FIFTH ERCIM WORKSHOP ON EMOBILITY

Marc Brogle, Xavier Masip Bruin,
Torsten Braun, Geert Heijenk (Eds.)

Universitat Politècnica de Catalunya
Advanced Network Architectures Lab (CRAAX)
Vilanova i la Geltrú, Catalonia, Spain, June 14, 2011

Published:	June 2011
	Technical University of Catalonia, Barcelona, Spain
Print:	PADISGRAF S.L.L CIF: 62.144.704
	Fleming, 27
	08800 Vilanova i la Geltrú
	Spain
	Tel: +34938100054
	padisgraf@padisgraf.com
	www.padisgraf.com
ISBN	978-84-920140-3-3

Preface

ERCIM, the European Research Consortium for Informatics and Mathematics, aims to foster collaborative work within the European research community and to increase co-operation with European industry. In the ERCIM eMobility workshop, current progress and future developments in the area of eMobility are discussed and the existing gap between theory and application closed. The fifth edition of eMobility workshop was hosted by the Advanced Network Architectures Lab (CRAAX) of the Technical University of Catalonia in Spain and took place on June 14, 2011

This volume contains scientific articles accepted for publication by eMobility technical program committee. The accepted contributions discuss several topics of the ERCIM eMobility working group including, pricing schemes for Mobile WiMAX systems, mobility support in publish/subscribe networks, automated merging in cooperative adaptive cruise control systems, content replication strategies for smart products, inquiry-based bluetooth parameters for indoor localization, efficiency analysis for multicast traffic distribution in PMIPv6 domains. The invited talk discussed opportunistic computation and its performance.

At this point, we want to thank all authors of the submitted papers and the members of the international program committee for their contribution to the success of the event and the high quality program. The proceedings are divided into two sections: regular papers and the invited talk section. The regular paper section features short papers that present work in progress and ongoing research, as well as full papers that elaborate in more detail the presented topics. All papers have been carefully selected in a peer review process.

We hope that all workshop delegates enjoy the scientific program and the beautiful region as well as the coast of Vilanova i la Geltrú. We further hope that many scientists, including the current participants, will continue to use the yearly ERCIM eMobility workshop as an event for the exchange of ideas and experiences. The next ERCIM eMobility workshop is scheduled for 2012.

General chairs:
Torsten Braun
Geert Heijenk

TPC chairs:
Marc Brogle
Xavier Masip Bruin

General chairs

Torsten Braun, University of Bern, Switzerland
Geert Heijenk, University of Twente, The Netherlands

TPC chairs

Marc Brogle, SAP AG (SAP Research), Switzerland
Xavier Masip Bruin, Technical University of Catalonia, Spain

Technical program committee

Francisco Barcelo-Arroyo, Universitat Politecnica de Catalunya, ES
Hans van den Berg, University of Twente, NL
Robert Bestak, Czech Technical University in Prague, CZ
Raffaele Bruno, Italian National Research Council, IT
Tao Chen, VTT, FI
Djamel Djenouri, CERIST research centre Algiers, Algeria
Jean-Marie Jacquet, University of Namur, BE
Dimitri Konstantas, University of Geneva, CH
Yevgeni Koucheryavy, Tampere University of Technology, FI
Saverio Mascolo, Politecnico di Bari, IT
Edmundo Monteiro, University of Coimbra, PT
Evgeny Osipov, Luleå University of Technology, SE
Vasilios Siris, FORTH-ICS, GR
Dirk Stähle, University of Wuerzburg, DE

Table of Contents

I Invited Talk

II Regular Papers

Part I

Invited Talk

Opportunistic Computation and its Performance

Erol Gelenbe

Intelligent Systems & Networks Group
Electrical & Electronic Engineering Dept.
Imperial College
London SW7 2BT, UK
e.gelenbe@imperial.ac.uk

Abstract. We propose Opportunistic Computation as a simple paradigm in which "agents" can be created *ex nihilo*. They then compute autonomously on their own and can transition or transform themselves into another agent, or they can terminate their computation, or two agents may "combine" and give rise to a new agent, or together they may jointly terminate their computation. This computational system can include N different types of agents, and at any instand during the computation the total number of agents of any type is unlimited. An agent that requests to combine with another agent to carry out a computational step will however terminate if its partner agent cannot be found. In this paper, a brief description of this proposed computational model is followed by a mathematical model of its performance. We then derive the probability distribution of the number and type of all agents that may be present in such a system. From this basic result, we can compute measures of interest such as memory occupancy and the amount of communication that takes place.

Keywords Parallel Computation, Opportunistic Computing, Performance Evaluation

1 Introduction

We borrow the term "Opportunistic Computation" (OC) from the field of computer-communications where there has been recent interest in "opportunistic or delay-tolerant communications (DTC)" [23] which has been discussed for a few years, In DTC mobile entities (such as vehicles or people) carry wireless communication devices which pass packets on to other devices carried by other people or vehicles, in the hope that these packets will eventually arrive at the desired destination thanks to the mobile entities' motion, and also to the successive "opportunistic" encounters and hops that the packets can make among the wireless communication devices carried by mobile entities. OC is also similar to "Chemical Computation Models" [22] where computational entities combine with each other to "co-compute", just like molecules in a chemical reaction combine to form larger or more complex molecules.

OC is similar to the concept of "coalition formation" in agent systems [14] in which coalitions are formed via a small set of compositional rules. OC is also a special kind of stochastic population model [5], in which the "agents" are individuals which can combine with other individuals to produce one offspring which then replaces its parents. Agents can also "die" or terminate, while they will also be destroyed if they need to combine with another agent of a specific type, but cannot find an appropriate agent with which they can do that. Such models also link to other areas such as viruses in the Internet [20], neural networks, and chemistry [3, 4, 18]. G-Networks [6, 8–10, 12, 16] are queueing network models related to OC, where "customers" are served one at a time in a finite set of queues, and certain customers called "negative customers", "triggers" or "signals", have a behaviour similar to the agents discussed in this paper. The major difference however is that in G-Networks, customers only interact with each other when they are at the head of a queue, while in OC binary interactions can occur simultaneously among *all* of the agents that are present in the system, and the effective rates of interaction depend on the *total* number of agents of each type.

1.1 A model of Opportunistic Computation

More formally, OC is a simple computational paradigm in which there are N types of agents $\mathbf{U} = \{U_1, \ ... \ U_N\}$, where each agent contains code and data. A subset $\mathbf{I} \subseteq \mathbf{U}$ will denote the "initial agent types" which may be used to initiate a computation.

We will use $U, V, W \in \mathbf{U}$ to denote different agent types. T is a symbol we will use to denote a terminated computation, while the symbol ϵ will represent the "empty" symbol or non-existent agent. The system is structured with a set of generation or re-write rules, which describe the creation, destruction, termination and transformations that the agents can undergo, so that:

- (0) An agent of some type U can be created *ex nihilo*, as presented symbolically by $\epsilon \Rightarrow U$, or
- (i) An agent of some type can compute autonomously on its own giving rise to a successor agent of another type, represented symbolically by $\odot U \Rightarrow V$, or
- (ii) An agent of some type computes autonomously and then terminates $\oslash U \Rightarrow T$, without creating a successor agent, or
- (iii) Two agents of types U and V combine and when the co-computation ends the two are replaced by one successor agent of the type W: $U \oplus V \Rightarrow W$, where W is not necessarily of a type distinct from U and V, or
- (iv) Two agents of types U and V combine in a computation and then terminate without leaving a successor agent $U \oplus V \Rightarrow T$, and we also have the cases where
- (v) The computation $U \oplus V \Rightarrow W$, will be aborted if either an agent of type U or of type V does not exist in the system at the time when the computation is launched, giving rise the the step $U \oplus \epsilon \Rightarrow \epsilon$ or $\epsilon \oplus V \Rightarrow \epsilon$, and similarly

– (vi) The computation $U \oplus V \Rightarrow T$, will be aborted because either an agent of type U or of type V does not exist in the system at the time when the computation is launched, giving rise to the computation step $U \oplus \epsilon \Rightarrow \epsilon$ or $\epsilon \oplus V \Rightarrow \epsilon$.

Note that $U,\ V,\ W$ above can refer to the same or to different types of agents.

A **computation in this model is therefore a tree**, where:

– The leaves are the initial agents in the computation, i.e. they are elements of **I**,
– The root is either ϵ, for an aborted computation, or T for a computation that terminates normally,
– The **intermediate nodes** of the tree are one of the symbols $\odot,\ \oslash,\ \oplus,\ \ominus$, and
– Each arc in the tree is labeled with one of the $U_i \in \mathbf{U}$ and a path in the tree may contain multiple instances of the same agent type U_i.

Furthermore, at any instant of time, many computations (i.e. many trees) may be active simultaneously in various stages of progress.

Examples of this model can include cases where if two agents combine to compute together, one could be purely code while the other would be purely data and the data's access path. But in general an agent in the OC framework is an entity which includes both code and data, and we assume that the hardware that is needed for the computation is always available, i.e. there are an unlimited number of identical hardware units that can execute any of the computational steps described above, and the communication delays or costs associated with all computations can be neglected. The opportunistic nature of OC means that if some agent needs some other agent to carry out its next computational step and does not find that agent, then the agent destroys itself so that it does not indefinitely occupy resources (such as memory space) in the "hope" that the agent that it requires will eventually appear. This means that some sequences of computation or "threads" will be wasted because they end in an aborted termination. Thus one of the questions we address in this paper is how we can attain a desired throughput of *normal* terminations despite the waste of some of the threads due to abortions. Since aborted threads will have consumed both memory space and computational time on the hardware up to the time that they are aborted, we are also interested in estimating the amount of waste that is incurred.

This paper is concerned with the performance of such systems. OC is obviously efficient in terms of time, since no computational step is held back for reasons of synchronisation: an agent will never wait for another agent to become available. If the agent needed by another agent is not immediately available, the computation thread will be aborted. At the same time, because many threads are being executed, the chances are that one can achieve the desired throughput in number of computations terminating correctly per-unit-time.

Thus we introduce a mathematical model for the performance of OC which allows us to explictly obtain the probability distribution of all types of agents that may be present in such a system, given a flow of starting agents at the input, which also provides an estimate for the memory occupancy of the system, the amount of communication that OC requires, and other performance metrics of interest such as the average execution time of a computation that results in a normal termination.

1.2 A Mathematical Model of the Performance of OC

For each type U_i of the N types of agents $\mathbf{U} = \{U_1, \ ... \ , U_N\}$ which compute in the manner described above. Among these, the subset $\{U_1, \ ... \ , U_I\}$, $I \leq N$ are the "initial types" of agents, and we associate a Poisson flow of rate $\lambda_i > 0$ of initiations (0) $\epsilon \Rightarrow U_i$ for $1 \leq i \leq I$ while $\lambda_i = 0$ for $i > I$. Note that if $\lambda_i = 0$ then this simply means that the scheme for constantly generating agents of type U_i is either unavailable or dormant.

Since *OC* is a form of automatic computation that does not have an external control, each of the agent types will initiate computations based on their own specific propensity (or reaction rate in a chemical reaction), and these will now be specified. Any one of the computation types described previously will take some time whose average value is given as the inverse of the rates indicated below. Thus in any time interval $[t, t + \Delta t[$, $t \geq 0$, the following events may occur:

- (i) With probability $\gamma_i \Delta t + o(\Delta t)$ an agent U_i carries out an autonomous execution and then terminates, i.e. $\oslash U_i \Rightarrow \epsilon$
- (ii) With probability $\beta_{ij} \Delta t + o(\Delta t)$ an agent U_i transforms itself into one of type U_j, i.e. $\odot U_i \Rightarrow U_j$, and we assume $i \neq j$ to avoid the trivial case where nothing will have changed when an individual is replaced by another one of the same type.
- (iii) With probability $\mu_{ijl} \Delta t + o(\Delta t)$ the agent of type U_i and the one of type U_j complete a computation resulting in an agent of type l, and $l \neq i, j$, i.e. $U_i \oplus U_j \Rightarrow U_l$,
- (iv) With probability $\delta_{ij} \Delta t + o(\Delta t)$ the computation of an agent U_i with U_j results in a successful termination of that computation, i.e. $U_i \ominus U_j \Rightarrow \epsilon$,

and the rates are non-negative $\gamma_i, \ beta_{ij}, \ \mu_{ijl}, \ \delta_{ij} \geq 0$, and they are finite.

The events (i) to (iv) are characterised by rates which relate to single individuals, while the probabilities that such events occur will depend on the *number* of individuals of each type that are present in the system. These rates are non-negative, but some may be zero if the corresponding events never occur. We also denote by r_i the total activity rate of an individual of type i:

$$r_i = \gamma_i + \sum_{j=1}^{N} \{\beta_{ij} + \delta_{ij} + \sum_{k=1}^{N} \mu_{ijk}\} \tag{1}$$

The vector representing the number of agents of each type at time $t \geq 0$ is denoted by $K(t) = (K_1(t), \ ... \ , K_N(t))$, $K_i(t) \geq 0$, while $k = (k_1, \ ... \ , k_N)$, is a vector of non-negative integers representing a particular value that may be taken by $K(t)$. We will describe the manner in which the system evolves over time by using the probability distribution $p(k,t) = Prob[K(t) = k]$ with some given initial condition $p(k,0)$. We also use the following notation. Let e_i be the N-vector that contains zeros in all positions except the $i-th$, whose value is $+1$. Then we define:

- $k_i^- = (k_1, \ ... \ , k_i - 1, \ ... \ k_N)$, or $k_i^- = k - e_i$, when $k_i > 0$,
- $k_i^+ = k + e_i$,
- $k_{ij}^{++} = k + e_i + e_j$, including the case when $i = j$,
- $k_{ij}^{+-} = k + e_i - e_j$ for $k_j > 0$,
- $k_{ijl}^{++-} = k + e_i + e_j - e_l$ for $k_l > 0$, including when (a) l is equal to i or j, and (b) $i = j$.

Using the assumption that the external arrivals of agents of each type are independent Poisson processes, and assumptions (i) through (iv) about tha manner in which agents interact with each other or terminate, we write the Chapman-Kolmogorov equations [7] for the system:

$$\frac{dp(k,t)}{dt} = \sum_i \{\lambda_i p(k_i^-, t) 1[k_i > 0] + \gamma_i (k_i+1) p(k_i^+, t) + \sum_j [\ \beta_{ij}(k_i+1) p(k_{ij}^{+-}, t) 1[k_j > 0] \quad (2)$$
$$+ \delta_{ij}((k_i+1)(k_j+1) p(k_{ij}^{++}, t) + (k_i+1) p(k_i^+, t) 1[k_j = 0])$$
$$+ \sum_{l \neq i,j} \mu_{ijl}((k_i+1)(k_j+1) p(k_{ijl}^{++-}, t) 1[k_l > 0] + (k_i+1) p(k_i^+, t) 1[k_j = 0])] - (\lambda_i + r_i k_i) p(k,t)\}$$

2 Product form solution

Our main result concerns the solution of the equations (2) in steady-state, and the joint probability distribution for the number of agents of each type, showing that it is in "product form". Because the flow equations that drive this solution are non-linear. we also prove that the solution exists and is unique.

Theorem 1 Consider the following system of non-linear equations:

$$\Lambda_i = \lambda_i + \sum_j \phi_j \beta_{ji} + \sum_j \sum_l \phi_j \phi_l \mu_{jli}, \ i = 1, \ ... \ N \quad (3)$$

where:

$$\phi_i = \frac{\Lambda_i}{r_i}, \ i = 1, \ ... \ N \quad (4)$$

If the following flow equation is satisfied:

$$\lambda \equiv \sum_{i=1}^{I} \lambda_i = \sum_{i=1}^{N} [\gamma_i \phi_i + \sum_{j=1}^{N} \phi_i \phi_j \delta_{ij}] \quad (5)$$

and the system of equations (3) has a non-negative solution $\Lambda_i \geq 0$, then (2) has a unique steady-state solution $p(k) \equiv \lim_{t\to\infty} p(k,t)$ given by:

$$p(k) = e^{-\sum_{i=1}^{N} \phi_i} \prod_{i=1}^{N} \frac{\phi_i^{k_i}}{k_i!} \tag{6}$$

$$q_i(k_i) = e^{-\phi_i} \frac{\phi_i^{k_i}}{k_i!}, if k_i \geq 0$$

2.1 Conditions for, and Consequences of Theorem 1

Let us first point to the main condition for this theorem and then deail some of the consequences that it leads to.

Main condition The condition that is stated for this product form solution is the stability condition (5): in the left-hand-side we have the rate at which initial agents are started by the computational process, while in right-hand-side of (5), the first term sums over all the normal termination events that will reduce by one the number of agents of type i, while the second term is the rate of normal terminations that reduce the number of agents by two. Thus (5) states the requirement that the total arrival rate of the initial agents to the system (the left-hand-side), be identical to the total *normal termination rate* (the right-hand-side) of agents.

Consequence 1 As a consequence of the theorem we can compute the **total aborted computation termination rate** or wastage rate of computations as:

$$\alpha = \sum_{i,j=1}^{N} \phi_i e^{-\phi_j} [\delta_{ij} + \sum_{l \neq i,j}^{N} \mu_{ijl}] \tag{7}$$

because $e^{-\phi}$ is the probability in steady-state that there are no agents of type i in the system.

Consequence 2 We can also estimate the *total average space requirement* of an OC. Indeed, if an agent of type U_i occupies on average of m_i memory units, then the average space requirements for OC for a given set of initiation rates λ_i, $i \in \mathbf{I}$ is given by:

$$M = \sum_{i=1}^{N} m_i \phi_i \tag{8}$$

where the ϕ_i is obtained from the simultaneous solutions of equations (4) and (3).

Consequence 3 Another measure of interest is the total inter-agent communication rate for OC. If b_{ij} denotes the average size of a message that is passed when an agent of type U_i is replaced by an agent of type U_j, and c_{ijl} is the size of the messages exchanged when an agent of type U_l replaces the agents of type U_i, U_j as a result of a co-computation, while d_{ij} is the size of messages exchanged when the co-computation of an agent of type U_i and U_j result in a termination, we can compute the total data rate for normal computation in OC as:

$$\nu = \sum_{i,j=1}^{N} \phi_i(\beta_{ij} b_{ij} + \phi_j[d_{ij}\delta_{ij} + \sum_{l\neq i,j}^{N} \mu_{ijl} c_{ijl}]) \tag{9}$$

and similarly we can compute because $e^{-\phi}$ is the probability in steady-state that there are no agents of type i in the system.

