A Collaborative Service Discovery and Service Sharing Framework for Mobile Ad Hoc Networks

Haidar Safa[1], Hassan Artail[2], Hicham Hamze[2], and Khaleel Mershad[2]

[1] Department of Computer Science
[2] Department of Electrical and Computer Engineering
American University of Beirut, Beirut, Lebanon
{hs33,hartail,hhh14,kwm03}@aub.edu.lb

Abstract. Service sharing and discovery play a relevant role in mobile ad hoc environments. Upon joining a self-organizing network, mobile nodes should be able to explore the environment to learn about, locate, and share the available services. In this paper, we propose a distributed and scalable service discovery and sharing framework for ad hoc networks. The proposed framework defines three types of nodes: service directories, service providers and requesting nodes. Service directory nodes act as mediators for lookup requests from requesting nodes. Joining service provider nodes register their services with the nearest service directory. A requesting node discovers the available services by submitting requests to its nearest service directory which determines the node providing the requesting service. The performance of the proposed model is evaluated and compared to the broadcast-based model that has been extensively studied in the literature.

Keywords: service discovery, mobile ad hoc networks, cooperative computing, ns2.

1 Introduction

Service discovery protocols like Service Location Protocol (SLP) of IETF [5], Sun's Jini [1], and IBM's Salutation [10] that have been developed to help applications discover remote services residing on machines in a wired network do not directly dwell on mobile ad-hoc network (MANET) environments where self-configurability is the key.

Several service discovery solutions for MANETs were recently proposed [2], [4],[6]. A distributed service discovery architecture that relies on a virtual backbone for locating and registering available services was presented in [6]. This architecture creates a mesh structure from a subset of a given network graph that includes the nodes acting as service brokers and also a subset of paths connecting them. Then it establishes sub-trees rooted at service requesting nodes and registering servers for efficient dissemination of the service discovery control messages. The disadvantage of this approach is that it totally relies on multicasting and broadcasting techniques for service discovery and registration. In [2] and

K. Li et al. (Eds.): NPC 2007, LNCS 4672, pp. 151–160, 2007.

[4], two semantic-routing-based service discovery schemes were described and are called Group-based Service Discovery (GSD) and Candidate Node Pruning enhanced Group-based Service Discovery Protocol (CNPGSDP), respectively. In GSD, services are classified into several groups and each server periodically generates service advertisement packets that can be forwarded into the network. To restrict the spreading range of these packets, the maximum number of hops they can travel is limited (denoted as d). The service advertisement packet contains not only information about the service provided by the sender, but also about the groups to which the services provided the sender's d-hop neighbors belong. Each node maintains a cache called Service Information Cache (SIC), which is used to store service advertisement packet temporally. This makes a node know not only about the services provided by the servers in its d-hop neighbor set, but also about the groups to which these services belong. When a node needs services and there is no matched services in its SIC, the node constructs a service request packet and forwards the packet towards some elaborately selected nodes in unicast mode. When receiving a packet sent by a node, each selected node should forward the packet further, unless the packet is matched or exceeded its hop limit. If the node that receives a new request packet finds a matched service, it sends out a service reply packet in unicast mode directly to the sender of the service request packet. The reply packet will be relayed to the source of the service request packet using a traditional ad-hoc routing protocol (like AODV and DSDV [8]) or along the reverse path after retracing the path traversed by the request packet. As to the CNPGSDP protocol, it was proposed to enhance GSD by reducing the number of unicast messages using a techniqye called Broadcast Simulated Unicast (BSU) in which several unicast request packets are replaced with one request packet transmitted in broadcast mode with all unicast receivers enclosed. These semantic-routing-based approaches have several issues. First, when no service group finds a match in the service cache, the request would still have to be broadcasted to the whole network. Second, the selective forwarding process might result in false forwards, meaning the request might be forwarded to a region where the service is no longer available (due to mobility of nodes) or has the right group but not the exact service.

We observe that these proposed protocols are either request broadcast-based, advertisement-based, or a combination of both. The broadcast-based solution, in which a service discovery request is broadcast throughout the network and the node that contains the service responds with a service reply, suffers from several drawbacks [11]: 1)it scales poorly with increasing network diameter and size; 2)it utilizes resources and computation power on all nodes; and 3)it is heavy on network bandwidth. In the advertisement-based solution the services advertise themselves to all of the nodes [9] and each node interested in discovering services will cache the advertisements. The advertisements are matched with service requests and a result is returned. In this solution, the cache size increases with the number of services while many of the nodes may have limited memory that does not allow them to store all the advertisements. Similar to the broadcast

method, this approach is also inefficient in terms of bandwidth usage, since the whole network has to be periodically flooded with advertisements.

