1-2007

BusinessFinder: Harnessing presence to enable live Yellow Pages for small, medium and micro mobile businesses

D. CHAKRABORTY

K. Dasgupta

S. Mittal

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

C. OBERLE

See next page for additional authors

DOI: https://doi.org/10.1109/MCOM.2007.284550

Published in IEEE Communications Magazine, 2007 January, Volume 45, Issue 1, Pages 144-151
https://doi.org/10.1109/MCOM.2007.284550

BusinessFinder: Harnessing Presence to Enable Live Yellow Pages for Small, Medium and Micro Mobile Businesses

Dipanjan Chakraborty, Koustuv Dasgupta, Sumit Mittal, and Archan Misra, IBM Research
Anuj Gupta, IBM Software Group
Eileen Newmark, IBM Systems and Technology Group
Christopher L. Oberle, IBM Global Services

Abstract

Applications leveraging network presence in next-generation cellular networks have so far focused on subscription queries, where "presence" information is extracted from specific devices and sent to entities who have subscribed to such presence information. In this article we present BusinessFinder, a service that leverages the underlying cellular presence substrate to provide efficient, on-demand, context-aware matching of customer requests to nomadic micro businesses as well as small and medium businesses having a mobile workforce. Presence, in the context of BusinessFinder, is not simply limited to phone location and device status, but also encompasses dynamic attributes of vendors (both "mobile" and "static"), such as their current availability and workload, expertise and reputation. Besides presenting the architecture and implementation of BusinessFinder with a centralized source of context, we also describe early work on a novel resource-aware query routing algorithm that can efficiently support BusinessFinder query semantics in distributed presence environments of the future.

Introduction

Presence, loosely defined as the network's ability to track and disseminate dynamic, contextual attributes of individual devices or users (such as a phone's location or an individual's status on an instant messenger client), is widely touted [1] as a "killer service" for next-generation cellular networks. Examples of presence in existing *user-user consumer applications* include live Instant Messaging connectivity status of designated contacts, Push-to-Talk (PoC) or walkie-talkie service [2] on a cellular network and "buddy alerts" (based on proximity of designated friends). Increasingly, presence is also becoming a generic interface for *user-application integration* across enterprise and service provider networks. This convergence of presence-based applications across both enterprise and telecom networks is being driven by the common adoption of signaling standards, such as the Session Initiation Protocol (SIP) [3] and SIMPLE [4].

In this article we present *BusinessFinder*, a service offering that leverages upon the underlying cellular presence substrate to provide *efficient*, *on-demand*, *context-aware* matching of customer requests to *nomadic* vendors. From our perspective, some of BusinessFinder's features make it one of the few early efforts to investigate the impact of presence-aware applications in the small-medium business (SMB) segment, especially in emerging economies (e.g., India, China, Russia, and Brazil) where small and medium businesses constitute the fastest growing segment of the economy. (As an illustration, [5] forecasts that the SMB sector in India will spend $7.7 Billion on IT in 2006, with an annualized growth of 26 percent that is three times the overall GDP growth rate). A key characteristic of such "emerging" markets is the highly decentralized and fragmented nature of consumer interaction with various business vendors — BusinessFinder is specifically targeted to enable easy, targeted interaction with vendors such as plumbers, florists, electricians, and auto mechanics who operate either as individuals (*micro businesses*) or from relatively small (less than five to ten individuals) shops (SMBs).

BusinessFinder enables the cell phone to be used not just as a traditional communication tool, but also as a business tool. Conceptually, BusinessFinder may be viewed as a "Live Yellow Pages" service that factors in the actual mobility of both *the requester* (the customer seeking a service) and the vendors (e.g., the electrician or plumber offering a service) to perform on-demand matching. BusinessFinder differs from existing location-based services (e.g., lookups for static restaurants, gas stations, and ATMs) by explicitly addressing *vendor nomadicity*. It considers the instantaneous (or predicted) relative locations of requesters and mobile vendors, as well as additional presence-driven attributes in the matching process. While not directly a focus of this article, BusinessFinder itself can become an intermediate "directory service" (e.g., as part of a "Web-based mashup" displaying dynamic vendor information on a map, with associated "click to call" semantics). Companies having a mobile workforce can also use the features of BusinessFinder to manage their resources and provide improved dispatch of their workforce to serve customer requests.

BusinessFinder has three key features specifically designed for conditions in emerging economies:

- It performs query lookups based on not just the changing location of individual vendors, but also their additional presence attributes, such as current workload, expected availability, service profiles and reputation, cost, and expertise. This ability to directly match consumer requests to individual mobile vendors is particularly critical in developing economies, where people typically do not use well-established business aggregators — for example, unlike the widespread use of the American Automobile Association (AAA) or General Motors' Onstar-based emergency automobile assistance in the United States, automobile-related assistance in India is typically provided via direct, ad hoc negotiations with the nearest mechanic — and mobile phones are often the only communication device used by vendors.
- Feedback about the service quality of individual vendors from prior users of BusinessFinder is explicitly used in the query matching process to return "better ranked" vendors, whenever available. Such reputation-based ranking is particularly important due to the "bazaarlike" interaction in emerging economies, where vendors can vary widely in quality and pricing attributes.
- Vendors are able to manipulate their presence attributes via a variety of channels. In particular, we emphasize Short Messaging System (SMS) and Interactive Voice Response (IVR) channels), allowing them to use BusinessFinder without mandating a complete rollout of IMS-enabled [6] IP handsets (whose market share in emerging telecom markets may be low for a while).