Consequence 4 Finally the average execution time of a normally terminating, or aborted, computation tree is also easily obtained. For a normally terminating computation tree, the average execution time is given via Little's formula as:

$$\tau_n = \frac{\sum_{i=1}^{N} \phi_i}{\lambda} \tag{10}$$

when the stability condition (5) is satisfied, because the initiation rate λ is also the completion rate of normally terminating computations. However, the average rate is of the aborted terminations is:

$$\tau_a = \frac{\sum_{i=1}^{N} \phi_i}{\alpha} \tag{11}$$

where α computed above is the total rate of aborted computations.

3 Conclusions

In this paper we have suggested a model of Opportunistic Computation, and then constructed a probability model of its performance. We have shown that under a stability condition, the performance model of OC has a product form solution. From the product form solution we have indicated how the memory space needs, communication needs, and average execution times of OC can be estimated. Because the product form depends on a non-linear flow equation, we have also proved the existence and uniqueness of the product form solution based on Brouwer's theorem.

We hope that this development and analysis can motivate much further work, both for practical experimentation of OC on large test-beds, and in terms of further analysis of cases of practical interest.

References

1. J. G. Kemeny and J. L. Snell "Finite Markov Chains", Van Nostrand Pub. Co., Princeton, NJ, 1965.
2. A. F. Bartholomay "The general catalytic queue process", In *Stochastic Models in Medicine and Biology* (ed. J. Garland), pp. 101142, University of Wisconsin Press, Madison WI, 1964.

3. D. Gillespie “General method for numerically simulating the stochastic time evolution of coupled chemical reactions”, *J. Computational Physics* vol. 22, pp. 403-434, 1976.
4. L.A. Segel “Modeling Dynamic Phenomena in Molecular and Cellular Biology”, Cambridge University Press, 1984.
5. P. Whittle “Systems in Stochastic Equilibrium”, John Wiley Ltd., Chichester, 1986.
6. E. Gelenbe “Réseaux stochastiques ouverts avec clients négatifs et positifs, et réseaux neuronaux”, *Comptes-Rendus Acad. Sciences de Paris*, t. 309, Série II, 979-982, 1989.
7. J. Medhi “Stochastic Models in Queueing Theory”, Academic Press Professional, San Diego, CA, 1991.
8. E. Gelenbe “G-Networks with signals and batch removal”, *Probability in the Engineering and Informational Sciences*, vol. 7, pp 335-342, 1993.
9. E. Gelenbe “G-networks: An unifying model for queuing networks and neural networks,” *Annals of Operations Research*, vol. 48, No. 1-4, pp 433-461, 1994.
10. J.M. Fourneau, E. Gelenbe, R. Suros “G-networks with multiple classes of positive and negative customers,” *Theoretical Computer Science*, vol. 155 (1996), pp.141-156.
11. S. Schnell and C. Mendoza “Closed form solution for time-dependent enzyme kinetics”, *J. Theoretical Biology* vol. 187, pp. 207-212, 1997.
12. E. Gelenbe, A. Labed “G-networks with multiple classes of signals and positive customers”, *European Journal of Operations Research*, vol. 108 (2), pp. 293-305, July 1998.
13. D. Gillespie “Approximate accelerated stochastic simulation of chemically reacting system”, *J. Chemical Physics* vol. 115, no. 4, pp. 1716-1733, 2001.
14. S. Aknine “A reliable algorithm for multi-agent coalition formation”, *Proc. IEEE Int'l. Symp. Intelligent Control/Intelligent Systems and Semiotics*, Cambridge, MA., pp. 290-295, 1999.
15. C. Rao, D. Wolf and A. Arkin “Control, exploitation and tolerance of intracellular noise”, *Nature* vol. 420, pp. 231-237, 2002.
16. E. Gelenbe, J.M. Fourneau “G-Networks with resets”, *Performance Evaluation*, vol. 49, pp. 179-192, 2002.
17. J.-M. Fourneau, E. Gelenbe “Flow equivalence and stochastic equivalence in G-networks”, *Computational Management Science*, vol. 1 (2), pp. 179-192, 2004.
18. P. Whittle “Networks: Optimisation and Evolution”, Cambridge University Press, Cambridge, 2007.
19. E. Gelenbe “Steady-state solution of probabilistic gene regulatory networks”, *Phys. Rev. E 76*, 031903, 2007.
20. E. Gelenbe “Dealing with software viruses: a biological paradigm” *Information Security Tech. Report* vol. 12, pp. 242-250, 2007.
21. T. Jahnke and W. Huisinga “Solving the chemical master equation for monomolecular reaction systems analytically”, *J. Math Biol.* vol. 54, pp. 1-26, 2007.
22. D. Soloveichik, M. Cook, E. Winfree, and J. Brook“Computation with finite stochastic chemical reaction networks”, *Natural Computing*, vol. 7, no. 4, pp. 615-633, Dec. 2008, 10.1007/s11047-008-9067-y
23. Anonymous “Delay Tolerant Networking”, Wikipedia, $http : //en.wikipedia.org/Delay - Tolerant_Networking$

Part II

Regular Papers

Inquiry-based Bluetooth Parameters for Indoor Localisation - an experimental study

D. C. Dimitrova, U. Bürgi, G. Martins Dias, T. Braun, and T. Staub

dimitrova|buergi|martins|braun|staub@iam.unibe.ch
University of Bern, Switzerland

Abstract. The ability to locate people in an indoor environment is attractive due to the many opportunities it offers to businesses and institutions, including emergency services. Although research in this area is thriving, still no single solution shows potential for ubiquitous application. One of the candidate technologies for localisation is Bluetooth owing to its support by a wide range of personal devices. This paper evaluates indoor signal measurements collected based on the Bluetooth inquiry procedure. Our goal was to establish how accurately a mobile device can be linked to the space (e.g., shop, office), in which it currently resides. In particular, we measured the Received Signal Strength Indicator and the Response Rate of an inquiry procedure for various positioning scenarios of the mobile devices. Our results indicate that the Bluetooth inquiry procedure can be successfully used to distinguish between mobile devices belonging to different spaces.

1 Introduction

The amount of information currently available in a modern society, from public transportation schedules and weather forecast to shopping discounts and cultural events, greatly exceeds one's capacity to process it and hence an appropriate content filtering is required. A major filtering criterion is one's location; a person is generally more interested in information, e.g., events or special offers, about its vicinity than in information associated with a remote location. The whole concept of Location Based Services (LBS), for example, rests on the assumption that offering one location-dependent services can increase, among others, generated profit and customer satisfaction, see [2]. Intuitively, the ability to determine one's location is crucial.

Outdoor positioning is dominated by the Global Positioning System (GPS), which offers a highly effective and affordable solution, provided on user-friendly devices. Indoor environments, however, still pose a challenge to the localisation paradigm and foster vigorous research by both academia and industry.

Various technologies have been proposed to tackle the problem of indoor localisation including infrared (e.g., [18]), ultrasound (e.g., the Active Bat system[1]) and Radio Frequency IDentification (e.g., [3]). Some authors, e.g., [7, 13,

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/

16], go a step further and use combined feedback from multiple technologies. Arguably, however, the main focus of the scientific community lies elsewhere. Large number of papers, among which [6] and [19], argue that Ultra Wide Band (UWB) radio offers the excellent means to determine one's location with high precision. Equally many studies campaign for the use of IEEE 802.11, e.g., [4, 8, 11], or Bluetooth, e.g., [9, 13, 12], both a Radio Frequency (RF) technology, due to their ubiquitous support by personal devices. A taxonomy overview of the technologies can be found in [15].

The number of proposed localisation techniques is equally great, the most often used being angulation, lateration and fingerprinting, see [5]. In angulation the location is a derivative of measured angles to fixed reference points. Lateration is based on the same concept but uses distances, which can be determined by various methods among which Time of Arrival (ToA), Time Difference of Arrival (TDoA), received signal strength (RSS) and hop count. Various modifications of each method have been proposed, e.g., [4, 14], as well as combinations thereof. Finally, a fingerprinting technique compares on-line measurements to an off-line database in order to determine location. Currently there are so many localisation proposals based on the fingerprinting technique that a separate taxonomy such as [10] is appropriate.

None of the currently available solutions for indoor location estimation is mature enough to offer ubiquitous applicability. The optimal choice of technology and localisation technique still depends on the application's requirements towards accuracy, cost and ease of deployment. For example, UWB radio can provide highly accurate positioning but is costly and requires device modifications. A more cost-efficient course is to use a RF-based technology with high commercial penetration ratio, e.g., Bluetooth. RF signals, however, are more susceptible to propagation effects, which introduces estimation imprecision.

We are interested in an easy to deploy, low-cost localisation solution with rough position granularity. In particular, we wish to accurately locate persons over the spaces of a large building, e.g., an exposition centre or a shopping mall, without requesting any cooperation from their devices, i.e., non-intrusive detection. Both the IEEE 802.11 standard and Bluetooth can meet our requirements. Currently we focus on the Bluetooth technology but we are aware that the IEEE 802.11 technology can benefit a localisation solution and intend to include it in a future study. Section 2 discusses in more detail our motivation and related work on indoor localisation with Bluetooth.

This paper presents our findings on a Bluetooth-based experimental deployment in a controlled environment. Data from scenarios including in-room and out-of-room positioning, is analysed. Our purpose is to identify the Bluetooth signal parameters which can provide a successful location estimation and to determine which technology-specific and environmental factors affect the process. We aim to gain sufficient insights, which to support us in the development of a scalable method for the localisation at room level of users over large indoor areas. The presented work has been performed within the Eureka Eurostars project Location-Based Analyzer, project no. 5533. It has been funded by the

Swiss Federal Office for Professional Education and Technology and the European Community.

The paper is organised in the following sections. In Section 2 we briefly summarise the state of the art in Bluetooth localisation and position our work. Section 3 describes the measurements set-up and the studied positioning scenarios. Results are discussed in Section 4 while in Section 5 we draw conclusions and identify open issues.

2 Bluetooth-based Localisation

Indoor location estimation based on Bluetooth is attractive mainly due to the large scale adoption of the technology by a wide range of devices, including mobile phones and personal assistants. Hence, a Bluetooth-based localisation system has the potential for quick, cost-efficient deployment without the need to modify the intended target devices. Any localisation algorithm requires certain input parameters from which it derives a target's position. A Bluetooth device can provide feedback on three status parameters in connection mode, namely, Link Quality (LQ), Receieved Signal Strength Indicator (RSSI) and Transmit Power Level (TPL). Methods that rely on these parameters face many challenges and showed little potential for practical application, see [12]. For example, there is no exact definition of LQ and its relation to Bit Error Rate (BER) is device-specific, see [17]. An RSSI reading is less ambiguous but unfortunately susceptible to power control mechanisms at the targets. Additionally, a general disadvantage of this group of methods is the requirement to establish a Bluetooth connection, which does not scale well as the number of targets increases, see [9].

A recent modification of the Bluetooth Core Specification[2] instigates new research on Bluetooth-based localisation. The RSSI reading returned by a Bluetooth inquiry, termed here inquiry-related RSSI, is not affected by power control and hence is a more reliable measure of a target's distance to the inquirer. Although lengthy - the inquirer needs to check all 32 Bluetooth radio channels - an inquiry procedure can monitor a larger number of targeted devices than a connection-based method. Some authors, e.g., [1], introduce as an additional measure the Response Rate (RR) of a Bluetooth inquiry, i.e., the percentage of inquiry responses to total inquiries in a given observation window.

A juxtaposition of the work done by others and our definition of the indoor localisation problem suggests that a solution based on the Bluetooth inquiry procedure fits best our needs. A short motivation follows. Our purpose is to develop a low-cost, easy to deploy system, which can locate persons with a precision at room-level. For these purposes Bluetooth offers a satisfying solution due to its ubiquitous support by personal devices. More specifically the inquiry procedure was chosen since it allows us to gather measurements without requesting active participation of the mobile devices. As a first step in the search of a localisation

[2] Available at https://www.bluetooth.org/Technical/Specifications/adopted.htm

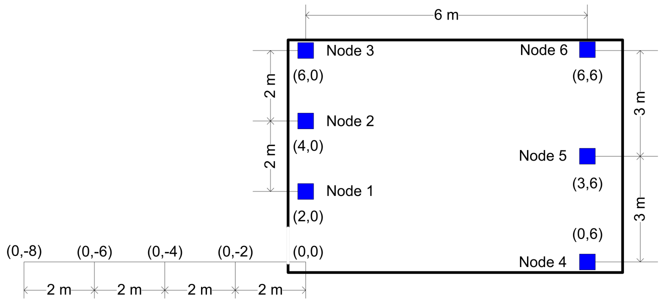


Fig. 1. Schematic of the measurements set-up.

solution we need to determine the bounds of the parameters to be used, i.e., their dependency on distance, obstacles or other factors. For this purpose we performed the experimental study presented in this paper. Since, in our opinion an optimal localisation algorithm will rely on data about both inquiry-related RSSI and RR, we monitor both parameters in the experimental measurements.

3 Experimental Set-up

In the deployed experimental set-up six reference nodes collect measurements based on the Bluetooth inquiry procedure from four mobile devices (MDs), whose position changes. Reference nodes are Bluetooth-enabled wireless sensors, whose position is fixed and known. In particular, OveroFire gumstixs nodes were used. Four smart phones were tracked - an HTC Desire (MD1), HTC Wildfire (MD2), an iPhone (MD3) and an LG e900 (MD4). All measurements are performed in an indoor controlled environment, i.e., the number and identity (MAC address) of the discoverable Bluetooth mobile devices is known. Note that the mobile devices are assumed to be in discoverable mode. We measured inquiry-related RSSI values and inquiry response rates since, as previously mentioned, their collection does not require the active participation of the monitored device.

A graphical representation of the set-up in shown in Figure 1. The wireless sensor nodes are indicated by labelled squares. Three sensors are located at the near end of the room (close to the entrance) at positions (2,0), (4,0) and (6,0); another three sensors are located at the far end of the room at positions (0,6), (3,6) and (6,6). The coordinates in a position pair (x, y) correspond to the distance in meters to the reference location (0,0).

A full grid deployment of sensor nodes may be more insightful in terms of measurements but it is in conflict with our goal to find a low-cost deployment scenario. Recall that we only want to accurately locate persons over building spaces and not precisely determine their positions.

Depending on the positioning of the mobile devices, we distinguish between *in-room* and *out-of-room* scenarios. In the former case the mobile devices were moved over positions (0,0), (2,0), (4,0) and (6,0) in circular manner; in the latter case the phones were moved over positions (-2,0), (-4,0), (-6,0) and (-8,0) outside the room. The exact moving patterns are indicated in Table 1. This specific choice of scenarios allows us to monitor the detection of subjects inside and outside a confine space, e.g., an office or a shop.

The mobile devices and the sensor nodes were positioned on the floor. In the room were present tables and chairs, which we expect to have effect on the propagation conditions. However, we have not explicitly studied the impact of the environment and the proximity to obstacles on the performance. Further, no special attention was given to the orientation of the devices towards the sensor nodes. The latter was explicitly chosen since no such awareness is yet feasible in a real deployment.

Table 1. Moving patterns for the in-room and out-of-room scenario.

	in-room				out-of-room			
MD1	(0,0)	(2,0)	(4,0)	(6,0)	(0,-2)	(0,-8)	(0,-6)	(0,-4)
MD2	(2,0)	(4,0)	(6,0)	(0,0)	(0,-4)	(0,-6)	(0,-8)	(0,-2)
MD3	(4,0)	(6,0)	(0,0)	(2,0)	(0,-6)	(0,-2)	(0,-4)	(0,-8)
MD4	(6,0)	(0,0)	(2,0)	(4,0)	(0,-8)	(0,-4)	(0,-2)	(0,-6)

4 Evaluation of Bluetooth Inquiry-based Parameters

In this section we present our findings on the inquiry-related RSSI values and RR values collected during the in-room and out-of-room scenarios. The measurements are accompanied by a short discussion. For simplicity we will use *RSSI* instead of *inquiry-related RSSI* in the rest of the discussion.

4.1 Received Signal Strength Indicator

We begin with the in-room scenario. For each of the mobile devices the RSSI values registered by each sensor node (SN) are continuously monitored while the devices are moved over positions (0,0), (2,0), (4,0) and (6,0) according to the moving patterns shown in Table 1. Each device has resided at each location for at least 1 min. Measurements were continuous, i.e., the radio channels were constantly scanned. An event can be defined as the detection of a device in a unit of time; thus multiple sensors can detect the same event time unit and one senor can detect the same device multiple times but only over different time units. Our observations show that most often there is one-to-one pairing between an event and an RSSI measurement.

First, we discuss the RSSI traces of a single mobile device, i.e., MD1, when moved over locations (0,0), (2,0), (4,0) and (6,0) in that order. As seen in Figure 2, SNs 2, 3 and 4 each registers a distinct, maximum RSSI value when MD1

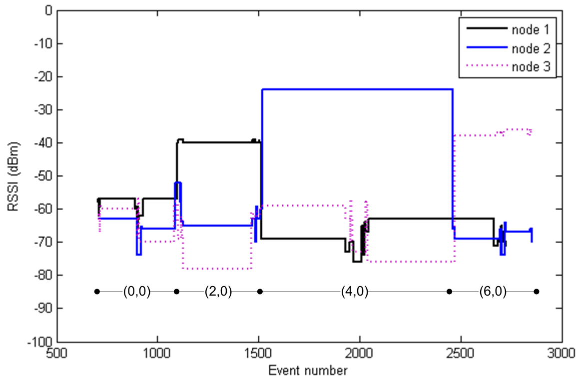


Fig. 2. Inquiry-related RSSI measurements of a single device (MD1) moving over positions (0,0), (2,0), (4,0) and (6,0) in that order.

is located next to it. However, no conclusive estimate can be derived when MD1 is at position (0,0). Hence, relying on absolute readings from a single node is vulnerable to device proximity to the node. A relative analysis of the readings of spatially disconnected nodes may be more robust. Therefore, we compare the max RSSI values measured by SNs 1, 2 and 3. Although SN1 has higher RSSI than the rest, suggesting that SN1 is closer to the target, no conclusive decision can be made. In order to increase the estimate precision we can further consider measurements from SNs 4, 5 and 6, see Figure 4(a). Clearly a SN in the proximity of an MD measures higher RSSI (-25 to -50 dBm for SNs 1, 2 and 3) than a remote SN (-55 to -70 dBm for SNs 4, 5 and 6).