In this work, we propose a scalable and distributed service discovery and sharing model (DSDSM) for self-organized networks. This model does not employ broadcasting for service requests and advertisements for service providers and as a result, avoids many of the issues that are associated with the above described approaches. The remaining part of the paper is organized as follows. Section 2 describes the proposed DSDSM model while Section 3 presents the simulations and analyses performed to evaluate the proposed model and describes the experimental results. Finally, Section 4 contains some concluding remarks.

2 Proposed Framework

2.1 Basic Concept

In the proposed Distributed Service Discovery and Sharing Model (DSDM) the network is composed of service directory nodes (SDs), service provider nodes (SPs), and requesting nodes (RNs). The main task of the Service Directory nodes (SDs) is to maintain a list of all the services provided by their nearby service provider nodes. We assume that every node that joins the network registers the services that it can provide with the nearest SD. A RN node that is requesting a service can discover the available services in the network by forming and submitting requests to its nearest SD node. SDs act as distributed indices for services by storing service description entries along with the addresses of the service provider. Service description includes information like type of service, signature, service provider, time to live (TTL). The RN can ask for a specific service (i.e., print service, scan service, internet service), a specific type of services (i.e., all music services, all food services), or all the services registered in the SD. If the SD receives a service discovery request and the requested service is not cached (i.e., a miss), the request is forwarded to the next nearest SD node to retrieve the required information. Upon receiving the response, the RN caches the service description so that the next time the node requests the same service it can retrieve the service description from its own cache.

Nodes can determine the nearest SD from the routing table assuming that table-driven proactive routing protocols such as DSDV are used [8]. Sending requests to or registering services with the nearest SD helps in minimizing delay and network congestion. If the contacted SD does not have the description for the requested service, it also forwards the request to its nearest SD that has not been visited yet by this request.

2.2 System Configuration

The SDs are the central component of the system and must be selected carefully. Preference is given to nodes with sufficient resources. Nodes summarize their capabilities by calculating a special score using the following parameters: availability

time in the network ($TIME$), battery life (BAT), available bandwidth (BW), and available memory for caching (MEM). To be considered a candidate SD, the node must meet a minimum criterion in each category. That is,

$$\{D_k\}|R_k^x > \Theta_x, \forall x \in \{TIME, BAT, BE, MEM\} \quad (1)$$

where $\{D_k\}$ is the set of candidate devices, R_k^x is a resource for device k, and Θ_x an empirically-set threshold for resource X. If $\{D_k\}$ includes more than one device, then the one with the maximum weighted score is selected. That is, if device j is the selected one, then

$$SC_j = max\{SC_k\}|SC_k = \sum \alpha_x R_k^x \quad (2)$$

where SC_j is node k's score, k refers to one of the devices satisfying the condition in (1), and α_x is the weight associated with resource X such that $\sum \alpha_x = 1$.

2.3 System Formation and Operation

The system will start with one SD then add SDs on demand. When the network starts every node joins the network by broadcasting a HELLO packet that includes its score. Using the data from HELLO packets, each node will know about the scores of the other nodes. The node with the highest score will be considered as the first SD. Nodes meeting the conditions (1) and (2) will be considered as potential SD candidates. The first SD broadcasts a *DSDSM Information Packet* (DIP) that contains itself as the only SD in the list of SDs. the list of SDs is the main parameter in the DIP packet which is broadcast only when this list changes. Node that joins the network after forming the network broadcast a HELLO packet that includes its score and requests the list of SDs from a nearby node by sending a unicast *SDs List Request Packet* (SDLRP). The nearby node replies by sending a unicast DIP. When the number of nodes in the network increases, the last joining SD will start a timer and send a *SD Assignment Packet* (SDAP) to the SD candidate node with the next highest score to assume the role of SD. If the latter accepts the invitation it broadcasts a DIP message with its identifier added to the list of SDs. This message will be interpreted by the inviting SD as acknowledgment message. Otherwise, if the timer expires without receiving the DIP message, the next SD candidate on the row will be invited. Moreover, DSDSM depends on the table-driven proactive routing protocols to detect nodes going offline. If an SD goes offline, the candidate SD node with the highest score will broadcast a DIP to announce its new role as an SD. To protect for situations in which this or other candidates take no action, all candidates that meet the condition in (1) start a timer after detecting the departure of an SD. The second-highest-score candidate will wait a period of T and will assume the role of an SD and broadcast a DIP if it hears nothing. The third-highest candidate waits a period of $2T$ before it sends a DIP if it hears nothing, and so on. The service provider (SP) holds for each service the SDs that it registered its service with, thus allowing for rebuilding SD entries