The rest of the article is organized as follows. The next section illustrates the current and future presence-based substrate (using a centralized presence server) from which BusinessFinder extracts a variety of vendor and requester context, and how BusinessFinder uses a standard Parlay [7] based interface (e.g., the Sprint Business Mobility Framework [8] or the Lucent iLocator service at http://www.lucent.com) to currently retrieve provider network information. We then present the functional architecture of BusinessFinder and related implementation-specific details, as well as provide some rough estimates on the revenue potential of BusinessFinder. Looking further toward the future, where the presence information of potentially millions of users is distributed across multiple presence servers (for reasons of scale and performance), we describe the research innovations we have developed to support BusinessFinder's services in a distributed-presence environment. Finally, we present conclusions and a discussion of open challenges.

Use Case Scenarios and Context Sources for BusinessFinder

BusinessFinder uses multiple contextual attributes to determine a "nearby and available" vendor for a requested service. The following scenarios capture the typical usage pattern of BusinessFinder:

Alice, currently located at point X, issues an SMS for "plumber", indicating her desire to locate a nearby plumber. The telecom service provider uses localization technology to track both her location, as well as the location of cell phones of registered plumber. Her location is then matched with that of a vendor (say, Harish), who is not just near to Alice, but also available. Availability may be expressed via either profiles (e.g., Harish indicates that he's available from 7 am–7 pm on weekdays) or via dynamic messages (e.g., Harish sends an SMS indicating that he's free to take another job). Once Harish serves Alice's request, the service also collects feedback from Alice to rank Harish and uses this ranking in future matches.

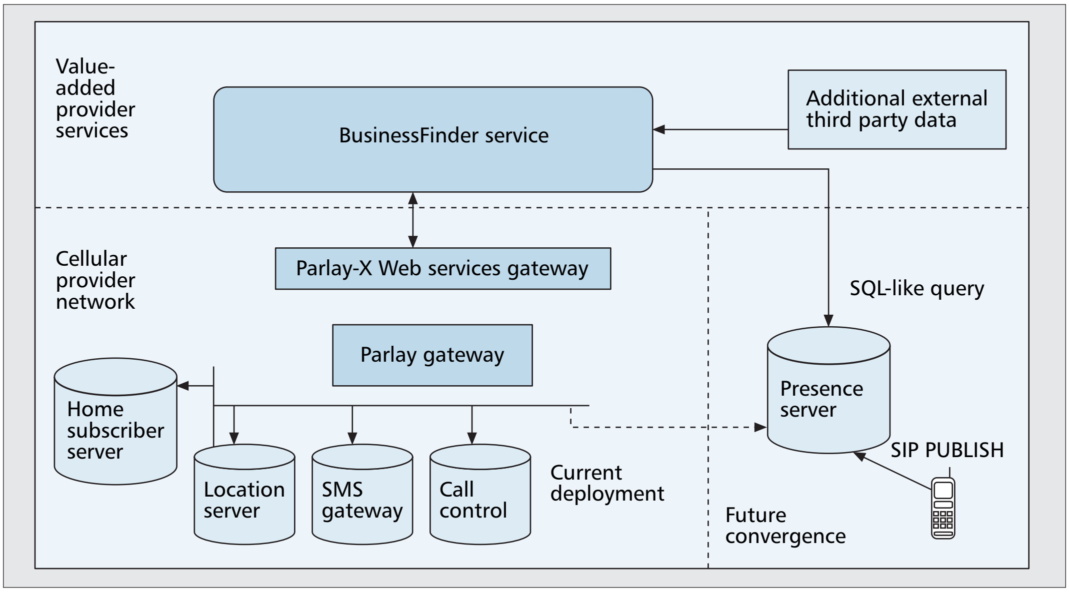
It is easy to envisage additional scenarios conforming to this "on-demand" service model. For example, a customer in an unfamiliar location with a car problem at night may use BusinessFinder to connect to a "locally available" auto mechanic, or an office executive desiring to pick up flowers en route to an evening engagement may connect to a "roaming florist" on his driving route. Interestingly, vendors themselves may act as customers requesting "matching" to other downstream vendors. As a concrete example, UNCTAD reports [9] how fishermen in the Indian state of Kerala currently call multiple ports to determine the best "current prices" for their nightly catch and dock their boat in one port or another. BusinessFinder can be used to automate this "matchmaking" process.

Context, Presence, and Their Integration with BusinessFinder

SIP-based presence is increasingly being viewed by providers and enterprises as a standard pub-sub mechanism for dynamic network-related events, with a Presence Server acting as an intermediate broker matching published data to subscriptions. The basic presence management substrate in SIP is subscription-based, where a subscriber (the interested party) issues a SUBSCRIBE message to the presence server specifying the events (from a designed URI of the form sip:user@domain) on which it requests notification. An individual *presentity* (the source of presence data) proactively publishes changes to its status via a PUBLISH message transmitted to the presence server; when the change in the presentity's state satisfies the predicate in the corresponding subscription, the presence server informs the interested party of the updated state

SIP-based presence is increasingly being viewed by providers and enterprises as a standard pub-sub mechanism for dynamic network-related events, with a Presence Server acting as an intermediate broker matching published data to subscriptions.