In order to establish the minimalistic view of the network we take away SNs 2 and 5; the RSSI information available for localisation purposes is shown at Figure 3. The phenomenon, previously observed for MD1 at location (0,0), exhibits again for location (4,0) - it is difficult to derive precise location estimate, which is even further complicated by the higher RSSI values of SNs 4 and 6 compared to SN3.

The relative analysis proves even more efficient when applied to the out-of-room scenario, see Figure 4(b). All devices were positioned outside the room and moved as indicated by Table 1. Due to the longer signal path to the sensors and the presence of walls the maximum measured RSSI values by all SNs are lower compared to the in-room scenario. In addition to lower values, the maximum measured RSSI of an MD outside the room varies much less over the SNs.

We can conclude that rough location estimates based on the Bluetooth inquiry procedure are feasible especially with large number of sensor nodes. The results about MD1 at location (0,0) and (4,0), in the four SNs case, however suggest that fine-granularity location estimation, i.e., a meter, may be challenging. In order to investigate the issue further measurements are necessary.

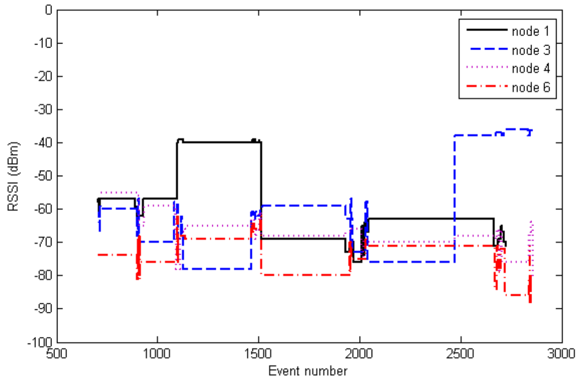
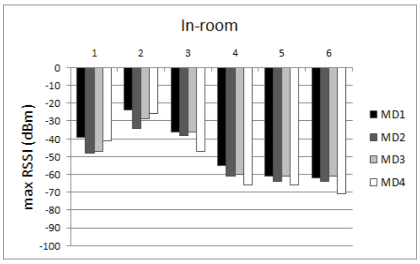
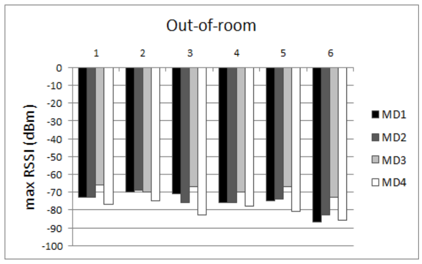


Fig. 3. Inquiry-related RSSI measurements of a single device (MD1) for four node deployment.



(a) in-room



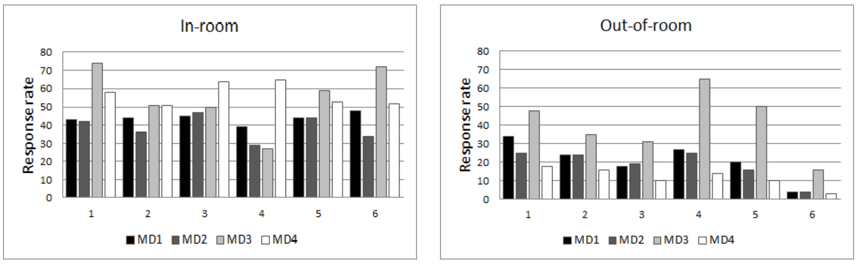
(b) out-of-room

Fig. 4. Maximum RSSI levels reported by the RSSI inquiry procedure for two cases: (a) in-room (MDs close to SNs 1, 2 and 3) and (b) out-of-room (MDs outside the room).

4.2 Response Rate

In addition to the RSSI values we have also collected measurements on the response rate of an inquiry again for the in-room and out-of-room scenario. The results are presented in Figures 5(a) and 5(b) respectively. It is interesting to observe a trend opposite to the RSSI measurements, namely, each MD registers only minor changes in RR from node to node for the in-room case while the differences are indicative for the out-of-room case. In the latter the RR seems to depend on the distance between MD and SN and the direction of signal propagation, i.e., SN4 (in line with the MDs) registers lower RR then SN1 but higher RR than SN3.

Another interesting observation is that the iPhone (MD3) has much higher RR than the other MDs. We explain that with the difference in how a device manages its 'discoverable' mode. In order to save battery the operation system may 'hide' the device after a pre-defined period of time. This was the case for



(a) in-room (b) out-of-room

Fig. 5. Response rate of Bluetooth inquiries for two cases: (a) in-room (MDs close to SNs 1, 2 and 3) and (b) out-of-room (MDs outside the room).

all MDs but the iPhone, which caused them to be 'hidden' during repositioning. After moved to the new location all phones were set in the discoverable mode.

5 Concluding Remarks

Based on the performed measurements we can conclude that an indoor localisation based on the Bluetooth inquire procedure is possible given that one wants to locate targets on a rough position granularity, i.e., at room-level. Finer location estimates, e.g., within few meters are in our opinion challenging. We also believe that both parameters inquiry-related RSSI and inquiry response rate should be used in combination to provide higher reliability of the estimate.

Although insightful the performed measurements introduce several concerns. The measurements suggest that both RSSI and RR may be device-specific or even depending on the particular sensor node. For example, SN2 has persistently higher detected RSSI than SN1 and SN3. These hardware oriented issues are accompanied by concerns about interference and location-specific propagation effects such as signal reflection. Further, we acknowledge the fact that higher node mobility can lead to changes in the measurements. In order to resolve these open issues we intend to perform more extensive data measurements.

References

1. M. S. Bargh and R. de Groote. Indoor localization based on response rate of Bluetooth inquiries. In *Proc. of 1st ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, MELT '08, pages 49–54. ACM, 2008.
2. P. Bellavista, A. Küpper, and S. Helal. Location-based services: Back to the future. *IEEE Pervasive Computing*, 7:85–89, 2008.
3. Byoung-Suk C., Joon-Woo L., Ju-Jang L., and Kyoung-Taik P. Distributed sensor network based on RFID system for localization of multiple mobile agents. In *Wireless Sensor Network*, volume 3, pages 1–9. Scientific Research, 2011.

4. M. Ciurana, F. Barceló-Arroyo, and S. Cugno. A robust to multi-path ranging technique over IEEE 802.11 networks. *Wireless Networks*, 16:943–953, 2010.
5. C. Fuchs, N. Aschenbruck, P. Martini, and M. Wieneke. Indoor tracking for mission critical scenarios: A survey. *Pervasive Mobile Computing*, 7:1–15, 2011.
6. S. Gezici, Zhi T., G.B. Giannakis, H. Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *Signal Processing Magazine, IEEE*, 22(4):70 – 84, 2005.
7. Y. Gwon and et al. Robust indoor location estimation of stationary and mobile users, 2004.
8. A. Haeberlen, E. Flannery, A.M. Ladd, A. Rudys, D.S. Wallach, and L.E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. of 10th annual international conference on Mobile computing and networking*, MobiCom '04, pages 70–84. ACM, 2004.
9. Simon Hay and Robert Harle. Bluetooth tracking without discoverability. In *Proc. of 4th International Symposium on Location and Context Awareness*, LoCA '09, pages 120–137, Berlin, Heidelberg, 2009. Springer-Verlag.
10. M.B. Kjaergaard. A taxonomy for radio location fingerprinting. In *Proc. of 3rd international conference on Location-and context-awareness*, LoCA'07, pages 139–156. Springer-Verlag, 2007.
11. A.M. Ladd, K.E. Bekris, A. Rudys, G. Marceau, L.E. Kavraki, and D.S. Wallach. Robotics-based location sensing using wireless ethernet. In *Proc. of 8th annual international conference on Mobile computing and networking*, MobiCom '02, pages 227–238. ACM, 2002.
12. A. Madhavapeddy and A. Tse. Study of bluetooth propagation using accurate indoor location mapping. ubicomp 2005. In *Proc. of 7th International Conference on Ubiquitous Computing (UbiComp*, pages 105–122, 2005.
13. A.K.M. Mahtab Hossain, H. Nguyen Van, Y. Jin, and W.S. Soh. Indoor localization using multiple wireless technologies. In *Proc. of Mobile Adhoc and Sensor Systems, MASS 2007.*, pages 1 –8, 2007.
14. I. Martin-Escalona and F. Barcelo-Arroyo. A new time-based algorithm for positioning mobile terminals in wireless networks. EURASIP Journal on Advances in Signal Processing, 2008.
15. K. Muthukrishnan, M. Lijding, and P. Havinga. Towards smart surroundings: Enabling techniques and technologies for localization. In *In: Proc. of 1st International Workshop on Location and Context-Awareness (LoCA), SpringerVerlag*, 2005.
16. N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proc. of 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 32–43. ACM, 2000.
17. M. Rondinone, J. Ansari, J. Riihijärvi, and P. Mähönen. Designing a reliable and stable link quality metric for wireless sensor networks. In *Proceedings of the workshop on Real-world wireless sensor networks*, REALWSN '08, pages 6–10. ACM, 2008.
18. R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10:91–102, 1992.
19. G. Zhang, S. Krishnan, F. Chin, and C.C. Ko. UWB multicell indoor localization experiment system with adaptive TDOA combination. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5, 2008.

Automated Merging in a Cooperative Adaptive Cruise Control (CACC) System

Wouter Klein Wolterink, Geert Heijenk, Georgios Karagiannis
University of Twente, Enschede, The Netherlands
{w.kleinwolterink, geert.heijenk, karagian}@utwente.nl

Abstract. Cooperative Adaptive Cruise Control (CACC) is a form of cruise control in which a vehicle maintains a constant headway to its preceding vehicle using radar and vehicle-to-vehicle (V2V) communication. Within the Connect & Drive[1] project we have implemented and tested a prototype of such a system, with IEEE 802.11p as the enabling communication technology. In this paper we present an extension of our CACC system that allows vehicles to merge inside a platoon of vehicles at a junction, i.e., at a pre-defined location. Initially the merging vehicle and the platoon are outside each other's communication range and are unaware of each other. Our merging algorithm is fully distributed and uses asynchronous multi-hop communication. Practical testing of our algorithm is planned for May 2011.

Keywords: automated merging, CACC, ITS, V2I, V2V

1 Introduction

Automated driving has long since been subject of research, especially when it comes to driving in platoon formation (see [1], [2], [3]). Current research generally focus on controlling the driving speed of a vehicle, thus keeping the headway to the preceding vehicle constant – steering is left to the human driver. One example of a platoon driving system is cooperative adaptive cruise control (CACC). Within the Connect & Drive[1] project we have implemented and tested a prototype CACC system, see Fig. 1.

Research on merging maneuvers within platoons can be found in e.g., [2] and [4]. However, their goal was to optimize the merging procedure from the point of the merger's benefits. Our approach focuses on the realization of a merging manoeuvre where the disturbances on the highway are minimized.

The goal of this paper is to present an extension to CACC that allows for automatic merging at a freeway junction. This extension consists of both hardware (an added road side unit, or RSU) and software (both on the RSU and the CACC vehicles). The RSU is responsible for tracking merging vehicles, estimate their arrival at the junction, and communicate this to the freeway vehicles. The CACC control algorithm has been adapted to allow for gap creation.

[1] This work is supported by the Dutch NL Agency/HTAS project Connect & Drive, Project no. HTASD08002

The outline of this paper is as follows. The key points of our CACC system are highlighted in Section 2. In Section 3 an overview of the merging application is given, identifying the different parts and their roles. In Section 4 the extended CACC control algorithm is specified. We conclude this paper in Section 5.

2 Cooperative adaptive cruise control

CACC is a form of cruise control in which the speed of vehicles is automatically controlled in a cooperative matter using a front-end radar and V2V communication. Because of the short reaction time of CACC compared to human drivers, vehicles can drive relatively close together (time headway < 1s), forming platoons. The goals of CACC include increasing the capacity of the road network and decreasing vehicle emissions. For details about the control aspects of our CACC system see [1].

The specific CACC system considered here has been based on 802.11p. All vehicles periodically (10 Hz) transmit a one-hop broadcast packet, containing necessary vehicle information such as its location, speed, and acceleration. Based on radar input and received broadcast packets the CACC control algorithm constantly adapts its desired acceleration to keep the vehicle's headway to its predecessor constant. The desired acceleration is the CACC's input to the engine controller. The desired headway can be set by the driver, or can be overruled by the CACC system for safety reasons.

Fig. 1. Four CACC operated vehicles during practical testing.

3 The merging application

Figure 2 gives a sketch of the considered merging scenario. A mixed CACC/non-CACC platoon is driving along the freeway. A merging vehicle (merger for short) approaches the freeway and will join the flow of traffic at the merge area, where it is expected to arrive at about the same moment as the platoon. The merging vehicle and the platoon are initially unaware of each other. For the merger to be able to join the flow of traffic, a gap within the platoon is required that is of sufficient size for the merger to merge inside the platoon. We refer to this gap as the *merging gap*. This gap must be created automatically by the CACC system, i.e., without any intervention of the CACC drivers. The gap must also be properly aligned with the merger, i.e., it should be at about the same position as the merger when the merger reaches the merge area. When this is the case the driver of the merging vehicle will manually perform the merge manoeuvre.

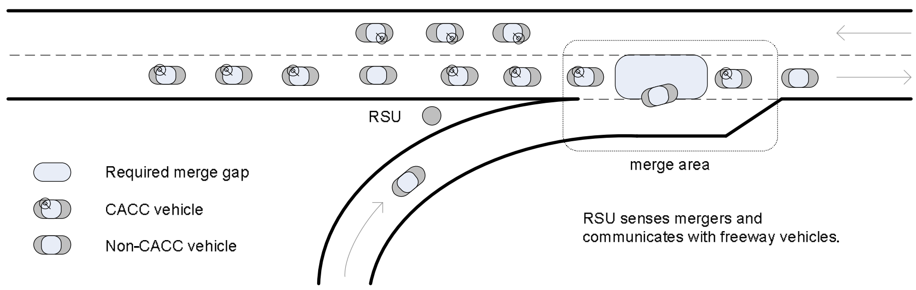


Fig. 2. The CACC merging scenario at a freeway junction.

To be able to judge when the approaching non-CACC vehicle will reach the merge area we employ an RSU that is able to sense the merging vehicle and estimate (i) its arrival time at the merge area, and (ii) the size of the required merging gap in the platoon. Details on how to perform such an estimation are out the scope of this paper. This could be utilizing vehicle-to-infrastructure communications, if the merger is equipped with communication capabilities.

Having performed the estimation the RSU communicates its outcome by means of periodical 802.11p broadcasts, similar to how CACC vehicles broadcast. In this way CACC vehicles that are within reception range of the RSU are made aware of the merger's approach. To support a larger communication range CACC vehicles include any estimation they have received directly from the RSU in their own broadcasts.

4 The extended CACC control algorithm

Figure 3 shows the state diagram of our extended CACC merging control algorithm, to decide whether or not a vehicle should create a merging gap. A gap is created by doubling the desired CACC headway. The goal of the algorithm is to have one vehicle create a gap, in a distributed fashion with asynchronous communication. Vehicles indicate that they are creating a gap by raising a flag in their periodical broadcast.

By default a vehicle operates in CACC mode with the default desired headway. When the vehicle receives a broadcast (either directly from the RSU or forwarded by a vehicle) that contains information about a new merging vehicle, the vehicle first checks if some other vehicle is already creating a gap for that specific merging vehicle. If not, then the vehicle estimates whether it will be inside the required merging gap. If so then it will double its desired CACC headway. It will keep this large headway until (i) someone has merged in front, (ii) the vehicle has passed the merge area, or (iii) a vehicle with a higher ID was detected creating a gap. In all cases the vehicle reverts back to CACC with default headway. Front-side merging is detected by the vehicle's radar.

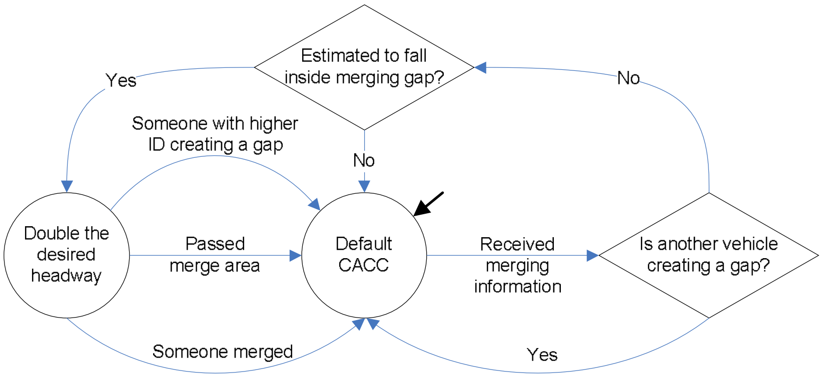


Fig. 3. State diagram of the extended CACC control algorithm.

5 Conclusions

We have presented a fully distributed CACC merging application that allows for automated merging using asynchronous communication. The application uses an RSU to detect mergers and to calculate the required merging gap. The extended CACC control algorithm ensures that a single merging gap is created inside the platoon. The merger may be non-CACC operated. Currently the algorithm has been implemented and tested in Simulink (see [5]) – practical tests are planned for May 2011.

In earlier work (see [6]) we investigated the communication aspects of our CACC merging application. In a follow-up project to Connect & Drive we wish to apply our experiences, both w.r.t. communication- and control engineering aspects, to develop an improved merging application that can be deployed on a large scale.

References

1. Naus, G., Vugts, R., Ploeg, J., Van de Molengraft, M., Steinbuch, M.: Cooperative adaptive cruise control, design and experiments. American Control Conference, USA, 2010.
2. Hsu, A., Sachs, A., Eskafi, F., Varaiya, P.: The Design of Platoon Maneuvers for IVHS. American Control Conference, 1991.
3. Dhevi Baskar, L., De Schutter, B., Hellendoorn, H.: Hierarchical Traffic Control and management with Intelligent Vehicles. In Proc. of IEEE IVS, 2007, pp. 834 -839.
4. Halle, S. Chaib-draa, B., Laumonier, J.: Car platoons simulated as a multiagent system. In: Proceedings of the 4th Worksshop on Agent-Based Simulation, 2003, pp. 57-63.
5. Simulink, http://www.mathworks.com/products/simulink/
6. Klein Wolterink, W., Heijenk, G.J., Karagiannis, G.: Constrained Geocast to Support Cooperative Adaptive Cruise Control (CACC) Merging. In: Proceedings of the Second IEEE Vehicular Networking Conference (VNC 2010), 13-15 Dec 2010, Jersey City.

