gateway table is due to the wait time of each node prior to reporting the newly released EPC. Hence, due to invisibility of these available EPCs to the other nodes, these resources will not be used when requested by a customer. Probability of UUB is formulated in Equation 7.3. Where γ is the number of Unjustified Update Blocking.

$$PUUB = \frac{\gamma}{N} \qquad \text{where} \quad 0 \leq \gamma \leq N \tag{7.3}$$

The total blocking probability (PT) is equal to the sum of the three types of blocking probabilities as shown in Equation 7.4

$$PT = PJUB + PUUA + PUUB \tag{7.4}$$

The ideal case is to have both UUA and UUB being equal to zero, denoting that the distributed routing information is 100% correct and accurately reflects the resource allocation in the EPCglobal network. The above mentioned blocking types are our major concern in performance evaluation since they are directly related to the performance of our proposed architecture.

7.4 Performance Evaluation

To evaluate our EPCglobal network architecture and its proposed discovery service, we have provided two major subsections, the first one is performance analysis which provide queuing analysis of the blocking probability for the EPC update requests. Whereas the second one provides an extensive simulation results to evaluate our EPCgloabl network architecture.

7.4.1 Performance analysis

In this section, we explain the queuing analysis of the blocking probability of the EPC status. All the used notations are defined in the List of Symbols at the beginning of this thesis. We aim at formulating the blocking probability in terms of threshold and the number of EPCs. As it is shown in Equation 7.8, the total blocking probability P can be defined as total number of update reservation requests blocked at a given AS over the arrival rate δ.

$$\delta_i = \delta \cdot \frac{N_i^{paths}}{N^{paths}} \tag{7.5}$$

$$\begin{aligned} \delta_i^j &= \frac{\delta_i}{N_E} \\ &= \frac{\delta \cdot N_i^{paths}}{N_E \cdot N^{paths}} \end{aligned} \tag{7.6}$$

$$\delta_{avg} = \frac{N_{avg}^{paths} \cdot \delta}{N^{paths}} \tag{7.7}$$

$$\begin{aligned} \mathcal{P} &= \frac{\sum_{i=1}^{N_A} B_i}{\delta} \\ &= \frac{\sum_{i=1}^{N_A} [\sum_{j=1}^{N_E} \delta_i^j \cdot [(1 - e^{-\delta_i^j \cdot \mu}) + [T \cdot (N_A - 1) \cdot F \cdot (1 - e^{\frac{-2\delta_{avg}}{T \cdot N_E}})]]]}{\delta} \end{aligned} \tag{7.8}$$

$$
\begin{aligned}
B_i &= \sum_{j=1}^{N_E} \delta_i^j \cdot [P(A_{i,j}) + P(B_{i,j})] \\
&= \sum_{j=1}^{N_E} \delta_i^j \cdot [P(ElapseTime < \mu) + P(B_{i,j})] \\
&= \sum_{j=1}^{N_E} \delta_i^j \cdot [(1 - e^{-\delta_i^j \cdot \mu}) + P(B_{i,j})] \\
&= \sum_{j=1}^{N_E} \delta_i^j \cdot [(1 - e^{-\delta_i^j \cdot \mu}) + [(N_A - 1) \cdot T \cdot F \cdot P_{pois}(X_{ad} \geq 1)]] \\
&= \sum_{j=1}^{N_E} \delta_i^j \cdot [(1 - e^{-\delta_i^j \cdot \mu}) + [T \cdot (N_A - 1) \cdot F \cdot (1 - P_{pois}(X_{ad} = 0))]] \\
&= \sum_{j=1}^{N_E} \delta_i^j \cdot [(1 - e^{-\delta_i^j \cdot \mu}) + [T \cdot (N_A - 1) \cdot F \cdot (1 - e^{\frac{-2\delta_{avg}}{T \cdot N_E}})]]
\end{aligned}
\tag{7.9}
$$

The total number of blocked update requests can be defined as shown in Equation 7.9, which is the result of summing two types of probabilities multiplied by δ_i^j defined in Equation 7.6. The first one is P($A_{i,j}$) which represents the probability of receiving two update requests for the same EPC at the same AS within a time frame less than μ whereas the second one is P($B_{i,j}$) representing the probability of receiving an advertisement that is initiated by one of the other autonomous systems in the network and received by this local AS_i reserving a given EPC. As it is shown in Equation 7.9, P($A_{i,j}$) can be expressed using a Poisson distribution modeling the reservation requests' arrival process in AS_i, it models the probability of receiving two update requests for the same EPC at a specific AS within a time frame which is equal to or less than μ. Equation 7.9 also shows that P($B_{i,j}$) can be expressed in terms of the following three probabilities. The probability of receiving an advertisement that is initiated by $N_A - 1$ other AS, the probability of EPC e_j is among the advertised list of EPCs T and finally the probability of EPC advertised as reserved F. The probability of receiving an advertisement from one or more other AS is assumed to follow Poisson distribution with a $\delta = \frac{2\delta_{avg}}{T \cdot N_E}$. In the arrival rate formulation, δ_{avg} is doubled since the counter at each node is incremented once an EPC is reserved or released. It is worth to mention that F is the probability of having an EPC status as reserved, the value used for F is 0.5. It is worth mentioning

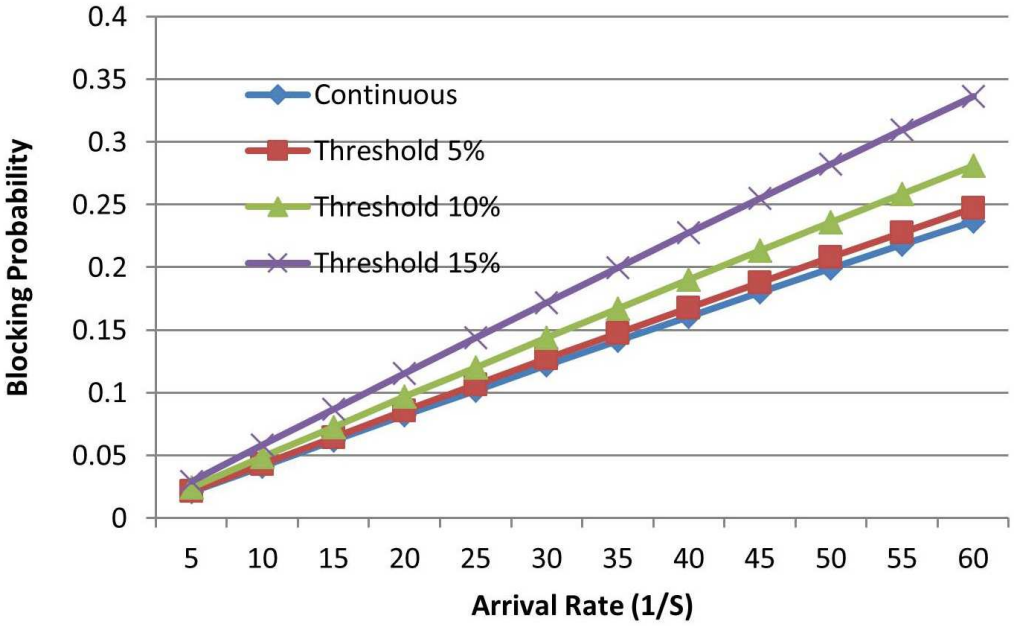


Figure 7.7 Blocking probability for different thresholds value and 100 EPC

that the status of an EPC can change within μ more than once before it is advertised. To show the effect of the threshold on the blocking probability, we have provided Figure 7.7. As seen in the figure, blocking probability increases with the threshold since more errors are expected in the routing table before the accurate statues of EPC is exchanged. This behavior matches our simulation results.

Figure 7.8 shows the blocking probability at different number of EPCs. Since more EPCs are available resulting in less blocking probability, probability of selecting the same EPC decreases. Therefore, blocking probability decreases as the number of EPCs increaes. This analysis is also coherent with our simulation results in the next subsection.

7.4.2 Simulation results

In our simulation, we have simulated multiple supply chains, each supply chain is a ring consisting of 8 nodes, each node represents an EPCglobal network with a gateway that connects the supply chain nodes together. All supply chain heads are connected together forming the well-known network architecture of the San Francisco network [141]. It is worthwhile to note that the supply chain size does not affect the results whereas the

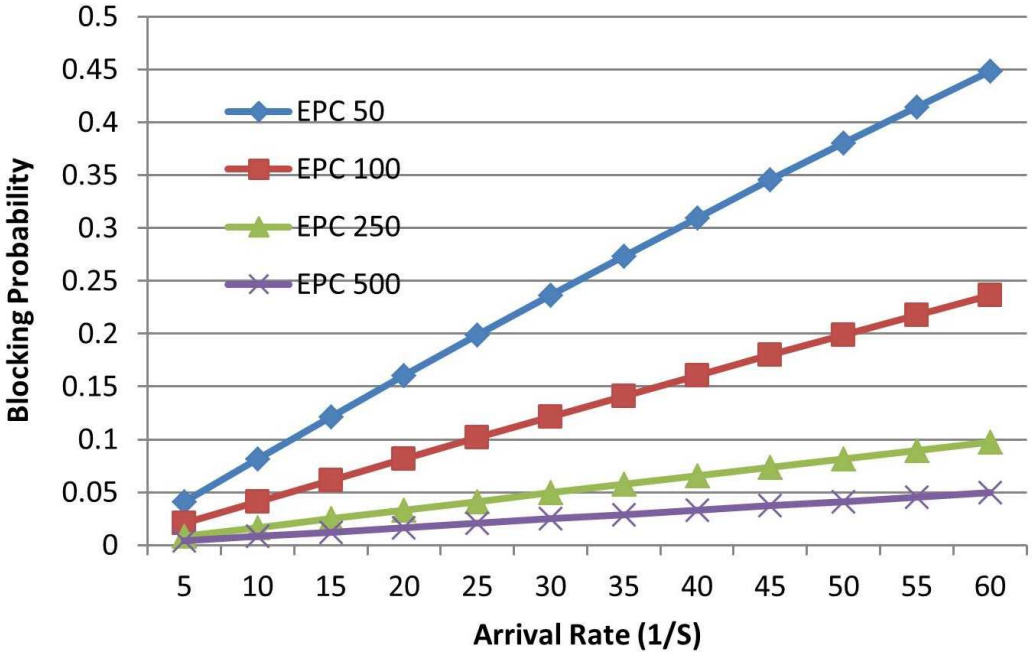


Figure 7.8 Blocking probability for different number of EPCs

number of EPCs has an impact on the performance as illustrated in the next subsection. We have defined two global database types, an inter-domain data base and a number of intra-domain data base. The global data base is used to keep track of the real time EPC status of all EPCs in the simulated network model. Thus, the global databases have precise information that can be used to verify the accuracy of the distributed routing information located in each supply chain head. On the other hand, the intra-domain database is a table that is used to keep track of the status of the EPCs in the distributed routing information of a single supply chain. For each supply chain head, an individual intra-domain database is maintained.

The requests are assumed to arrive following a Poisson distribution with varying arrival rates. Each request includes the number of the requested EPCs and the destination node. The selection of the destination node is uniformly distributed over the total number of supply chain heads. Each point in the plots represents the average of five runs.

In the gateway of an EPCglobal network node, the update advertisement is done based on a threshold denoting the amount of information that has changed since the last update. In this chapter, we investigate the impact of triggering the advertisement based on the threshold values of 5, 10 and 15% each of which denotes the percentage of change

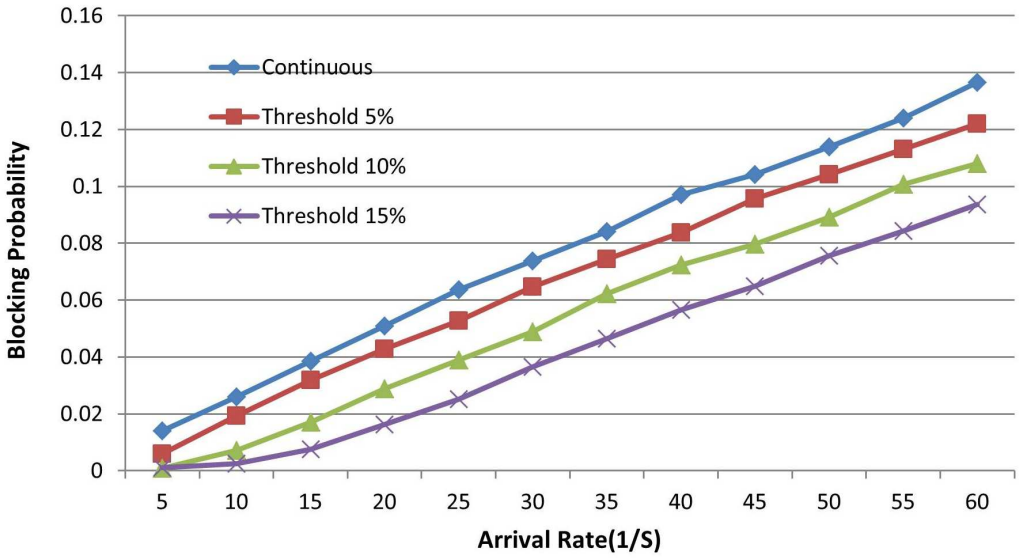


Figure 7.9 Justified Update Blocking for 8-node supply chains and 100 EPC shipped

in the distributed database in the gateways. We compare the threshold-based advertisement to continuous refreshing which advertises every single change in the routing table of the gateway.

7.4.2.1 Blocking probability

Figure 7.9 presents the JUB under varying arrival rates for a network of supply chain equal to eight and the number of shipped EPCs is 100. As observed in the figure, the higher the threshold the lower the JUB since more errors are expected in the status of EPCs. Thus, if an EPC status shows availability in the routing table, it will lead to UUA blocking. The same phenomenon is expected to occur for the UUB. It is clearly shown that when continuous update threshold is used, JUB increases. This is due to the fact that the routing information is accurate as the routing information is being exchanged upon every single change in any routing table; hence the other blocking types will not occur as the routing information is always up-to-date.

Figure 7.10 illustrates the performance results for UUA blocking when the number of shipped EPCs is 100. The reason for UUA blocking is that the changes at other routing tables have not been reflected in all routing tables. As mentioned above, the higher the threshold is, the higher the UUA blocking is as more errors will take place in the routing

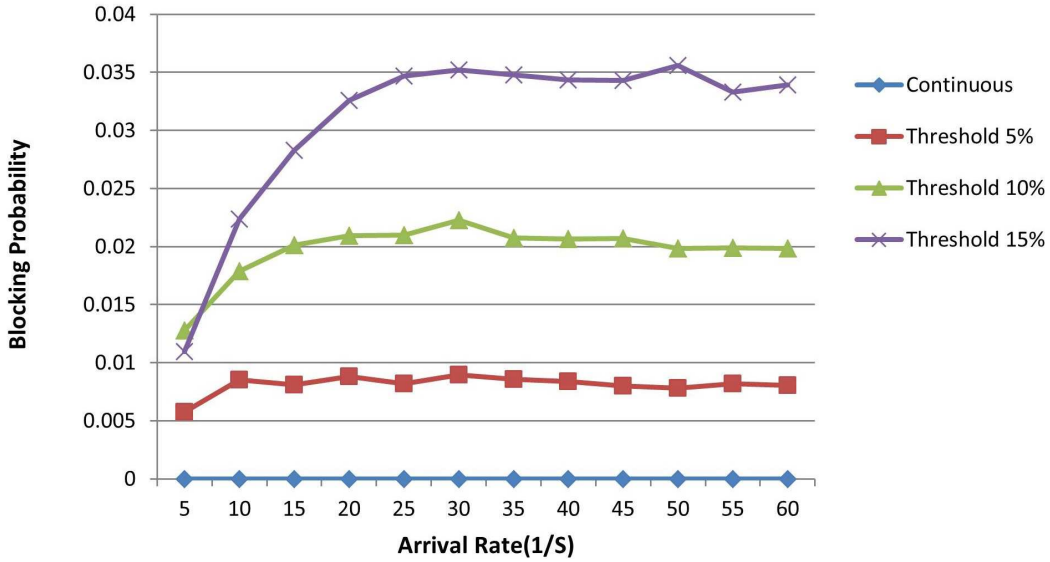


Figure 7.10 Unjustified update acceptance for an 8-node supply chains and 100 EPC shipped

table. In other words, UUA occurs if the status of real time sessions of certain EPCs have not been reflected in other routing tables. Hence, a local routing table may chose an EPC with TRUE status even though the EPC is already under another update session by another node. Thus, UUA blocking is expected across the update session path. As it is also clear, the higher the rate of update session requests, the higher the UUA blocking due to less available free EPC.