As initially implemented, BusinessFinder uses the Parlay 2.0 interface specifications to not just retrieve the basic presence data, such as a mobile user's location or a mobile device's presence status (off-hook/on-hook), but also to control the requester's interaction with the BusinessFinder service.



■ **Figure 1.** *BusinessFinder integration with network context/presence and external data sources.*

via a NOTIFY message. The presence information is published in an XML-based extensible format, and can thus include an arbitrary number of presence attributes (including non-network information).

As service provider networks become IMS-compliant, the requisite "context" information needed by BusinessFinder will typically be available from a centralized SIP-based presence server. However, since BusinessFinder is not based on long-live subscription semantics, the presence server's role is to act as an up-to-date repository of the presence state for its "registered users." In BusinessFinder, there is no use of the SIP SUBCRIBE or NOTIFY messages; a requester only issues a "snapshot query" (via a BusinessFinder-specific API) not for an explicitly specified presentity, but rather the SIP URI of a presentity that best satisfies the request predicates. BusinessFinder acts as a "data/query overlay" on the raw presence data, typically using SQL-like queries (with spatio-temporal predicates) over the raw presence data stored in the presence server database.

In the nearer term, in the absence of an IMS-based infrastructure, the context information required by BusinessFinder is typically distributed across multiple network elements. In fact, as the examples with mobile fishermen illustrate, the matching is also likely to need access to additional, decentralized, Web-based information systems (e.g., providing the current pricing details for a specific port). Accordingly, Fig. 1 illustrates the natural integration of BusinessFinder with a presence infrastructure, as well as with current legacy network-based and external information systems elements. As the figure shows, the network information for various users and devices is stored in different centralized servers, such as the Location Server, the SMS Gateway and the Home Subscriber Server (HSS) and exposed via an application gateway, using standardized interfaces such as Parlay or JAIN [10]. Under this currently used model, BusinessFinder is thus a value-added service that retrieves the required context information from the underlying network elements via the Parlay-based provider gateway interface. *According to the planned evolution of presence [1], the presence server will eventually become a central repository for all forms of "context," including location, phone status, the individual's availability, and so forth, and BusinessFinder can function by interfacing solely with the presence server.*

BusinessFinder: Functional Architecture, Implementation, and Business Case

As initially implemented, BusinessFinder uses the Parlay 2.0 interface specifications [7] not just to retrieve the basic presence data, such as a mobile user's location or a mobile device's presence status (off-hook/on-hook), but also to *control* the requester's interaction with the BusinessFinder service. The Parlay gateway mediates with the SMS Messaging Gateway (SMSC) to provide callbacks and notification when a requester sends an SMS to the "BusinessFinder service." BusinessFinder enables a vendor (individual or SMB) to exercise fine-tuned control over their presence information and edit their profile and/or schedule via multiple modalities, such as voice, SMS, or through a hosted application on the Web. We have also augmented BusinessFinder to retrieve context from a presence server — in this case, the location information and cellphone status of vendors and requesters are directly retrieved from a presence server using a SIP-based interface.

The Internal Architecture of BusinessFinder

Figure 2 shows the internal architectural components of BusinessFinder. The heart of the BusinessFinder service is the *Matchmaker* component that accepts a request for a service (e.g., Find me an electrician) and tries to match it to a "nearby,

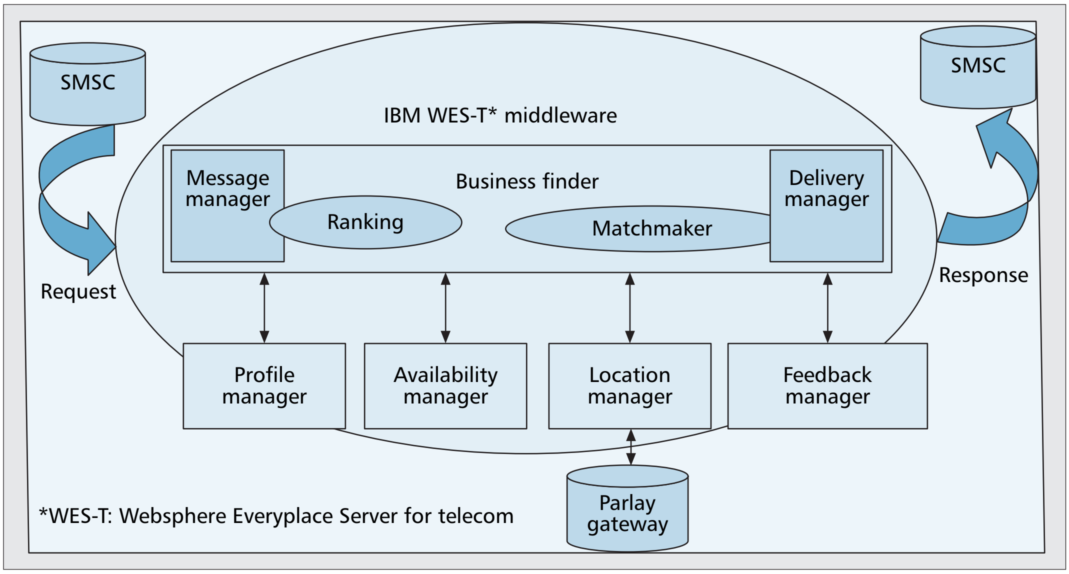


Figure 2. *Internal components of BusinessFinder.*

The MatchMaker component in BusinessFinder uses a plurality of attributes in the vendor matching process. In particular, our current implementation uses the attributes of availability, location (at multiple granularities), expertise, and vendor reputation.

