Agent-enabled Collaborative Downloading: Towards Energy-efficient Provisioning of Group-oriented Services

Iva Bojic, Vedran Podobnik, and Mario Kusek

University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
{iva.bojic, vedran.podobnik, mario.kusek}@fer.hr

Abstract. In this article we propose a novel approach, enabled by software agents, in mobile service provisioning process: energy-efficient collaborative downloading. The main idea is that mobile users, represented with their agents and corresponding profiles, interested in the same content download some parts directly from a service server and others afterwards locally exchange among themselves to reduce overall energy consumption. Collaborative downloading is formed as an auction where mobile users' agents compete to determine which parts of a requested content to download directly from the service server and which to exchange with other users. Our main motivation for conceiving collaborative downloading was to lower the overall energy consumption of users' mobile devices. Simulation results demonstrate that the proposed innovation can save up to 75% of the energy of mobile devices' batteries.

Keywords: user collaboration, software agents, self-organization, ad hoc network, mobile service provisioning, energy-efficiency.

1 Introduction

The emergence of the Next Generation Network (NGN) has enabled provisioning of advanced services to mobile users [1] but also provoked a growth of mobile services' complexity [2]. With a growth of mobile services' complexity, there is a need for larger battery capacities and/or lower energy consumption [3]. This article proposes a novel approach, enabled by software agents, in mobile service provisioning: energy-efficient collaborative downloading. Concept of collaboration among users while downloading is not a novel one. In [4] is proposed SPAWN, a cooperative strategy for content delivery and sharing in future vehicular networks. However, the idea behind the SPAWN is *communication efficient,* while our innovation in mobile service provisioning is motivated by achieving *energy efficiency*. VC-MAC mechanisms introduced in [5] are for a novel protocol that utilizes the concept of cooperative communication tailored for vehicular networks. While VC-MAC leverages the broadcast nature of the wireless medium to maximize the system throughput, our approach utilizes the same concept to achieve better energy-efficiency by building Bluetooth *ad hoc* network. Bluetooth network extends and complements the General Packet Radio Service (GPRS) mobile network, but consumes much less energy than GPRS-based communication (shown in Tab. 1 [6]).

Table 1. Energy consumption in GPRS and Bluetooth networks [6]

	Send (*μjoules/bit*)	Recieve (*μjoules/bit*)
Bluetooth	0.064	0.064
GPRS	15.647	1.422

2 System Architecture

The proposed system architecture is illustrated in Fig. 1. Let $\mathcal{I} = \{i_1, \cdots, i_\mathcal{N}\}$ denote the set of mobile users that are subscribers of a certain telco and let $\mathcal{J} = \{j_1, \cdots, j_\mathcal{M}\}$ denote a set of mobile users in one mobile *ad hoc* network. Set $\mathcal{J}$ is always a subset of set $\mathcal{I}$, while $\mathcal{M} = |\mathcal{J}|$. On the service server (e.g. telco's server or 3rd party provider's server) content $\mathcal{K}$ is divided into smaller parts $\mathcal{K} = \{k_1, \cdots, k_\mathcal{P}\}$.