Figure 7.11 illustrates UUB, and similar to UUA blocking, UUB also occurs due to out of date routing information. The following scenario can assist understanding this phenomenon. At a given node the requested EPC status seems to be FALSE; hence it has never been used even though the updates are done frequently. The actual value of the EPC status is to be updated as this EPC has been released and the status of this free EPC has not been reflected in the local routing table of the requested node. This frequency of this phenomenon is higher when the rate of the updates is low which results in higher UUB.

Figure 7.12 presents the total blocking probability for each threshold. Continuous update mode can be considered as a benchmark; hence the desired situation is expected to behave similar to the benchmark where the total blocking probability is low and equal only to the JUB. On the other hand, under the threshold-based advertisement, the higher

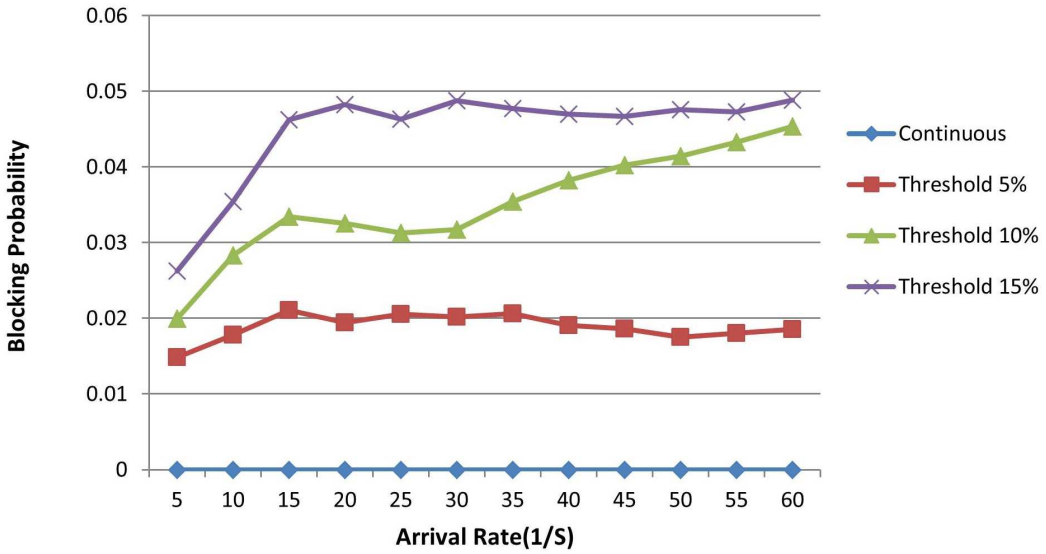


Figure 7.11 Unjustified Update Blocking for 8-node supply chains and 100 EPC shipped

the threshold is, the higher the blocking probability experienced due to out of date information in the routing tables.

Finally, Figs. 7.13, 7.14, 7.15, 7.16 present the JUB, UUA, UUB and the total blocking probability when the number of EPCs is 250. As we have mentioned, the number of nodes of each supply chain does not affect the results as the supply chain is connected through a supply chain head. As a result of having more EPCs available, all types of blocking probability decrease. As expected also, the higher the update session requests is the higher the blocking probability is.

7.4.2.2 Euclidean distance

In order to evaluate the difference between the routing tables, we calculate the average Euclidean distance (EU) between all tables as shown in Equation 7.10 where N is the total number of the nodes in the network, and D_{ij} is the root square value of the squared number of differences between two entries in two routing tables. Figure 7.17, 7.18 present the EU for EPC 100 and EPC 250, respectively for a supply chain. Both graphs demonstrates the impact of threshold-based advertisement on the EU, and shows the direct relation between the threshold and the EU distance. This is due to the fact that

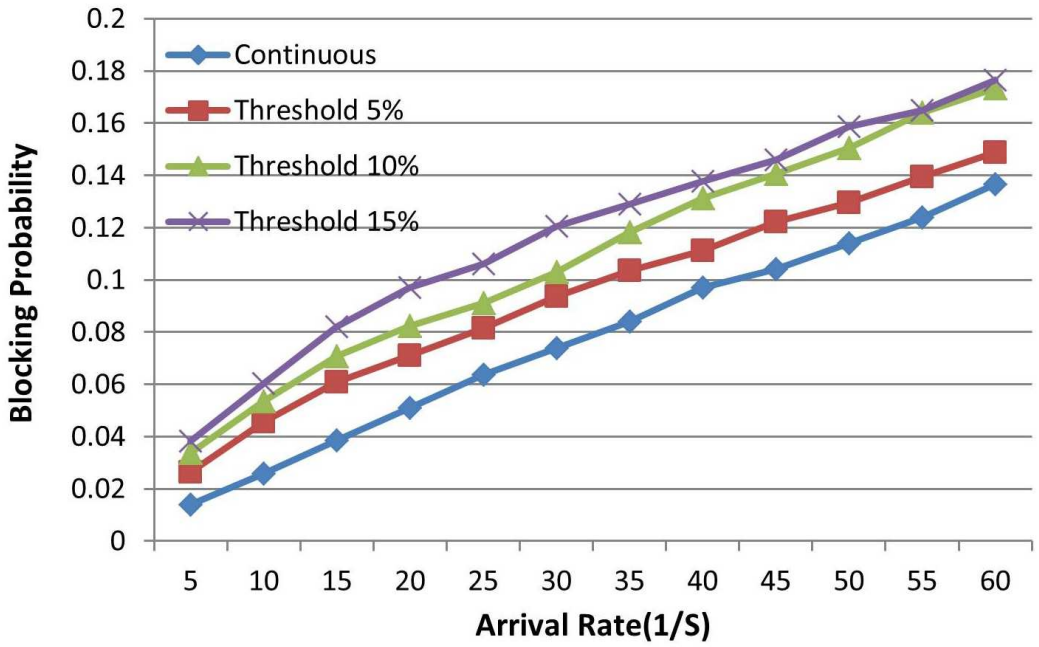


Figure 7.12 Total blocking for 8-node supply chains and 100 EPC shipped

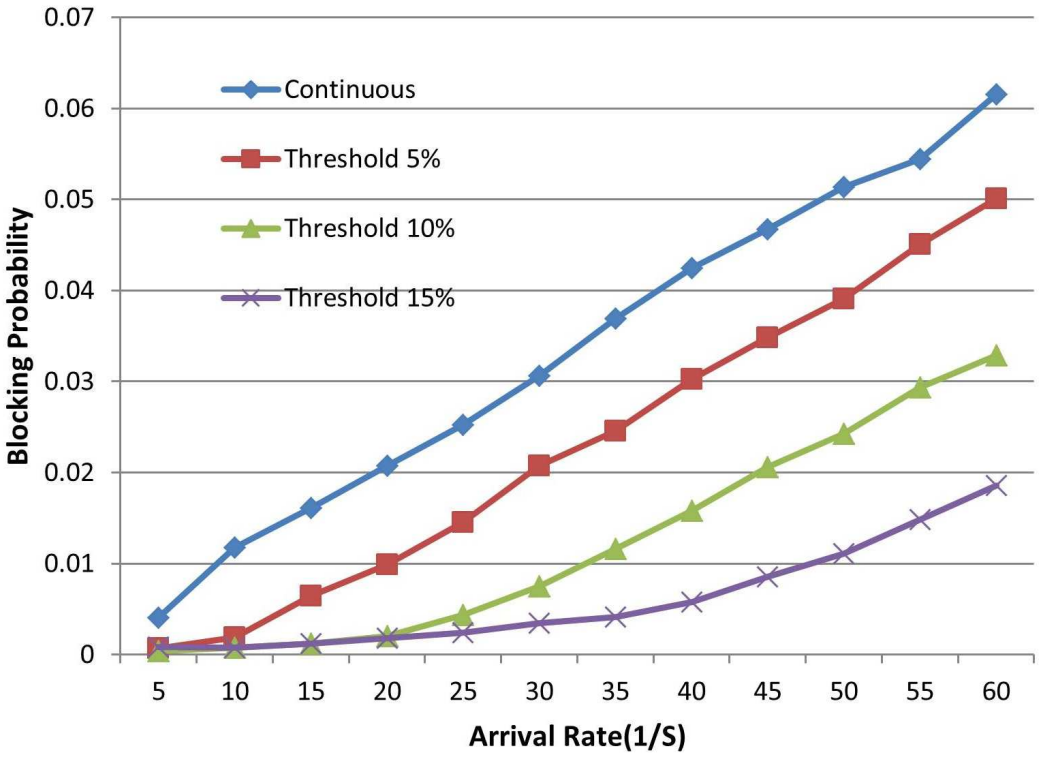


Figure 7.13 Justified Update Blocking for 8-node supply chains and 250 EPC shipped

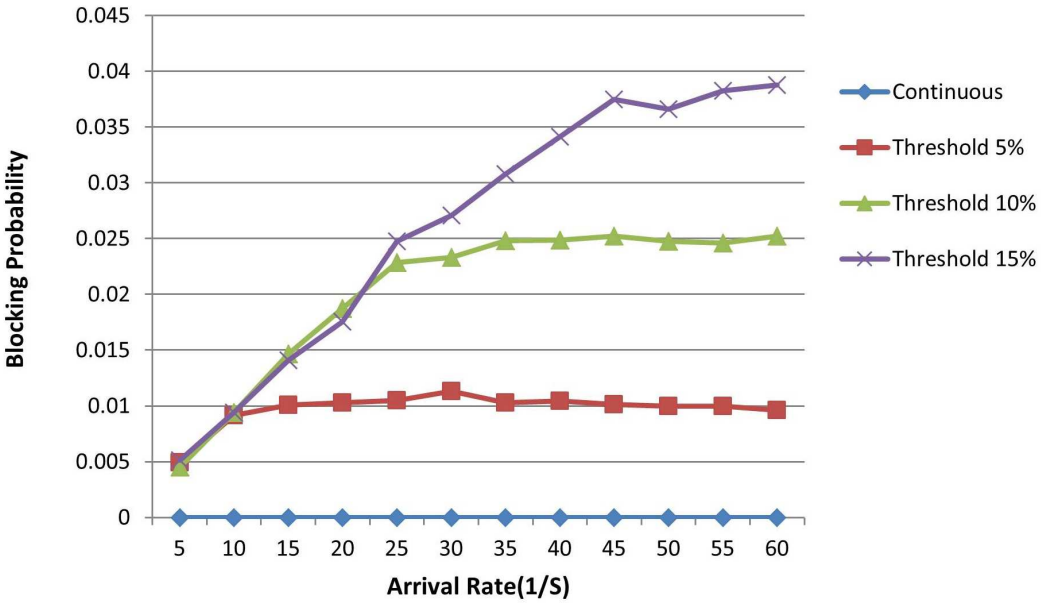


Figure 7.14 Unjustified Update Acceptance for 8-node supply chains and 250 EPC shipped

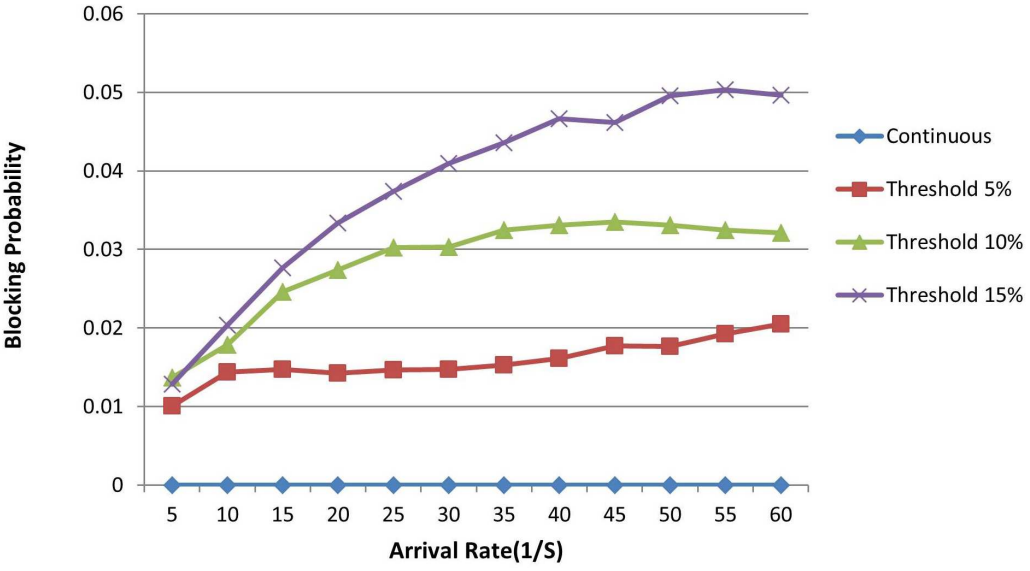


Figure 7.15 Unjustified Update Blocking for 8-node supply chains and 250 EPC shipped

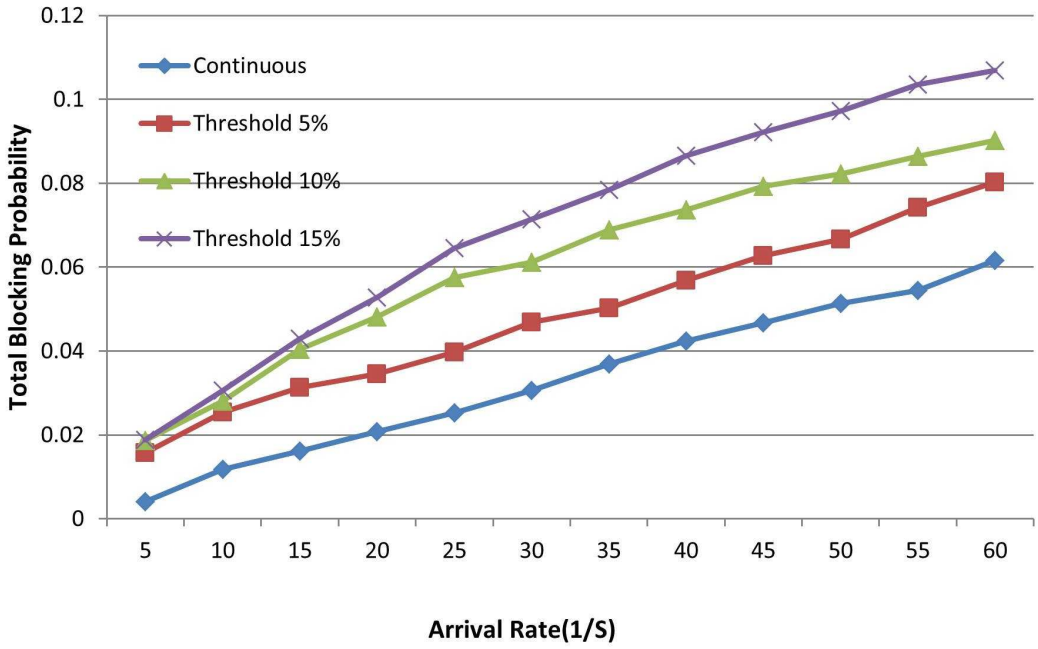


Figure 7.16 Total Blocking for 8-node supply chains and 250 EPC shipped

higher threshold causes less frequent updates leading to higher differences among the routing tables.

$$EU = \frac{\sum_{i=0}^{N_A-1}[\sum_{j=0}^{N_A-1} D_{ij}]}{N^2} \tag{7.10}$$

7.4.2.3 Number of Updates

Figure 7.19 and 7.20 illustrate the number of updates under each arrival rate for 100 and 250 EPCs, respectively. As expected, continuous update leads to huge number of advertisements compared to threshold-based advertisement. As seen in the graph, the threshold range between 5% and 10% introduces the most feasible results for 100 EPCs as the number of advertisements can be decreased by 80% compared to continuous update under the maximum arrival rate that has been tested. The same phenomenon is observed for 250 EPCs where the number of advertisements can be reduced by 90% under the maximum arrival rate that has been tested.