when an SD goes offline. Upon receiving the DIP from the replacement SD, the concerned SPs reply by sending a *SD Registration Packet* (SDRP) that contains the descriptions that were used to reference the lost SD. The DIP will also serve to inform nodes about the replacement and prompt them to update their SD lists. If an SP goes offline, the SDs will detect its departure when the routing protocol updates their routing tables, and will update their entries accordingly.

The message sequence diagram of the DSDM model is shown in Fig. 1. It consists of three phases. In the top *registration* phase each SP willing to share its services sends an SDRP to its nearest SD. In the *discovery* phase the RN asks its nearest SD for a specific service by sending a *Service Request Packet* (SRP) to its nearest SD. If this SD does not have a matching service, it adds its address to the SRP to indicate that it has been visited and then sends this modified SRP to the nearest SD that has not been checked yet. This continues until a hit occurs where the SD replies with a *Service Reply Packet* (SREP) that comprises the service description including the service provider address. In the bottom *invocation* phase, the RN sends a *Service Invoke Packet* (SINVK) to the SP node (SN) implementing the service. The RN gets the result from the SP via a *Service Response Packet* (SRESP).

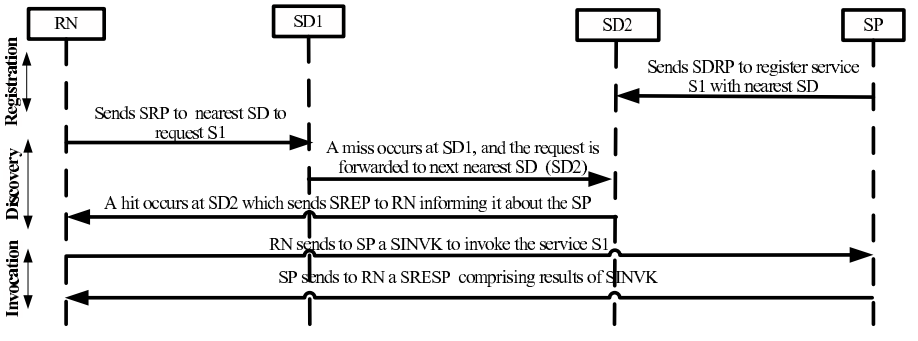

Fig. 1. Message sequence diagram in DSDM

3 Simulations

3.1 Simulation Parameters and Environment

To study the performance of the proposed approach, we have conducted several simulation scenarios and experiments using the network simulator ns-2 [7]. In these simulations, the distributed coordination function (DCF) of IEEE 802.11 is used as the underlying MAC protocol. The radio interface is based on Lucent's Wave LAN technology with 100 meters of nominal propagation range and 2 Mbps of nominal bit rate. The network topography is 1000×1000 m^2 in which 100 nodes were randomly deployed. The node speed (V_{max}) varied from 0 to 2 meters/sec using the Random way point model (RWP).

The duration of each simulation scenario is set to 1000 seconds. The first 150 seconds are given for the DSDV routing agents to populate the routing tables of each node. Then the mobile nodes that are willing to share their services starts registering them with their nearest service directory (SD) node by sending SDRP messages. We assume that 20 services are provided by a group of mobile nodes chosen randomly and the number of SDs (N_{SD})has been assigned in the network.

After the registration period is complete, the RNs start submitting SRP (service request packets) messages at a request rate (λ_i= 0.1 req/sec) per request node. These requests ask for diverse services using a Zipf-like access pattern, which has been used frequently to model non-uniform distributions [12]. In Zipf law, a service ranked i ($1 \leq i \leq n_s$) is accessed with probability $1/(i^{\Theta} \sum_{k=1}^{n_s} 1/K^{\theta})$ where Θ ranges between 0 (uniform distribution) and 1 (strict Zipf distribution). The access pattern is also location-dependent in the sense that nodes around the same location tend to choose the same set of services (i.e., have similar interests).