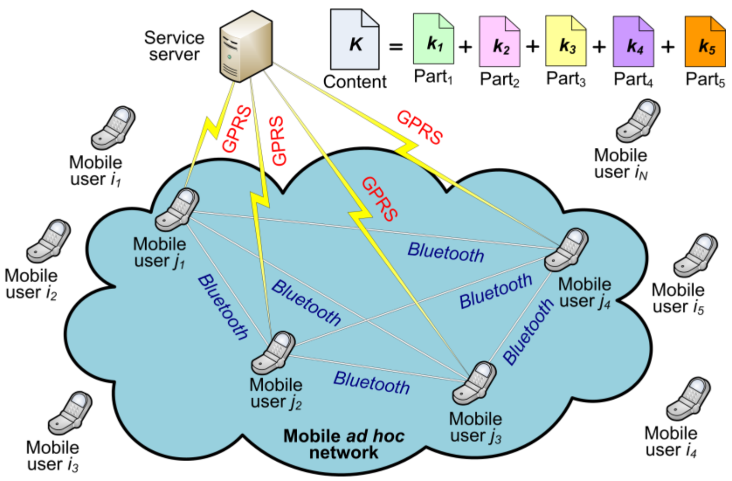

Figure 1 System architecture

2.1 User and Service Profiles

Each mobile user is presented by *user profile* while each service is represented with *service profile*. In our system, these profiles are based on the *Composite Capabilities/Preferences Profile* (CC/PP) and *User Agent Profile (UAProf)* standards. Semantic matchmaking is the process of comparing two objects represented through semantic profiles, resulting with a number within a certain interval. In our case, we compare doubles user-service and rate their similarity with a number between 0 and 1 (our matchmaking algorithm is presented in [7] and [8]). Such comparison enables the telco (i.e. mobile service provider) to enhance the provisioning of group-oriented services.

2.2 Agent-enabled User Clustering

A software agent [9][10][11] is a program which autonomously acts on behalf of its principal, while carrying out complex information and communication tasks that have been delegated to it. From the owner's point of view, agents improve her/his efficiency by reducing the time required to execute personal and/or business tasks. A system composed of several software agents, collectively capable of reaching goals that are difficult to achieve by an individual agent or a monolithic system, is called a Multi-Agent System (MAS) [12].

Agent communication has to be regulated by a certain set of rules in order to effectively conduct a conversation. This necessity compelled *Foundation for Intelligent Physical Agents*[1] (FIPA) to develop *Agent Communication Language*[2] (ACL). The ACL defines universal message format and provides protocols for a great variety of possible situations.

An agent must possess some intelligence grounded on its knowledge base, reasoning mechanisms and learning capabilities. The intelligence of an agent is a prerequisite for all its other characteristics. Depending on the assignment of a particular agent, there are differences in types of information contained in its knowledge base. However, generally this information can be divided into two parts – the owner's profile and the agent's knowledge about its environment. It is very important to notice that the agent's knowledge base does not contain static information. Adversely, the agent continuously updates its owner's profile according to its owner's latest needs.

The proposed MAS is consisted of six different types of agents:

- *Repository Agent* (RA). The RA receives user profiles from Provider Agent and stores them into a semantic database. Additionally, the RA implements the algorithm for user clustering (i.e. the identification of a set of $\mathcal{M}$ mobile users which are physically close to each other and interested in the similar service). Although the calculation of set $\mathcal{J}$ from set $\mathcal{I}$ can be done in various ways, for user clustering we are using following steps:
    1. calculating similarities between the service profile and each user profile within the required distance (see [7] and [8]);
    2. ranking users according to semantic similarities in a descending order;
    3. possible candidates are determined by setting a similarity threshold (i.e. 60% of the highest value);
    4. enquiring possible candidates about their interest in the service;
    5. all the possible candidates who answered affirmative are added to the set $\mathcal{J}$.
- *Provider Agent* (PA). The PA manages users' profiles and communicates with the RA and SA.

---

1 Foundation for Intelligent Physical Agents: *http://www.fipa.org*

2 ACL specification: *http://www.fipa.org/specs/fipa00061/XC00061E.html*

- *Server Agent* (SA). The SA sends REQUEST (for group) message to PA requesting for group. After RA makes semantic matchmaking and group formation, PA sends names of users in group to SA. Each mobile user is represented by her/his MA.
- *Mobile Agents* (MAs). The MA receives ACL messages from SIP_to_ACL Agent, sends them to SA and then forwards them to ACL_to_SIP Agent.
- *ACL_to_SIP Agent* (ASA) and *SIP_to_ACL Agent* (SAA). Since the communication between server and mobile devices uses *Session Initiation Protocol* (SIP), SAA and ASA convert ACL messages to SIP and SIP to ACL messages, respectively.