Leveraging Process Models to Optimize Content Placement - An Active Replication Strategy for Smart Products

Markus Miche[1], Marcus Ständer[2], and Marc Brogle[1]

[1] SAP Research Switzerland
Kreuzplatz 20, 8008 Zürich, Switzerland
{markus.miche, marc.brogle}@sap.com

[2] Telecooperation Group, Technische Universität Darmstadt
Hochschulstrae 10, 64289 Darmstadt, Germany
staender@tk.informatik.tu-darmstadt.de

Abstract. Along the entire product lifecycle, users are overwhelmed by the increasing number of features and the diversity of technical products. *Smart* products with embedded computing and networking functionality are a promising approach for tackling this issue. Smart products make use of process models to interact with and guide their users in a proactive manner. While smart products require huge amounts of content across their lifecycle to realize user guidance, they only possess limited storage capacities. Hence, there is a need for specific mechanisms to distribute content required by smart products in order to make it available and accessible with low latency. This paper presents an active replication strategy that leverages process models associated with smart products to optimize content placement. The proposed strategy results in enhanced content availability and query efficiency, and leads to improved user-perceived performance.

Keywords: Smart Products, Content Replication, Workflow Management System, Distributed Storage

1 Introduction

Technical products ranging from consumer goods such as microwaves, to vehicles and airplanes are characterized by an increasing number of functions, features, and customization options. This bears a new level of complexity for users dealing with them. Hence, there is a need for novel technologies to assist and guide users in all phases of the product lifecycle from manufacturing to use and maintance up to refurbishment and disposal.

More than one decade after Mark Weiser's visionary article "The Computer of the 21st Century" [14], one can encounter an increasing number of intelligent physical objects in everyday life. This ranges from objects equipped with smart labels such as RFID or NFC tags to *smart products* with embedded computing, sensing, and networking functionalities. Smart products are able to communicate amongst each other without relying on pre-installed communication infrastructures. Moreover, based on distributed process models, smart products are able to interact with and assist their users as well as to autonomously work together to fulfill certain tasks [11]. A common approach for modelling such distributed processes is the XML-based process definition language XPDL [13]. Hence, smart products represent a promising technological advance to approach the above-stated issue.

In order to assist and guide their users, smart products require a lot of information. This includes pre-constructed content such as graphical user interface elements, manuals of different formats (e.g., text, audio, or video), as well as executable code. Moreover, smart products make use of information acquired by sensors that may be either attached to them or available in their environment. To achieve a high-level of user-perceived performance when interacting with smart products, this content has to be highly available and accessible with low latency. In a perfect world where smart products possess "infinite" storage capacity and are always connected to backend systems via broadband communication technologies, these objectives would be easily achievable. However, especially regarding mass production, smart products are typically resource-constrained with respect to their storage, communication, and processing capabilities. Hence, despite the technological advances, smart products are in general neither able to store all information required during their lifecycle on-board, nor are they capable of connecting to business systems at all times. Consequently, as stated by [4], there is a need for "*[..] intelligent data staging and pre-staging, so that data can be placed close to where the users will be when they need it (particularly in slow or unreliable communications situations)*".

This paper presents an active content replication strategy, which leverages structure and states of workflows to predict future content needs and optimize content placement. For this purpose, the paper presents an extended annotation schema for XPDL workflows that enables modeling of workflow-related content needs. Moreover, an iterative prediction algorithm is proposed that estimates future content needs based on the above-mentioned annotations taking into account dynamics of smart products environments (i.e., cooperation of smart products cannot be strictly planned at workflow initialization). Finally, a content placement mechanism is presented that collects and places content where it will most likely be accessed in upcoming activities. This novel replication strategy approaches the shortcomings of pure reactive "on-demand" data staging strategies. It enhances content availability and query efficiency in order to eventually optimize user-perceived performance when interacting with smart products.[3]

---

[3] The trade-off between content availability and consistency in distributed systems stated by the CAP theorem is out of scope of this paper. An integration of the

The remainder of this paper is structured as follows: Section 2 describes existings content replication strategies and points out their limitations with respect to the challenges of smart products environments. Thereafter, Section 3 presents an architecture for smart products covering a distributed storage framework plus the workflow management system Methexis. The main contribution and results of an initial evaluation are presented in Sections 4 and 5, respectively. The paper concludes in Section 6 with an outlook on future work.

2 Related Work

Replication strategies are part of most distributed storage mechanisms. Replication strategies are applied in Content Distribution Networks (CDN) to reduce access delay of client requests and for balancing load among replica servers. Opportunistic networks make use of content replication in order to distribute content between intermittently connected nodes according to the store-carry-and-forward paradigm. Finally, Peer-to-Peer (P2P) content distribution systems apply replication strategies to enhance content availability, durability, and query efficiency.

Based on a detailed study on related work, an overview of design decisions for replication strategies is presented in [9]. With respect to the replication strategy presented in Section 4, the two most important attributes of this overview are *replication schedule*, i.e., when to replicate content, and *replication knowledge*, i.e., which information to use for determining number and placement of replicas. First, replication schedule captures *static* approaches, which assume a priori knowledge about access distribution, and *dynamic* approaches that adapt number and placement of replicas during runtime. Second, replication knowledge distinguishes between *reactive* replication strategies that purely rely on past observations (e.g., access history) and *active* replication strategies that further include estimates on upcoming content needs [7].

While static replication strategies with a priori knowledge about upcoming content needs are not applicable to dynamic smart products environments, most existing dynamic replication concepts apply pure reactive strategies [6, 12]. There are only few approaches that consider active replication strategies in order to improve number and placement of replicas which eventually enhances the above-stated quality attributes.

MDCDN, the mobile dynamic CDN proposed by [1] applies the *statistical demand forecasting* method "double exponential smoothing" as part of an active replication strategy. This prediction enables nodes in MDCDN to dynamically pre-fetch content that is likely being requested in the upcoming period, thus reducing latency of future requests. *Sequence predicition algorithms* represent

proposed replication strategy with the consistency maintenance concepts presented in [9] is subject to future research.

similar means for estimating upcoming content needs. As an example, the algorithm FxL proposed by [5] enables nodes to not only respond to requests with the actual result but to moreover dispatch the result of the request with the highest probability of being requested next based on analyses of related request histories. This additional information is cached by the requester and – in most cases – leads to reduced acccess latency of future requests. Other approaches leverage *social information* to optimize number and placement of replicas. While ContentPlace applies social-oriented policies such as "most frequently visited" [3], the concept proposed in [8] analyzes data from social networks in order to distribute and share content among users. Finally, [2] utilizes worklets, which represent self-contained sub-workflows including execution rules, to explicitly model upcoming content needs. These rules are evaluated taking into account context information as well as user profiles.

However, to the knowledge of the authors, none of the above-presented approaches copes with the dynamics of smart products environments. While the basic idea of the proposed strategy is comparable with the approach presented in [2], it further captures dependencies between activities as well as context-dependent content needs. Moreover, the proposed prediction algorithm approaches the dynamics of smart products environments by analyzing usage history information and by performing iterative reevaluations taking into account dynamically changing context information. Hence, the proposed active replication strategy combines proven concepts, but adapts and extends them with respect to the specifics and challenges of smart products.

3 Architecture Description

The proposed workflow-based active replication strategy is based on a distributed storage framework and the workflow management system Methexis. The main components of these two modules regarding the replication strategy as well as their interrelations are depicted in Fig. 1. Moreover, the illustration presents the relation of the two modules to the communication middleware and the context manager of the generic platform for smart products that is currently being developed in the course of the SmartProducts project [10].[4] The communication middleware organizes smart products in a hybrid P2P overlay network, which captures their resource limitations and heterogeneity, and provides event-based communication according to the publish/subscribe messaging pattern.

Workflow Management System. The workflow management system Methexis is a lightweight version of the Open Business Engine. Its basic structure is organized in a layered architecture enabling the separation between workflow *Administration Tools*, the executing *Workflow Engine*, and (third party) services

[4] For the sake of clarity, the relation to other modules such as the Interaction Manager and the Access Control are not reflected. For more information, see `http://www.smartproducts-project.eu`.

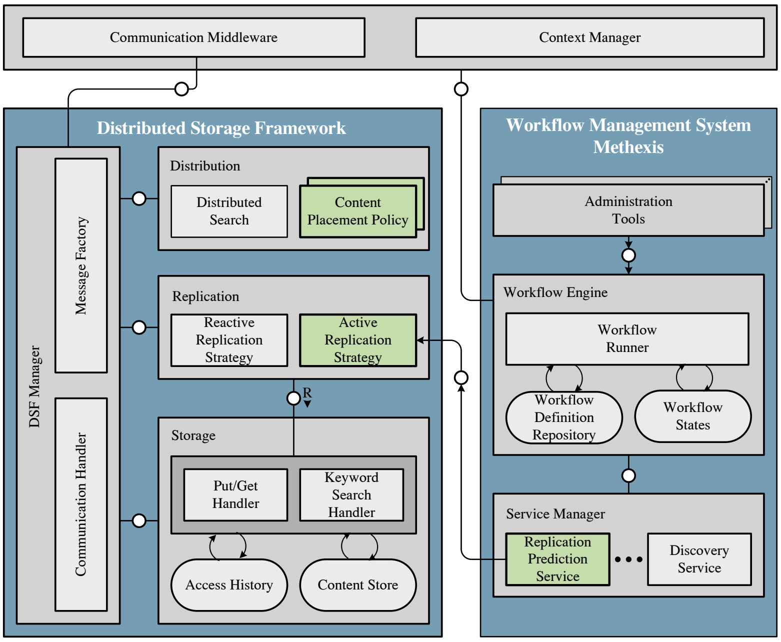


Fig. 1. Architecture View

that can be plugged in using Methexis' *Service Manager*. Administration tools can make use of a client API to get information from or send commands to the engine (e.g., getting lists of workflow definitions, starting a workflow instance).

The middle layer provides the central functionality of Methexis: the XPDL model and the workflow engine. XPDL models are standardized representations of workflows. The workflow engine plans, checks, and manages execution and states of workflows. For example, if an activity is finished then the workflow engine is responsible for checking conditions of outgoing transitions and deciding on the transitions to follow. The engine is furthermore responsible for finding and starting workflows on remote products using the communication middleware.

On the lowest layer, a service programming interface provides the possibility to enrich the functionality of the engine with different kinds of services. In this paper, Methexis is extended with a new service: the *Replication Prediction Service*. It enables predicting future activities of a given workflow as well as deriving related content needs.

Distributed Storage Framework. The *DSF Manager* provides the public API of the distributed storage framework and orchestrates its core components. Based on a communication handler and a message factory, it hides any communication specifics from the core components of the distributed storage framework and provides the protocol of the latter. The *Storage* component encapsulates the on-board content store and enables local storage (put) and retrieval (get) of content. Moreover, it stores content-related metadata to facilitate keyword search. According to the concept MFR proposed by [6], the component maintains access histories in order to determine content to be stored off-board in case of limited storage capacity. Based on the network topology maintained by the communication middleware and its content location and routing functionality, the *Distribution* component of the distributed storage framework enables location and retrieval of content stored off-board (i.e., on other smart products or backend systems). This includes distributed search functionality as well as off-board content placement policies.

The *Replication* component provides the actual replication functionality in order to enhance content availability, durability, and query efficiency. It covers a reactive strategy that adapts Top-K MFR described in [6] using access history information maintained by the storage component. This reactive strategy is complemented by the novel workflow-based active replication strategy (see Section 4). As depicted in Fig. 1, this replication strategy makes use of the *Replication Prediction Service* provided by Methexis and the off-board placement policies of the *Distribution* component in order to optimize number and placement of replicas according to the estimation of future content needs.

4 A Workflow-based Replication Strategy

4.1 Extended Workflow Annotation Schema

XPDL workflows consist of activites and transitions, which connect activites and may be annotated with conditions. By default, the schema of XPDL workflows supports the definition of simple and complex types of *extended attributes* that can be used to annotate activities with additional information. In order to model activity-related content needs such as user interface elements, information for involved users, or executable code, a specific complex type of extended attribute has been defined. The corresponding part of the extended workflow schema is depicted in Fig. 2.

The schema supports two ways of modeling content needs. In case activity-related content is explicitely known at design time, the item *itemId* can be used to specify content identifiers. In case an activity requires content that is generated/updated by another activity, i.e., there is a dependency to another activity, the optional attribute *dependency* can be used to model dependencies to other activities. Content needs that are not explicitely known at design time can be modeled using the item *itemMetadata*. This item can be associated with a set of metadata modeled as key-value pairs with keys being defined according to a controlled vocabulary of content-related metadata attributes.

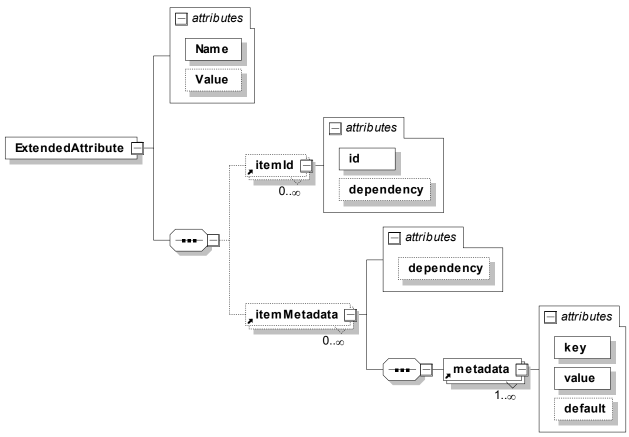


Fig. 2. Schema Extension for Annotating Activities with Content Needs

Corresponding values can be modeled either explicitly or using a parameterized approach, which enables the incorporation of context information. As an example, imagine information to be presented to the user involved in an activity. Instead of statically defining a certain format (e.g., `value=''audio''`), the parameterized approach enables taking into account user preferences (e.g., `value=''{userprofile.preferredFormat}''`). Finally, if user preferences cannot be resolved when determining content needs, the attribute *default* of *itemMetadata* enables the definition of a default value (e.g., `default=''audio''`). This way, the extended schema is not limited to explicit specification of content needs but moreover enables implicit, context-aware modeling taking into account dynamics of smart products environments.

4.2 Prediction Algorithm

Prediction of future activities in order to derive upcoming content needs, which is simply referred to as *predicting future content needs* in the following description, is easy if a workflow is locally executable and if its structure is purely sequential. However, smart products are able to execute more complex workflows, including splits in the control flow as well as activities that can be executed by remote smart products. An algorithm for analyzing and predicting future content needs of such context-aware and distributed workflows has to take different aspects into consideration: (i) how many future activities shall be taken into account, (ii) how to determine which smart product will execute a remote activity, and (iii) when the estimation should be accomplished in order to minimize, e.g., the workload of the responsible product. In the following, an approach to these questions is introduced and explained based on the workflow depicted in Fig. 3.

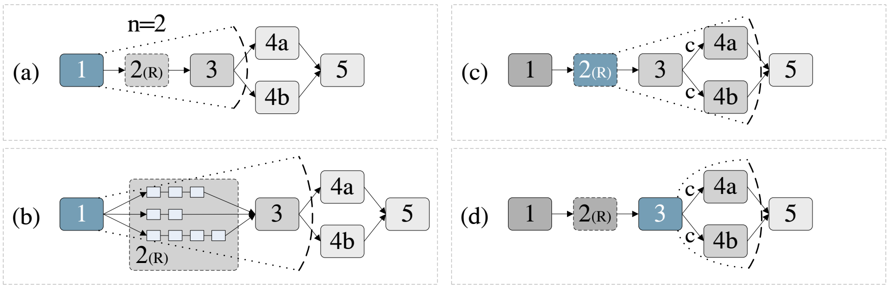


Fig. 3. Prediction Strategy

The basic idea of the algoritm is to predict future content needs at the start of each activity. When activities with content annotations are found, the distributed storage framework is notified and all content annotations are passed for the next processing step explained in Section 4.3. To meet the requirements of limited storage capacity and to reduce content misplacement, a *prediction boundary* n (see dashed sector in Fig. 3a) is introduced. The prediction boundary describes the *maximum* number of future activites (i.e., hops in the workflow graph), the prediction algorithm is allowed to check for annotated content needs (middle-gray in Fig. 3).

In case of *uncertainty*, the algorithm reduces the *prediction range* r, which is the actual number of activites the algorithm predicts content needs for, with $r \leq n$. This happens in case of *OR* splits. While *AND* splits do not influence prediction – all transitions need to be followed –, *OR* splits have conditioned transitions that often relate to context information, i.e., not all transitions might be followed. For this reason, it is difficult to evaluate conditions that are too far in the future, taking into account the dynamics of smart products environments. To approach this challenge, the algorithm makes use of history information in order to find the most probable transition, i.e., the transition that has been followed most often in the history period under consideration. Moreover, to limit the risk of content misplacement, the prediction range is reduced to cover only the activity following the *OR* split (see Fig. 3d). When the activity preceding the *OR* split is being executed, context becomes more reliable and the actual condition can be evaluated to increase prediction accuracy.

Another source of uncertainty are *remote activities*. In general, activities can be separated into local activities and remote activities. Local activities, such as activity 1 in Fig. 3, are executed on the product that runs the workflow, while remote activities are performed by other smart products in the environment (see activity 2 in Fig. 3). Since remote activities are assigned to remote smart products during runtime, it is not possible to reliably predict the product that will execute the activity. To cope with this challenge, the discovery service of the workflow engine, see Fig. 1, is used to identify smart products available in the environment that might be suitable candidates for executing the remote activity at a later stage (see Fig. 3a and 3b). This approach is combined with history information about earlier usage of the available smart products in the context of the given workflow. If history information is available, the prediction of future content needs is limited to the first activity of the most probable remote workflow. Otherwise, no prediction will be performed because of the high level of uncertainty. However, as depicted in Fig. 3b, this does not affect the local prediction range, since the local control flow is not influenced by the assignment of remote activities.

Hence, the prediction algorihtm has three main steps. First, at the beginning of each activity, the algorithm determines the prediction range taking into consideration the specific rules for conditioned transitions. Second, for all activites within this range, the algorithm checks for activities that need to be reevaluated because they are followed by context-dependent *OR* splits and for activities that have not been predicted in previous iterations. For the latter, the algorithm derives activity-related content needs taking into consideration the rules for remote activities and forwards them to the distributed storage framework. Finally, the algorithm checks predicted activities for *content dependencies*. Content dependencies might occur, in case activities require content that is generated by previous activities of the workflow. In case of dependencies related to the active activity, the prediction algorithm is re-invoked after the activity has been accomplished in order to reevaluate and resolve content dependencies.