3.2 Delay Estimation

We studied the delay of the system as the number of service directories (N_{SD}) varied between 1 and 10. The trend of the service invocation delay is illustrated in Fig. 2 (a). Both, the average service discovery delay and the service invocation delay (invocation delay includes also discovery delay) are plotted in the figure. Naturally, as the number of SDs is increased, the average service invocation and discovery delays in the system increase as a function of N_{SD}, and this is mainly attributed to the increasing number of hops that the SRP packets traverse in order to find the particular services. In Fig. 2 (b) shows the average service invocation delay of DSDSM for N_{SD}=3 and N_{SD}=6 and compared to the invocation delay when employing a broadcast scheme.

3.3 Bandwidth

To study the traffic load on each SD, we assume that T_{SD_i} is the traffic traversing SD_i. T_{SD_i} is obtained by computing the number of each incoming and outgoing packets traversing the node SD_i. The average traffic for each SD is $\sum_{i=1}^{N_{SD}} T_{SD_i}/N_{SD}$ and is estimated simply by dividing the sum of such traffic by the number of service directories in the system, N_{SD}.

This process gets repeated for each value of N_{SD}. For each value of N_{SD}, averages of the results of 10 different trials were plotted in Fig. 2(c). Each trial lasted for 650 seconds and corresponds to a scenario in which a set of service providers were chosen randomly. Fig. 2(c) shows that as the number of service directories increases, the average traffic load at each SD decreases. This is because with multiple SDs, it is likely that the description of the desired service will be found at a closer SD. Fig. 2(d) shows that as the number of service directories increases, the traffic overhead incurred at each SD increases. This is can be justified since as the number of SDs increases, the average number of SDs that are traversed increases. Fig. 2(d) shows also that with more SDs the effect

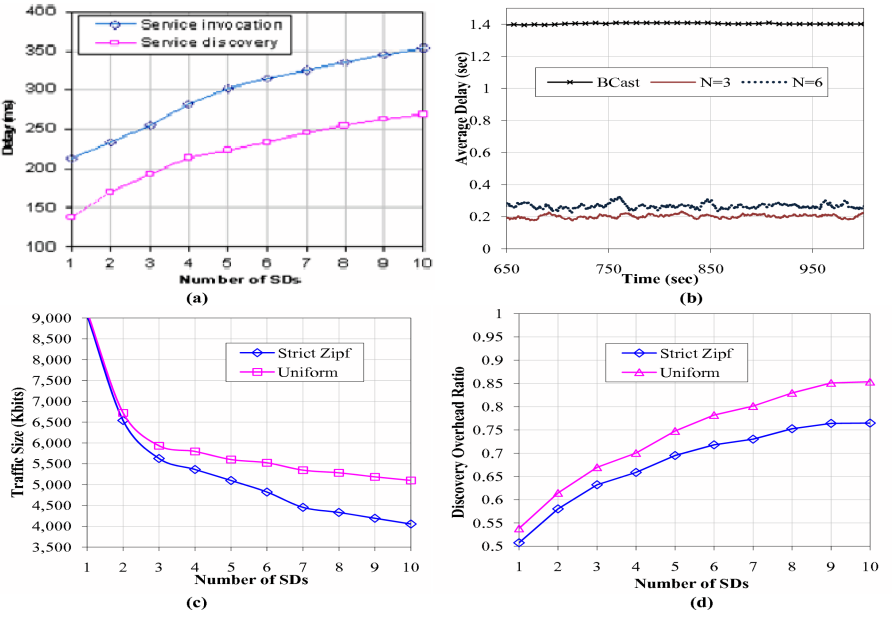

Fig. 2. (a) Service discovery delay and service invocation Delay versus N_{SD}, (b)Average service invocation delay, (c)SD Traffic versus N_{SD},(d) Overhead ratio

of the strict-Zipf distribution diminishes since popular services could be found across a growing number of SDs.

3.4 Estimating the Maximum Number of SDs

We observe that as the number of SDs in the system increases, the average delay to receive a response for a request will also increase. Intuitively, this is because the service request packet will traverse more SDs on average. It follows that a limit must be set on the number of SDs so as to prevent the number of SDs to increase indefinitely. One way to compute this limit is to set an upper limit on the average delay in the system ($E[\tau]$) and make this limit equal to the average delay of accessing a point at the corner of the topography and getting back the reply from it ($E[\tau_{corner}]$). Thus, the maximum number of SDs (N_{SD}^{max}) can be set as follows:

$$N_{SD}^{max} = max(N_{SD}) | E[\tau] < E[\tau_{corner}] \tag{3}$$

We now have to derive several parameters that are involved in computing N_{SD}^{max}: 1) The expected number of hops between any two nodes in the network ($E[H]$), 2) The expected number of hops within the SDs system ($E[H_{SD}]$)(different from $E[H]$ because of the nearest node selection), 3) The expected number of hops to the corner of the topography ($E[H_{CO}]$), and 4) The response time of the system.