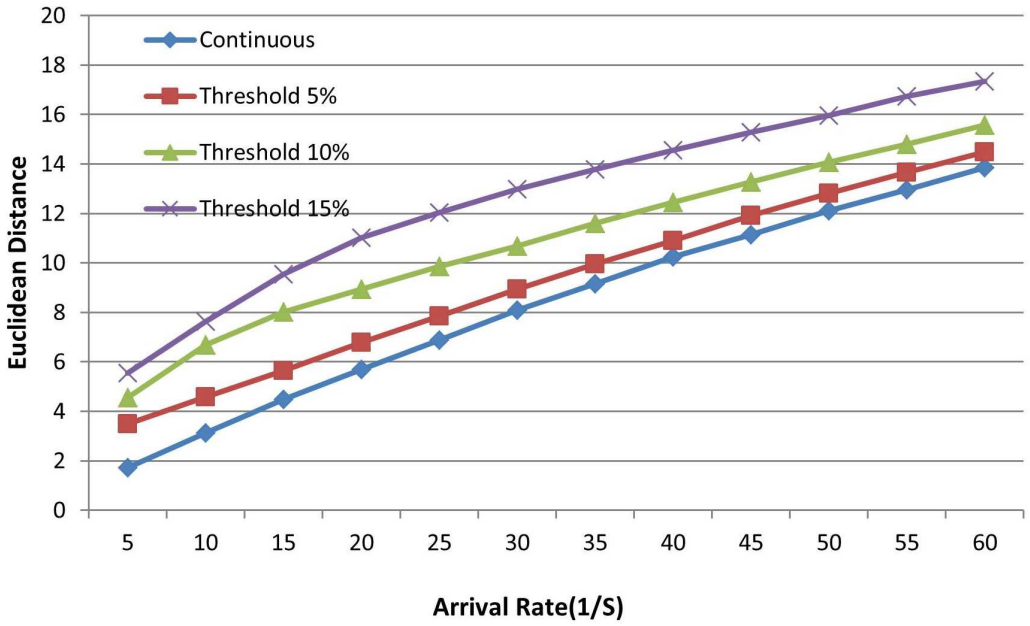


Figure 7.17 Average Euclidean distance between routing tables of 100 EPCs

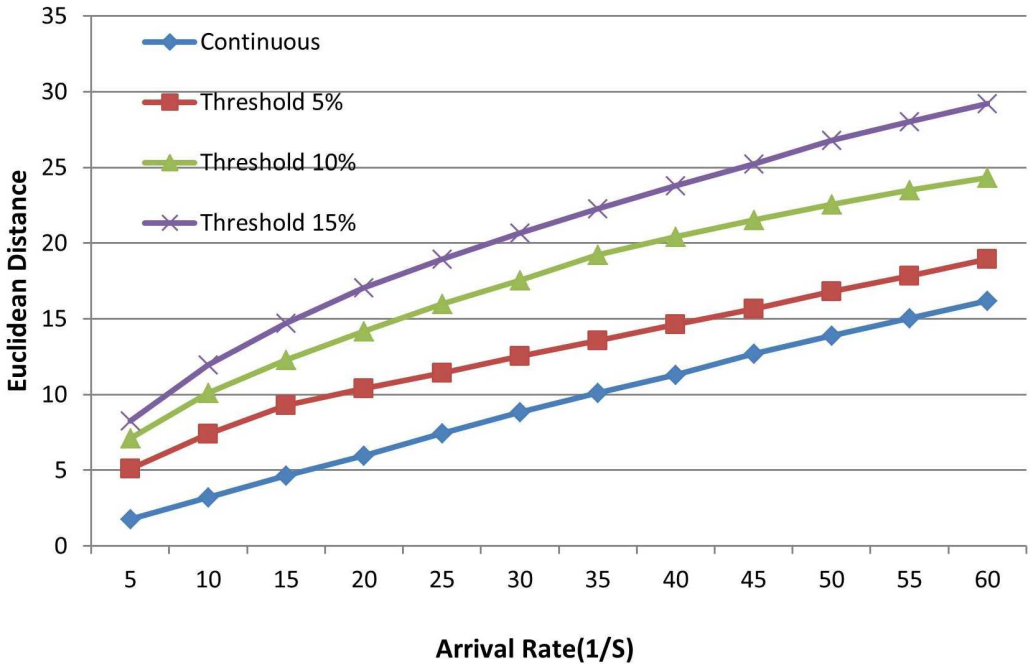


Figure 7.18 Average Euclidean distance between routing tables of 250 EPCs

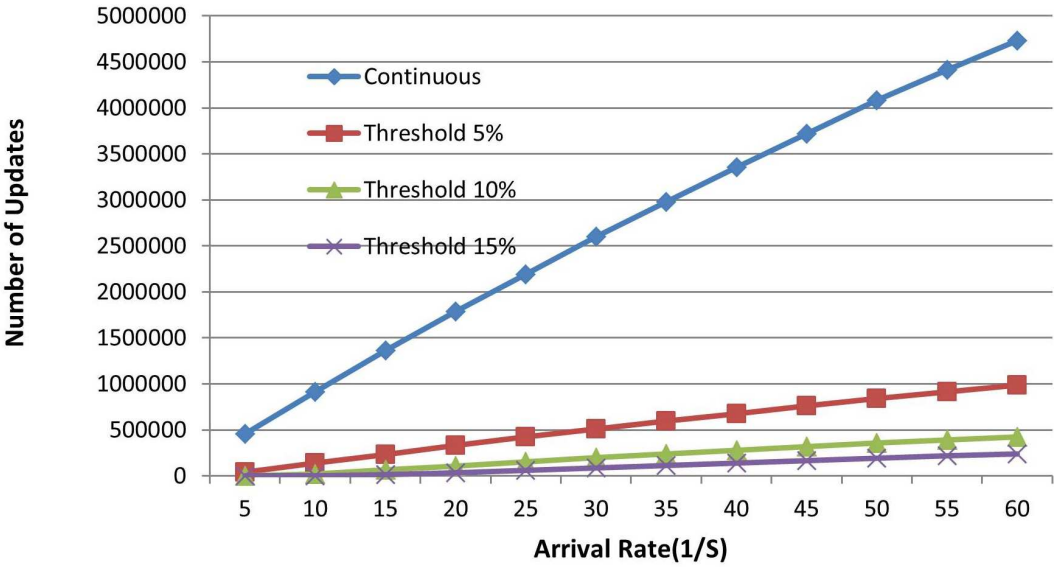


Figure 7.19 Number of updates for 100 EPCs

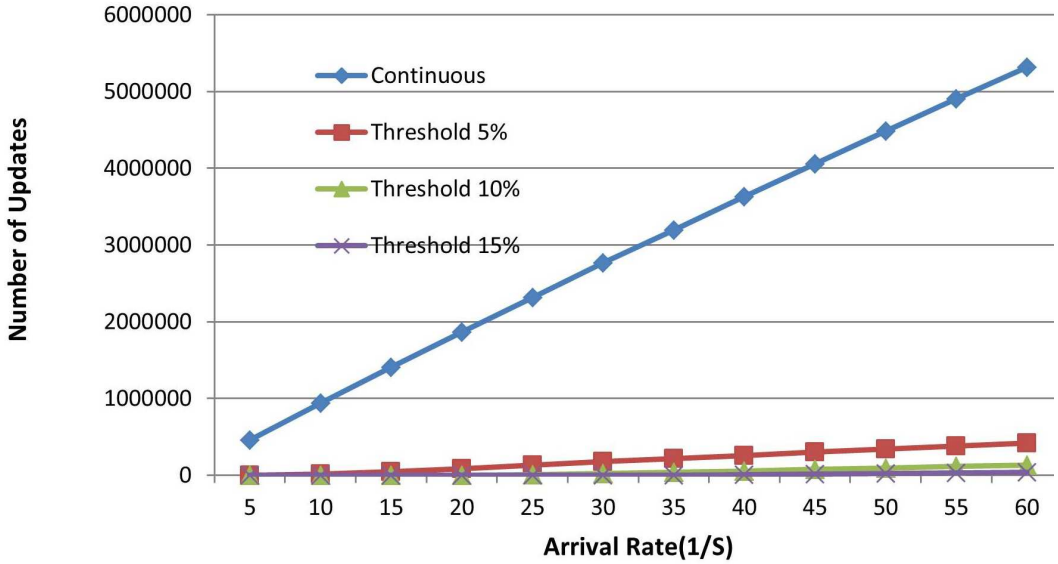


Figure 7.20 Number of updates for 250 EPCs

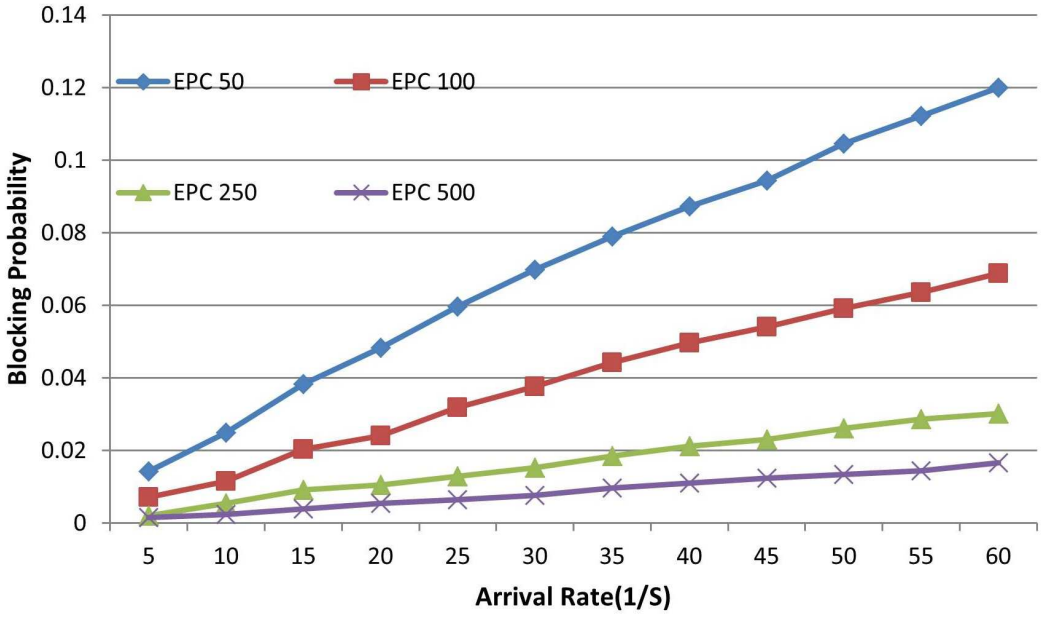


Figure 7.21 Justified Update Blocking due to intra-domain effect under various EPCs

7.4.2.4 Intra-domain blocking effect

In this section, we investigate the impact of the intra-domain blocking for a supply chain of 8 nodes while the intra-domain reservation status of EPCs are not shared with other domains introducing non-zero intra-domain blocking probability. Figure 7.21 illustrates JUB due to intra-domain blocking effect for 50, 100, 250 and 500 EPCs. As seen in the figure, number of EPCs demonstrate inverse relation with the blocking probability since the probability of having an available EPC is higher when the number of EPCs is high.

Figure 7.22 shows the UUA blocking probability which performs similar to JUB. UUA occurs if inter-domain routing table sees an EPC status as available although the EPC is not available in real-time, and its information has not been exchanged yet.

It is worth to mention that UUB is zero as the inter-domain routing tables are up-to-date as continuous update is being done to update inter-domain routing tables; hence any unavailable EPC will not be chosen and hence no UUB. Figure 7.23 shows the total blocking probability, as expected the higher the EPC number the lower the blocking probability due to the reasons that have been explained above.