4.3 Content Retrieval and Placement

The retrieval and placement of content needed by upcoming activities is the final step of the workflow-based active replication strategy. Based on the input provided by the prediction algorithm, the mechanism optimizes content placement in order to enhance content availability and query efficiency using the primitives of the hybrid P2P overlay network maintained by the communication middleware (see Section 3).

For each local activity, the mechanism analyzes content needs taking into account context information in case of parameterized values and checks its local availability. In case the required content is not available locally, the mechanism tries to create a local replica. For this purpose, it dispatches a distributed search request to the parent node of the active node (i.e., its super-node in the hybrid P2P overlay network). In case the requested content is stored in the super-node's cluster (i.e., stored either by the super-node or by at least one of its children), the super-node sends a replica to the active node. Otherwise, the super-node distributes the search request to other super-nodes using the content location and routing functionality of the P2P overlay network, collects and analyzes the results, and sends a replica back to the active node. If the replica cannot be stored by the active node because of storage limitations, it is placed on its super-node.

For remote activities, the mechanism tries to place a replica of the required content on the super-node of the active node (or locally, in case it is a super-node). This is based on the assumption that workflows are generally executed in spatially limited environments, in which nodes are organized in only a few clusters. Hence, it is likely that the active node and nodes whose content needs have been predicted belong to the same or spatially close cluster. Placing content into the cluster of the active node using the above-described mechanism enhances content availability and query efficiency in most cases. Even further, the procedure saves storage capacity on the target node (remember that super-nodes are normally elected based on different attributes such as storage capacity or uptime).

The proposed content retrieval and placement approach is applicable to most existing hybrid P2P overlay networks without posing additional requirements. An example of how this functionality was applied to a hybrid P2P overlay network, with super-nodes being organized in a structured way, is provided in [9].

5 Initial Qualitative Evaluation

In order to assess the impact of the proposed replication strategy on user-perceived performance, an initial qualitative evaluation has been conducted. Similar to a cost/benefit analysis, the overhead of the replication strategy (e.g., inaccurate predicion of future activities, misplacement of content) is evaluated and compared to the potential enhancement of content availability and query efficiency.

By pre-staging content required by upcoming activities of a workflow, the proposed replication strategy clearly enhances content availability and reduces access latency of future requests. Especially with respect to smart products being equipped with low storage capacity and low-bandwidth communication technologies, the proposed replication strategy is highly valuable. In a proof of concept implementation of the distributed storage framework and Methexis done in the course of the SmartProducts project, the on-demand location and retrieval of activity-related content strongly affected user-perceived performance. Hence, the "proactive" fetching of content used in future activities has great potential to enhance performance of workflows executed in smart products environments.

Despite the above-stated potential, the proposed replication strategy may result in certain overhead in the data flow of the workflow. First, in the case of context-dependent OR splits and remote activites, the forecasts of the prediction algorithm are made under uncertainty. By integrating history information in the prediction algorithm (e.g., which path of an OR split has been used how often with recent decisions being given extra weight), this uncertainty and the resulting overhead can be reduced in the long term. Second, the dynamics of smart products environments – in particular mobility and churn – may result in additional overhead. While mobility of smart products can merely lead to non-optimal placement of content (e.g., a smart product for which content has been pre-staged changes its super-node), churn can result in content being pre-staged for a smart product that is no longer involved in the execution of the workflow. The prediction algorithm approaches these challenges by performing iterative reevaluations of its estimations and by limiting the prediction of content needs related to remote activities to only the first activity. Moreover, the active replication strategy decouples data flow from the actual control flow of workflows. This way, imperfect replication and placement of activity-related content solely affects data flow, which – except for the pre-fetching overhead – is even in the worst case still of similar performance than traditional on-demand staging approaches regarding content availability and query efficiency. Hence, except for the computation complexity of the prediction algorithm, the control flow of the workflow is not affected by the proposed replication strategy.

6 Conclusion & Outlook

In order to enhance content availability and query efficiency in dynamic smart products environments, this paper proposes a novel active replication strategy that leverages distributed processes modeled as XPDL workflows to predict future content needs and optimize content placement. For this purpose, the paper makes three main contributions: (i) an extension of the XPDL workflow schema that allows explicit, implicit, and parameterized modeling of activity-related content needs, (ii) a prediction algorithm that estimates content needs of future activities in dynamic environments taking into account conditioned transitions and context-dependent remote activities, and (iii) a content retrieval and placement mechanism that analyzes predicted activities and related content needs in order to retrieve and place content close to where it will be most likely accessed in the upcoming workflow execution.

Moreover, the paper presents the results of an initial qualitative evaluation that points out the benefit of the proposed replication strategy. A more comprehensive evaluation including comparison with related work is subject for future research. For this purpose, a simulation model is currently being developed that reflects the specifics of smart products environments. Furthermore, future research includes enhancements of the replication strategy based on the lessons learned during simulation - in particular with respect to the prediction of content needs and the optimal content placement given the limited resources of smart products. For example, the replication strategy could take into consideration past activities of users, which could be maintained as part of the user profile, in order to reduce the strategy's overhead (e.g., should a certain information be pre-fetched in case a user has already seen it during a previous execution of the workflow). Results of the simulation-based evaluation as well as enhancements of the workflow-based active replication strategy will be presented in future publications.

Acknowledgement

Part of this research has been funded under the EC 7th Framework Programme, in the context of the SmartProducts project (231204).

References

1. Aioffi, W.M., Mateus, G.R., Almeida, J.M., Mendes, D.S.: Mobile dynamic content distribution networks. In: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems. pp. 87–94. Venice, Italy (2004)
2. Bahrami, A., Wang, C., Yuan, J., Hunt, A.: The workflow based architecture for mobile information access in occasionally connected computing. In: IEEE International Conference on Services Computing. pp. 406–413. Chicago, IL, USA (2006)
3. Boldrini, C., Conti, M., Passarella, A.: ContentPlace: social-aware data dissemination in opportunistic networks. In: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems. p. 203–210. MSWiM '08 (2008), ACM ID: 1454541
4. Franklin, M.: Challenges in ubiquitous data management. In: Informatics: 10 Years Back, 10 Years Ahead. p. 24–33. Springer Berlin/Heidelberg (2001)
5. Hartmann, M., Schreiber, D.: Prediction algorithms for user actions. In: Lernen - Wissen - Adaption. pp. 349–354. Halle, Germany (2007)
6. Kangasharju, J., Ross, K., Turner, D.: Optimizing file availability in Peer-to-Peer content distribution. In: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. pp. 1973–1981 (2007)
7. Karlsson, M., Karamanolis, C.: Choosing replica placement heuristics for wide-area systems. In: Distributed Computing Systems, 2004. Proceedings. 24th International Conference on. pp. 350–359 (2004)
8. Kröner, A., Schneider, M., Mori, J.: A framework for ubiquitous content sharing. Pervasive Computing, IEEE 8(4), 58–65 (2009)
9. Miche, M., Erlenbusch, V., Allocca, C., Nikolov, A., Mascolo, J.E., Golenzer, J.: D4.1.3: Final concept for storing, distributing, and maintaining proactive knowledge securely. Deliverable D4.1.3, EU FP7 SmartProducts (2011)
10. Miche, M., Schreiber, D., Hartmann, M.: Core services for smart products. In: 3rd European Workshop on Smart Products. p. 1–4. Salzburg (2009)
11. Mühlhäuser, M.: Smart products: An introduction. In: Constructing Ambient Intelligence. vol. 11, pp. 158–164. Darmstadt, Germany (2007)
12. Shen, H.: IRM: integrated file replication and consistency maintenance in P2P systems. IEEE Transactions on Parallel and Distributed Systems 21(1), 100–113 (2010)
13. Webster, P., Uren, V., Ständer, M.: Shaken not Stirred: Mixing Semantics into XPDL. 5th International Workshop on Semantic Business (2010)
14. Weiser, M.: The computer of the 21st century. SIGMOBILE Mob. Comput. Commun. Rev. 3, 3–11 (1999)

A Dynamic Pricing and Admission Control Scheme for Heterogeneous Services in Mobile WiMAX Systems

Fatma Ghandour, Mounir Frikha, and Sami Tabbane

Mobile Network Multimedia Radio Research Unit
Tunisian High School of Communications
Parc Technologique Elghazala, Tunisia
gh_fatma7@yahoo.fr
m.frikha@supcom.rnu.tn
sami.tabbane@supcom.rnu.tn

Abstract. In this paper, we propose a new pricing model for fixed and variable bandwidth demand services in multiservice WiMAX networks. The user's satisfaction level and the operator's revenue are the two keys for the pricing scheme we propose. We define a satisfaction degree metric which is used to calculate the price per resource unit for traffic with variable bandwidth demand. In moment of congestion, an interactive pricing technique is applied between connections of different types of service in order to cover the revenue decrease caused by the unsatisfied services. This pricing scheme is joined with an admission control algorithm that is needed to determine the type of connection that should be accepted when resources are limited. Numerical results proved the effectiveness of our approach by offering a reduced price for the unsatisfied services while at the same time keeping a stable revenue for the operator.

Keywords: Pricing, Wimax, Admission Control

1 Introduction

Through the last years, the success of broadband wireless access has incited the operators to give more interest to new technologies and recent standard like WiMAX, which can provide a variety of real time and non real time service over wide area and with high data rates. The two most known versions of WiMAX standard are 802.16d [1] for fixed systems and 802.16e [2] for mobile systems. The fixed version of the standard defines four types of traffic flow : Unsolicited Grant Service (UGS) for T1/E1 type telecom voice, real-time-polling services (rtPS) for video streaming and teleconferencing, non-real-time-polling services (nrtPS) for file transfer, and best effort (BE) services for non-delay sensitive communication. The mobile version added a fifth class of service called extended real-time-polling services (ertPS) and which is especially designated for voice traffic with silence suppression. For each of those service types, the standard specified a set of quality of service (QoS) parameters to be negotiated between the Subscriber Station (SS) and the Base Station (BS) and respected during the connection time.

With this variety of services and in a variable network conditions, designing a suitable pricing model for the users is not an easy task for the operator. In fact, When deploying this type of networks with reduced CAPEX, the operator aims at increasing its revenue while keeping attractive prices and ensuring a satisfying quality of service for the users. To face this tradeoff, operators are generally looking for new pricing models that can keep a continuous QoS provisioning while being profitable at the same time. In literature, we can find some pricing schemes that have been proposed for wireless environments. When applied to WiMAX networks, those schemes need to be adjusted to fit the requirements of such a multiservice environment, and generally, they only can be adequate for one or two service types. However, operators deploying WiMAX networks look for a complete pricing scheme, that englobe the five classes of service defined in the standard, and at the same time it need to be dynamic and take care of the network changing conditions. In this context, we propose in this work a dynamic pricing scheme that relates the operator's revenue to the QoS level offered to each user. In this scheme, the per unit of time is computed instantaneously according to the user's satisfaction degree, using dynamic pricing windows, so that he does not have to pay high for low QoS level. On the other hand, the choice of the type of service that will be accepted when resources become limited has a great impact on the increase or the decrease of the operator's revenue. That's why we associate the scheme with an admission control that decides which new connection will be added based on the system state.

In the rest of this paper, we will give a brief overview of some related work followed by a detailed description of the system model and the proposed scheme. Numerical results are analyzed in section 4 and the conclusion is given in section 5.

2 Related Work

Researchers are now giving more and more importance to pricing topic due to the diversity of wireless access technologies. The work in [9] presents a rich survey about the available pricing schemes that can be applied to wireless services. The authors give different classification of those parameter-based approaches and find out that the priority based ones are the most efficient pricing strategies from both of the user and the operator point of view. In [8], the authors were interested in heterogeneous networks, where different wireless access technologies coexist. They classified the services under wireless networks into 2 categories: Premium with fixed price and Best effort with dynamic price. A game theoretic approach was used to resolve the problem of competitive pricing models proposed by the operators for the best effort services, for two cases : when all of the service providers offer their proposed prices simultaneously and when one service provider can offer its prices before the others.

In literature, the proposed pricing schemes for WiMAX systems are generally joined together with admission control or resources allocation strategies. For example in [7] a cost based admission control was proposed using the Competitive On Line (COL) function which assigns a cost to a service based on its required amount of bandwidth. The authors in [10] were interested to OFDMA-based 802.16 systems, and investigate how the operator have to charge the allocated resources in order to control demand and maximize the revenue, but the work is only limited to two service classes. In [4], a pricing model has been proposed for real time services in WiMAX (UGS and rtPS), where the price per unit of time was related to a valuation function that depends on the transmission rate of each service flow. This model was extended for non real time services in [5] where the author proved that adopting a variable pricing model per allocated symbol performs better than a fixed pricing model. In [6], the authors presented two pricing-based distributed algorithms and related the system utility function to the transmission rate and the channels conditions.

3 Proposed pricing scheme

3.1 Network model

We consider a multiservice Mobile WiMAX environment, where the five classes of services (UGS, ertPS, rtPS, nrtPS, and BE) are carried simultaneously in the network. The operator aims at increasing its total revenue coming from those services with different QoS requirements. Therefore, he needs a pricing strategy that should take the service classifications and the traffic characteristics into account. In fact, traffic can be either bursty or continuous (with fixed rate). In WiMAX system, since bursty traffic has variable bandwidth demand that can be satisfied or not, depending on the available resources, services with QoS constraints need to be bordered by a minimum and a maximum rate value, which is the case for nrtPS, rtPS and ertPS services classes. For continuous traffic, it requires a constant granted amount of bandwidth under all network conditions: this is the UGS class. We classify the five classes of service defined for Mobile WiMAX into three types:

- Type 1, for Bursty traffic with QoS requirement: nrtPS, rtPS and ertPS classes.
- Type 2, for continuous traffic: UGS class.
- Type 3, for Bursty traffic with no QoS requirement: BE Class.

Based on this service classification, we propose an interactive pricing model that allows the operator to associate a *"Price Window"* for each service class, among which the price of a resource unit per unit of time will be calculated instantaneously after checking the satisfaction level of each user.

3.2 Pricing Scheme

The basic idea consists of assigning a specific $Price\ Window$ to each class of service. We note PW_j as the $Price\ Window$ of service class j and P_j^{min} and P_j^{max} as the borders of the window. The system will periodically calculate the instantaneous price of a resource unit Pu_j^t for this class of service, which must be within PW_j. The price associated to connection i that carries this type of service at time t is equal to the price of a resource unit per unit of time, multiplied by the obtained rate R_i :

$$P_{x,i}^t = Pu_{x,i,j}^t \times R_i \tag{1}$$

Where j refers to the service class, i to the connection, and x to the type of service according to the previous classification (type 1, type 2 or type 3).

3.3 The Instantaneous Price per Resource unit

This parameter needs to be updated periodically in order to associate each connection with the price value that corresponds to its satisfaction level at that time. Therefore we extend our definition of the $Satisfaction\ Degree$ metric that was originally defined for nrtPS traffic type in our previous work [11] to englobe all the other services. This metric can be considered as a QoS measure, since it consists of the ratio between the obtained and the requested QoS level. Equation 2 gives the general expression of the $Satisfaction\ Degree$ of connection i at time t, noted SD_i^t:

$$SD_i^t = \frac{R_i^t}{\tilde{R_i}^t \wedge R_i^{max}} \times (1 - \frac{D_i^t}{D_i^{max}}) \times (1 - \frac{J_i^t}{J_i^{max}}) \tag{2}$$

Where $\tilde{R_i} \wedge R_i^{max}$ refers to the minimum value between the requested rate ($\tilde{R_i}$) and the maximum sustained traffic rate (R_i^{max}) of user i. While D_i^t, J_i^t, D_i^{max} and J_i^{max} refer respectively to the measured delay, the measured jitter, the maximum tolerated delay and the maximum tolerated jitter.

Note that, since delay and jitter are not among the QoS constraints of nrtPS services, we can easily retrieve the same $Satisfaction\ Degree$ expression that we gave in [11]. For type 2 service (UGS), since it is supposed to be granted its maximum sustained traffic rate, only delay and jitter are considered in its $Satisfaction\ Degree$ expression, whereas for type 3 service (BE), since there is no QoS requirements, the service is considered as satisfied during the connection lifetime. Hence, the following expressions define the $Satisfaction\ Degree$ for each of the three types of services:

$$SD_i^t = \begin{cases} \frac{R_i^t}{\tilde{R_i}^t \wedge R_i^{max}} \times (1 - \frac{D_i^t}{D_i^{max}}) \times (1 - \frac{J_i^t}{J_i^{max}}) \ i \in \text{type 1} \\ (1 - \frac{D_i^t}{D_i^{max}}) \times (1 - \frac{J_i^t}{J_i^{max}}) \ i \in \text{type 2} \\ \infty \ i \in \text{type 3} \end{cases}$$

We will give in what follows the expression of the instantaneous price per resource unit that corresponds to each service type.

Connections of type 1 For this type of traffic composed of nrtPs, rtPS and ertPS services, the price $P_1^t(i)$ for connection i at time t depends on the calculated value of the *Satisfaction Degree*, SD_i^t. In the best cases, when the service is granted the rate that it needs and when jitter and delay are too small compared to the maximum tolerated values, the user is satisfied and the price should be very close to its maximum value, which we denoted as P_i^{max} and which represents the upper bound of the associated *Price Window*, (PW_i). If delay or jitter reach their maximum tolerated values, the *Satisfaction Degree* will be equal to zero, and, in this case, the operator will propose the lower bound of PW_i, P_i^{min}, as a price for this service flow. Between those two extreme states, the price changes dynamically within PW_i, according to the user's satisfaction level. In all cases, the instantaneous price will be computed as follows:

$$Pu_{1,i}^t = min(Pu_i^{min} \times (\ln(SD_i^t) + 1), Pu_i^{max}) \tag{3}$$

Connections of type 2 UGS connections always have the highest priority and are granted a fixed amount of bandwidth under all network states. Their associated *Satisfaction Degree* is only affected by the delay and jitter. Those two parameters may increase slightly due to channel fluctuation or to network congestion, but in general, they are kept low due to the use of E1/T1 transmission for that type of service. That's why, it should get higher *Satisfaction Degrees'* values compared to the first service type. Hence, we propose an interactive pricing model between those two types, in order to cover the revenue decrease caused by the first one. We correlate the price assigned to UGS connections at time t to the $Average\ Satisfaction\ Degree$ value of all type 1 connections. If N is the number of active connections of type 1, the $Average\ Satisfaction\ Degree$ value at time t, noted ASD^t, is:

$$ASD^t = \frac{\sum_{i=1}^{N} SD_i^t}{N} \tag{4}$$

But fist, the system has to fix the set of type 2 connections that can participate in this recompensation procedure. To do that, it has to verify that the concerned UGS connection gets better QoS level than type 1 connections, by comparing its corresponding SD^t value to the computed ASD^t value. According to this comparison, the system will decide if this connection will get the minimum price (Pu_i^{min}) or it will make part of the group of type 2 connections whose the price will slightly increase along the *Price Window* to cover the revenue decrease caused by the unsatisfied connections. Their associated price is calculated according to equation 5. Hence, the ASD^t value will act as a dynamic threshold for type 2 connections to take account of the changing network conditions.

$$Pu_{2,i}^t = max(Pu_i^{max} \times (1 - \ln(ASD^t)), Pu_i^{min}) \tag{5}$$

Connections of type 3 Since this type of traffic (BE traffic) is the last one to be served and do not affect the satisfaction level of other traffic types, we do not assign a price window but a fixed value, P_3, for the price per resource unit. In fact, this simple pricing technique, where the total price value is linearly proportional to the obtained bandwidth, is more appropriate for this traffic with no QoS requirements. The total price will be equal to P_3, multiplied by the allocated amount of bandwidth, as it is expressed in 2.