3 Collaborative Downloading: the SOMA

Mobile users collaborate to reconstruct content $\mathcal{K}$. Collaborative downloading is formed as a set of auctions where users compete to determine which parts of content $\mathcal{K}$ to download directly from the service server and which to exchange with other users. In this article we propose Self-Organized Market Algorithm (SOMA) for solving this problem. Energy-efficient collaborative downloading based on SOMA has two phases: *market formation* and *auctions*.

3.1 Phase one: SOMA market formation

Inputs to the first SOMA phase are sets $\mathcal{J}$ and $\mathcal{K}$, while the output is defined as an association of every element in set $\mathcal{K}$ with one of the users from set $\mathcal{J}$. It is important to note that in this phase of SOMA, every element of set $\mathcal{K}$ is associated with only one user from set $\mathcal{J}$: this is the user which has to download the corresponding part of the requested content directly from the service server via GPRS and who will act as the initial seller of that content part in SOMA's second phase. This is also the reason why this first step of SOMA is called market formation.

SOMA's market formation phase begins with each mobile user j_a calculating her/his own utility function u_{j_a}:

$$u_{j_a} = w_{bc_1} \cdot bc_{j_a} + w_{ss} \cdot ss_{j_a}, \forall j_a \in \mathcal{J} \tag{1}$$

where $bc_{j_a} \in [0,100]$ indicates the *battery charge* level and $ss_{j_a} \in [0,100]$ the *signal strength* level of user j_a's mobile device, while w_{bc_1} and w_{ss} are non-negative constants defining the importance of battery charge level and signal strength level, respectively. While the bc_{j_a} and ss_{j_a} values are individual characteristics of every particular mobile user j_a, w_{bc_1} and w_{ss} are predefined by the telco and are common to all mobile users $\mathcal{J}$.

Afterwards, each mobile user j_a collects the values of the utility functions u_{j_b} ($\forall j_b \in \mathcal{J}, b \neq a$) of all other users. User j_a can then calculate the *number of content parts* np_{j_a} which she/he has to download directly from the service server as

$$np_{j_a} = round\left(\mathcal{P} \cdot \frac{u_{j_a}}{\sum_{b=1}^{\mathcal{M}} u_{j_b}}\right), \forall j_a \in \boldsymbol{J} \quad (2)$$

where

$$\sum_{a=1}^{\mathcal{M}} np_{j_a} = \mathcal{P} \quad (3)$$

Formula (3) denotes that sum of parts downloaded on all mobile devices in the mobile *ad hoc* network is equal to entire content.

3.2 Phase two: SOMA auctions

The input to the SOMA auctions phase is the output from the SOMA market formation phase, while the output is defined as the system state in which all users from set $\mathcal{J}$ possess all the parts of set $\mathcal{K}$ and are therefore able to reconstruct the requested content. In this algorithm phase, parts are exchanged between mobile users via Bluetooth. Each user that possesses at least one part of the content is a seller in the auctions and bids together with other sellers. Each user missing at least one part of the content is a buyer in the auctions and starts auctioning for the parts she/he wants to buy.

Seller bid prices are not explicitly determined by sellers but are implicitly calculated by buyers. Buyer j_a calculates bid price $p_{j_a j_b k_c}$ for part k_c of seller j_b as follows

$$p_{j_a j_b k_c} = pc_{j_b k_c} + dc_{j_a j_b k_c}, \forall k_c \in \mathcal{K}, \forall j_a, j_b \in \mathcal{J} \quad (4)$$

where $pc_{j_b k_c}$ presents user j_b's *production costs* for part k_c and $dc_{j_a j_b k_c}$ denotes the *delivery costs* for part k_c from seller j_b to buyer j_a.