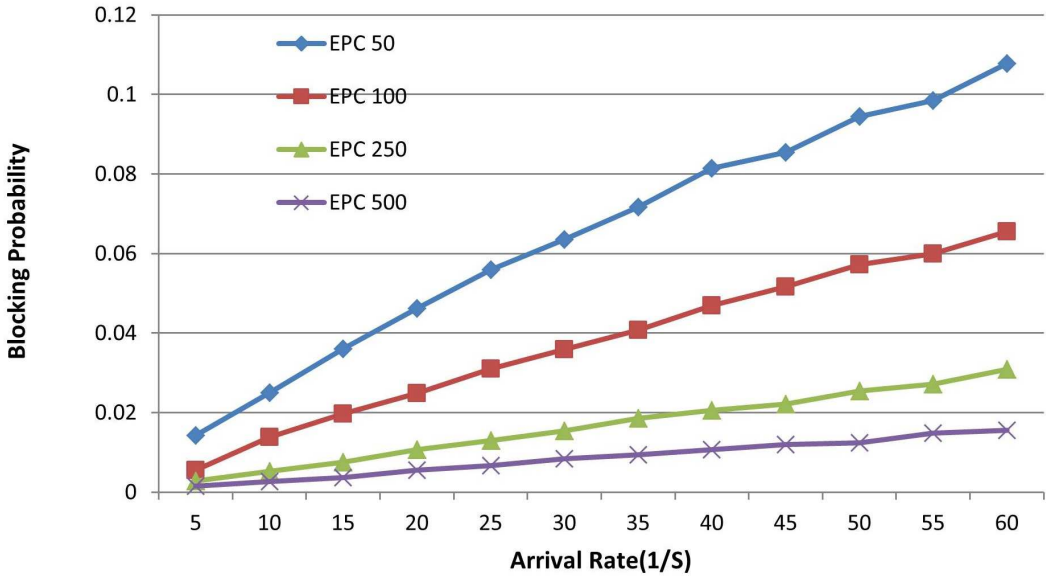


Figure 7.22 Unjustified update acceptance due to intra-domain effect under various EPCs

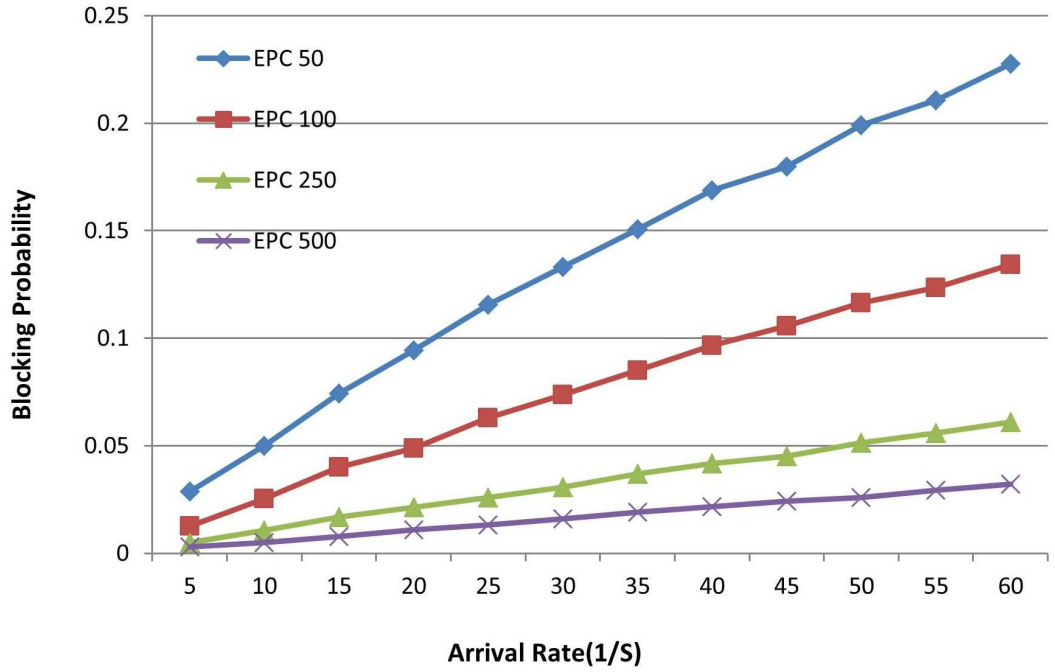


Figure 7.23 Total blocking due to intra-domain effect under various EPCs

7.5 Conclusion

We have proposed an extended architecture of the EPCglobal network, which provides the flexibility to connect multiple supply chains. Furthermore, the extended architecture enables implementation of a distributed discovery service in which all the information about any object can be tracked across multiple supply chains. It also defines an interaction mechanism between intra and inter-domain routing protocols such as EPC-OSPF and EPC-BGP, in order to exchange the status of the EPCs. Inter and Intra-domain interaction is performed without exchanging any vital information about the intra-domain connectivity of the supply chain. We have proposed a threshold-based EPC status update scheme which triggers status advertisement due to excessive number of changes in the EPC status at a node. We have evaluated our proposed architecture under a mesh network of supply chains in terms of update blocking probability. We have compared the continuous update and threshold-based update of the EPCs at the supply chain head considering 5%, 10% and 15% as the threshold values to trigger advertising the changes. Through numerical results, we have shown that there exists a trade-off between the blocking probability and the communication / computing overhead due to the number of advertisements and EPC status updates. We have further shown that threshold-based advertisement can overcome this trade-off if the threshold ratio is set between 5-10 %. We have also evaluated the intra-domain effect on the blocking probability, and we have shown that not advertising the intra domain EPC status will lead to a reasonable amount of blocking.

Chapter 8

Inventory Management as a Service for Supply Chain Stakeholders

Supply Chain refers to all the facilities involved and the activities carried out to make a product item or a service available for final purchase by end customers. Efficient management of these facilities and activities, referred to in literature as Supply Chain Management (SCM), plays a crucial role in organizing enterprise processes, and hence increasing operational efficiency of the enterprise [142]. On a company level, SCM consists of compiling local observed data (e.g., inventory level, order response time, etc.), and data related to the current trend of the market (e.g., positive/negative feedback from end consumers, demand forecasting, etc.), in order to make appropriate decisions, aiming at increasing profit while maintaining an acceptable level of customer satisfaction.

Inventory management is a crucial part in supply chain management. If a company holds too much inventory, it minimizes the likelihood of stock disruption, but in return, the storage cost and the inventory investment cost get higher. If it holds too little inventory, then the stock-out risk gets higher, along with potential loss of customers with good will. Either case can cost big money. This is the reason why one of the most important decisions that companies have to make frequently is related to their replenishment policy. In particular, companies have to decide when would be “the best time” to place an order, and what is “the best quantity” to order. Obviously, the decisions

made have to minimize the likelihood of stock disruption, decreasing shortage costs to the minimum.

In this chapter, we propose a cloud-centric platform for supply chain inventory optimization. The proposed platform gathers data related to product items throughout their life cycle, including that pertaining to the context of their final purchase by end consumers. The collected data is then compiled and analyzed, to help stakeholders in a supply chain manage their inventories in an optimal way. By optimal, we mean that stakeholders aim at balancing two contradicting objectives; namely the objective of avoiding situations in which a product goes out of stock, and that of minimizing the inventory threshold triggering an order of such product.

As a result of the increasing complexity of supply chains and the uncertainty involved in demand forecasting, the goal of minimizing the likelihood of stock disruption is not a simple one to achieve. In this chapter, we propose an integrated cloud-centric platform aiming at offering "real-time and efficient inventory management" as a service for supply chain stakeholders, through compilation of more accurate forecasting of the future demand and other locally observed parameters provided by supply chain stakeholders. The offered service first computes the optimal (smallest) threshold minimizing the likelihood of stock disruption, and then it notifies the corresponding stakeholder to act accordingly.

As illustrated in Figure 8.1, the proposed service differs from the conventional lookup services provided by the Object Naming Service (ONS) [7], and Discovery Services [50, 100, 101], in two aspects:

- First, conventional lookup services, i.e., ONS and Discovery Services, do not involve one of the most influencing actors in a supply chain; i.e., the end consumer. The proposed platform herein offers end consumers secured accounts where they can have access to their purchase history, and provide their feedback and reviews regarding their purchases. As a result, the platform becomes a valuable channel for listening to all the actors in the market.

- Second, while conventional lookup services, i.e., ONS and Discovery Services, are offering only precise traceability of product items in a supply chain, the proposed platform offers also efficient real-time inventory management services to supply chain stakeholders.

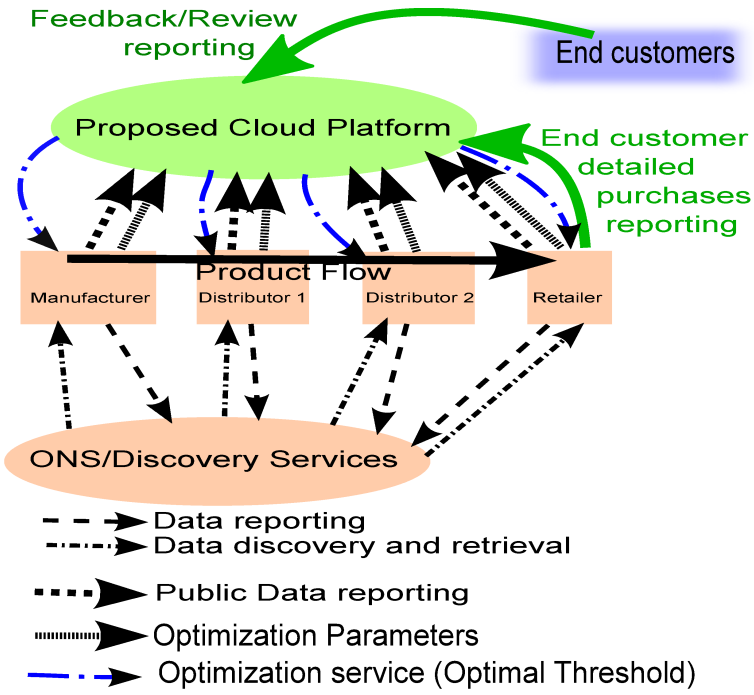


Figure 8.1 The proposed cloud platform vs ONS/Discovery Services

Supply chain management can and should benefit from the unlimited capabilities and resources of cloud computing. Cloud providers offer storage (Infrastructure as a Service) and computing power (Platform as a Service and Software as a Service) in a cost-efficient easy-to-access way. This is enabling companies to out-source significant portions of their Information Technology (IT) systems, such as Enterprise Resource Planning (ERP) systems and Customer Relationship Management (CRM) systems. Email services and social network services, being a success-story cloud-based services, represent an inspiration for our proposed platform.

The objective of this work is to provide supply chain stakeholders with inventory management as a cloud service. This is carried out through efficient replenishment schemes derived collaboratively among stakeholders, on the basis of real-time information flow. This information flow is realized through a cloud-based platform where all data, regarding all transactions (including retail transactions) in all supply chains, is stored and

compiled. Moreover, end consumers can also provide their feedback/review pertaining to a given product category, which would help in improving the forecasting model.

In the remainder of this chapter, the terms “stock-out”, “out of stock” and “stock disruption” are used interchangeably. likewise, the terms “inventory optimization” and “inventory management” are also used interchangeably.

8.0.1 Outline

This chapter is structured as follows. Section 8.1 describes in detail the architecture of our proposed cloud-centric platform. In Section 8.2, we present the inventory optimization service offered by the proposed platform. A detailed description of the inherent probabilistic optimization model is also provided. In Section 8.3, we present the analytical evaluation of our proposed inventory optimization service. Section 8.4 concludes this chapter.

8.1 Proposed Cloud-Centric Platform

In this section, we first introduce the detailed architecture of our proposed cloud-centric platform. which will store all information related to each and every transaction in a supply chain, including transactions carried out by end consumers. Within a given supply chain, such information enables each and every stakeholder to be aware of the order answering rate and the deterioration rate of all its partners in the supply chain. Then, we explain how the proposed platform can be extended into a feedback/review repository, which will certainly help in improving the forecast model.

8.1.1 System Architecture

As illustrated in Figure 8.2, three authorities work together to deploy the platform. Governmental authorities uniquely identify each user, and then associate each user with a unique household identified by its unique address. Financial authorities associate each user with their payment cards; debit, credit and prepaid cards. Business authorities

create an account for each company, identified by its own company prefix [6, 124], so that public EPC data, regarding the product items exchanged in a supply chain, are reported on the platform.

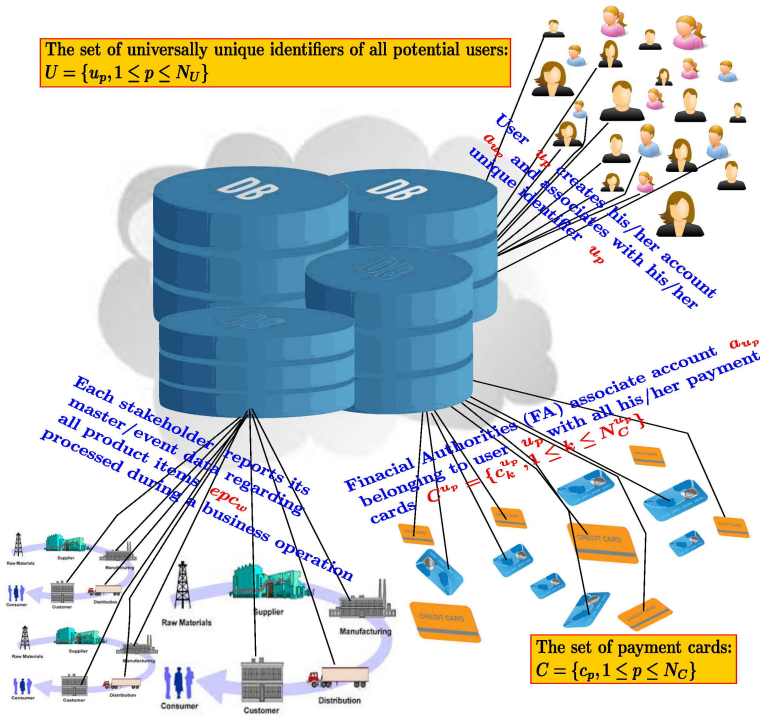


Figure 8.2 Overall architecture of the proposed cloud platform

8.1.2 Consumers' Transactions and Feedback

As mentioned earlier, users choose to opt-in or out of the proposed platform. Once a user u decides to create and activate their account a_u, they start receiving detailed data regarding all their purchases, in the form of a document denoted $d_{c,r,L} = (c, r, timestamp, location, L)$, where r denotes the retailer's identifier, c denotes the payment card identifier used by user u to pay the current purchase, and L denotes a soft copy of the receipt of the current purchase. $L = (e_i, quantity, price\ per\ unit), 1 \leq i \leq N_L$ refers to the list of the EPC codes of all the purchased items, the quantity, the price per unit, etc. N_L denotes the number of product items in the receipt L.

Figure 8.3 illustrates data reported by the retailers to the cloud-based platform regarding purchases of the registered users. It also shows how a given user connects to his/her

account and gets access to all his/her purchases including detailed receipts, location and timestamp of purchase, the selling retailer and the card used for payment.

Users are also invited to provide their feedback, regarding the items they bought, in the platform. This information will enable the platform to score products based on the users' feedback. Such score will give stakeholders hints with regards to the expected business trends relating to each product category, and then act accordingly.

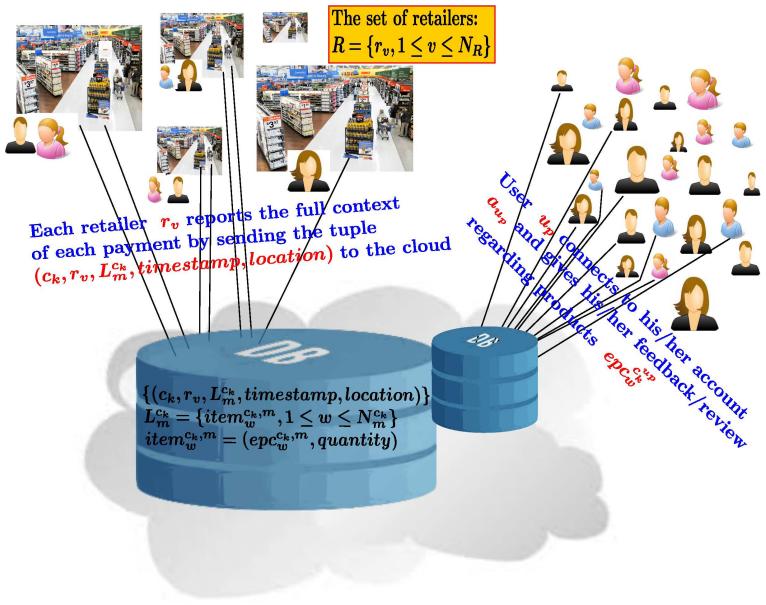


Figure 8.3 Users' reviews/feedback and transactions reported to the platform

8.2 Proposed Inventory Optimization Cloud Service

In this section, we describe the model of our proposed cloud-based service for real-time inventory optimization. This service is offered by the platform presented in Section 8.1, which enables each and every stakeholder to be aware of the order answering rate and the deterioration rate of all its partners within a supply chain. The suggested model targets a global optimization of replenishment policies of all stakeholders within a supply chain, in order to minimize their stock disruption likelihood.

8.2.1 Inventory Optimization Model

In a supply chain, each stakeholder keeps track of its inventory deterioration rate in order to avoid stock-outs. Let us consider a supply chain with N stakeholders $\{S_1, S_1, ..., S_N\}$. Figure 8.4 illustrates the parameters of each stakeholder in the supply chain and its relationship with its immediate supplier S_{i-1} and its N_i immediate customers $\{S_{i+1}^1, S_{i+1}^2, ..., S_{i+1}^{N_i}\}$ in the supply chain.