well-ranked, available" vendor. The other major components of BusinessFinder are listed below:

Profile Manager: This component is responsible for storing the profiles of individual vendors, including attributes such as their names, phone numbers, availability schedule (e.g., Mon–Fri 9 am–6 pm), their skills and expertise, charges, and so on. This information is usually provided by the vendor when s/he registers with BusinessFinder (typically via the Web, but also via SMS based cellular channels).

Location Manager: This is responsible for interacting with the Parlay gateway (or presence server) to obtain and track (using both poll-based or change-triggered notification mechanisms) the location of individual registered vendors and requesters. Location granularity could vary across different implementations, ranging from geographical coordinates (if GPS was available) to cell-tower (or base station) IDs or triangulation-based techniques.

Availability Manager: This component is responsible for controlling and modulating the availability of an individual vendor. Availability Manager collects information from various sources (some relatively static, others dynamic) and infers the availability status of vendors. Besides prespecified availability schedules, vendors are allowed to explicitly send notifications stating their status (e.g., a vendor sends an SMS that s/he is available after finishing a job). BusinessFinder also tracks vendors that have been "matched" to requesters, making them automatically "unavailable' for a predetermined duration.

To support models where a vendor may be unable (e.g., while active on a plumbing chore) or not literate enough to use text-based interaction from a cell phone, BusinessFinder uses an Interactive Voice Response (IVR) component that allows individual vendors to navigate through personalized menus and alter their presence state. The IVR component translates such voice-based interaction (e.g., plumber Bob saying "I'll be busy till 5 pm") into the appropriate presence events and publishes them to either the Presence Server or the Availability Manager (for non-IMS implementations). A structured, low-complexity voice interaction is important to manage the challenges arising from the existence of marginally-literate vendor populations and wide linguistic diversity in countries like India.

Feedback Manager: It is responsible for collecting feedback from the requester about the service provided by a vendor, and is then used for implementing the appropriate ranking algorithms to determine an individual vendor's "reputation." As reputation management has been widely studied and used (e.g., the Internet auction site eBay), BusinessFinder can utilize one of several well-known algorithms for distributed reputation and trust management (e.g., see [11]).

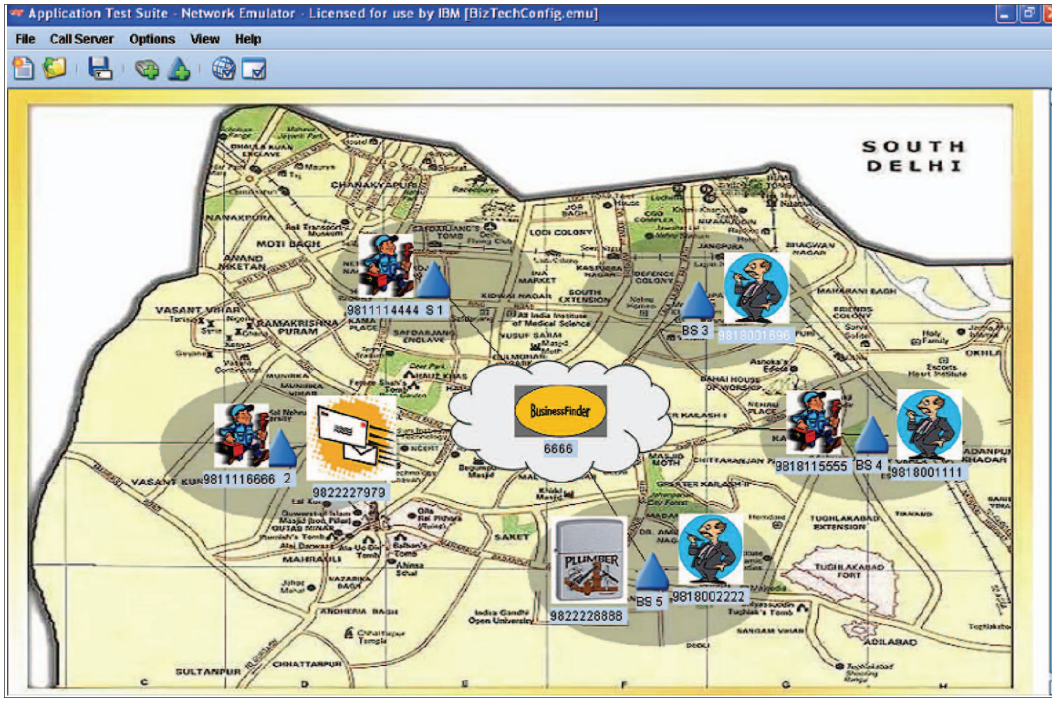
Message Manager and Delivery Managers: These components mediate the interaction of users with BusinessFinder via different channels-intercepting incoming traffic (e.g., voice, SMS) to BusinessFinder and delivering "matched" information to the requester.