To derive $E[H]$, we assume a rectangular topography with an area a×b (where $a < b$). Two nodes are capable of forming a direct link if the distance S between

them is less than the node maximum transmission range r_0 . Using stochastic geometry, the probability density function (pdf) of x (x is a random variable denoting the straight line distance between two nodes) is given in [3] as

$$P_x(x) = \frac{4x}{a^2b^2}(\frac{\pi}{2}ab - ax - bx + 0.5x^2)\, For\, 0 \leq x < a < b \tag{4}$$

It is concluded that if two nodes are at distance S_0 from each other, the number of hops between them when there is an infinite number of nodes would tend towards x_0/r_0. Hence, $E[H]$ is equivalent to dividing expected distance between two nodes $E[x]$ by r_0. E[H] is given in [3] as

$$E[H] = \frac{0.521a}{r_0} \tag{5}$$

$E[H]$ represents the expected number of hops when only one destination choice is available. Finding the expected number of hops to nearest SD node, is like selecting the minimum from a set of multiple independent random variables. Given N_{SD} random variables represented by a vector of random variables $X = (x_1, x_2, x_3, ..., x_{N_{SD}})$, the pdf of selecting the minimum of X is:

$$P_{min(x)} = \sum_{i=1}^{N_{SD}} P_X(x_i = r | x_j > r, \forall j \neq i) = N_{SD} P_{x_i}(x_i = r)\left(P_{x>r}(r)\right)^{N_{SD}-1} \tag{6}$$

Referring to the pdf in expression (4), the probability of the distance being greater than a value r is $P_{x>r}(r) = \int_x^\infty P_x(x)dx$. Hence, the expected distance to the nearest SD given N_{SD} choices is:

$$E[P_{min(X)}] = \int_0^\infty rP_{min(x)}(r)dr = N_{SD}\int_0^\infty rP_x(r)\left(\int_r^\infty P_x(r)dr\right)^{N_{SD}-1} \tag{7}$$

It follows that the expected number of hops to the nearest SD is $E[P_{min(x)}]$ divided by r_0 and given as

$$E[H_{SD_{nearest}}] = \frac{N_{SD}}{r_0}\int_0^\infty rP_x(r)dr\left(\int_r^\infty P_x(r)dr\right)^{N_{SD}-1} \tag{8}$$

To calculate the average number of hops to get to the SD with the desired service data from an RN, we multiply P_i, the probability that SD_i has the desired service data, with the average number of hops to contact each SD and then take the sum. For simplicity, P_i can be set to $\frac{1}{N_{SD}}$. Hence, the expected number of hops to get to the SD with the service data is:

$$E[H_{SD_{Data}}] = \sum_{i=1}^{N_{SD}} P_i E[H_{SD_{nearest}}] \tag{9}$$

To derive E$[H_{CO}]$, the expected number of hops to the corner of the topography, we calculate first the expected distance from a node to the corner of the topography, $E[S_{CO}]$, using tables of integration and the Mathematica software:

$$E[S_{CO}] = \int_0^a \int_0^a \frac{1}{a^2}\sqrt{x^2+y^2}dxdy = \frac{1}{3}\left[\sqrt{2} + log(1+\sqrt{2})\right] a = 0.7652a \quad (10)$$

Next, to get $E[H_{CO}]$ we divide $E[S_{CO}]$ by the transmission range r_0:

$$E[H_{CO}] = E[S_{CO}]/r_0 = 0.7652a/r_0 \quad (11)$$

With a topography of $1000 \times 1000m^2$ and a transmission range of $100m$, $E[H_{CO}]$ is therefore 7.65 hops.

To compute the average system delay $E[\tau]$, we assume that T_{in} is the average delay of transmitting packets between nodes in a MANET. For simplicity we assume that T_{in} takes into account factors such as node location, packet size, number of hops, and congestion. The delay for accessing a SD node that has the service description is $T_{in} \times E[H_{SDdata}]$, plus an additional delay of $T_{in} \times E[H]$ that is incurred to transmit the reply back to the RN. Then $E[\tau] = T_{in}(E[H_{SD_{Data}}] + E[H])$. Similarly, the delay for accessing a SD at the corner and getting back the reply is $E[\tau_{corner}] = T_{in}(2E[H_{CO}])$.