Stakeholder keeps answering orders initiated by its N_i immediate consumers Upon reaching the predefined threshold th_i, stakeholder S_i initiates an order to its immediate supplier S_{i-1}.

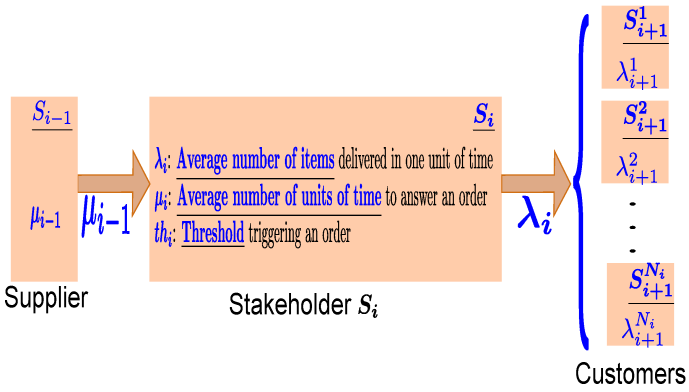


Figure 8.4 Stakeholder parameters and its relationship to its supplier and customer

8.2.1.1 Assumptions

Our probabilistic model assumes the following:

- All the stakeholders define the same unit of time U, such as one week or one month. Their parameters λ_i and $\mu_i, 1 \leq i \leq N$ are defined with regards to one unit of time.
- For each stakeholder S_i, the delivery process is a Poisson process with rate λ_i. This assumption has been widely adopted in related literature [143, 144]. Although in reality a stakeholder would be supplier of more than one customer, as illustrated in Figure 8.4, the assumption that its delivery process is a Poisson process is reasonable. The reason being that the sum of two Poisson processes of

rates λ_1 and λ_2, is a Poisson process with rate $\lambda_1 + \lambda_2$. Hence if Stakeholder S_i answers its N_i consumers orders, each following a Poisson process with rate $\lambda_{i+1}^j, 1 \leq j \leq N_i$, then we can assume that it answers all the orders following a Poisson process with rate $\lambda_i = \sum_{j=1}^{N_i} \lambda_{i+1}^j$.

- A stakeholder S_i has one and only one supplier with regards to a given product category. Although this assumption may seem strong, it is still reasonable in real life since companies tend to do business with suppliers who can answer their orders fully, otherwise they would switch to a more reliable supplier. In [145], a similar assumption has been made to model a supply chain, in a game-theory-based model, with only two agents (players); a retailer and his supplier. Moreover, this assumption simplifies the optimization model.

- Given an order initiated by stakeholder S_i, the inter-arrival time of the corresponding shipment (in terms of U) from its immediate supplier S_{i-1} follows an exponential distribution with arrival rate $1/\mu_{i-1}$; (μ_{i-1} being the average inter-arrival time between two successive order shipments initiated by S_{i-1}). This assumptions has also been used in related literature [144].

8.2.1.2 The Model

We propose a probabilistic model which computes the probability that stakeholder S_i observes a stock-out with regards to a given product category, right after the threshold th_i is reached, and consequently, an order of λ_i units has been directed to the immediate supplier S_{i-1}.

$$
\begin{aligned}
P(SO_i) &= P(th_i \; items \; sold \; before \; shipment \; is \; received) \\
&= P(X_{i,Y_{i-1}} > th_i) \\
&= \mathrm{P}\Big(\Big[(Y_{i-1} \leq \Delta_{th_i}) \; \textbf{\textit{and}} \; (E[X_{i,Y_{i-1}}]_0^{th_i} > th_i) \Big] \\
&\quad \textbf{\textit{or}} \; \Big[(Y_{i-1} > \Delta_{th_i}) \; \textbf{\textit{and}} \; (E[X_{i,Y_{i-1}}]_{th_i}^{+\infty} > th_i) \Big] \Big) \\
&= \mathrm{P}\Big((Y_{i-1} \leq \Delta_{th_i}) \; \textbf{\textit{and}} \; (E[X_{i,Y_{i-1}}]_0^{th_i} > th_i) \Big) \\
&\quad + P\Big((Y_{i-1} > \Delta_{th_i}) \; \textbf{\textit{and}} \; (E[X_{i,Y_{i-1}}]_{th_i}^{+\infty} > th_i) \Big) \\
&= \mathrm{P}_{(Y_{i-1} \leq \Delta_{th_i}) \cdot P(E[X_{i,Y_{i-1}}]_0^{th_i} > th_i)} \\
&\quad + P(Y_{i-1} > \Delta_{th_i}) \cdot P(E[X_{i,Y_{i-1}}]_{th_i}^{+\infty} > th_i)
\end{aligned}
\tag{8.1}
$$

$$
P(Y_{i-1} \leq \Delta_{th_i}) = 1 - e^{-\frac{1}{\mu_{i-1}} \cdot \Delta_{th_i}} \tag{8.2}
$$

$$
P(Y_{i-1} > \Delta_{th_i}) = e^{-\frac{1}{\mu_{i-1}} \cdot \Delta_{th_i}} \tag{8.3}
$$

$$
\begin{aligned}
E[X_{i,Y_{i-1}}]_0^{th_i} &= \int_0^{th_i} x \cdot \frac{1}{\mu_{i-1}} \cdot e^{-\frac{1}{\mu_{i-1}} \cdot x} dx \\
\mathrm{P}_{(E[X_{i,Y_{i-1}}]_0^{th_i} > th_i)} &= 1 - P(E[X_{i,Y_{i-1}}]_0^{th_i} \leq th_i) \\
&= 1 - \sum_{k=0}^{th_i} \frac{\lambda_u^k}{k!} \cdot e^{-\lambda_u} \\
&\quad \text{where } \lambda_u = E[X_{i,Y_{i-1}}]_0^{th_i} \cdot \lambda_i
\end{aligned}
\tag{8.4}
$$

$$
\begin{aligned}
E[X_{i,Y_{i-1}}]_{th_i}^{+\infty} &= \int_{th_i}^{+\infty} x \cdot \frac{1}{\mu_{i-1}} \cdot e^{-\frac{1}{\mu_{i-1}} \cdot x} dx \\
P(E[X_{i,Y_{i-1}}]_{th_i}^{+\infty} > th_i) &= 1 - P(E[X_{i,Y_{i-1}}]_{th_i}^{+\infty} \leq th_i) \\
&= 1 - \sum_{k=0}^{th_i} \frac{\lambda_v^k}{k!} \cdot e^{-\lambda_v} \\
&\text{where } \lambda_v = E[X_{i,Y_{i-1}}]_{th_i}^{+\infty} \cdot \lambda_i
\end{aligned}
\tag{8.5}
$$

Using Equations 8.1, 8.2, 8.3, 8.4 and 8.5, we derive the Equation 8.6 which computes the probability that stakeholder S_i observes a stock-out after the threshold th_i is reached and an order is initiated. This probability corresponds to the probability that all the th_i items left in the inventory are sold out before a shipment is received, in answer to the previously initiated order. Equation 8.6 confirms the dependence of the stock-out probability for stakeholder S_i on the parameters λ_i, μ_{i-1} and th_i.

$$
\begin{aligned}
P(SO_i) = &\left[\left(1-e^{-\frac{1}{\mu_{i-1}} \cdot \Delta_{th_i}}\right) \cdot \left(1-\sum_{k=0}^{th_i} \frac{\lambda_u^k}{k!} \cdot e^{-\lambda_u}\right)\right] \\
&+ \left[\left(e^{-\frac{1}{\mu_{i-1}} \cdot \Delta_{th_i}}\right) \cdot \left(1-\sum_{k=0}^{th_i} \frac{\lambda_v^k}{k!} \cdot e^{-\lambda_v}\right)\right] \\
&\text{where } \lambda_u = \lambda_i \cdot \int_0^{th_i} x \cdot \frac{1}{\mu_{i-1}} \cdot e^{-\frac{1}{\mu_{i-1}} \cdot x} dx \\
&\text{and } \lambda_v = \lambda_i \cdot \int_{th_i}^{+\infty} x \cdot \frac{1}{\mu_{i-1}} \cdot e^{-\frac{1}{\mu_{i-1}} \cdot x} dx
\end{aligned}
\tag{8.6}
$$

8.2.2 Inventory Optimization as a Cloud Service

8.2.2.1 Real-time Statistical Computation of Parameters

Once all the data is gathered in the proposed cloud-based platform described in Section 8.1.1, delivery rates λ_i are computed real-time for each stakeholder S_i, based on the purchase rate at the level of the last stakeholder in the supply chain; which is the retailer. Table 8.1 presents the notations used to compute the purchase rate at the retailer's level $\lambda_{retailer}$, using the cloud platform illustrated in Figure 8.2 and Figure 8.3. $\lambda_{retailer}$ is

computed with regards to a given category y, purchased by end users within a time frame $[t_1, t_2]$, in a location l_1. Equation 8.7 computes this parameters $X_{y,l_1,t_1,t_2} = \lambda_{retailer}$. The cloud platform uses Equation 8.7 and Equation 8.6, along with the parameters introduced by each stakeholder, will enable all the involved stakeholders in a given supply chain (The retailer, the distributor, the manufacturer, etc.) to find the optimal threshold.

Table 8.1 Decision variables used in Parameter Computation

$f(e_i, y)$	$=$	$\begin{cases} 1, & \text{if EPC } e_i \text{ is of product category } y \\ 0, & \text{otherwise} \end{cases}$
$g(e_i, L_m^{c_k})$	$=$	$\begin{cases} 1, & \text{if EPC } e_i \text{ belongs to list } L_m^{c_k} \\ 0, & \text{otherwise} \end{cases}$
$a(t, t_1, t_2)$	$=$	$\begin{cases} 1, & \text{if timestamp } t \text{ comes between } t_1 \text{ and } t_2 \\ 0, & \text{otherwise} \end{cases}$
$b(l, l_1)$	$=$	$\begin{cases} 1, & \text{if location } l \text{ belongs to location } l_1 \\ 0, & \text{otherwise} \end{cases}$

Equation 8.8 illustrates users' average feedback score of a given product category y within a time frame $[t_1, t_2]$, in a location l_1. Such statistic may enable the manufacturer to predict the product future and hence to make strategic decisions with respect to the activity of manufacturing/assembling product y. Moreover, other retailers will be able to efficiently predict optimal quantities of product y to be supplied in the next order, based on the most recent feedback of consumers.

Other supply chain stakeholders, such as logistics providers, will also be able to efficiently reschedule their resources (e.g., distribution trucks and warehousing space) in advance based on the predicted orders from retailers, per location and per time frame.

$$X_{y,l_1,t_1,t_2} = \sum_{p=1}^{N_u} \sum_{k=1}^{N_c^{up}} \sum_{m=1}^{N_L^{c_k}} \sum_{i=1}^{N_{L_m^{c_k^{up}}}} \Big(f(e_i, y) \cdot g(e_i, L_m^{c_k^{up}}) \cdot a(time(L_m^{c_k^{up}}), t_1, t_2) \cdot b(location(L_m^{c_k^{up}}), l_1) \Big) \tag{8.7}$$

$$F_{y,l_1,t_1,t_2} = \frac{\sum_{p=1}^{N_u} \sum_{k=1}^{N_c^{up}} \sum_{m=1}^{N_L^{c_k}} \sum_{i=1}^{N_{L_m^{c_k^{up}}}} \Big(f(e_i, y) \cdot g(e_i, L_m^{c_k^{up}}) \cdot a(time(L_m^{c_k^{up}}), t_1, t_2) \cdot b(location(L_m^{c_k^{up}}), l_1) \cdot fb(u_p, e_i) \Big)}{\sum_{p=1}^{N_u} \sum_{k=1}^{N_c^{up}} \sum_{m=1}^{N_L^{c_k}} \sum_{i=1}^{N_{L_m^{c_k^{up}}}} \Big(f(e_i, y) \cdot g(e_i, L_m^{c_k^{up}}) \cdot a(time(L_m^{c_k^{up}}), t_1, t_2) \cdot b(location(L_m^{c_k^{up}}), l_1) \Big)} \tag{8.8}$$

8.2.2.2 Real-time Probabilistic Computation of the Inventory Threshold

For each stakeholder S_i, the proposed cloud platform, illustrated in Figure 8.2, computes statistically the parameters λ_i and μ_{i-1}. Then, it computes the stock-out probability using Equation 8.6 for a set of possible values of the inventory threshold until it finds the inventory threshold for which the stock-out probability is less than a given probability threshold p_i, referring to the stock-out probability threshold that stakeholder S_i is willing to accept. The up-to-date optimal inventory threshold th_i is then conveyed

by the cloud platform to stakeholder S_i, in order to update its parameters. This real-time update of the inventory threshold would enable stakeholder S_i to avoid a potential stock-out.

8.3 Results

For the definitions of the notations used in this section, the reader shall refer to Section 8.2.1.2 and Table 8.1. Figure 8.5 and Figure 8.6 illustrate the stock-out probability in terms of the threshold triggering an order request th_i, for different values of the deterioration rate λ_i, and for the replenishment rates $\mu_{i-1} = 0.8$ and $\mu_{i-1} = 1.6$ respectively. On the x-axis, the inventory threshold triggering an order request th_i is given. Both graphs show that when the deterioration rate λ_i increases while the same inventory threshold th_i is used, the stock-out probability increases Exponentially. For example, when λ_i changes from 60 to 70 in Figure 8.5 (increase of less than 17%), while keeping the same optimal threshold $th_i = 51$ computed for $\lambda_i = 60, \mu_{i-1} = 0.8$, then the stock-out probability raises from a value ≤ 0.001 to over 0.05. The probability increase is of more than 4900%.

Figure 8.7 and Figure 8.8 illustrate the stock-out probability in terms of the deterioration rate λ_i for different values of the inventory threshold triggering an order request th_i, and for the replenishment rates $\mu_{i-1} = 0.8$ and $\mu_{i-1} = 1.6$ respectively. On the x-axis, the deterioration rate λ_i is given. Both graphs show that when th_i decreases while the same λ_i is used, the stock-out probability increases exponentially. For example, when th_i changes from 36 to 28 in Figure 8.7 (decrease of less than 29%), while keeping the same optimal $\lambda_i = 40$ computed for $th_i = 36, \mu_{i-1} = 0.8$, then the stock-out probability raises from a value ≤ 0.001 to over 0.1. The probability increase is of more than 9900%.

The results show how crucial it is for a supply chain stakeholder S_i to detect real-time changes in its own deterioration rate or in its immediate supplier replenishment rate μ_{i-1}. Real-time detection of those changes would enable stakeholders to change their replenishment policy as soon as possible. Such real-time reaction allows stakeholders to avoid abrupt and sharp stock disruption, or at least to alleviate its effects.

8.4 Conclusion

In this chapter, we have proposed an integrated cloud platform offering real-time and accurate inventory management as a service for supply chain stakeholders. Using the parameters loaded by each stakeholder into the platform, and using the data collected by the platform, the optimization service computes, via a probabilistic model, the optimal "inventory threshold", to be considered by supply chain stakeholders in their replenishment policy. By optimal we mean that it minimizes the stock disruption likelihood, while minimizing the allocated resources. Analytical results have illustrated the efficiency of the inventory optimization service offered by our proposed cloud-centric platform.