The MatchMaker component in BusinessFinder uses a plurality of attributes in the vendor matching process — in particular, our current implementation uses the attributes of availability, location (at multiple granularities), expertise, and vendor reputation. In the matching process, BusinessFinder attempts to trade off between the potentially conflicting goals of returning a proximate vendor and one that is highly ranked by weighing each of these attributes — the actual assignment of weights is the prerogative of the service provider.

BusinessFinder Implementation Details

We have developed the BusinessFinder application, comprising of a set of vendors and a set of customers interacting with the core BusinessFinder logic through a Parlay-based telecom simulator (details of which are available at http://www.openapisolutions.com). This simulator provides the basic features of a real telecom infrastructure, such as SMS capability, initiation and reception of calls, determination of the cur-

For a live deployment, we envision the use of short codes that are commonly used in developing regions for SMS-based value added services. It is also possible to have a menu-driven interface on the phone or a richer client GUI (e.g., iMMS) to enable easier end-user interaction with BusinessFinder.



■ **Figure 3.** *Snapshot of the emulated environment used by the BusinessFinder prototype.*

rent location of a mobile phone, and so on. Our implementation uses the following Parlay APIs:

- SENDSMS: Send an SMS, specified by a String message to the specified address (or address set), specified by Addresses.
- GETRECEIVEDSMS: Retrieve all the SMS messages received according to specified criteria.
- GETLOCATIONFORGROUP: Initiate a retrieval activity, where one or more terminals, or groups of terminals, may have their locations determined.

Figure 3 shows a snapshot of the Business Finder scenario in the prototype system along with a subset of SMS message exchanges between a vendor and a requester. It depicts a region of South Delhi with a host of vendors (both micro businesses as well as SMBs) and a number of customers requesting for these vendors, using SMS to send and receive requests and updates. Interaction requests to BusinessFinder are expressed through appropriately formatted messages. For example, an SMS of "MATCH Plumber" to the number 6666 indicates a request to BusinessFinder for a nearby available plumber. For a *live* deployment, we envision the use of short codes that are commonly used in developing regions for SMS-based value added services. It is also possible to have a menu-driven interface on the phone (e.g., incorporating service catalogues in the SIM card menu), or a richer client GUI (e.g., iMMS) to enable easier end-user interaction with BusinessFinder.

BUSINESS REVENUE POTENTIAL OF BUSINESSFINDER

Next we present a simple business calculation, illustrating (very roughly) in Fig. 4 the additional revenue that a network provider deploying the BusinessFinder "dynamic yellow pages" service may hope to generate. The estimate assumes a mobile population of around 20 million in 2006 going up to around 40 million in 2010, with 1 percent of this population being the vendors subscribed to BusinessFinder. We estimate (conservatively) that only 2 percent of the mobile phone subscribers would initially use the BusinessFinder service, generating only two queries per year. As the popularity of such services grows, the number of people using this service will be around one-fifth of the total subscriber base, generating around 10 queries per year (the increase in service usage that we assume is similar to the trend forecast for the use of data services in [12]). We also assume, in this calculation, Rs. 6 (15 cents) being charged per query on the BusinessFinder service (this is the current price for a premium SMS service in India), a fee of Rs. 100 ($2.50) per year for listing a vendor on BusinessFinder and a revenue of Rs. 50 (~$1) from advertisements to users.

This revenue does not consider additional fees that may be generated through the exploitation of BusinessFinder's directory service in other third-party applications. It should be noted that BusinessFinder requires no additional capital investment in the network (the evolution to the presence-based infrastructure will take place independent of BusinessFinder), and only minimal software upgrades (since it can work with existing legacy network elements). Thus, preliminary analysis seems to suggest that BusinessFinder can provide an attractive return on investment (RoI) to a network provider. More importantly, an effective implementation can provide significant spillover benefits by increasing customer loyalty and reducing user churn.

BUSINESSFINDER IN DISTRIBUTED PRESENCE ENVIRONMENTS

The architectural specification of BusinessFinder in earlier sections assumed a centralized model, consisting of a single centralized Presence Server or a single Parlay gateway interfacing with a single centralized Location Server. However, for reasons of scale, as the number of nodes advertising presence (i.e., the population of *presentities*) increases, the presence information of different mobile users would need to be partitioned on a *distributed* set of presence servers. In such a distributed presence environment, the following three characteristics of BusinessFinder's "Find me the nearest available vendor" query raise additional challenges:

- The ephemeral nature of presence information
- Spatio-temporal changes in the distribution of vendors across presence servers
- Snapshot queries as opposed to subscription queries (i.e., "Find me a plumber now" rather than "Continually notify me of nearby plumbers"), which makes traditional publish-subscribe algorithms inapplicable

