Bluetooth Location Networks

F. J. González-Castaño, J. García-Reinoso
Departamento de Ingeniería Telemática, Universidad de Vigo,
ETSI Telecomunicación, Campus, 36200 Vigo, Spain,
{javier,reinoso}@det.uvigo.es

Abstract— **In this paper, we propose a Bluetooth Location Network (BLN) for location-aware or context-driven mobile networks. Some examples of context-driven network users are walking individuals that shop in a mall (m-commerce), people visiting a museum (e-museums) or professionals that interact with exhibition stands. We assume that, in any of those scenarios, there exist *service servers* that need to know user location in real-time, to send context-oriented information to user handhelds when necessary. The BLN transmits position information to service servers, without user participation, and may also be used to transmit context-oriented information to user terminals. The BLN is not subject to line-of-sight constraints and its base technology is available for existing commercial handhelds, as an option. BLN users carry either a Bluetooth-enabled handheld or a Bluetooth *badge* (in the latter case, the BLN provides location services to any kind of user terminal). The BLN is composed by small, completely independent Bluetooth nodes (no wires), which establish an spontaneous network topology at system initialization. The BLN can coexist with Bluetooth devices that are not part of the location system, such as printers or headphones. We evaluate BLN performance with IBM's BlueHoc simulator.**

Keywords—**Location-awareness, context-driven services, user positioning, Bluetooth**

I. Introduction

A. Motivation

In this paper, we propose a Bluetooth Location Network (BLN) for location-aware or context-driven mobile networks.

For instance, m-commerce (mobile e-commerce, [1]) for cell phones or PDAs [2], [3] has a promising future. Datamonitor [4] has predicted that the US m-commerce market will grow 1,000 percent up to 1.2 billion US$ by 2005. In a typical m-commerce scenario, customers walk around a large commercial area or mall carrying wireless PDAs. A PDA client allows its user not only to purchase items, make reservations or request information, but also to receive (possibly context-driven) store coupons, advertisements, advice and guidance.

Another interesting application field is electronic guidance. Exhibition visitors receive specific information associated to their current location. See [5] for a review of current initiatives.

In any of those scenarios, there may exist information servers that need to know user location in real-time, and send context-oriented information to user handhelds when necessary.

The BLN transmits position information to *service servers*, without user participation, and may also be used to transmit context-oriented information to user terminals. The BLN is not subject to line-of-sight constraints and its base technology is available for existing commercial handhelds, as an option. As a fully operational data network, the BLN admits alternative uses like as a security network when the target area is closed to the public, or as a spare network for emergencies.

BLN users carry either a Bluetooth-enabled handheld [6], [7] or a Bluetooth badge (in the latter case, the BLN provides location services to any kind of user terminal). The BLN is composed by small, completely independent Bluetooth nodes (no wires), which establish an spontaneous network topology at system initialization. The BLN can coexist with Bluetooth devices that are not part of the location system, such as printers or headphones. We analyze BLN performance with IBM's BlueHoc simulator.

B. Background

• *Outdoor systems*: Cell phone location services [8] and GPS [9]. These systems are only valid outdoors (specially GPS). In that case, they are quite effective, and possibly the best choice.

For indoor applications, there exist several alternatives:

• *RF triangulation-oriented systems*: Pinpoint's 3D-iD system [10] is based on *cell controllers*, which interact with RF *tags*. The controller is linked to several antennas, that receive tag identifiers. User position can be determined by processing tag signals received by different antennas. From our point of view, it has two disadvantages: It needs coaxial cables to link controllers with antennas, and cannot be used as a data network.

• *Beacons*: HP's CoolTown [11] is based on IR beacons, which push position-dependent URLs into handheld IR ports. The system is user-dependent, since he must aim the infrared port to location beacons. It could be argued that this is not a drawback, since automatic detection of location information (without user participation) may have severe consequences in terms of nuisance value. A good example happens when users are annoyed because the Web page they are viewing is suddenly supplanted by one advertising frozen peas from a grocery store nearby. Obviously, we expect the system to be *reasonable*. Consider, for example, a museum, where updates are associated to new halls, once the user enters them, and imagine that the update is the previous page with a tiny flashing icon at its bottom meaning "do you want to update context information"? On the other hand, asking the user to locate IR beacons each time he enters a room full of visual distractions may be tiring, and signaling beacons with large red arrows unsightly. Possibly, depending on the specific application, user-dependent line-of-sight IR systems may be more advantageous than user-independent ones or viceversa, and both approaches deserve consideration. Other examples of IR systems are [12], [13].