3.4 The Instantaneous Revenue

The Instantaneous Revenue is the total operator's revenue, noted Re, calculated based on the sum of the price paid by all active connections at time t :

$$\begin{aligned} Re^t &= \sum_{i=1}^{n} P_{1,i}^t + \sum_{i=1}^{m} P_{2,i}^t + \sum_{i=1}^{k} P_{3,i}^t \\ &= \sum_{i=1}^{n} Pu_{1,i}^t \times (R_i) + \sum_{i=1}^{m} Pu_{2,i}^t \times (R_i) \\ &\quad + \sum_{i=1}^{k} Pu_{3,i}^t \times (R_i) \end{aligned} \tag{6}$$

Where n, m and k represent respectively the number of type 1, type 2 and type 3 connections, and R_i represents the rate allocated to connection i.

3.5 Admission Control Scheme and Discussion

Admission control schemes are classically proposed to limit the number of connections in order to avoid potential network congestion. They are generally based on the idea of rejecting a new request when resources become limited, regardless of the type of this new incoming service. This is definitely the best strategy if there is a single new request at that time. But what if many requests arrive while the system can only accept some of them? will it choose them randomly, or it will prefer some users or some service type? The idea that we propose is to accept the requests coming from the type of service that have the highest price per resource unit at that time, in the logic of giving the resources to those who pay the most. In fact, in perfect conditions, where all connections are satisfied, the instant price per resource unit take the borders values of the *Price Windows* and the Instantaneous Revenue reaches the average value that the operator looks to maintain. Its expression is simplified to the following equation :

$$Re^+ = \sum_{i=1}^{n} Pu_{1,i}^{max} \times (R_i) + \sum_{i=1}^{m} Pu_{2,i}^{min} \times (R_i) + P_3 \times \sum_{i=1}^{k} R_i \tag{7}$$

In the other extreme case, where all type 1 connections are unsatisfied and the *Average Satisfaction Degree* value is decreased, the Instantaneous Revenue Value takes the following expression:

$$Re^{-} = \sum_{i=1}^{n} Pu_{1,i}^{min} \times (R_i) + \sum_{i=1}^{m} Pu_{2,i}^{max} \times (R_i) + P_3 \times \sum_{i=1}^{k} R_i \qquad (8)$$

Through this system's transition from one state to the other, the operator wants to maximize its profit, which is heavily impacted by the number of connections for each of the three service types. We propose to implement an acceptance priority order, based on the price of the service, to be applied whenever there are too many requests for new connections, while resources are not enough to satisfy all of theme.

In a static pricing scheme, the task is easy since prices per resource unit are fixed earlier by the operator. Here, the system will prefer the type of connection that have the highest price. However, this can lead to a big service discrimination, if the system keep going under congestion for longtime, since only one type of service will be accepted. In this case, the operator can for example fix a threshold for the number of accepted connections per type of service; once this threshold is reached, the system moves to the second type as classified in the acceptance priority order.

For the dynamic pricing scheme that we proposed, the system has to update its acceptance priority order frequently, since the price per resource unit is related to the instantaneously changing satisfaction level of the users. Suppose that the admission control is simply based on the comparison of the instantaneous price per resource unit for each type of service, just like the static pricing scheme. Let us consider the case where the operator fixes a price window for type 1 and type 2 services, where the minimum border of type 1 service is higher than the maximum border of type 2 service. Here, the accepted new connection will be of type 1. If the network goes under instability state, adding a new connection of type 1 will lead to a decrease of the ASD^t value, since the new connection will mostly get a low SD value under such network conditions. Hence, an admission control based on a simple comparison between the prices per resources unit of each type of service is profitable for the operator, but it may disturb the system state. That's why we propose an admission control that looks at the system state before deciding about the acceptance of new connections. In fact, according to equation 7 and 8, when type 1 connections become more and more satisfied and the revenue expression tends to Re^{+} value, it is more profitable to increase the number of type 1 connections since they pay with their highest instant price per resource unit. But when the number of satisfied connections is decreasing and the ASD^t value is low, The choice of type 2 connection is more safe for the system, and profitable enough for the operator, since it will pay by the maximum border of its $PriceWindow$. A threshold (Th) related to the ASD^t value needs to be fixed to mark the transition between normal and instable state. The two following steps summarizes this algorithm :

– If ASD^t is higher than Th, the system is in a stable state. The acceptance priority order is type 1 at first and then type 2 and then type 3.
– If ASD^t is less than Th, the acceptance priority is type 2 and then type 1. Type 3 connections are not accepted here until that the system regain its stable state.

4 Case study

In order to point out the system recompensation aspect especially between type 1 and type 2 services in the proposed interactive pricing scheme, we took the example of a network where there is initially 6 actives connections: 5 connections of type 1 and 1 connection of type 2. We suppose that during a 50 seconds of simulation, 3 new connections requests are received every 10 seconds whereas the system has to choose one at each time due to limited network resources. Table 1 gives the *Price Window* values affected to those services. All connections of type 1 are assumed to have the same value.

Table 1. QoS parameters

Nodes	Pu_i^{min}	Pu_i^{max}
Type 1	7	11
Type 2	10	13
Type 3	0	6

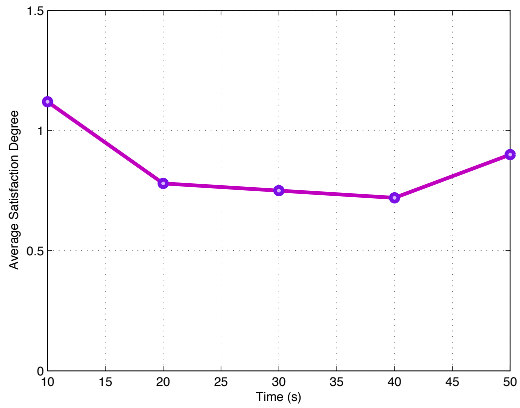


Fig. 1. Average Satisfaction Degree variation over time.

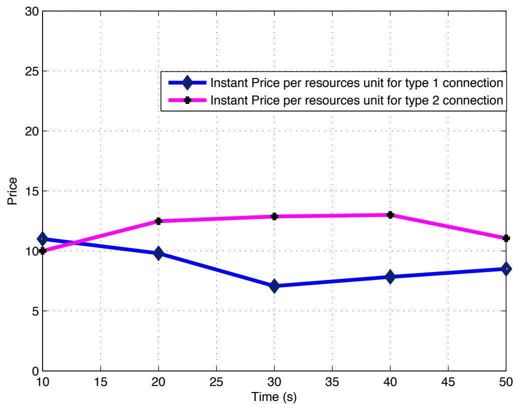


Fig. 2. Instant Price per resource unit variation over time

In figure 1, we present the instant variation of the ASD^t value through the simulation time. We fixed the threshold *Th*, used in the admission control algorithm to 0.7. According to this curve, the system starts to loose its stability state after 20 seconds of simulation, where the ASD^t value starts to decrease. The instant price per resource unit for both of type1 and type 2 connections are depicted in figure 2. We can note that at t= 20, seconds, the instant price per resource unit of type 1 connection is going down to its minimal value Pu_i^{min}, whereas the price of type 2 connection starts to increase slightly to recompense this revenue decrease. At t=10 seconds, the Base Station receives the first three requests for connection establishment. With the proposed CAC, since the ASD^t is high, the acceptance priority is given to type 1 connection whereas for the following requests, the connection that will be accepted is type 2 since it will pay with its maximum price value Pu_i^{max}. We present in figure 3 the Instantaneous Revenue variation with and without the proposed admission control. We can note that our scheme visibly enhances the instant operator's revenue by being selective in the choice of the new connections to add, according to the system state.

5 Conclusion

In this work, we presented a pricing model combined by an admission control scheme that aims at keeping high the operator's revenue especially in moments of instability in a multiservice WiMAX system. The model is based on inter services recompensation mechanism to cover the revenue lost caused by a dissatisfied connections at a given time, using *Prices Windows* instead fixed prices values. Within this model, we give different expressions for the instant price per resource unit for all the defined classes of services of Mobile WiMAX systems,

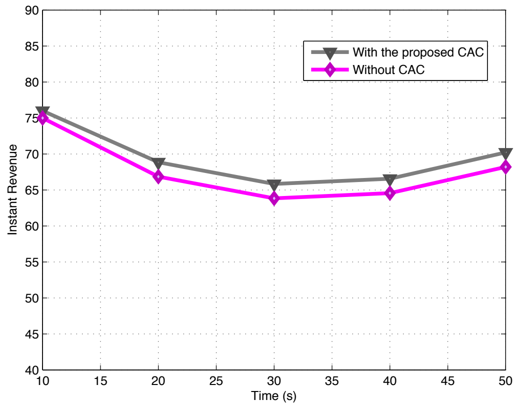


Fig. 3. Instantaneous Revenue variation over time

which we classified into three types, according to their QoS requirements. The proposed admission control scheme helps the system to choose the type of service that should be accepted according to the instant prices values, in a way to give resources to the service that pay more, if the system state allows it . This model can be adjusted to fit the requirements of future wireless technologies.

References

1. IEEE 802.16-2004, IEEE Standard for local and metropolitan area networks, Air Interface for Fixed Broadband Wireless Access Systems. Oct 2004.
2. IEEE 802.16e, IEEE Standard for local and metropolitan area networks, Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed. 2005.
3. Business Case Models for Fixed Broadband Wireless Access based on WiMAX Technology and the 802.16 Standard. White Paper. In: WiMAX Forum, 10 October 2004.
4. A. Belghith, L. Nuaymi, and P. Maille: Pricing of Real-Time Applications in WiMAX Systems. In: IEEE 68th Vehicular Technology Conference, VTC2008-Fall, Calgary, Canada, 21-24 September 2008.
5. A. Belghith, L. Nuaymi, and P. Maille: Pricing of differentiated-QoS services WiMAX networks. In: IEEE Global Communications Conference, GLOBECOM 2008, New Orlean USA, 30 Novembre-4 December 2008.
6. J. Lee, M. Chiang, and A. Robert: Utility Maximization and Pricing-Based Distributed Algorithms for Rate-Reliability Tradeoff. In: the 25th IEEE International Conference on Computer Communications INFOCOM 2006, April 2006 Page(s):1 - 13.
7. B.-J. Chang Y.-L. Chen, and C.-M. Chou: Adaptive Hierarchical Polling and Cost-based Call Admission Control in IEEE 802.16 WiMAX. In: Wireless Communica-

tions and Networking Conference, 2007, WCNC 2007, pp. 1954 - 1958, Hong Kong, 11-15 March 2007.
8. D.Niyato and E.Hossain: A Game Theoretic Analysis of Service Competition and Pricing in Heterogeneous Wireless Access Networks. In: IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 7, NO. 12, DECEMBER 2008
9. C.A.Gizelis and D.Vergados: A Survey of Pricing Schemes in Wireless Networks. In: IEEE COMMUNICATIONS SURVEYS and TUTORIALS, 2010.
10. H.Jung and B.Tuffin: Pricing for Heterogeneous Services in OFDMA 802.16 Systems. In: The Sixth International Conference on Wireless On-Demand Network Systems and Services, WONS 2009.
11. F.Ghandour, M.Frikha and S.Tabbane: Pricing, Resources Allocation and Scheduling for WiMAX Systems. In: the Fifth International Symposium on Wireless Pervasive Computing, ISWPC 2010, Modena, Italy, May 2010.

On the efficiency of a dedicated LMA for multicast traffic distribution in PMIPv6 domains

Luis M. Contreras[1], Carlos J. Bernardos[1], and Ignacio Soto[2]

[1] Departamento de Ingeniería Telemática, Univ. Carlos III de Madrid, Spain
contreras.uc3m@gmail.com, cjbc@it.uc3m.es
[2] Departamento de Ingeniería de Sistemas Telemáticos, Univ. Politécnica de Madrid, Spain
isoto@dit.upm.es

Abstract. IP multicast allows the efficient support of group communication services by reducing the number of IP flows needed for such communication. Proxy Mobile IPv6 (PMIPv6) is a network-based mobility management solution, where the functionality to support the terminal movement resides in the network. Recently, a baseline solution has been adopted for multicast support in PMIPv6. Such base solution has inefficiencies in multicast routing because it may require multiple copies of a single stream to be received by the same access gateway. Nevertheless, an alternative solution to support multicast in PMIPv6 avoids this issue. This paper evaluates by simulation the scalability of both solutions under realistic conditions, and provides an analysis of the sensitivity of the two proposals against a number of parameters.

Keywords: Multicast, PMIPv6, Mobility.

1 Introduction

IP multicast allows the efficient support of group communication services (one-to-many or many-to-many) over IP networks. Applications like TV distribution take advantage of this extension of the IP protocol which facilitates the delivery of a single copy of a data stream to multiple listeners interested in receiving the same content simultaneously. The increasing generalization in the use of multicast has also triggered the need for supporting IP multicast in mobile environments.

Mobility management support in IP-based networks is a topic that has received considerable attention in recent years. A first approach to cover this issue was based on host-based solutions, mainly Mobile IPv6 (MIPv6) [1], where IP mobile terminals or Mobile Nodes (MNs) are aware of their IP mobility and have to perform operations in order to maintain their ongoing communication sessions. As a further step, a network-based solution has been standardized, called Proxy Mobile IPv6 (PMIPv6) [2], where the functionality to handle the movement of the MNs resides in the network.

Group communication is out of the scope of the PMIPv6 standard specification. Recently, a base solution has been adopted for multicast service delivery in PMIPv6 domains [3]. This baseline solution has been built on the capabilities of the existing multicast and mobility protocols, and considers the use of the PMIPv6 core entities to serve both unicast and multicast traffic, without any further improvement for a better adaptation to the multicast case. As consequence, the base solution has inefficiencies, specially the well known tunnel convergence problem, which implies that multiple copies of a single stream may be received by the same access gateway. To solve this issue, the architecture defined in [4] proposes a separate core entity being the unique responsible of serving multicast traffic to the access nodes, then guaranteeing that only one copy of the same stream is certainly received. This paper presents a comparative analysis of both solutions, providing some insights in the potential benefits of using dedicated infrastructure for multicast service.

2 Background on PMIPv6 and multicast

2.1 Proxy Mobile IPv6

PMIPv6 is based on MIPv6, reusing much of its concepts and packet formats. In PMIPv6, mobility support is provided by some specific network entities, namely Mobile Access Gateway (MAG) and Local Mobility Anchor (LMA).

The MAG takes care of the mobility signaling on behalf of the MNs attached to its links, tracking the MNs as they move, while the LMA stores all the routing information needed to reach the MNs in the PMIPv6 domain by associating each MN with the MAG that the MN is using. A tunnel between the LMA and the MAG allows the transfer of traffic from and to the MN. Using PMIPv6, the MN can move across a PMIPv6 domain changing its access link, while keeping its IP address.

The MN is registered by the MAG in the LMA by sending a Proxy Binding Update message (an extension of the MIPv6 Binding Update). The LMA then assigns one or more home network prefixes to the MN. The LMA acknowledges the registration process with a Proxy Binding Acknowledgment message that is sent from the LMA to the MAG, containing the home network prefixes allocated to the MN. The MAG completes the configuration to serve the MN traffic by setting up the appropriate forwarding rules for the downlink/uplink traffic to/from the MN.

Different LMAs can coexist in a certain PMIPv6 domain, for instance as a mechanism to perform load balancing. Figure 1 shows several MNs maintaining associations with distinct LMAs. The LMA forwards the traffic for a certain MN to the correct MAG using the configured tunnel. The MAG decapsulates the packets and forwards them to the MN transparently. In the opposite direction, traffic coming from the MN is encapsulated by the MAG (which is the MN's default router) and decapsulated by the LMA that routes it towards the final destination.

2.2 Multicast basics in access networks

By means of IP multicast, a number of receivers located anywhere in the network can subscribe to a content in the form of a multicast session group. The content is distributed using a particular data stream forming a multicast flow. A single copy of such flow is carried on every link in the network along the multicast path dynamically created to reach the interested receivers. The data stream is replicated on the routers where the multicast path topologically diverges.

The multicast source of the data stream does not maintain any subscription list of interested receivers. The source simply sends the data stream to an arbitrary group of hosts represented by an IP multicast address. The receivers indicate their interest in receiving certain content by explicitly joining the multicast group. The Multicast Listener Discovery (MLD) [5] defines the control messages for managing the group membership process in IPv6. Multicast protocols distinguish between multicast receiver (host part) and multicast router (network part) functionalities. Basically, the host part is devoted to the group subscription management, while the router part is focused on building and maintaining the multicast tree.

A multicast router in the receiver's sub-network will capture the control messages for joining or leaving a multicast group. In some cases, the router can act as a proxy [6] for the group membership indications of the receivers connected to it, instead of the multicast router role described above. This typically occurs in aggregation networks, where the first-hop router concentrates the traffic of a huge number of receivers. The proxy performs the router part of the group membership protocol on each downstream interface, while it plays the host role on the upstream interface towards the next multicast backbone router. The proxy is in charge of summarizing the subscription demand of the receivers.