We now present our ongoing research on a resource-aware query routing architecture designed to handle BusinessFinder semantics on the "distributed presence" architecture of the future. We assume that an individual presence server manages the presence data for a specific partition of the overall geographic space. In this architecture, all vendors currently resident in a specific zone transmit their presence updates to the corresponding presence server. (We make this assumption since most BusinessFinder queries are location-sensitive; for conventional subscriptions to location-independent SIP URIs, a separate set of servers can perform static load sharing by a priori partitioning of the SIP URI namespace.) Moreover, the request for a particular type of vendor is also issued by a requester to the presence server corresponding to the requester's current location. If this presence server currently has a pool of "available" vendors (i.e., vendors currently in the same zone as the requester), the matching process is conceptually similar to that described above. However, in cases where the local pool is exhausted (due to 'random' spatio-temporal variations in both BusinessFinder requests and the available vendor pool) BusinessFinder's query needs to be potentially routed to alternative presence servers to retrieve available vendors from nearby areas. This problem is challenging, since the BusinessFinder query is only implicitly specified ("nearest available plumber"), and the URI that best matches the predicate of the query (i.e., nearest to the requester, with high ranking and available) is *time-varying* and must be *dynamically determined*.

We have developed a Resource-Aware Query Routing (RAQR) algorithm [13] to support such distributed BusinessFinder queries. Fundamentally, RAQR aims to strike a compromise between extremes of:

- Broadcast: In this approach, each BusinessFinder query is be forwarded to all the presence servers (since we do not know which set of servers actually have available vendors), and the resulting set of available vendors could be retrieved and then matched.
- Presence data flooding: Here, each presence update from each vendor is broadcast (flooded) to all the presence servers, so that each presence server is aware of the status of all vendors (global consistency), and is able to resolve the "nearest vendor" locally.

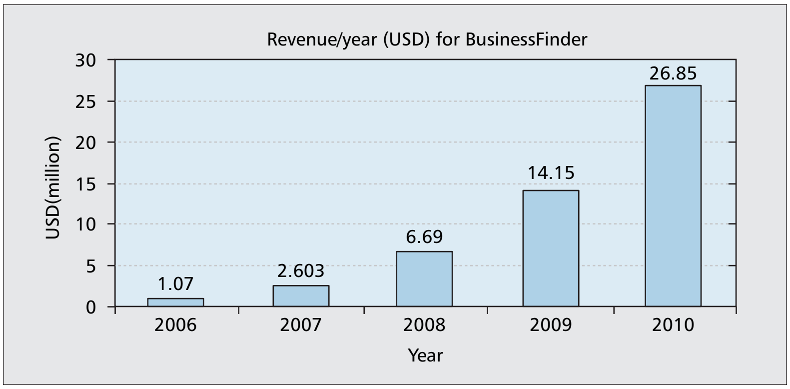


Figure 4. *Revenue projection from BusinessFinder.*

Both these approaches are, however, inefficient. Broadcast methods needlessly distribute each query to all servers, while presence data flooding ends up flooding the entire set of dynamic presence updates from a large population of users over the entire network.

RAQR solves the problem by essentially keeping a presence server aware of the identity of *nearby servers with an excess pool of available vendors*, so that, upon failure of local matching, a server is able to route its query only to this precomputed subset of servers that might potentially satisfy the BusinessFinder query. The main feature of RAQR is the *on-demand formation* of the *gradients*, pointing to servers having a surplus of resources. Gradients are created for a particular server only when it is projected to face a crunch in its local resources (relative to the request arrival load) and are removed whenever the crunch condition ceases to exist (either due to an increase in the arrival rate of new vendors or a reduction in the arrival load of requests). In RAQR, an originating server injects an INTEREST message informing its need for a certain resource type that is progressively diffused over the network of presence servers. A node receiving the INTEREST message responds directly to the originating server with a GRADIENT-OFFER message. The node also relays the INTEREST if it is unable to completely satisfy the original request rate-the parameters of this relayed message are adjusted to reflect the residual demand. On receiving the GRADIENT-OFFER responses, the originating server uses the quality-of-response (QoR) values of the servers to determine the best set of servers (those handling nearby regions and with good availability of high-quality vendors) and sets up direct gradients to them (using a RESERVE message). Of course, as the resource levels (of available vendors) fluctuate, nodes that had earlier offered their services may WITHDRAW their offer. Similarly, if the originating server feels that the