• *Client processing*: In these systems, clients calculate their position using beacon signals as references. For example, MIT Cricket [14] uses a combination of RF and ultrasonic inputs. The authors claim that, by freeing service servers from location processing, their system is more scalable. On the other hand, in an user-independent Cricket location system, user handhelds should interact with Cricket non-standard hardware to extract position information, which should be sent to the service servers via some data network. Such interaction is not considered, and may increase handheld complexity.

• *Pinging*: In these systems, beacons trigger terminal pulse transmission. In AT&T Bat [15], beacons are linked to DSP-controlled sensor arrays that receive ultrasonic pulses from user terminals, determine their position, and transmit it to service servers via a specialized data network. This approach is user-independent and the data network can be shared with other applications. However, Bat terminals are non-standard, and sensor arrays are considerably complex.

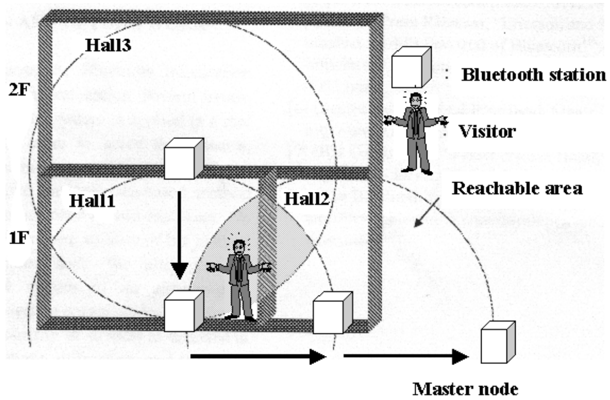

Fig. 1. Cooperative Bluetooth location

Note: It should be understood that the philosophy in section I-A is not necessarily the best one for *any* application. Each system, including ours, has clear advantages over the remaining ones, based on its *own* assumptions. For example, Bat sensor arrays have a high location precision. Cricket is extremely simple, and works well in case of user-dependent location services. A single 3D-iD cell controller may handle a medium-sized store. Finally, Cooltown may use IR ports present in most handhelds.

C. Bluetooth location networks

In this paper, we define a Bluetooth Location Network (BLN) that fulfils the requirements in section I-A. We assume that users carry a Bluetooth-enabled terminal, or any mobile terminal and a Bluetooth location badge. Users must access the Web/WAP service server from their handhelds, and enter their badge address. By doing so, the Bluetooth address of the badge becomes *valid* from the BLN's point of view (obviously, the location network must work even if invalid addresses are present, as we will see later). The service servers associate the user's IP address or WAP session to his badge number, for all subsequent transactions. The badge (or the Bluetooth modem in the user's terminal) interacts with the BLN, which provides service servers with real-time user position. The service servers may use this information to push URLs into user terminals via TCP/IP sockets, or to update WAP cards. Thus, no client action is required to generate context-driven updates.

Bluetooth was also selected as the base technology for the information offering system in [16]. Although the authors claimed some of the advantages we enumerated in section I-A, they also stated that Bluetooth range does not provide enough location precision. Consider the example in figure 1. In principle, if the three Bluetooth stations detect the user modem, the user could be located in any hall if considering the full coverage (even outside halls 1, 2 and 3). The key point in our philosophy is establishing a *cooperative location network*. The network does not only transmit exhibition information to user terminals, but also the addresses of the Bluetooth stations that detect them to a *master node*. In the example in figure 1, the master node will determine that the user is located inside the grey region. Note that most of that region intersects the hall were the user is actually located. This is interesting, because this particular arrangement could not be solved by the non-cooperative system in [16]. The cooperative BLN in this paper is intended to cover 2D target areas, although it can be generalized to cover 3D ones, as we will comment later.

The rest of this paper is organized as follows: the next section describes BLN protocols. Section III analyzes BLN performance. Finally, section IV concludes.