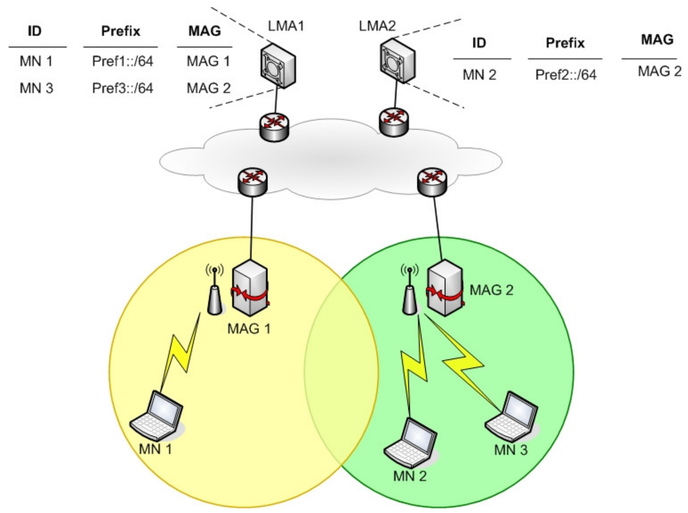


Fig. 1. Network entities in Proxy Mobile IPv6

2.3 Multicast in PMIPv6

Multicast is out of the scope of the PMIPv6 standard specification. This produces inefficiencies when distributing contents to multiple receivers (individual copies per MN). Several solutions have been proposed supporting multicast for Mobile IP networks [7]. However, they are not directly applicable to PMIPv6 because of the specificities of network-localized management environments, where the MN is not aware of network layer changes. A new approach is needed to provide multicast in PMIPv6 domains. With such aim, the MULTIMOB working group was chartered at the Internet Engineering Task Force (IETF) to specify a solution for multicast listener mobility compatible with the PMIPv6 and multicast standards.

Two kinds of solutions for multicast subscription in PMIPv6 can be differentiated: the remote subscription, where the MN gets the multicast data stream from the LMA, and the local subscription, where the MN directly obtains the multicast stream from the access router. According to the current MULTIMOB charter terms, only the remote subscription case is considered by now.

The base solution [3] provides a way to manage multicast traffic delivery to MNs unaware of their mobility. The MN expresses its interest in joining or leaving a multicast group by sending MLD control messages to the MAG, which acts as the first hop router for the MN. The MAG maintains the individual multicast status of the MN and handles the multicast traffic towards it accordingly to the MLD messages received. The MAG incorporates the functionality of MLD proxy, summarizing the group subscription requests of the MNs connected to it.

With the remote subscription model, the multicast traffic reaches the MNs after going through the corresponding LMA (note that there might be multiple LMAs in the same domain). A distinct MLD proxy instance is then defined per LMA connected to the MAG, in such a way that every MAG-LMA tunnel is part of a separate MLD proxy domain. For every proxy instance in the MAG, the tunnel interface pointing to the LMA becomes the proxy upstream interface, whereas the links towards the MNs are the corresponding downstream interfaces of each instance. The LMA is the entity in charge of interacting with the multicast infrastructure out of the PMIPv6 domain.

The summarization of control messages in upstream that the MAG performs is applied per set of MNs associated with a certain LMA, as the different proxy instances of the same MAG are isolated one from the other. The LMA maintains the multicast state of every tunnel interface. Such status reflects the summarized view offered by the MAG on behalf of the attached MNs bound to the LMA. A multicast stream will be delivered over the tunnel or removed from it according to the aggregated behavior of the MNs attached to the MAG.

The LMA, the MAG and the tunnel linking them are all part of the multicast tree. This branch will be common to every multicast tree providing any content subscribed by an MN in a MAG and associated to a particular LMA. It makes possible to send a single copy of a data stream per group of MNs demanding the same content.

3 Dedicated LMA for multicast traffic

In the base solution, the MAG can receive multiple copies of the same stream if several attached MNs associated to different LMAs demand simultaneously the same content. The Figure 2(a) shows this issue. Each LMA will forward the same stream to a certain MAG by using their corresponding LMA-MAG tunnel. Under these circumstances, multiple copies arrive to the MAG, creating tunnel convergence problem.

As an enhancement to this situation, a dedicated multicast LMA (M-LMA) [4] can be deployed to act as the unique mobility anchor for the multicast traffic to all the attached MNs in the PMIPv6 domain, whereas the unicast traffic is handled by the rest of LMAs, as in the PMIPv6 specification.

The M-LMA maintains a tunnel with every MAG in the domain for all multicast channels. This tunnel will be the path followed by the multicast traffic towards the MAG for all the MNs attached to it, thus avoiding the tunnel convergence problem, because only one stream copy arrives to the MAG. The Figure 2(b) represents graphically this solution.

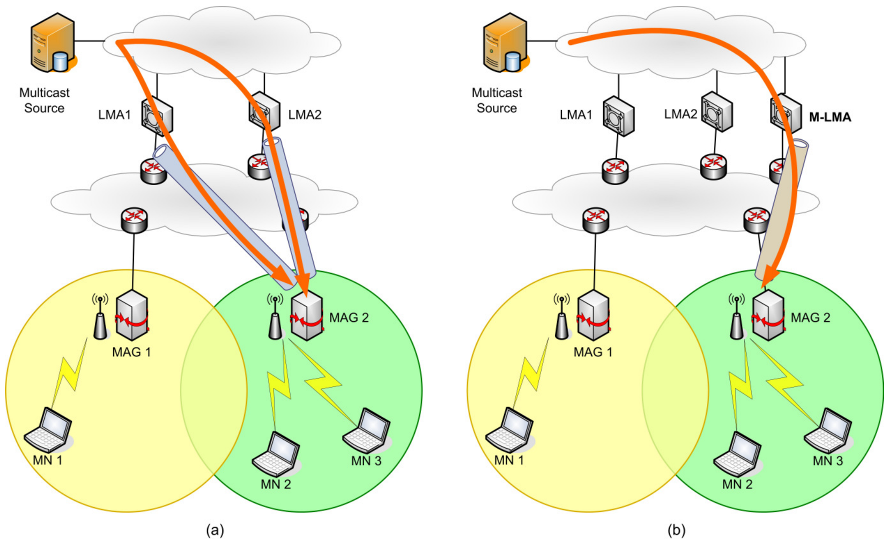


Fig. 2. Tunnel convergence problem in base solution (a) vs. M-LMA proposal (b)

A number of advantages can be envisioned, such as the simplification of the multicast distribution tree topology or the significant reduction of the network bandwidth needed for providing multicast service within a PMIPv6 domain

The base solution specification includes a comparison with the M-LMA proposal which roughly evaluates the performance of each of them in a couple of extreme scenarios. The population of MNs in the comparison is set to 1 million which are uniformly distributed among 200 MAGs. These MAGs are then connected to 4 LMAs, for the base solution, or 1 LMA, for the M-LMA approach.

The comparison is based on three metrics: the number of simultaneous streams delivered per LMA, the total number of streams in the network for serving the connected MNs, and the number of repeated streams at MAGs to account for the tunnel convergence issue. The comparison is presented in Table 1.

This analysis is a theoretical exercise to study system scalability, but we argue that it does not represent a realistic scenario, so the conclusions on scalability obtained from it are not well supported. First, the model of the multicast channel demand (all the MNs subscribed to the same content in one case, all the MNs subscribed to a different content in the other) does not follow a realistic pattern, neither in terms of subscription behavior nor in terms of multicast flow volume. Second, the number of active multicast MNs and the MAGs supporting them seems to be unrealistically high.

In this paper we provide a more realistic comparison of both systems to better characterize the efficiency and scalability of them.

Table 1. System comparison with extremal settings in [3]

Case scenario	Multicast scheme	Redundant streams per MAG	Simultaneous stream per LMA	Total streams in the domain
Each MN subscribed to a different content	Base solution	0	250.000	1.000.000
	M-LMA	0	1.000.000	1.000.000
All MNs subscribed to the same content	Base solution	3	200	800
	M-LMA	0	200	200

4 Simulation framework

4.1 Channel popularity model

The channel preference in IPTV-like systems is commonly modeled [8, 9, 10] by a power-law distribution known as Zipf function. The Zipf function states that the occurrence of a certain event (here, the tuning of a multicast channel) is determined by:

$$k \times \left(1 / r^{\alpha} \right), \tag{1}$$

being k a constant, r the rank or popularity of the event in the distribution, and α the factor which characterizes the skewness of the distribution. Then, the frequency or probability that predicts the eligibility of an event is provided by:

$$\frac{\left(1 / r^{\alpha} \right)}{\sum_{i=1}^{R} \left(1 / i^{\alpha} \right)}, \tag{2}$$

where R is the total number of ranked elements. As α increases, the popularity of the first ranked events increases, while the distribution tail concentrates less occurrences. We will consider also here a Zipf-type function to model the channel demand by the MNs in the domain.

4.2 Description of the simulation

We have used the numerical computation tool Octave [11] to simulate the multicast channel demand in a PMIPv6 domain, and to calculate the number of streams required per LMA-MAG tunnel defined in the system. The simulation focuses on the streams delivered in a certain time instant, thus not including the impact of user mobility during a longer observation period. The methodology followed here to simulate the system demand is basically the same approach followed in [10].

We firstly obtain a uniform random variable in the range (0,1) with as many samples as the number of MNs considered in the simulation (all are supposed to be multicast active and tuned to a channel). We then map the random variable to a Zipf function defined by both the skewness parameter α and the number of channels C, in order to simulate the MNs channel subscription. At this step we already have a picture of the channel demand in the PMIPv6, where each MN is associated with a tuned channel. Now, it is time to compare the behavior of the architectures under consideration.

To do that, we sequentially split twice the MNs in different groups, first accordingly to the number of LMAs, and then accordingly to the number of MAGs, in such a way that we form a matrix with the following elements: number of MNs, channel subscribed by the MNs, MAG where the MN is attached to, and LMA where the MN maintains a service association. In these conditions we are able to calculate the number of multicast streams needed per LMA-MAG tunnel in the domain, and to obtain the metrics previously used for scalability comparison.

The process is entirely repeated several times with random and independent sets of MNs subscriber choices (more than 100 times) guaranteeing convergent results in terms of mean and standard deviation values. The results obtained are the average values of the metrics under study.

4.3 Simulation parameterization

In order to develop a sensitivity analysis of both network solutions for the metrics defined above, we will consider different values for the parameters in the simulation.

In each iteration within the simulation routine, the same Zipf-like distribution representing the MN channel preference is used to coherently compare the impact of the distinct parameters against the considered network solution. The only case where different sets of subscriptions are compared is the case for the sensitivity on the number of MNs in the system, as it necessarily requires a distinct simulation universe. In our simulation we study the demand created for 6.000 and 12.000 MNs, respectively, all requiring multicast service.

The scenarios used for the analysis consists of a unique LMA, in the case of the M-LMA approach; and 2 LMAs or 4 LMAs, for the case of the base solution. In the later case, the variation in the number of LMAs provides an insight on the LMA scalability for multicast stream replication. Some other impacts can be evaluated, as follows:

- The impact of the user preferences on the system can be determined by the variation on the skewness factor of the Zipf distribution. The values considered in the simulation for α are 0.6 and 0.9 [8, 9, 10].
- The impact of the service provider content offer can be modeled by the variation on the number of eligible channels, that is, in the variation of C. In this simulation it takes values of 150 and 300 channels.
- The impact of the network access capillarity can be modeled by the number of MAGs in the domain. We study the impact of having 6 or 12 MAGs in the system.

We will confront every simulation scenario respect to the architecture defined by the number of LMAs taken into account. The different scenarios will be identified in the rest of the paper according to the rule defined by *{number of MNs, skewness factor, number of channels, number of MAGs}* (e.g., the scenario identified as *{6000, 0.6, 150, 6}* means that we are considering 6000 MNs which create a channel demand defined by a Zipf distribution with skewness factor of 0.6 over 150 channels, and that are evenly distributed in 6 MAGs).

5 System analysis

The following sub-sections show the sensitivity of the two solutions regarding to the parameters of the simulation. We first focus on the metrics of the number of streams per LMA, and the number of total streams in the PMIPv6 domain. The number of repeated streams per MAG is analyzed separately.

5.1 Impact of the content offer

Figure 3(a) depicts the impact of the content offer in the number of average streams per LMA. According to the figure, the number of streams per LMA grows with the number of channels accessible for the MNs. More channels in the system mean more distinct multicast streams needed to serve the MNs demand (all the channels are susceptible of having MNs tuned on them in our simulation).

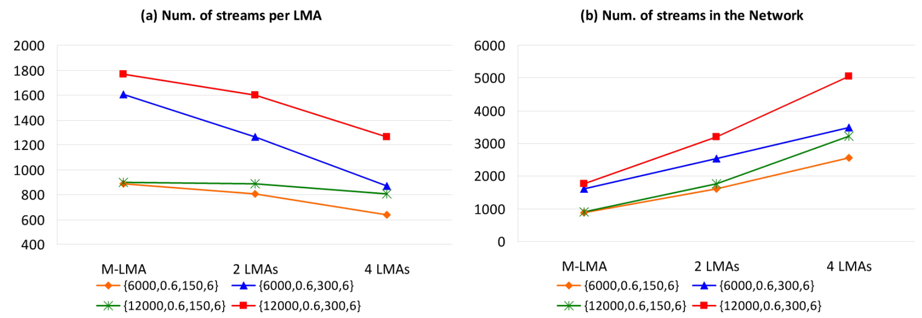


Fig. 3. Impact of multicast channel offer – (a) Number of streams per LMA; (b) Number of streams in the network

The requirements in terms of stream reception for the LMAs in the base solution are certainly more scalable than the M-LMA approach, improving the scalability as the number of served MNs grows. In the case of the base solution, the MNs are split among the LMAs, so if no MN associated with a particular LMA is subscribed to a particular channel, the LMA would not require receiving that channel. In the M-LMA approach, the unique LMA serves all the MNs, which means that it has to receive all the channels subscribed by any of the MNs in the PMIPv6 domain. However, the requirements for replicating multicast streams in the LMA in the base solution are not far from those needed by the M-LMA. This is because each LMA in the base solution has to send a copy of a channel to a MAG as soon as at least one MN associated with that LMA and attached to the MAG requires that channel. Popular channels will be replicated by every LMA and sent to each MAG that receives multiple copies of the channel, while in the M-LMA solution just one stream will be sent from the M-LMA to each MAG.

In fact, Figure 3(b) shows that the M-LMA scales better from the network perspective because the M-LMA architecture requires much less streams than the base solution to serve the same set of MNs. The increment in the number of streams as more channels are offered by the service provider is translated in a progressive increment of the average number of streams required by the base solution. The situation becomes worst as the number of LMAs in the domain increases.

5.2 Impact of the user preferences

Figure 4(a) presents the impact of α in the number of streams transmitted per LMA. As a general rule, greater values of α reduce the average number of streams per LMA. This is basically motivated by the shift in the Zipf distribution which accumulates the audience in the first ranks of the distribution. Some channels in the tail will not be necessarily tuned by any MN associated with an LMA and attached with some MAG, reducing the number of streams per LMA. It can also be observed that the streams per LMA increases with the number of MNs, because some of the previous channels not demanded before in some tunnels become now requested.

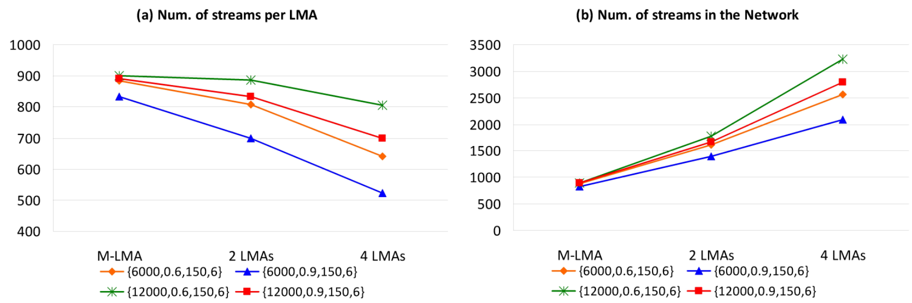


Fig. 4. Impact of the skewness factor α – (a) Number of streams per LMA; (b) Number of streams in the network

A relevant result is that the LMA scalability requirements are quite similar in both the base solution and the M-LMA approach. This is more notorious when the number of MNs increases. In the base solution, the MNs attached to a MAG are associated to different LMAs. Each one of these LMAs have to serve their respective MNs demand which will be composed of a wide set of common channels, thus driving to the tunnel convergence problem. This simulation confirms that when the number of MNs grows with respect to the number of channels, which is a reasonable configuration in multicast environments, the set of common channels that have to be managed by every LMA also grows, minimizing the scalability advantage of having several LMAs, while suffering the problem of the multiplication of stream copies in the domain.

Figure 4(b) shows the total average number of streams in the network for different values of α. It becomes also clear that the number of total streams decreases when α

increases due to the shift of the Zipf distribution which describes the MNs channel demand. The same arguments as before are applicable here. The total number of streams in the network grows significantly with the number of LMAs serving the multicast traffic. The volume of streams required for the service by the 4 LMAs case is close to 3 or 4 times (depending on α) larger than the resources needed by the M-LMA approach, where less tunnels are involved in the multicast service delivery.

5.3 Impact of the network access capillarity

The impact of the number of MAGs in the PMIPv6 domain can be observed in Figure 5. The number of streams per LMA increases if the MNs are distributed among more MAGs with the same channel demand. So, as the number of tunnels increases, the average number of streams per LMA also increases. In the M-LMA case, this implies more streams per (M-)LMA because only one device supports all the growth. For the base solution it can be highlighted that each LMA supports less streams, as the load is shared among the LMAs, but the difference in terms of number of streams supported per device (LMAs vs. M-LMA) are smaller than expected, because there is not a clean split of the load among the LMAs, much of it is replicated in every LMA.

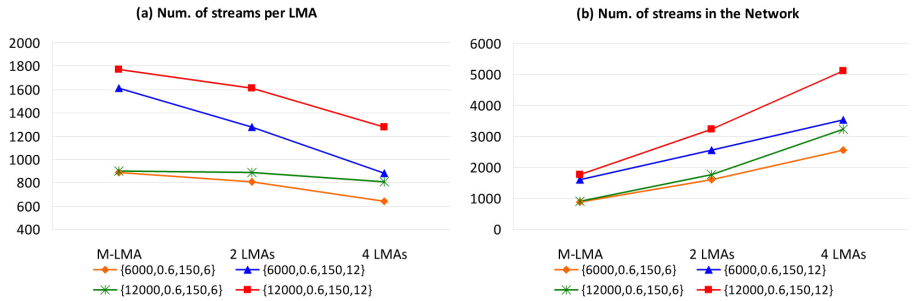


Fig. 5. Impact of the access capillarity – (a) Number of streams per LMA; (b) Number of streams in the network

On the other hand, the increment in the number of MAGs in the domain also impacts on the number of the total streams in the network. The number of streams in the network increases with the number of LMAs and MAGs.