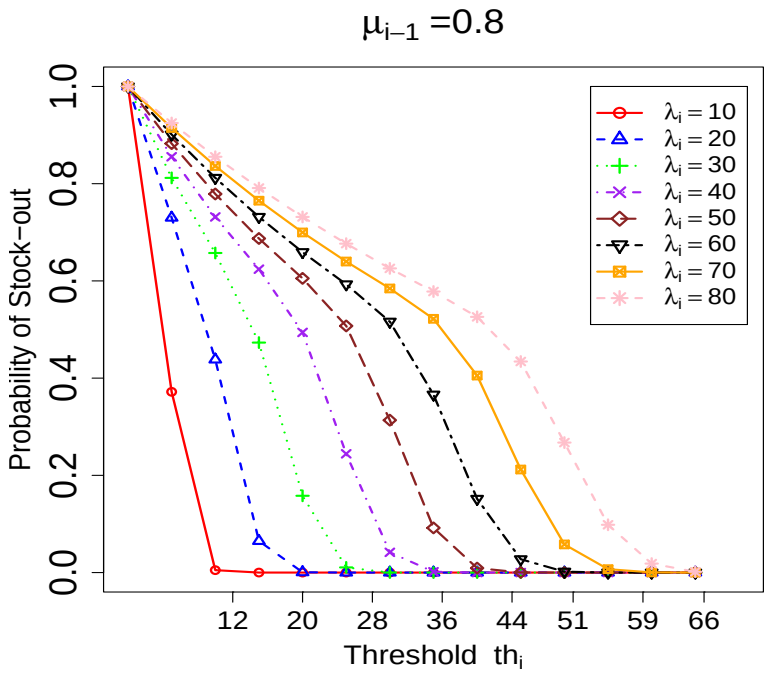


Figure 8.5 Probability of stock-out for stakeholder S_i, defined by Equation 8.6, in terms of the threshold value th_i, for $\mu_{i-1} = 0.8$ and different values of the delivery rate λ_i (refer to the List of Symbols and Notations at the beginning of this thesis for the definition of the notations)

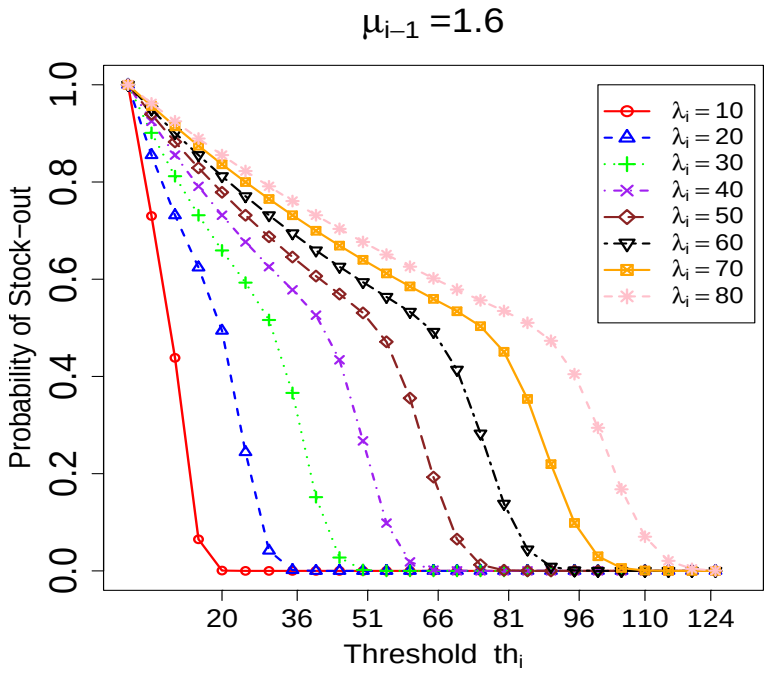


Figure 8.6 Probability of stock-out for stakeholder S_i, defined by Equation 8.6, in terms of the threshold value th_i, for $\mu_{i-1} = 1.6$ and different values of the delivery rate λ_i (refer to the List of Symbols and Notations at the beginning of this thesis for the definition of the notations)

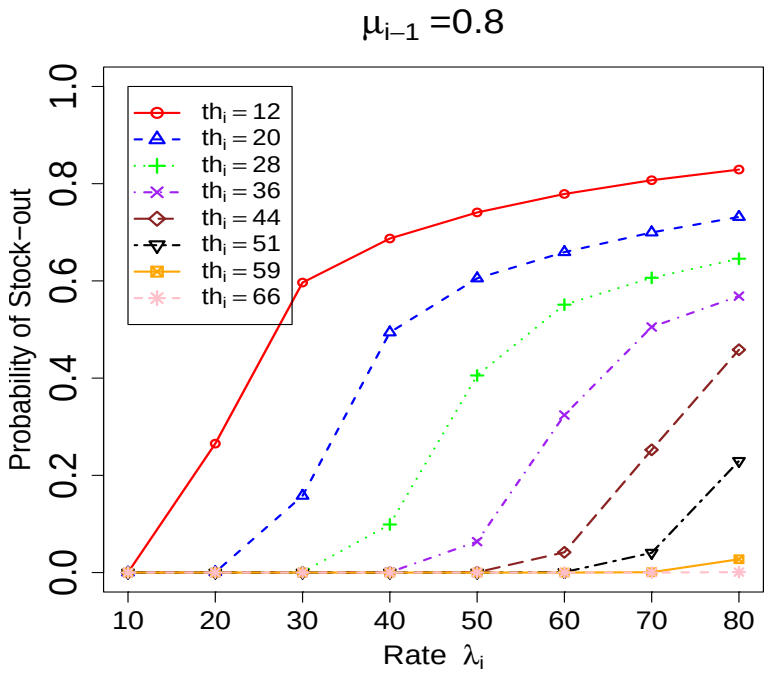


Figure 8.7 Probability of stock-out for stakeholder S_i, defined by Equation 8.6, in terms of the delivery rate λ_i, for $\mu_{i-1} = 0.8$ and different values of the threshold th_i (refer to the List of Symbols and Notations at the beginning of this thesis for the definition of the notations)

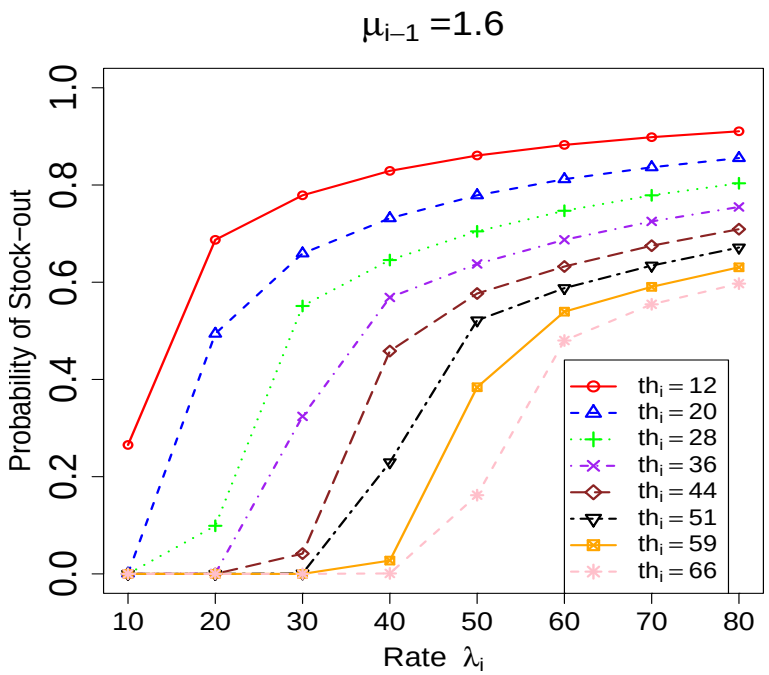


Figure 8.8 Probability of stock-out for stakeholder S_i, defined by Equation 8.6, in terms of the delivery rate λ_i, for $\mu_{i-1} = 1.6$ and different values of the threshold th_i (refer to the List of Symbols and Notations at the beginning of this thesis for the definition of the notations)

Chapter 9

Conclusions and Future Work

9.1 Conclusions

We conclude this thesis with a brief review of the problems that have been or may be encountered in implementing Discovery Services in an IoT environment, with a perspective on the future business infrastructure, represented by the EPCglobal Network. We also provide an overview of our proposed solutions to those problems, each followed by a summary of its performance evaluation. Two major problems have been addressed in this thesis. The first problem resolves around the security of Discovery Services. The second problem relates to the efficiency of those Discovery Services in terms of their scalability and query responsiveness. Addressing these two problems is extremely vital for the business infrastructure, given the sensitive nature of the data involved and given the trend of its incessant growth. Like all research work, this effort is merely a milestone in a never-ending endeavor.

Data exchanged amongst supply chain stakeholders is very sensitive. Unauthorized access to such data may have a fatal effect on the competitiveness of the targeted organization, and hence on its survival. Man-in-the-Middle attacks and replay attacks are examples of security attacks aiming at unauthorized access to private data. DoS attacks are security attacks in which one or many compromised systems are used to target a single system, the goal being to flood its resources in an attempt to disrupt the services it is offering. We have proposed three security schemes in order to mitigate these attacks.

The proposed security schemes aim at detecting suspicious lookup queries considering the originator's business habits, such as the frequency of this type of queries and the average number of items processed by the originator in a predefined period of time. Detection of suspicious queries is inspired by the well-studied "Anomaly Detection" approach. Upon receiving a lookup query, two scores are computed; one representing the probability that this query is "safe" and the other representing the probability that it is "risky". The state with the highest score is considered the predicted state. We have conducted extensive experiments to evaluate our proposed security schemes. The results have shown that our first proposed scheme, consolidating two or more features to compute the states' scores (PSA algorithm and PPA algorithm), perform better than a reference algorithm which takes only one feature into consideration at a time. The performance have been measured in terms of the detection rate, the false alarm rate and the accuracy. The results have also shown that our CHMM-based security scheme enhances even better the detection rate, the false alarm rate and the accuracy. We have observed that, in the case the two states (safe and risky) are distinguishable enough, the CHMM-based security scheme makes great use of its learning capabilities of data sequentiality to perform much better than the reference algorithm. Limitations of our proposed schemes lay in the assumptions made to simplify the models. These assumptions need to be loosened or even canceled.

From an architecture standpoint, although a number of distributed architectures have been proposed in the literature, both their scalability and their responsiveness remain vulnerable, mainly because of their reliance on non-clustered or poorly-clustered distributed architectures. We have introduced and compared two architectures for distributed and scalable Discovery Services; a Flat Distributed Architecture (FDA), which speaks for the majority of the distributed architectures that have been proposed in the literature so far, and a Hierarchical Distributed Architecture (HDA) representing the core contribution of our work. The performance evaluation has been performed both analytically and experimentally. The experimental performance evaluation has been carried out both through a simulation on Planetsim and an emulation on Planetlab. The results have shown that HDA is more scalable than FDA, and its lookup responsiveness is much better. Network scalability has been evaluated in terms the number of hops required per lookup query, while lookup responsiveness has been evaluated in terms the time required per lookup query. The assumption made regarding hierarchical naming

of geographical regions is reasonable given that it is already partially satisfied in the EPCglobal framework.

We have also proposed an extended architecture for the EPCglobal network which has the ability to connect multiple supply chains forming a mesh network. The proposed architecture runs a distributed Discovery Service enabling users to track any object regardless of its location. Furthermore, it eliminates the need of exposing vital information about intra-domain connectivity of the supply chain. Numerical results have shown a trade-off between blocking probability and communication computing overhead due to EPC update messages. We have shown that this trade-off can be addressed by selection of appropriate thresholds in EPC status advertisement. We have also provided a queuing analysis of the blocking probability for the EPC update requests of the EPCglobal network, we have formulated the blocking probability in terms of the threshold and the number of EPCs involved. We have shown that the queuing analysis results match our simulation results.

Finally, we have looked at the extensibility of Discovery Services into cloud-centric services capable of providing supply chain stakeholders with real time and accurate information with regards to their inventories. To achieve this goal, we have proposed an integrated cloud platform offering real-time and accurate inventory management as a service for supply chain stakeholders. Using the parameters loaded by each stakeholder into the platform, and using the data collected by the platform, the optimization service computes the optimal "inventory threshold", to be considered by supply chain stakeholders in their replenishment policy. By optimal we mean that the proposed replenishment threshold minimizes the stock disruption likelihood, while minimizing the allocated resources. Analytical results have confirmed the efficiency of the inventory optimization service offered by our proposed cloud-centric platform in allowing stakeholders to avoid abrupt and sharp stock disruption, or at least to alleviate its effects. Limitations of our proposed cloud-based framework lay in the assumptions made to simplify the model. These assumptions need to be loosened or even canceled.

9.2 Future Work

In this thesis we spot light on some of the issues resolving around deployment of efficient and secured Discovery Services in the EPCglobal Network. The expected widespread use of the lookup service in the future business infrastructure brings up multiple challenges related to the unprecedented and incessantly growing amount of data to be handled, the exponentially increasing frequency at which the service is being called, and also the exceptionally large flow of highly sensitive EPC information. These challenges define potential perspectives for extending our work.

From a security standpoint, our work on query state inference can be extended in three different ways. First, although the assumptions made to simplify our models are quite reasonable, it would be interesting to look into loosening, or even canceling, some of those assumptions and then redesign the inference algorithms within an even more realistic settings. Second, other anomaly detection techniques, based on different models such as neural networks and Bayesian networks, can be tried and assessed. Third, it would be of extreme benefit to try those security schemes with genuine datasets collected by genuine supply chain stakeholders in a realistic settings. This would put the proposed security schemes to the ultimate test which would provide final insights with regards to their assessment, and hence to their usability in the future business infrastructure.

Our work on the architecture design aspect of Discovery Services can also be extended in at least two ways. First, in our proposed hierarchical distributed architecture, the number of messages exchanged needs to be minimized to reduce overall network overhead. One way to achieve this is to piggyback multiple messages originated by a given P2P network in one inter-level exchanged message. Second, design of Discovery Services needs to be more cloud-oriented in order to benefit from the unlimited capabilities and resources of cloud computing. Email services and social network services, being success-story cloud-based services, represent an inspiration for such cloud-oriented design.

References

[1] European Commission and EPoSS working Group RFID, “Internet of things 2020. a roadmap for the future,” September 2008.

[2] E. Borgia, “The internet of things vision: Key features, applications and open issues,” *in Elsevier Computer Communications Journal*, vol. 54, no. 1, pp. 1–31, December 2014.

[3] L. Atzori, A. Iera and G. Morabito, “The internet of things: A survey,” *in Elsevier Computer Networks Journal*, vol. 54, no. 15, pp. 2787–2805, October 2010.

[4] G. Shen and B. Liu, “The visions, technologies, applications and security issues of internet of things,” in *Proceedings of the 2011 International Conference on E-Business and E-Government (ICEE ’11)*, pp. 1–4, Shanghai, China, 2011.

[5] H. Zhang and L. Zhu, “Internet of things: Key technology, architecture and challenging problems,” in *Proceedings of the 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE ’11)*, vol. 4, pp. 507–512, Shanghai, China, 2011.

[6] K. Traub et al., “The EPCglobal architecture framework,” http://www.gs1.org/gsmp/kc/epcglobal/architecture, ver. 1.5. [Accessed September 25, 2014].

[7] GS1 EPCglobal, “EPCglobal object name service (ONS),” http://www.gs1.org/gsmp/kc/epcglobal/ons, January 2013, ver. 2.0.1. [Accessed September 25, 2014].