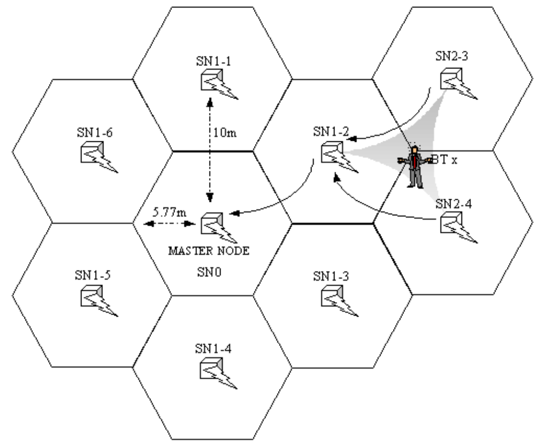

Fig. 2. Bluetooth Location Network

II. BLN protocols

A. BLN configuration

The BLN is composed by mobile badges and static Bluetooth units (located at the ceiling, for example). We will refer to the latter as *static nodes*. Static nodes (SNs) are arranged in a network that covers the whole target area. Hexagonal tiling is a typical solution in 2D cellular network planning, which we have followed in this research (figure 2). Other arrangements where any SN has at most seven closest neighbors located less than 10 m away could be used as well (for class 2 Bluetooth modems) [18]. For example, meshes or k-ary 3-cubes [20] for 2D or 3D target areas, respectively.

Each cell in figure 2 has an area of 86.55 m^2. SN units scan their surroundings periodically, by means of Bluetooth inquiry calls [17]. All SNs are organized in a radial scatternet around a master node, SN0, connected to the service servers (not shown). The remaining SNs are arranged in "circular" layers around SN0. The notation SNX-Y stands for the Bluetooth address of SN Y in layer X. In layer X, SNX-1 is placed right above SN0, and the remaining Y values increase clockwise. Our example shows the six cells in the first layer, SN1-1 to SN1-6, and two cells in the second layer, SN2-3 and SN2-4. Each SN is a slave of all six surrounding neighbor SNs.

All SNs perform inquiry cycles periodically, to publish their existence. If SN a detects an inquiry from SN b, and b is not currently listed in a's routing table, a must send its *minimum distance to the master node* in number of hops to b (a *distance packet*: 186-bit DM1 packet [17] carrying a 1-byte *distance field*). All SN minimum distances are set to ∞ at power up, excepting the master node's, which is set to 0. Thus, the master node initiates the configuration by sending 0-hop distance packets to its neighbors. Later, if a SN performing an inquiry cycle does not receive an answer from one of its neighbors that was previously listed in the routing table, it deletes the corresponding entry. If this changes its minimum distance to the master node, the SN transmits a new minimum distance packet to all its slaves.

Whenever a SN receives a distance packet, it searches its routing table to check if the corresponding distance is lower than its current lowest distance to the master node. If so, the SN builds a new distance packet and transmits it to all its slave SNs, excepting those included in minimum-distance routes to the master node. This algorithm is similar to the *split horizon* algorithm [21].

Therefore, the configuration process also restarts in case of SN failures, and propagates changes from the failure neighborhood (possibly only affecting a BLN region).

If a SN receives a distance packet, it must update its routing table. The routing table stores pairs of neighbor SN addresses and their distances to the master node, and is

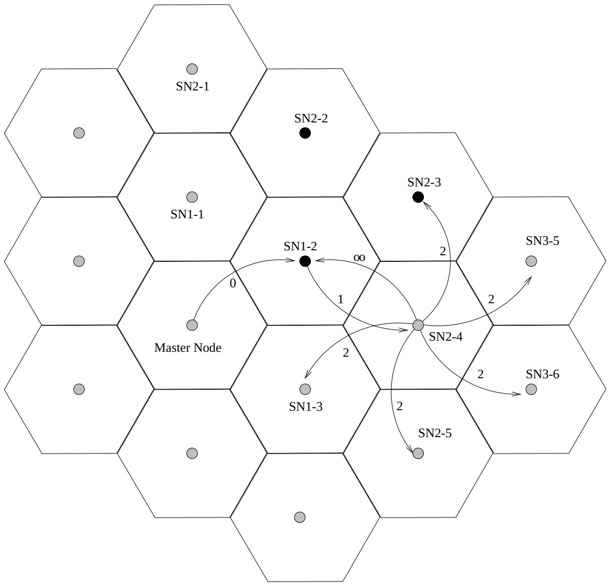

Fig. 3. BLN configuration

Next hop	Prev. distance	New distance
SN1-2	∞	2
SN1-3	∞	∞
SN2-5	∞	∞
SN3-5	∞	∞
SN3-6	∞	∞

TABLE I

Routing table of SN2-4

sorted by distance. Thus, the best path to the master node is always on top of the table.