The production cost is calculated by a seller as follows

$$pc_{j_b k_c} = w_{bc_2} \cdot \left(100 - bc_{j_b}\right) + w_f \cdot f_{j_b k_c} \quad (5)$$

where $f_{j_b k_c} \in [0,100]$ represents the *antipheromone* for user j_b's part k_c, while w_{bc_2} and w_f are non-negative constants defining the importance of the battery charge level and the *antipheromone*, respectively. After user j_b sells part k_c, $f_{j_b k_c}$ is set to the highest possible value (i.e. 100). As time passes, the *antipheromone* value decreases: after a while *antipheromone* is equal to 0 (i.e. the lowest possible value) and stays that way until user j_b sells part k_c again. The usage of the *antipheromone* creates balance in the market.

The delivery cost is calculated by buyer as

$$dc_{j_a j_b k_c} = w_{pb} \cdot pb_{j_a j_b} + w_{ps} \cdot \left(\mathcal{P} - ps_{j_a j_b}\right) \quad (6)$$

where $pb_{j_a j_b}$ represents the number of content parts that user j_a has already bought from user j_b and $ps_{j_a j_b}$represents the number of content parts that user j_a has already sold to user j_b, while w_{pb} and w_{ps} are non-negative constants defining the importance of $pb_{j_a j_b}$and $ps_{j_a j_b}$, respectively.

4 Simulation: A Proof-of-Concept Scenario

Our simulation results compare today's standard approach to mobile service provisioning (i.e. *plain mode*) with *collaborative download mode*. In *plain mode* each mobile user communicates only with a service server and downloads the whole desired content autonomously via certain mobile network technology. In *collaborative download mode* mobile users interested in the same content collaborate and download the desired content together. Content on the service server is divided into several parts and each part can be downloaded independently from the service server via certain mobile network technology and afterwards exchanged with other mobile users via certain *ad hoc* network technology.

4.1 SOMA pre-phase: group formation

Before the SOMA begins, group of users in collaboration must be specified (i.e. *SOMA pre-phase: group formation*, described in section 2.2). Fig. 2 shows agents' communication during the group formation phase. For clarity, we have not included ASA and SAA in Fig. 2, but every message sent by mobile user firstly goes through SAA who converts SIP message into ACL message and forwards it to corresponding MA. Analogously, every message sent to mobile user firstly goes through ASA who