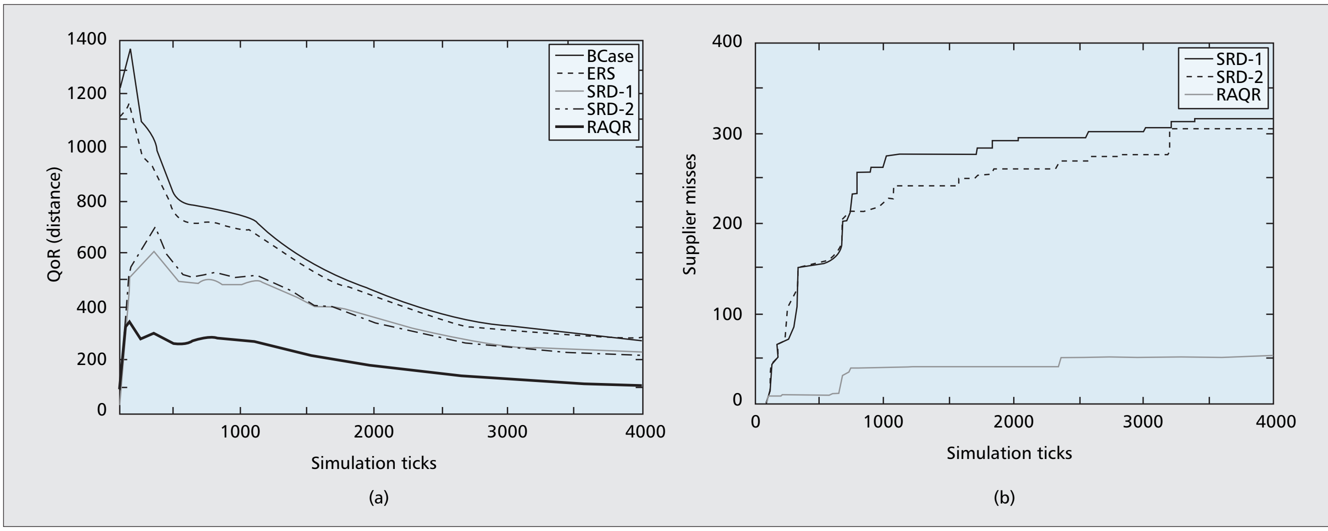


Figure 5. *Comparative performance of RAQR in terms of: a) average QoR; b) supplier misses.*

resource crunch is over, it may issue CANCEL messages to servers on which it had previously reserved resources. Note that the QoR metric is based on a weighted set of attributes, including vendor's reputation, the number of available vendors at the target server, and their distances from the requesting server. The assignment of weights to each such attribute is the prerogative of the service provider. For a detailed description of RAQR, the reader is requested to refer to [13].

Figure 5 illustrates the performance of RAQR against three alternative approaches listed in Table 1 for resolving BusinessFinder queries in a distributed presence environment. The studies were carried out with a set of 500 vendors moving randomly (based on the Random Waypoint model) over a (2500 ¥ 2500) grid, partitioned among 25 presence servers. While the vendor movement was random, the query generation (by mobile requesters) reflected both spatial and temporal skews. This was achieved by making the arrival rate of BusinessFinder requests to each server an independent normal distribution, with spatially varying means and temporally varying variances. We plot two different metrics:

- The *average QoR* of the responses, where the QoR of a successful query was measured solely based on the geographic distance between the requester and the matched vendor
- *Supplier miss*, which captures the number of instances where a BusinessFinder query is targeted toward a presence server that itself has no resources (vendors) to spare

Note that the simulations only consider the overhead of matching — that is, the process of determining a suitable vendor once a request has been received by BusinessFinder. In reality, when requests and responses between users and the BusinessFinder middleware are transported via SMS, the total turnaround latency would be another metric of interest (although it is most likely that the "matching latency" is a trivially small value compared to the SMS transmission delays). RAQR yields a QoR that is at least 50 percent better than other approaches (a lower QoR implies a better or "closer" match), while having a "supplier miss" that is 1/5th-1/6th that of the SRD-1 and SRD-2 algorithms. While not shown, the traffic overhead of RAQR (i.e., total signaling traffic needed to either maintain gradients, exchange presence status or forward BusinessFinder queries) is about 1/10th of the BCast approach and almost similar to SRD-approaches. The graphs demonstrate that effective support of BusinessFinder application semantics in telecom environments *will require not just the development of middleware components external to the core telecom infrastructure, but also enhancements to the core routing of presence information inside the network.*

Conclusions and Challenges

BusinessFinder is an instance of a presence-leveraging service that is specially targeted to the huge, nomadic, micro-business and SMB market segments in developing economies. Key innovations in BusinessFinder include its ability to track and use the location of both requesting users and vendors to match users to nearby vendors, its ability to use a variety of channels (such as IM, SMS, or voice) to capture the true availability of such nomadic vendors, and its use of community-feedback to eliminate poor-performing vendors from its directory. The architecture itself offers a bridge between low-cost basic data services and the higher-value, more sophisticated presence-based network platforms which will be introduced over the next three to five years. As the number of phones and individuals acting as presence "sources" grows, the presence information itself will be distributed. Our simulation-based studies demonstrate that our RAQR algorithm can perform significantly better BusinessFinder query "matching" on such a distributed set of presence servers at only 10 percent of the overhead encountered by more traditional broadcast-based algorithms.

While we have focused principally on the presence management-related aspects of BusinessFinder, a successful offering must address other challenges, including development of local-language support (both text-based and IVR) for

Name of algorithm	Behavior of algorithm
Broadcast	Query flooding, effectively transmitting each query to all servers.
Expanding Ring Search (ERS)	Searches for a nearby server with available vendors in incremental fashion (first one-hop neighbors, then two-hop, etc.).
Selective Resource Dissemination *n*-Hop (SRD-*n*)	Each server periodically disseminates information of its available vendor pool to all *n*-Hop neighbors (n = 1 implies presence data flooding).