[8] D. Atkins and R. Austein, “Threat analysis of the domain name system (DNS),” IETF, Request for Comments - RFC 3833, August 2004.

[9] B. Fabian and O. Günther, "Distributed ONS and its impact on privacy," in *Proceedings of the 2007 IEEE International Conference on Communications (ICC '07)*, Glasgow, UK, 2007.

[10] S. Evdokimov, B. Fabian and O. Günther, "Multipolarity for the object naming service," in *Proceedings of the 1st international conference on the Internet of Things (IOT '08)*, pp. 1–18, Zurich, Switzerland, 2008.

[11] B. Fabian, T. Ermakova and C. Muller, "SHARDIS: A privacy-enhanced discovery service for RFID-based product information," *in IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 707–718, August 2012.

[12] S. Chalasani and R. Boppana, "Data architectures for RFID transactions," *in IEEE Transactions on Industrial Informatics*, vol. 3, no. 3, pp. 246–257, August 2007.

[13] T. Staake, F. Thiesse and E. Fleisch, "Extending the EPC network: the potential of RFID in anti-counterfeiting," in *Proceedings of the 2005 ACM symposium on Applied computing (SAC '05)*, pp. 1607–1612, New York, NY, USA, 2005.

[14] M. Schapranow, A. Zeier, F. Leupold and T. Schubotz, "Securing EPCglobal object name service - Privacy enhancements for anti-counterfeiting," in *Proceedings of the second International Conference on Intelligent Systems, Modelling and Simulation (ISMS '11)*, pp. 332–337, Washington, DC, USA, 2011.

[15] Y. Cao, D. Wang and H. Sheng, "PTSP: A lightweight EPCDS platform to deploy traceable services between supply-chain applications," in *Proceedings of the 1st Annual Conference on RFID Eurasia*, pp. 1–5, Istanbul, 2007.

[16] M. D. D. Amorim, S. Fdida, N. Mitton, L. Schmidt and D. Simplot-Ryl, "Distributed planetary object name service: Issues and design principles," INRIA, Research Report 7042, September 2009.

[17] G. Yu, X. Du and S. Zheng, "A distributed supply chain discovery service," in *Proceedings of the 2011 International Conference on Computational and Information Sciences (ICCIS '11)*, pp. 445–448, Chengdu, China, 2011.

[18] B. Fabian, “Implementing secure P2P-ONS,” in *Proceedings of the 2009 IEEE International Conference on Communications (ICC ’09)*, pp. 1–5, Dresden, Germany, 2009.

[19] B. Fabian and O. Günther, “Security challenges of the EPCglobal network,” *in Communications of the ACM*, vol. 52, no. 7, pp. 121–125, July 2009.

[20] J. Müller, J. Oberst, S. Wehrmeyer, J. Witt, A. Zeier and H. Plattner, “An aggregating discovery service for the EPCglobal network,” in *Proceedings of the 2010 Hawaii International Conference on System Sciences (HICSS ’10)*, pp. 1–9, Washington, DC, USA, 2010.

[21] Y.-J. Gong, M. Shen, J. Zhang, O. Kaynak, W.-N. Chen and Z.-H. Zhan, “Optimizing RFID network planning by using a particle swarm optimization algorithm with redundant reader elimination,” *in IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 900–912, November 2012.

[22] B. Wang and M. Ma, “A server independent authentication scheme for RFID systems,” *in IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 689–696, August 2012.

[23] O. Ondemir, M. Ilgin and S. Gupta, “Optimal end-of-life management in closed-loop supply chains using RFID and sensors,” *in IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 719–728, August 2012.

[24] H. Ning, H. Liu and L. Yang, “Cyberentity security in the internet of things,” *in IEEE Computer Journal*, vol. 46, no. 4, pp. 46–53, April 2013.

[25] G. Hurlburt, J. Voas and K. Miller, “The internet of things: A reality check,” *in IT Professional*, vol. 14, no. 3, pp. 56–59, May 2012.

[26] GS1 EPCglobal, “GS1 EPC tag data standard,” http://www.gs1.org/gsmp/kc/epcglobal/tds, May 2013, ver. 1.7. [Accessed September 25, 2014].

[27] Z. Xing-liang and X. Shi-lian, “Application research of EPC network in reverse logistics,” in *Proceedings of the 2012 International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII ’12)*, pp. 369–372, Sanya, 2012.

[28] W.-C. Ku and Y.-H. Chen, “Improvement of EPC-C1G2 RFID authentication protocols,” in *Proceedings of the 1st IEEE International Conference on Communications in China (ICCC ’12)*, pp. 226–230, Beijing, 2012.

[29] GS1 EPCglobal, “GS1 EPCglobal tag data translation,” http://www.gs1.org/gsmp/kc/epcglobal/tdt, October 2011, ver. 1.6. [Accessed September 25, 2014].

[30] GS1 EPCglobal, “EPC information services (EPCIS) specification,” http://www.gs1.org/gsmp/kc/epcglobal/epcis, September 2007, [Accessed September 25, 2014].

[31] GS1 EPCglobal, “Discovery services standard (in development),” http://www.gs1.org/gsmp/kc/epcglobal/discovery, [Accessed September 25, 2014].

[32] R. Itsuki and A. Fujita, “Consideration for efficient RFID information retrieval in traceability system,” in *Procedings of the IEEE Conference on Emerging Technologies Factory Automation (ETFA’09)*, pp. 1–4, Mallorca, 2009.

[33] A. Oram, *Peer-to-peer: Harnessing the Benefits of a Disruptive Technology*. O’Reilly, 2001.

[34] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM ’01)*, pp. 149–160, New York, NY, USA, 2001.

[35] H. Deng and Y. Chen, “Realization of RFID resolution service using ”EPCIS Directory + PML” mode with structured cache,” in *Proceedings of the 2010 International Conference on Supply Chain Management and Information Systems (SCMIS ’10)*, pp. 1–5, Hong Kong, 2010.

[36] D. Huang, M. Verma, A. Ramachandran and Z. Zhou, “A distributed ePedigree architecture,” in *Proccesdings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS ’07)*, pp. 220–230, Sedona, AZ, 2007.

[37] Y. S. Kang, I. H. Park, J. Rhee and Y. H. Lee, "MongoDB-based repository design for IoT-generated RFID/sensor big data," *in IEEE Sensors Journal*, vol. 16, no. 2, pp. 485–497, January 2016.

[38] T. D. Le, S. H. Kim, M. H. Nguyen, D. Kim, S. Y. Shin, K. E. Lee and R. da Rosa Righi, "EPC information services with No-SQL datastore for the internet of things," in *Proceedings of the 2014 IEEE International Conference on RFID (RFID '14)*, pp. 47–54, Orlando, FL, 2014.

[39] G. Jianli, "Application research of push service in EPCIS system," in *Proceedings of the 2015 International Conference on Measuring Technology and Mechatronics Automation*, pp. 196–201, Nanchang, 2015.

[40] M. Mealling and R. Daniel, "The naming authority pointer (NAPTR) DNS resource record," IETF, Request for Comments - RFC 2915, September 2000.

[41] G. Lee, J. Shin, D. Park and H. Kwon, "Discovery architecture for the tracing of products in the EPCglobal network," in *Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '08)*, pp. 553–558, Shanghai, China, 2008.

[42] O. Kodym, F. Benes and J. Svub, "EPC application framework in the context of internet of things," in *Proceedings of the 2015 International Carpathian Control Conference (ICCC '15)*, pp. 214–219, Szilvasvarad, 2015.

[43] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, "DNS security introduction and requirements," IETF, Request for Comments - RFC 4033, Mar. 2005.

[44] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, "Resource records for the DNS security extensions," IETF, Request for Comments - RFC 4034, Mar. 2005.

[45] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, "Protocol modifications for the DNS security extensions," IETF, Request for Comments - RFC 4035, Mar. 2005.

[46] A. Friedlander, A. Mankin, W. D. Maughan and S. D. Crocker, "DNSSEC: A protocol toward securing the internet infrastructure," *in Communications of the ACM*, vol. 50, no. 6, pp. 44–50, June 2007.

[47] S. Ariyapperuma and C. J. Mitchell, "Security vulnerabilities in DNS and DNSSEC," in *Proceedings of the second International Conference on Availability, Reliability and Security (ARES '07)*, pp. 335–342, Washington, DC, USA, 2007.

[48] B. Fabian, O. Günther and S. Spiekermann, "Security analysis of the object name service," in *Proceedings of the 2005 International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous*, pp. 71–76, Santorini Island, Greece, 2005.

[49] J. Sun, H. Zhao, H. Xiao and G. Hu, "Lightweight public key infrastructure and service relation model for designing a trustworthy ONS," in *Proceedings of the 2009 IEEE/ACIS International Conference on Computer and Information Science (ICIS '09)*, pp. 295–300, Shanghai, China, 2009.

[50] M. Young, "Extensible supply-chain discovery service concepts (draft 04)," IETF, Internet Draft, August 2008.

[51] C. Bo and W. Dong, "Design and implementation of efficient distributed name service in EPC network," in *Proceedings of the 2013 IEEE Conference on Anthology*, pp. 1–6, China, 2013.

[52] University of Cambridge, AT4 Wireless, BT Research and SAP Research, "High level design for discovery services, BRIDGE Project," http://www.bridge-project.eu/, August 2007, [Accessed November 1st, 2015].

[53] M. ngel Guijarro, G. Arrebola, J. J. Cantero, E. Garca, F. J. Nez, J. Baos, M. Harrison, C. Condea, H. Casalprim and J. Torrecilla, "Working prototype of serial-level lookup service, BRIDGE Project," http://www.bridge-project.eu/, February 2008, [Accessed November 1st, 2015].

[54] M. Lorenz, J. Mueller, M.-P. Schapranow, A. Zeier and H. Plattner, "A distributed EPC discovery service based on peer-to-peer technology," in *Proceedings of the*

2011 European Workshop on Smart Objects: Systems, Technologies and Applications; RFID SysTech 2011, pp. 1–7, Dresden, Germany, 2011.

[55] U. Barchetti, A. Bucciero, M. De Blasi, L. Mainetti and L. Patrono, "Implementation and testing of an EPCglobal-aware discovery service for item-level traceability," in *Proceedings of the 2009 International Conference on Ultra Modern Telecommunications (ICUMT '09)*, pp. 1–8, St. Petersburg, 2009.

[56] "Fosstrak: Open source RFID platform," http://fosstrak.github.io/, April. 2007, [Accessed November 1st, 2015].

[57] P. Gaj, J. Jasperneite and M. Felser, "Computer communication within industrial distributed environment - a survey," *in IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 182–189, February 2013.

[58] A. Al-Fagih, F. Al-Turjman, W. Alsalih and H. Hassanein, "A priced public sensing framework for heterogeneous IoT architectures," *in IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 133–147, June 2013.

[59] F. Li, M. Vogler, S. Sehic, S. Qanbari, S. Nastic, H.-L. Truong and S. Dustdar, "Web-scale service delivery for smart cities," *in IEEE Internet Computing Journal*, vol. 17, no. 4, pp. 78–83, July 2013.

[60] E. Karmouch and A. Nayak, "A distributed constraint satisfaction problem approach to virtual device composition," *in IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 1997–2009, November 2012.

[61] S. Beier, T. Grandison, K. Kailing and R. Rantzau, "Discovery services-enabling RFID traceability in EPCglobal networks." in *Proceedings of the 2006 International Conference on Management of Data (COMAD '06)*, pp. 214–217, Delhi, India, 2006.

[62] J. Mitsugi, Y. Sato, M. Ozawa and S. Suzuki, "An integrated device and service discovery with UPnP and ONS to facilitate the composition of smart home applications," in *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT '14)*, pp. 400–404, Seoul, 2014.

[63] Open Connectivity Foundation (OCF), "UPnP device architecture 2.0," http://www.upnp.org/specs/arch, February 2015, ver. 2.0 [Accessed May. 16, 2016].

[64] S. Balakrichenan, A. Kin-Foo and M. Souissi, "Qualitative evaluation of a proposed federated object naming service architecture," in *Proceedings of the 2011 International Conference on Internet of Things and International Conference on Cyber, Physical and Social Computing (iThings/CPSCom '11)*, pp. 726–732, Dalian, 2011.

[65] H.-C. Hsiao and C.-W. Chang, "A symmetric load balancing algorithm with performance guarantees for distributed hash tables," *in IEEE Transactions on Computers*, vol. 62, no. 4, pp. 662–675, April 2013.

[66] P. Manzanares-Lopez, J. P. Muñoz Gea, J. Malgosa-Sanahuja and J. C. Sanchez-Aarnoutse, "An efficient distributed discovery service for EPCglobal network in nested package scenarios," *In Elsevier Network and Computer Applications Journal*, vol. 34, no. 3, pp. 925–937, May 2011.

[67] A. Talevski, E. Chang and T. S. Dillon, "Reconfigurable web service integration in the extended logistics enterprise," *in IEEE Transactions on Industrial Informatics*, vol. 1, no. 2, pp. 74–84, May 2005.

[68] H. K. Chan and F. T. S. Chan, "Early order completion contract approach to minimize the impact of demand uncertainty on supply chains," *in IEEE Transactions on Industrial Informatics*, vol. 2, no. 1, pp. 48–58, February 2006.

[69] M. Li, Z. Zhu and G. Chen, "A scalable and high-efficiency discovery service using a new storage," in *Proceedings of the 2013 IEEE Conference on Computer Software and Applications (COMPSAC '13)*, pp. 754–759, Kyoto, 2013.

[70] P. Liu, N. Kong, Y. Tian, X. Lee and B. Yan, "A DHT-based discovery service for RFID network," in *Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings '14), Green Computing and Communications (GreenCom '14) and Cyber, Physical and Social Computing(CPSCom '14)*, pp. 344–347, Taipei, 2014.

[71] Z. Hongtao and W. Dong, "Design and analysis of an EPC discovery service based randomized architecture," in *Proceedings of the 2013 IEEE Conference on Anthology*, pp. 1–5, China, 2013.

[72] T. Kelepouris, M. Harrison and D. McFarlane, "Bayesian supply chain tracking using serial-level information," *in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 5, pp. 733–742, September 2011.

[73] M. G. Khair, A. Dahbi and H. T. Mouftah, "EPCglobal network distributed discovery services using extended version of BGP," University Of Ottawa, Ottawa, ON, Tech. Rep. TR-2012-05, September 2012.

[74] M. G. Khair, B. Kantarci and H. T. Mouftah, "Distributed discovery services via EPC-BGP for mobile RFID," in *Proceedings of the 2013 IEEE International Conference on Communications (ICC '13)*, pp. 3678–3682, Budapest, Hungry, 2013.

[75] S. Evdokimov, B. Fabian, S. Kunz and N. Schoenemann, "Comparison of discovery service architectures for the internet of things," in *Proceedings of the 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '10)*, pp. 237–244, Newport Beach, CA, USA, 2010.

[76] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess and D. Savio, "Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *in IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, July 2010.

[77] K. Romer, B. Ostermaier, F. Mattern, M. Fahrmair and W. Kellerer, "Real-time search for real-world entities: A survey," *Proceedings of the 2010 IEEE*, vol. 98, no. 11, pp. 1887–1902, November 2010.

[78] W. He and L. Xu, "Integration of distributed enterprise applications: A survey," *in IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 35–42, 2014.

[79] Y. Wu, Q. Sheng, H. Shen and S. Zeadally, "Modeling object flows from distributed and federated RFID data streams for efficient tracking and tracing," *in*

IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 10, pp. 2036–2045, October 2013.