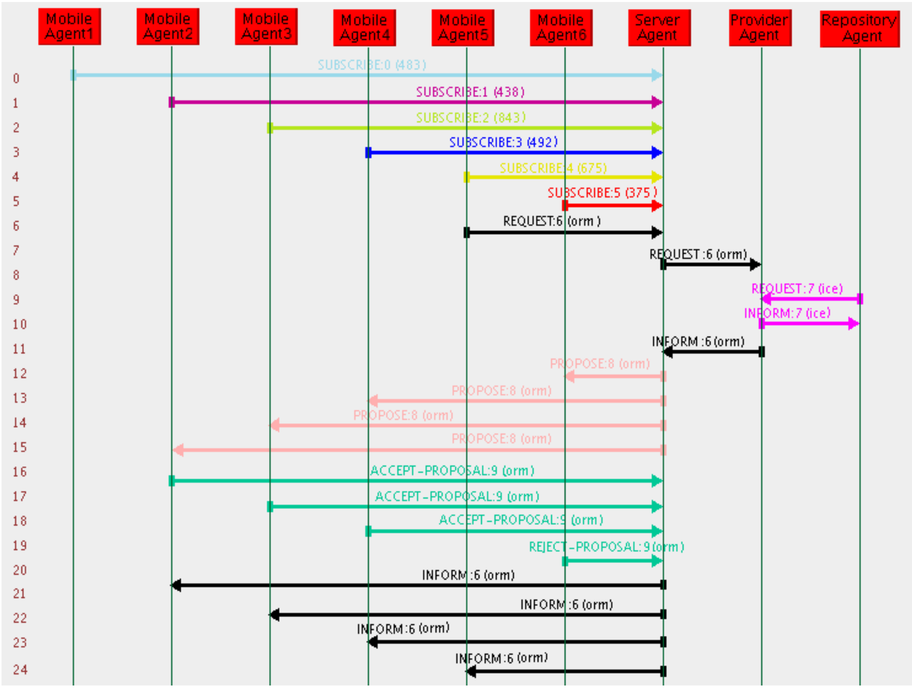

Figure 2 Group formation

makes new SIP message from received ACL message. In our proof-of-concept scenario there are six mobile users ($\mathcal{I} = \{i_1, i_2, i_3, i_4, i_5, i_6\}$) that are subscribers of a telco. Each user first sends *SUBSCRIBE* message to log-in. Afterwards, the whole system is on hold until one of the users requires (i.e. i_5) the certain (group-oriented) service. When such request occurs, the belonging MA sends *REQUEST* (for service) message to the SA. Afterwards, the SA sends *REQUEST* (for group) message to the PA who fetches user profiles stored in the semantic repository through *REQUEST* (for users) message sent to RA. After having performed the location filtering and clustering, the PA returns IMEIs of users in the same group as the requesting user i_5, because for them it is most likely that they would share user's i_5 interest in requested service. SA sends *PROPOSE* (for service) messages to all members of formed group, except to initiating user i_5 (i.e. users i_2, i_3, i_4 and i_6). SA is then blocked until it gets all of the answers or timeout occurs. It is very important to set a timeout to prevent a deadlock. Each MA can accept the proposition by sending back an *ACCEPT_PROPOSAL* (for service) message, or refuse it by sending *REJECT_PROPOSAL* (for service) message. All messages that SA did not receive before timeout occurred, are considered as *REJECT_PROPOSAL* (for service) messages. At last, the SA then sends *INFORM* (for group) message to those MAs which confirmed their interest in the movie (i.e. $\mathcal{J} = \{j_1, j_2, j_3, j_4\}$, where $j_1 \equiv i_2, j_2 \equiv i_3, j_3 \equiv i_4$ and $j_4 \equiv i_5$). Afterwards the download can be started with the SOMA phase one (i.e. market formation).

4.2 Phase one: SOMA market formation

After group formation, in our system there are four mobile users ($\mathcal{M} = 4$) while the content on the server is divided into five parts ($\mathcal{P} = 5$) where each part is size of 67 KB. Data in Tab. 2 shows battery charge bc_{j_a} and signal strength ss_{j_a} levels of each mobile device at moments t_0 and t_2.

Table 2. The battery charge and signal strength levels at moments t_0 (beginning of the SOMA) and t_2 (some moment during the SOMA phase two)

	j_1		j_2		j_3		j_4	
	t_0	t_2	t_0	t_2	t_0	t_2	t_0	t_2
bc_{j_a}	68	58	26	20	65	63	37	34
ss_{j_a}	92	92	54	53	55	66	83	88

Each mobile user j_a calculates the number of parts that she/he must download using the data in Tab. 2, formula (1) with parameters set to $w_{bc_1} = w_{ss} = 0.5$ and formula (2)

$$np_{j_1} = 2, np_{j_2} = np_{j_3} = np_{j_4} = 1 \tag{7}$$

Users then download designated parts directly from the service server via GPRS. The resulting distribution of parts at the end of the SOMA market formation phase (i.e. moment t_1) is shown in Tab. 3. Mobile user j_a sets *antipheromone* value for downloaded parts to the lowest possible value. During the second SOMA phase users exchange content parts via Bluetooth. Pairs of exchanging users are determined by an auction mechanism.

Table 3. *Antipheromone* values at moments t_1 and t_2

	j_1		j_2		j_3		j_4	
	t_0	t_2	t_0	t_2	t_0	t_2	t_0	t_2
k_1	0	100		31				35
k_2	0	66		6				22
k_3			0	56		51		
k_4		68			0	11		
k_5		12				89	0	74