With the above information, we can now apply the expression in (3) to determine the value of N_{SD}^{max}. After substituting $E[\tau]$ and $E[\tau_{corner}]$ in (3) and plugging in the values of $E[H]$, $E[H_{SD_{Data}}]$, and $E[H_{corner}]$ using (5), (9) and (11) respectively, we get the inequality:

$$E[H_{SD_{Data}}] - 1.0094a/r_0 < 0 \quad (12)$$

The expression in (12) was evaluated for different values of N_{SD} using Matlab and plotted in Fig. 3. As illustrated in the figure, the upper bound of N_{SD}, N_{SD}^{max}, is 7 for the chosen threshold (i.e.,E$[\tau_{corner}]$).

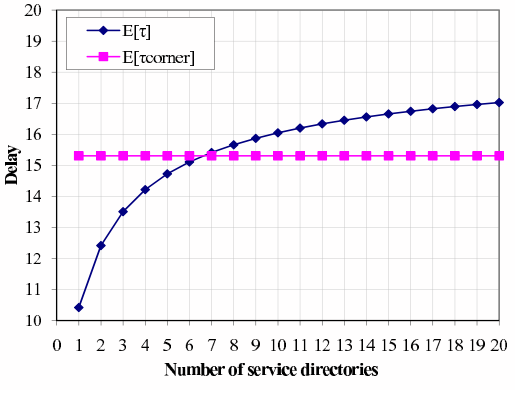

Fig. 3. $E[\tau]$ versus $E[\tau_{Corner}]$ for a 1000×1000 m^2 topography with r_0 set to 100

4 Conclusion

This paper described a Distributed Service Discovery and Sharing Model baptized as DSDSM. In the proposed model, service provider nodes register their services with the nodes designated as service directories which act as mediators for lookup requests from requesting nodes. A node requesting a particular service contacts its nearest service directory which will in turn forward the request to another service directory in a sequential manner if it does not know who offers the service. Once the requesting node has the service description and the address of the service provider, it can invoke the service directly from the service provider. Both analytical and experimental performance evaluation were conducted to study the average response time and bandwidth consumption of the system, while focusing on the service discovery functionality. The performance of the proposed model was compared to that of the broadcast-based service discovery model that was proposed in the literature, and was found to outperform it by significant amounts.

Acknowledgments. This work was supported in part by a grant the Lebanese National Council For Scientific Research (no. 111135 022141, 2006/2007).

References

1. Arnold, K., Osullivan, B., Scheifler, R., Waldo, J., Wollrath, A.: The Jini Specification (The Jini Technology). Addison-Wesley, Reading, Mass (1999)
2. Chakraborty, D., Joshi, A., Yesha, Y., Fin, T.: Toward Distributed Service Discovery in Pervasive Computing Environments. IEEE Transactions on mobile computing 5(2), 97–112 (2006)
3. Bettstetter, C., Eberspacher, J.: Hop distances in homogeneous ad hoc networks. In: Proc. of the 57th IEEE semiannual Vehicular Technology Conf., pp. 2286–2290. IEEE Computer Society Press, Los Alamitos (2003)
4. Gao, Z., Wang, L., Yang, M., Yang, X.: CNPGSDP: An efficient group-based service discovery protocol for MANETs. Computer Networks (in Press)
5. Guttman, E., Perkins, C., Veizades, J.: Service Location Protocol. RFC 2165 (1997)
6. Kozat, U., Tassiula, L.: Service discovery in mobile adhoc Networks: an Overall perspective on architectural choices and network Layer support issues. Adhoc Networks 2(1), 23–44 (2004)
7. NS-2 simulator (2006), `http://www.insi.edu/nsnam/ns`
8. Perkins, C.: Ad Hoc Networking, 2nd edn. Addison-Wesley, Reading (2004)
9. Ranganathan, A., Campbell, R.: Advertising in a pervasive environment. In: Proc. Second ACM Int'l Workshop Mobile Commerce, pp. 10–14. ACM Press, New York (2002)
10. The Salutation Consortium Inc: Salutation Architecture Specification Part 1, Version 2.1 (1999), `http://www.salutation.org`
11. Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., Sheu, J.-P.: The broadcast storm problem in a mobile ad hoc network Source. Wireless Networks 8(2-3), 153–167 (2002)
12. Zipf, G.: Human Behavior and the Principle of Least Effort. Addison-Wesley, Reading (1949)