Table 1. *Alternative approaches for resolving BusinessFinder queries.*

mobile devices, especially to ensure adoption by a less-literate workforce, adequate control of vendor behavior to avoid unwarranted spamming (e.g., a "matched vendor" using the requester's phone number for future unsolicited advertisements), or "denial of service" scenarios (e.g., a vendor making requests to competing vendors, thus flagging them as "unavailable"), and a more rigorous analysis of whether user requests for specific services can be segmented by their sensitivity to metrics such as distance, reputation, and pricing, and how such segmentation can be reflected in appropriate QoR formulations. We will continue to research these issues and refine the BusinessFinder application logic to accommodate specific concerns articulated by candidate service providers.

References

[1] J. Fontana, "Presence Applications Poised for Takeoff," Network World, Sept. 2004; http://www.networkworld.com/news/2004/090604specialfocus.html

[2] 3GPP Alliance, "Push-to-Talk Over Cellular (PoC) Services (Stage 2)," TR 23.979 v6.2.0; http://www.3gpp.org

[3] J. Rosenberg et al., "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002; http://rfc.sunsite.dk/rfc/rfc3261.html

[9] J. Roche, "Mobile Trading Comes of Age," UN Conference on Trade and Development, Sept. 2002.

[5] AMI Partners, http://www.ami-usa.com/ami/sections/Archives.aspx?sectionId=1, AMI Press Release, Feb. 2006.

[11] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," *Proc. World Wide Web Conf.*, May 2003.

[7] "The PARLAY Group: Bridging Telecom and IT," http://www.parlay.org

[8] J. Roche, "Mobile Trading Comes of Age," *UN Conf. Trade and Development*, Sept. 2002.

[6] G. Camarillo and M.-A. García-Martín, The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, New York: Wiley.

[8] Sprint Business Mobility Framework: http://www.parlay.org/en/docs/may2004mm/Opening_Plenary/05Sprint_IBM.pdf

[4] M. Lonnfors, E. Leppanen, and H. Khartabil, "SIP Presence Information Data Format," IETF SIMPLE Working Group, Internet draft, http://xml.coverpages.org/draft-lonnfors-simple-binpidf-00.txt

[13] D. Chakraborty, K. Dasgupta, and A. Misra, "Efficient Querying and Resource Management Using Distributed Presence Information in Converged Networks," *Proc. 7th Int'l. Conf. Mobile Data Mgmt.*, Nara, Japan, May 2006.

[12] "Mobile Application Platforms and Operating Systems," Informa Telecoms and Media Report, 2nd ed., London, U.K., 2005.

[10] J2SLEE and the JAIN Initiative, http://java.sun.com/products/jain/

Additional Reading

[1] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, 1996.

Biographies

DIPANJAN CHAKRABORTY (cdipanjan@in.ibm.com) is a research staff member at IBM India Research Laboratory. He received a Ph.D. degree in computer science from the University of Maryland, Baltimore County in 2004. His research is in the areas of mobile and pervasive computing environments, next-generation network protocols, and peer-to-peer systems with special interests in the fields of service discovery, information aggregation and composition, and ad hoc/sensor networks. He is also working in the area of business process management. His thesis was in the area of service discovery and composition for pervasive environments.

KOUSTUV DASGUPTA (kdasgupta@in.ibm.com) is a researcher in the Telecommunications Research and Innovation Center at IBM India Research Laboratory. His current projects include presence-based infrastructure for converged networks and social networks analytics for cellular service providers. He received a Ph.D. degree in computer science from the University of Maryland, Baltimore County in May 2003. His research interests and contributions focus on a wide range of large-scale networked computer systems including Internet systems, next-generation converged networks, sensor networks, storage systems, Web services, and enterprise grids.

ARCHAN MISRA is a research staff member at the IBM T. J. Watson Research Center, Hawthorne, NY, where he works on pervasive and presence-based network infrastructures. His current research projects include middleware for SIP-based "rich presence" in converged networks, wireless mesh networking, and remote healthcare monitoring. He received a Ph.D. degree in electrical and computer engineering from the University of Maryland at College Park in May 2000. He is an editor of *IEEE Wireless Communications* and currently chairs the IEEE Computer Society's Technical Committee on Computer Communications.

SUMIT MITTAL has been a technical staff member with IBM India Research Laboratory, New Delhi since August 2004. He received a Master of Science degree in computer science from Rice University, Houston, Texas, in 2005 and a B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Kharagpur in 2001. His current research interests are in the fields of distributed and pervasive computing, and various aspects of Web service composition and execution.

ANUJ GUPTA holds an M.Sc. (computer science) from Allahabad University and is currently working as manager, Rational Products in the India Software Laboratory, IBM. He has worked in various roles, including senior designer, project manager, and information architect. He has defined IT strategies for large organizations and managed multiple software development projects including product development from start to finish: concept, definition, design, development, and quality assurance of *N*-tier applications.

EILEEN NEWMARK is an expert in user-centered design and the development of digital media and network-based products for enterprise and consumer markets. Her 15 years of experience in the industry span business and product development from definition and analysis of market opportunities through prototype design, full product development, and marketing.

CHRIS OBERLE is a software engineer with more than 10 years of experience designing and developing applications. As a computer science graduate from North Carolina State University, he developed a passion for object-oriented design and the Java programming language. Recently he has been focusing his attention on search engine optimization techniques and is working on a new book in a related field.