[80] B. R. Ray, J. Abawajy and M. Chowdhury, "Scalable RFID security framework and protocol supporting internet of things," *in Elsevier Computer Networks Journal*, vol. 67, no. 4, pp. 89 – 103, July 2014.

[81] M. Yang and Y. Yang, "An efficient hybrid peer-to-peer system for distributed data sharing," *IEEE Transactions on Computers*, vol. 59, no. 9, pp. 1158–1171, September 2010.

[82] M. Yang and Y. Yang, "Applying network coding to peer-to-peer file sharing," *in IEEE Transactions on Computers*, vol. 63, no. 8, pp. 1938–1950, August 2014.

[83] B. Gedik and L. Liu, "A scalable peer-to-peer architecture for distributed information monitoring applications," *in IEEE Transactions on Computers*, vol. 54, no. 6, pp. 767–782, June 2005.

[84] S. Li, L. D. Xu and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *in IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, November 2013.

[85] Y. A. Ridhawi and A. Karmouch, "QoS-based composition of service specific overlay networks," *in IEEE Transactions on Computers*, vol. 64, no. 3, pp. 832–846, March 2015.

[86] M. Deshpande, K. Kim, B. Hore, S. Mehrotra and N. Venkatasubramanian, "ReCREW: A reliable flash-dissemination system," *in IEEE Transactions on Computers*, vol. 62, no. 7, pp. 1432–1446, July 2013.

[87] A. M. Haubenwaller and K. Vandikas, "Computations on the edge in the internet of things," *in Elsevier Procedia Computer Science Journal*, vol. 52, pp. 29 – 34, 2015.

[88] S. Abdelwahab, B. Hamdaoui, M. Guizani and T. Znati, "Cloud of things for sensing as a service: Sensing resource discovery and virtualization," in *Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM '15)*, pp. 1–7, San Diego, CA, 2015.

[89] S. K. Datta and C. Bonnet, “Search engine based resource discovery framework for internet of things,” in *Proceedings of the 2015 IEEE Global Conference on Consumer Electronics (GCCE ’15)*, pp. 83–85, Osaka, 2015.

[90] B. Djamaa, M. Richardson, P. Barker and I. Owens, “Discovery of things: A fully-distributed opportunistic approach,” in *Proceedings of the 2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–5, Glasgow, 2015.

[91] D. Georgakopoulos, P. P. Jayaraman, M. Zhang and R. Ranjan, “Discovery-driven service oriented iot architecture,” in *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*, pp. 142–149, Hangzhou, 2015.

[92] S. Abdelwahab, B. Hamdaoui, M. Guizani and T. Znati, “Cloud of things for sensing-as-a-service: Architecture, algorithms, and use case,” *in IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2016.

[93] R. Petrolo, S. G. Bonifacio, V. Loscri and N. Mitton, “The discovery of relevant data-sources in a smart city environment,” in *Proceedings of the 2016 IEEE International Conference on Smart Computing (SMARTCOMP ’16)*, pp. 1–5, St. Louis, MO, 2016.

[94] W. T. Lunardi, E. de Matos, R. Tiburski, L. A. Amaral, S. Marczak and F. Hessel, “Context-based search engine for industrial iot: Discovery, search, selection, and usage of devices,” in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–8, Luxembourg, 2015.

[95] S. K. Datta, R. P. F. D. Costa and C. Bonnet, “Resource discovery in internet of things: Current trends and future standardization aspects,” in *Proceedings of the 2015 IEEE World Forum on Internet of Things (WF-IoT ’15)*, pp. 542–547, Milan, 2015.

[96] E. Rapti, C. Houstis, E. Houstis and A. Karageorgos, “A bio-inspired service discovery and selection approach for iot applications,” in *Proceedings of the 2016 IEEE International Conference on Services Computing (SCC ’16)*, pp. 868–871, San Francisco, CA, 2016.

[97] H. Rahman, T. Kanter and R. Rahmani, “Enabling distributed context entity discovery for an internet-of-things platform,” in *SAI Intelligent Systems Conference (IntelliSys), 2015*, pp. 350–354, Nov 2015.

[98] J. Li, N. Zaman and H. Li, "A decentralized locality-preserving context-aware service discovery framework for internet of things," in *Services Computing (SCC), 2015 IEEE International Conference on*, pp. 317–323, New York, NY, 2015.

[99] P. Liu, N. Kong, Y. Tian, X. Lee and B. Yan, "CUIAS - A user identity authentication service for discovery service," in *Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings '14), Green Computing and Communications (GreenCom '14) and Cyber, Physical and Social Computing(CPSCom '14)*, pp. 95–101, Taipei, 2014.

[100] F. Thompson, "Extensible supply-chain discovery service schema," IETF, Internet Draft, August 2008.

[101] F. Thompson, "Extensible supply-chain discovery service commands," IETF, Internet Draft, August 2008.

[102] Z. Wang and H. Zhu, "A client privacy preserving discovery service scheme," in *Proceedings of the 2013 International Conference on Advanced Information Networking and Applications Workshops (WAINA '13)*, pp. 935–940, Barcelona, 2013.

[103] F. Zhu, W. Zhu, M. Mutka and L. Ni, "Private and secure service discovery via progressive and probabilistic exposure," *in IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 11, pp. 1565–1577, November 2007.

[104] "DNS related RFCs," August 2012. [Online]. Available: http://www.isc.org/community/reference/RFCs/DNS

[105] M. Kaashoek and D. Karger, "Koorde: A simple degree-optimal distributed hash table," in *Proceedings of the 2003 International Workshop on Peer-to-Peer Systems (IPTPS '03)*, pp. 1–10. Berkeley, CA, USA: Springer Berlin / Heidelberg, 2003.

[106] GS1 EPCglobal, "EPCglobal certificate profile specification," http://www.gs1.org/gsmp/kc/epcglobal/cert, June 2010, ver. 2.0.1 [Accessed September 25, 2014].

[107] D. Bell and L. LaPadula, "Secure computer systems: Unified exposition and multics interpretation," MITRE Corporation, Bedford, MA, Technical Report MTR 2997, Rev. 1, March 1975.

[108] D. Bell and L. LaPadula, "Secure computer systems: Mathematical foundations," MITRE Corporation, Bedford, MA, Technical Report MTR 2547, Vol. I, March 1973.

[109] R. Jain and N. Abouzakhar, "Hidden markov model based anomaly intrusion detection," in *Proceedings of the 2012 International Conference on Internet Technology And Secured Transactions (ICITST '12)*, pp. 528–533, London, 2012.

[110] J. Hu, X. Yu, D. Qiu and H.-H. Chen, "A simple and efficient hidden markov model scheme for host-based anomaly intrusion detection," *in IEEE Network Journal*, vol. 23, no. 1, pp. 42–47, January 2009.

[111] Y. Yasami, M. Farahmand and V. Zargari, "An ARP-based Anomaly Detection Algorithm Using Hidden Markov Model in Enterprise Networks," in *Proceedings of the 2007 International Conference on Systems and Networks Communications (ICSNC '07)*, p. 69, Cap Esterel, 2007.

[112] W. Wang, X.-H. Guan and X.-L. Zhang, "Modeling program behaviors by hidden markov models for intrusion detection," in *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics (ICMLC '04)*, pp. 2830–2835, Shanghai, China, 2004.

[113] W.-K. Wong, A. Moore, G. Cooper and M. Wagner, "Rule-based anomaly pattern detection for detecting disease outbreaks," in *Proceedings of the 2002 National Conference on Artificial Intelligence*, pp. 217–223, Menlo Park, CA, USA, 2002.

[114] W.-K. Wong, A. Moore, G. Cooper and M. Wagner, "Bayesian network anomaly pattern detection for disease outbreaks," in *Proceedings of the 2003 International Conference on Machine Learning*, pp. 808–815, 2003.

[115] E. Keogh, J. Lin, S.-H. Lee and H. Van Herle, "Finding the most unusual time series subsequence: Algorithms and applications," *in Knowledge and Information Systems Journal*, vol. 11, no. 1, pp. 1–27, December 2006.

[116] S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: An application to sensor data," *in Knowledge and Information Systems Journal*, vol. 11, no. 2, pp. 137–154, February 2007.

[117] M. Al-Subaie and M. Zulkernine, "Efficacy of hidden markov models over neural networks in anomaly intrusion detection," in *Proceedings of the 2006 International Computer Software and Applications Conference (COMPSAC '06)*, pp. 325–332, Chicago, IL, 2006.

[118] C. Bennett and K. Campbell, "A linear programming approach to novelty detection," *in Advances in Neural Information Processing Systems*, vol. 13, no. 13, p. 395, 2001.

[119] C. C. Aggarwal, "On abnormality detection in spuriously populated data streams," in *Proceedings of the SIAM Conference on DataMining*, pp. 80–91, Newport Beach, CA, 2005.

[120] W. Khreich, E. Granger, R. Sabourin and A. Miri, "Combining hidden markov models for improved anomaly detection," in *Proceedings of the 2009 IEEE International Conference on Communications (ICC '09)*, pp. 1–6, Dresden, 2009.

[121] E. Trentin and M. Gori, "Robust combination of neural networks and hidden markov models for speech recognition," *in IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1519–1531, November 2003.

[122] P. Zhang and H. Li, "Hybrid model of continuous hidden markov model and multi-layer perceptron in speech recognition," in *Proceedings of the 2009 International Conference on Intelligent Computation Technology and Automation (ICICTA '09)*, pp. 62–65, Changsha, Hunan, 2009.

[123] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Readings in Speech Recognition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, pp. 267–296.

[124] GS1 EPCglobal, "GS1 company prefix list," http://www.gs1.org/barcodes/support/prefix_list, [Accessed September 25, 2014].

[125] P. G. López, C. Pairot, R. Mondéjar, J. P. Ahulló, H. Tejedor and R. Rallo, "PlanetSim: A New Overlay Network Simulation Framework." in *SEM*, ser. Lecture Notes in Computer Science, vol. 3437, pp. 123–136. Springer, 2004.

[126] High Performance Computing Virtual Laboratory, "Hardware:SW," http://wiki.hpcvl.org/index.php/Hardware:SW, [Accessed June 30, 2015].

[127] Compute Canada, Compute Ontario, "High performance computing virtual laboratory," https://www.hpcvl.org/, [Accessed June 30, 2015].

[128] Z. Li, Y. Xie, C. Wu and B. Ding, "A security query protocol of ons in epc system," in *Proceedings of the 2012 International Conference on Anti-Counterfeiting, Security and Identification (ASID '12)*, pp. 1–6, Taipei, 2012.

[129] R. Quilez, N. Mitton, M. de Amorim and N. Pauvre, "Prototyping a multi-root ONS," in *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW '12)*, pp. 159–163, Paris, 2012.

[130] S. Turchi, L. Ciofi, F. Paganelli, F. Pirri and D. Giuli, "Designing EPCIS through linked data and REST principles," in *Proceedings of the 2012 International Conference on Software, Telecommunications and Computer Networks (SoftCOM '12)*, pp. 1–6, Split, 2012.

[131] C.-W. Tseng, C.-M. Chang and C.-H. Huang, "Complex sensing event process of IoT application based on EPCglobal architecture and IEEE 1451," in *Proceedings of the 2012 International Conference on the Internet of Things (IOT '12)*, pp. 92–98, Wuxi, 2012.

[132] H. Wang, Y. Li, Z. Zhang and Z. Cao, "Two-level path authentication in epcglobal network," in *Proceedings of the 2012 IEEE International Conference on RFID (RFID '12)*, pp. 24–31, Orlando, FL, 2012.

[133] A. Awad, R. German and F. Dressler, "Exploiting virtual coordinates for improved routing performance in sensor networks," *in IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1214–1226, September 2011.

[134] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proceedings of the*

2001 IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, pp. 329–350, London, UK, 2001.

[135] J. Doyle, *OSPF and IS-IS: Choosing an IGP for Large-Scale Networks*. Addison-Wesley Professional, 2005.

[136] IETF, Network Working Group, "RFC5340, OSPF for IPv6," http://tools.ietf.org/html/rfc5340, July 2008, [Accessed November 1st, 2015].

[137] IETF, Network Working Group, "RFC4271, A Border Gateway Protocol 4 (BGP-4)," http://www.ietf.org/rfc/rfc4271, January 2006, [Accessed November 1st, 2015].

[138] R. White, D. McPherson and S. Sangli, *Practical BGP*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[139] A. Elmokashfi, A. Kvalbein and C. Dovrolis, "BGP churn evolution: A perspective from the core," *in IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 571–584, April 2012.

[140] E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras and P. Vervier, "Visual analytics for BGP monitoring and prefix hijacking identification," *in IEEE Network Journal*, vol. 26, no. 6, pp. 33–39, November 2012.

[141] S. Sarkar, H.-H. Yen, S. Dixit and B. Mukherjee, "A novel delay-aware routing algorithm (DARA) for a hybrid wireless-optical broadband access network (WOBAN)," *in IEEE Network Journal*, vol. 22, no. 3, pp. 20–28, May 2008.

[142] S. Kamble, A. Desai and P. Vartak, "Data mining and data warehousing for supply chain management," in *Proceedings of the 2015 International Conference on Communication, Information Computing Technology (ICCICT '15)*,, pp. 1–6, Mumbai, 2015.

[143] S. Jain and N. Srinivasa Raghavan, "Analysis of base-stock controlled production-inventory system using discrete-time queueing models," in *Proceedings of the 2005 IEEE International Conference on Automation Science and Engineering (CASE '05)*, pp. 37–42, 2005.

[144] D.-P. Song, “Optimal integrated ordering and production policy in a supply chain with stochastic lead-time, processing-time, and demand,” *in IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2027–2041, September 2009.

[145] I. Slimani and S. Achchab, “Game theory to control logistic costs in a two-echelon supply chain,” in *Proceedings of the 2014 International Conference on Logistics and Operations Management (GOL ’14)*, pp. 168–170, Rabat, 2014.

List of Publications

Refereed Journal Papers

- A. Dahbi, M. G. Khair, B. Kantarci and H. T. Mouftah, ”Threshold-based Distributed Discovery Services for EPCGlobal Network,” submitted to Elsevier Computer Communications Journal.
- A. Dahbi and H. T. Mouftah, ”Improving Scalability and Responsiveness of Discovery Services in the Future IoT-based Business Infrastructure,” submitted to IEEE Access Journal.

Refereed Conference Papers

- A. Dahbi, M. G. Khair and H. T. Mouftah, “Secured distributed discovery services in the EPCglobal network,” In Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, 2013, pp. 2959-2963.
- A. Dahbi and H. T. Mouftah, “A Distributed EPC Data Discovery Service in the EPCglobal Network,” 2012 Fourth Annual WiSense Workshop, Ottawa, Canada.
- Abdelmounaim Dahbi, Mazen George Khair, and Hussein Talaat Mouftah. 2014. An enhanced security scheme for query state inference in EPCglobal discovery services. In Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications (DIVANet ’14). ACM, New York, NY, USA, 119-125.

- A. Dahbi, M. G. Khair and H. T. Mouftah, “A Hidden Markov Model security scheme for query state inference in discovery services,” In Proceedings of the 2014 Global Information Infrastructure and Networking Symposium (GIIS), Montreal, QC, 2014, pp. 1-6.

- A. Dahbi and H. T. Mouftah, “A Hierarchical Architecture for Distributed EPCglobal Discovery Services,” In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-7.

- A. Dahbi and H. T. Mouftah, “Supply chain efficient inventory management as a service offered by a cloud-based platform,” In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, 2016, pp. 1-7.