5.4 Repeated streams per MAG

The M-LMA approach does not suffer from the tunnel convergence issue. There is only one LMA to forward multicast traffic towards the MAGs, thus avoiding stream repetition at the MAGs.

The situation is the opposite in the case of the base solution. Several tunnels, one per LMA, can feed the MAG with the same multicast channel. This happens when a MAG has attached MNs that are associated to different LMAs and subscribed to the same content.

Table 2. Average number of repeated streams per MAG

Sensitivity to	Scenario	2 LMAs	4 LMAs
-	{6000, 0.6, 150, 6}	121,24	133,20
Zipf factor α	{6000, 0.9, 150, 6}	94,08	107,60
Number of Channels	{6000, 0.6, 300, 6}	154,39	184,64
Number of MAGs	{6000, 0.6, 150, 12}	78,79	94,07
-	{12000, 0.6, 150, 6}	145,21	148,48
Zipf factor α	{12000, 0.9, 150, 6}	129,50	138,53
Number of Channels	{12000, 0.6, 300, 6}	239,34	263,95
Number of MAGs	{12000, 0.6, 150, 12}	121,19	133,07

Table 2 quantifies the tunnel convergence issue at the MAG, that is, the average number of simultaneously repeated streams in different scenarios. The trend in the number of repeated streams at a MAG is the same as the one followed by the total number of streams in the network. The number of repeated streams decreases with the increment of α, and with the increment in the number of MAGs. On the contrary, it increases when more channels are accessible in the system.

6 Conclusions

In this paper we have compared two approaches to provide multicast content to MNs within a PMIPv6 domain. Neither solution requires modifications to PMIPv6 or to multicast protocols. One is the base solution adopted by the IETF. In this base solution, an LMA serving the unicast traffic also acts as the first multicast router for the mobile nodes in the domain, and the MAGs act as MLD proxies. A PMIPv6 domain can use more than one LMA to serve the visiting MNs, splitting the load among the LMAs both for unicast and multicast traffic. The alternative solution is using a dedicated LMA for multicast traffic, different from the LMAs for unicast traffic, while the MAGs keep their role as MLD proxies.

An initial comparison of both solutions in [3] suggests that the base solution has better scalability properties. In this paper we show that in realistic scenarios this is not the case. The base solution tries to reduce the load in the LMAs, splitting the load by having several of them. However, for multicast traffic, the number of channels that an LMA has to serve does not decrease linearly with the reduction of the number of MNs associated to that LMA. The key factor is the set of channels subscribed by any MN associated to the LMA and attached to certain MAG. So reducing the number of MNs that an LMA has to attend does not result in the expected load reduction. In fact, as the number of MNs increases in the PMIPv6 domain, we have less advantage for having several LMAs, as each of them will probably manage all the multicast channels (or at least the popular ones) anyway. In fact we are just duplicating the work at the different LMAs. The base solution also increases the work load at the MAGs, because of the tunnel convergence issue, in which a MAG will receive several copies of the same flow for MNs subscribed to the same content but bound to different LMAs.

In contrast to this, the dedicated LMA solution for multicast traffic has several advantages: separation of the management of the multicast traffic and the unicast traffic in different LMAs, reduced load in the MAGs, and significant reduction in multicast traffic load within the PMIPv6 domain. This last advantage is due to the duplication that happens in the base solution with several LMAs forwarding the same multicast traffic to every MAG, a situation that does not happen in the solution with a dedicated multicast LMA.

Further work will focus on a more complete characterization of the performance of M-LMA proposal by including user mobility impact on the simulations.

Acknowledgements. The work of Luis M. Contreras and Ignacio Soto has been partially supported by the Ministry of Science and Innovation of Spain under the I-MOVING project (TEC2010-18907) and the EU Celtic UP-TO-US project (TSI-020400-2010-114).

7 References

1. D. Johnson, C. Perkins, and J. Arkko: Mobility Support in IPv6. IETF RFC 3775. (2004)
2. S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil: Proxy Mobile IPv6. IETF RFC 5213. (2008)
3. T.C. Schmidt, M. Waehlisch, and S. Krishnan: Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains. IETF RFC 6224. (2011)
4. J.C. Zuniga, A. Rahman, L.M. Contreras, C.J. Bernardos, and I. Soto: Support Multicast Services Using Proxy Mobile IPv6. IETF Internet draft, draft-zuniga-multimob-smspmip-05 (work-in-progress). (2011)
5. R. Vida and L. Costa: Multicast Listener Discovery Version 2 (MLDv2) for IPv6. IETF RFC 3810. (2004)
6. B. Fenner, H. He, B. Haberman, and H. Sandick: Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD) – Based Multicast Forwarding ("IGMP/MLD Proxying"). IETF RFC 4605 (2006)
7. I. Romdhani, M. Kellil, H.-Y. Lach, A. Bouabdallah, and H. Bettahar: IP Mobile Multicast: Challenges and Solutions. IEEE Communications Surveys & Tutorials, 6(1). (2004)
8. Z. Avramova, D. De Vleeschauwer, S. Wittevrongel, and H. Bruneel: Dimensioning Multicast-Enabled Networks for IP-Transported TV Channels. ITC 2007, LNCS, vol. 4516, pp. 6–17. (2007)
9. N. Sinha, R. Oz, and S. Vasudevan: The Statistics of Switched Broadcast. Proceedings of Conference on Emerging Technologies, Tampa, USA. (2005)
10. D. T. van Veen, M. K. Weldon, C. C. Bahr, and E. E. Harstead: An analysis of the technical and economic essentials for providing video over fiber-to-the-premises networks. Bell Labs Technical Journal, 10 (1), pp. 181-200. (2005)
11. GNU Octave, http://www.gnu.org/software/octave (access date: January 2011)

A Selective Neighbor Caching Approach for Supporting Mobility in Publish/Subscribe Networks*

Vasilios A. Siris, Xenofon Vasilakos, and George C. Polyzos

Mobile Multimedia Laboratory
Department of Informatics
Athens University of Economics and Business, Greece
{vsiris, xvas, polyzos}@aueb.gr

Abstract. We present a selective neighbor caching approach for supporting mobility in publish/subscribe networks. According to the approach, a mobile's subscriptions are transmitted to a subset of brokers that are neighbors of the current broker that the mobile is connected to. Our key contribution is the definition of a target cost function and an intelligent procedure for selecting the subset of neighbors. The advantage of our proposal is that it reduces the buffering costs, since not all neighbor brokers cache items, while still obtaining the gains of proactive caching, by caching items in a subset of neighbor brokers that a mobile has a high probability to connect to.

1 Introduction and Motivation

In a publish/subscribe network, subscribers express their interests for information items through subscriptions. The event notification service is responsible for locating publishers of items matching the subscriptions, and for initiating the forwarding process from the publishers to the subscribers. When subscribers are mobile, they can attach to the network at different points (locations), and they can experience disconnections. Mobility in publish/subscribe networks is typically supported through *brokers*, which handle a mobile's subscription and cache items that match the subscriptions, when a mobile changes location or is disconnected.

Based on when and how subscriptions and matching items are transferred from an old to a new broker when a subscriber moves, approaches for supporting mobility in publish/subscribe networks can be categorized as follows [7]: reactive approaches [3, 6, 11, 10], durable subscriptions [1, 5], and proactive (or prefetching) approaches [2, 7]. With reactive approaches, when a mobile disconnects from a broker, the broker continues to cache items that match the disconnected mobile's subscriptions. When the mobile reconnects to a new broker, it informs the new broker of the old broker's id. The new broker then communicates with the

* This work was supported in part by the EU FP7 project PURSUIT, under contract ICT-2010-257217.

old broker, requesting all items that were cached in the old broker during the disconnection period. A disadvantage of this approach is that cached items need to be transferred from the old broker to the new broker; this incurs a capacity cost for transferring the cached items, and a delay for the new broker to start forwarding items to the mobile subscriber, since the new broker must receive all the items cached in the old broker before it can start forwarding.

With durable subscriptions, brokers maintain a mobile's subscriptions and cache items matching these subscriptions, independent of whether the mobile is connected to the broker or not. Without any additional mechanisms, in order to avoid loosing items, all brokers within a domain that a mobile can possibly connect to would need to maintain subscriptions and cache items matching them; this would incur significant buffering costs.

Proactive approaches for supporting mobility work as follows: A mobile's subscriptions are transmitted one broker ahead of its current broker *before* the mobile disconnects or moves; Selection of which brokers to transmit subscriptions to can be based on prediction [2], or knowledge of each broker's neighboring brokers that are one hop ahead of it [7]; the work in [2] does not propose nor evaluates a specific algorithm for prediction. According to the scheme in [7], when a mobile disconnects all neighboring brokers start caching items that match the mobile's subscriptions. Hence, when the mobile connects to one of these brokers, it can quickly receive items that were transmitted during the disconnection period. Proactive approaches trade-off buffer space for reduced delay in forwarding items to subscribers. Hence, proactive approaches can be more appropriate for applications where delay is important, such as real-time (e.g., voice, teleconferencing, critical event notification, and gaming) and streaming applications.

The proposal in this paper falls in the proactive approach category, and its main difference compared to [7] is that only a subset of neighbor brokers maintain a mobile's subscriptions and cache items matching them. Our key contribution is the definition of a target cost function and an intelligent procedure for selecting the subset of neighbors. The advantage of our proposal is that it reduces the buffering costs, since not all neighbor brokers cache items, while still obtaining the gains of proactive caching, by caching items in a subset of neighbor brokers that a mobile has a high probability to connect to.

A selective neighbor caching approach has been proposed for reducing the handover delay in WLANs [8]. The motivation is that, even when the number of neighboring APs is large, at most 3 or 4 access points are targets of the handoffs [8]; it is likely that a similar motivation applies for publish/subscribe networks, since the mobility of users is the cause for subscriber movement in both cases. However, the application of such an idea for supporting mobility in publish/subscribe network is different due to the different nature of the problem, since in publish/subscribe networks the objective is to forward to subscribers items that match their subscriptions. As a result, the model and corresponding procedure for selecting the subset of neighbors is fundamentally different. We also note that mobility prediction has been used to improve QoS support during handovers in cellular networks, e.g., see [4, 9].

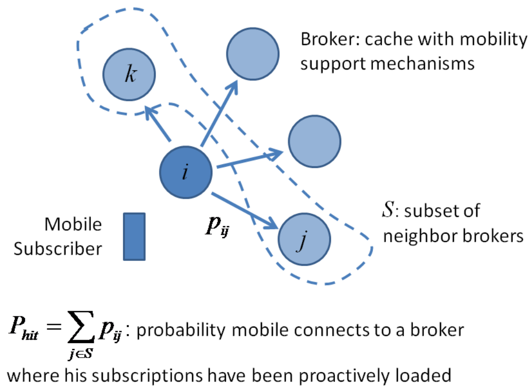


Fig. 1. A mobile's current broker i pro-actively transmits the mobile's subscriptions to a subset S of i's neighbor brokers. The subset of neighbors is selected to minimize the cost function (1).

2 Selective Neighbor Caching for Supporting Mobility

Consider a mobile that is initially connected to broker i. Assume that broker i knows the probabilities p_{ij} that the mobile connects to broker $j \in J$ after it disconnects from i, where J is the set of i's neighbor brokers; these probabilities can be obtained from information that broker i receives from its neighbors, as we will see later.

Broker i needs to select the subset of neighbor brokers to which it will proactively send the mobile's subscriptions. The probability that the mobile connects to a broker in the subset $S \subseteq J$ is $P_{hit} = \sum_{j \in S} p_{ij}$. Broker i selects the subset S^* that minimizes the cost function

$$P_{hit} \cdot C_{hit} + (1 - P_{hit}) \cdot C_{miss} + N(P_{hit}) \cdot C_{cache} \,, \tag{1}$$

where $N(P_{hit})$ is the minimum number of neighbor brokers to achieve P_{hit}; we discuss how this function is computed below. C_{hit} is the cost (e.g., delay) for a mobile to receive items from a broker that had proactively cached items matching the mobile's subscriptions; the probability for a mobile to connect to such a broker is P_{hit}. $C_{miss} > C_{hit}$ is the cost for a mobile to receive items from their original publishers, which occurs with probability $1 - P_{hit}$; we assume that the cost for receiving items directly from the publishers is the same for all publishers. The factors C_{hit}, C_{miss} can be estimated, e.g., from measurements of delay to obtain items from a local broker or a distant publisher, respectively. Finally, C_{cache} is the cost for a broker to cache a mobile's subscriptions and the matching items. If we assume that the memory requirements for storing subscriptions is small, then C_{cache} depends linearly on the memory requirements for storing a single item.

$N(P_{hit})$ is constructed as follows: without loss of generality, assume that all neighbors $j \in J$ are ordered with decreasing p_{ij}, i.e., $p_{ij} \geq p_{ij'}$ for $j < j'$. From the set of probabilities we can compute the stepwise function $N(\sum_{k=1}^{n} p_{ik}) = n$, where $n = 1, \dots |J|$. Note that each value of $P_{hit} = \sum_{k=1}^{n} p_{ik}$ uniquely determines a subset of neighbor brokers $\{1, \dots n\}$. With a linear search over all discrete values of P_{hit} (or a binary search if the number of values is large), we can find $P^*_{hit} = \sum_{k=1}^{n^*} p_{ik}$ that minimizes the cost (1), hence the subset of neighbor brokers that will proactively receive the mobile's subscriptions is $S^* = \{1, \dots n^*\}$.

When a mobile disconnects from its current broker i, then i notifies the neighbor brokers in the set S^* to start caching items that match the disconnected mobile's subscriptions. Additionally, until it receives an acknowledgement from all brokers in S^*, it forwards to these brokers the items it receives that match the disconnected mobile's subscriptions[1]. This ensures that when the mobile connects to a new broker included in S^*, the new broker has cached all items transmitted during the mobile's disconnection period; hence, these items can be immediately forwarded when the mobile connects to the new broker. When the mobile connects to a new broker, it informs the new broker of the old broker's id. In turn, the new broker informs the old broker that the mobile has reconnected, and subsequently the old broker informs all brokers in S^* to stop caching items. The above connection and disconnection actions are the same as the ones for the proactive approach in [7]. Moreover, through the above communication, broker i learns its neighbor brokers and obtains information in order to compute the probabilities p_{ij}; in the beginning, broker i will not know its neighbors and can initially assign equal probabilities when it first learns of their existence. The estimation of the above probabilities can be enhanced with information such as location, orientation, and road/path topology, which have been used for mobility prediction in cellular networks [4, 9].

In the above description, we have implicitly assumed that a broker selects a set of neighbor brokers that may be different for different mobiles. Alternatively, the subset of neighbor brokers can be the same for all mobiles, or can be the same for mobiles belonging to a specific category, which exhibit similar mobility patterns. Also, we have assumed that all brokers are on the same level; this, for example, means that the cost for transmitting items from any broker to a subscriber is the same. Alternatively, we can have multiple levels of brokers. In this case, the cost (e.g., delay) for a broker at a higher level to transmit items to a subscriber is higher. On the other hand, a broker at a higher level can potentially serve a larger number of mobile attachment points, hence using brokers at a higher level can reduce the subset of brokers that need to maintain subscriptions and cache matching items. The cost model presented in this paper can be extended to the case of multiple levels of brokers.

3 Conclusion

This paper describes our initial work on supporting mobility in publish/subscribe networks using an approach whereby a mobile's subscriptions are proactively

[1] We assume that the time to receive an ack is small and/or the number of matching items until an ack is received is small, hence do not consider the cost for transmitting these items to the neighbor brokers in the cost function.

cached in a subset of neighboring brokers. In particular, we present a model and the corresponding procedure for selecting the subset of neighbors to cache a mobile's subscriptions.

We are currently implementing the proposed model and procedure in a simulation environment, in order to investigate its performance, in terms of delay, and capacity/buffer overhead for various scenarios that correspond to different costs C_{miss}, C_{hit}, C_{cache} and different mobility patterns, and to quantify the gains in terms of reduced buffer requirements compared to the approach in [7].

References

[1] Java Message Service (JMS) API Specification. Available at `http://download.oracle.com/javaee/1.3/jms/tutorial/`.

[2] I. Burcea, H. Jacobsen, E. Lara, V. Muthusamy, and M. Petrovic. Disconnected Operation in Publish/Subscribe Middleware. In *Proc. of IEEE International Conference on Mobile Data Management (MDM)*, 2004.

[3] M. Caporuscio, A. Carzaniga, and A. L. Wolf. Design and Evaluation of a Support Service for Mobile Publish/Subscribe Applications. *IEEE Transactions on Software Engineering*, 29(12):1059–1071, December 2003.

[4] M.-H. Chiu and M. A. Bassiouni. Predictive Schemes for Handoff Prioritization in Cellular Networks Based on Mobile Positioning. *IEEE Journal on Selected Areas in Communications*, 18(3):510–522, March 2000.

[5] U. Farooq, E. Parsons, and S. Majumdar. Performance of Publish/Subscribe Middleware in Mobile Wireless Networks. In *Proc. of International Workshop on Software and Performance (WOSP)*, 2004.

[6] L. Fiege, F. C. Gartner, O. Kasten, and A. Zeidler. Supporting mobility in content-based publish/subscribe middleware. In *Proc. of Middleware 2003, LNCS, volume 2672*, 2003.

[7] A. Gaddah and T. Kunz. Extending mobility to publish/subscribe systems using a proactive caching approach. *Mobile Information Systems*, 6:293–324, 2010.

[8] S. Pack, H. Jung, T. Kwon, and Y. Choi. SNC: A Selective Neighbor Caching Scheme for Fast Handoff in IEEE 802.11 Wireless Networks. *ACM Mobile Computing and Communications Review*, 9(4):39–49, October 2005.

[9] W.-S. Soh and H. S. Kim. QoS provisioning in cellular networks based on mobility prediction techniques. *IEEE Communications Magazine*, 41(1):86–92, January 2003.

[10] V. Sourlas, G. S. Paschos, P. Flegkas, and L. Tassiulas. Mobility support through caching in content-based publish/subscribe networks. In *Proc. of IEEE/ACM Int'l Conference on Cluster, Cloud and Grid Computing*, 2010.

[11] J. Wang, J. Cao, J. Li, and J. Wu. MHH: A Novel Protocol for Mobility Management in Publish/Subscribe Systems. In *Proc. of IEEE Int'l Conference on Parallel Processing (ICPP)*, 2007.

Author Index