4.3 Phase two: SOMA auctions

At some moment t_2 user j_2 starts a new auction for missing parts (i.e. parts k_4 and k_5) and collects sellers' production costs. Sellers calculate production costs using formula (5) with the associated parameters set to $w_{bc_2} = w_f = 0.5$ and the data from Tab. 2 and Tab. 3

$$pc_{j_1k_5} = 27,\ pc_{j_3k_4} = 24, pc_{j_4k_5} = 70 \quad (8)$$

Now user j_2 can calculate sellers' delivery costs using formula (6) and the data from Tab. 4

$$dc_{j_2j_1k_5} = 3.5,\ dc_{j_2j_3k_4} = 2, dc_{j_2j_4k_5} = 2 \quad (9)$$

Finally, user j_2 can calculate sellers' bids using formula (4) with data calculated with formulas (8) and (9)

$$p_{j_2j_1k_5} = 30.5,\ p_{j_2j_3k_4} = 26,\ p_{j_2j_4k_5} = 72 \quad (10)$$

Since, the price $p_{j_2j_3k_4}$, according to (10), is the lowest, user j_2 decides to download part k_4 via Bluetooth from user j_3.

Table 4. The number of bought and sold parts for user j_2 at moment t_2

	j_1	j_2	j_3	j_4
pb_{j_2}	2	0	0	2
ps_{j_2}	0	1	1	0

In our simulation overall energy consumption is calculated as a size of chunk multiplied by mobile devices' characteristic energy consumption per bit shown in Tab. 1. Chunk is the information unit which includes only data packets in *plain mode* and both data packets and *gossip messages* in *collaborative download mode*. Gossip messages are used to propagate content availability information through users' bids in SOMA auctions. Despite gossip messages that present message overhead, the presented *collaborative download mode* consumes only 0.48 J, while the same *four-mobile-user* scenario in *plain mode* consumes 1.52 J of mobile users' energy.

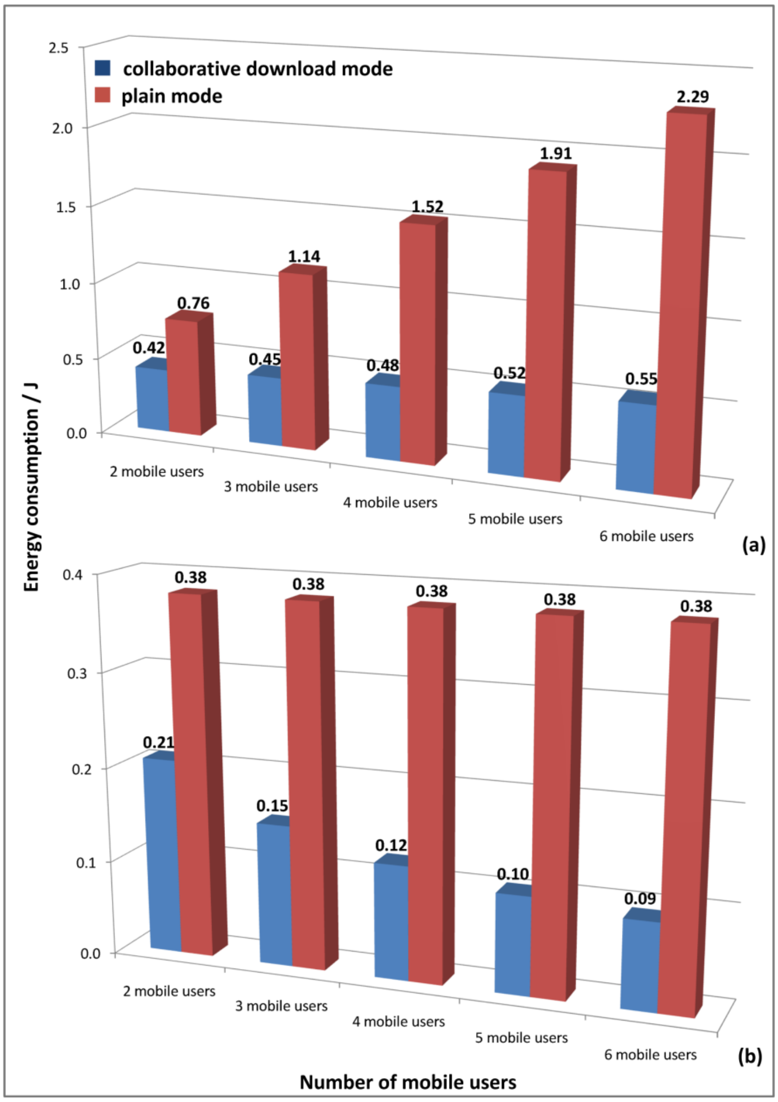

Figure 3 Energy consumption: (a) in the system, (b) per mobile user

Additionally, energy consumption in system of two to six users is shown in Fig. 3(a): for six users *collaborative download mode* saves 75% of mobile devices' energy. From Fig. 3(b) it is obviously that in plain mode each user downloads whole content autonomously using GPRS, no matter how many users are in the system. However, in *collaborative download mode*, whole content is downloaded in *ad hoc* network just once using GPRS (see formula (3)) and for further chunk exchange Bluetooth is used. Since, Bluetooth communication spends less energy than GPRS, total sum of exchanged chunks divided with amount of users in the system is decreasing with greater number of users in the system.

5 Conclusion

In *ad hoc* networks (e.g. Bluetooth-based network), every node acts as a router for other nodes. Thus, collaboration at the network level can be resonated at the application level. In this article we propose SOMA, an energy-efficient Bluetooth-based algorithm for users' collaboration in mobile service provisioning. The simulation results show a significant decrease of energy consumption of mobile users for *collaborative download mode* in front of *plain mode*. This is of prime importance since limited energy supply is one of the greatest challenges faced by both mobile users and telcos.

Acknowledgements. The authors acknowledge the support of research project "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639), funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Lee, M., Lee, J., Cho, Y.: How a Convergence Product Affects Related Markets: The Case of the Mobile Phone. ETRI Journal, 31(2), 215-224 (2009)
2. Yoon, J.-L.: Telco 2.0: A New Role and Business Model. Communications, 45(1), 10-12 (2007)
3. Pentikousis, K.: In Serach of Energy-Efficient Mobile Networking. IEEE Communications 48(1), 95-103 (2010)
4. Nandan, A., Das, S., Pau G., Gerla M.: Co-Operative Downloading in Vehicular Ad Hoc Wireless Networks. In: Proceedings of the 2nd Annual Conference on Wireless On-demand Network Systems and Services (WONS'05), St. Moritz (Switzerland), pp. 32-41 (2005)
5. Zhang, J., Zhang, Q., Jia, W.: A Novel MAC Protocol for Cooperative Downloading in Vehicular Networks. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '07), Washington (USA), pp. 4974-4978 (2007)
6. O'Hara, K.J., Nathuji, R., Raj, H., Schwan, K., Balch, T.: AutoPower: Toward Energy-aware Software Systems for Distributed Mobile Robots. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA'06), pp. 2757-2762 (2006)
7. Vrdoljak, L., Bojic, I., Podobnik, V., Kusek, M.: The AMiGO-Mob: Agent-based Middleware for Group-oriented Mobile Service Provisioning. In: Proceedings of the 10th International Conference on Telecommunications (ConTEL'09), Zagreb (Croatia), pp.97-104 (2009)
8. Podobnik, V., Petric, A., Trzec, K., Galetic, V.; Jezic, G.: Agent-based Provisioning of Group-oriented Non- linear Telecommunication Services. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 198-204. Springer, Heidelberg (2008)
9. Cockayne, W., Zyda, M.: Mobile Agents. Manning Publications, Greenwich (USA) (1998)
10. Bradshaw, J.: Software Agents. MIT Press, Cambridge (USA) (1997)
11. Podobnik, V., Petric, A., Jezic, G.: An Agent-Based Solution for Dynamic Supply Chain Management. Journal of Universal Computer Science 14(7), 1080-1104 (2008)
12. Jennings, N., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. Journal of Autonomous Agents and Multi-Agent Systems 1(1), 7-36 (1998)