Figure 3 depicts a BLN region. We describe two configuration steps to illustrate the general procedure.

1. Once the master node initiates the process, the distance from SN1-2 to the master node changes. Node SN1-2 transmits the distance change to all its slaves.
2. SN2-4 receives the distance packet from SN1-2. Since this distance plus 1 is lower than the current minimum distance, SN2-4 increments the received distance by 1, stores it in its routing table, and sorts the table again. Table I shows the evolution of the routing table of node SN2-4.

As we said previously, if a SN detects that the minimum distance to the master node has changed, it builds a new distance packet with this minimum distance, and transmits it to its slaves (neighbors) except to those who are in the minimum distance path (in our example, SN1-2). Those SNs receive a *infinite* distance packet, to prevent loops.

Remark 1: If a SN has less than seven neighbors less than 10 m away, it is possible to implement permanent links with them (the seven-slave transmission constraint holds). This is valid for hexagonal-tiling, mesh and k-ary 3-cube BLNs.

Remark 2: A simple authentication handshake avoids connection establishment with invalid Bluetooth modems, which are considered invalid badges for simplicity. Typically, invalid badges will answer inquiry cycles with FHS packets, which is relatively harmless (see section II-B below). However, in case they answered with another kind of packet, they would be easily detected by the authentication handshake and rejected.

Remark 3: Badges do not try to establish data connections with SNs. They simply answer inquiries with FHS packets, which does not violate the seven-slave constraint.

	Before inq.		After inq.	
Bluetooth addr.	Detected	New	Detected	New
BD-3	NO	NO	YES	NO
BD-7	NO	NO	NO	NO
BD-11	NO	NO	NO	NO
BD-13	NO	NO	YES	NO
BD-17	NO	NO	YES	NO
BD-19			YES	YES

TABLE II

SN cache evolution

B. BLN location protocol

The main goal of the BLN is to track user movements in the target area. To meet that goal, all SNs have to send inquiries and collect badge responses. Every SN has a cache where it stores badge addresses. When it detects a response from a badge whose address was not in the cache, it builds a 366-bit DM1 *location packet* that carries its own address and the badge's address (64+64 bits), and transmits it to the SN on top of its routing table.

For example, table II (second and third columns) shows the current SN cache state (BD-X identifies the badge with address X) when the SN is performing an inquiry cycle. Before the cycle starts, the "*detected*" and "*new*" columns are unmarked (set to NO).

When a badge detects an inquiry, it answers with a FHS packet. The SN extracts the badge address from the FHS packet and checks it in the cache. If the address is already listed, the corresponding *detected* column is marked (set to YES). Otherwise, a new row with the address is added and both the *detected* and *new* columns are marked with YES. When the inquiry cycle ends, (i) all marked (YES) *new* columns are switched to unmarked (NO) state, and location packets for the corresponding entries are transmitted to the master node to report that new badges have entered SN range. (ii) All entries with unmarked (NO) *detected* column are deleted, and a location packet for each one of them is transmitted to the master node to report that the corresponding badges are now out of SN range.

Location packets carry two Bluetooth addresses: SN address and badge address. The packets have a bit to report if the badge arrives to or leaves the cell.

It should be understood that the SN that detects a badge is in charge of building location packets. All SNs placed along the transmission path to the master node simply relay them to the SN on top of their routing tables.

Table II, in its fourth and fifth columns, represents a possible SN cache state after the inquiry cycle. A new badge, BD-19, was detected. Consequently, a location packet with BD-19 payload will be sent to the master node, and its *new* column will be changed to NO. Two badges, BD-7 and BD-11, have left the cell, and the corresponding location packets will be sent to the master node with the *detected* bit set to 0. Their entries will be removed. Badges BD-3, BD-13 and BD-17 are still around, but do not trigger location packet transmission.

Remark 4: SN responses to SN inquiry cycles are ignored by the location protocol, because the corresponding SNs are either listed in the routing table of the requesting SN or will be listed after establishing a master-slave link (this may happen, for example, when a dead SN is replaced).

Remark 5: Obviously, invalid badges will answer to SN inquiries, and will generate location packets. However, those

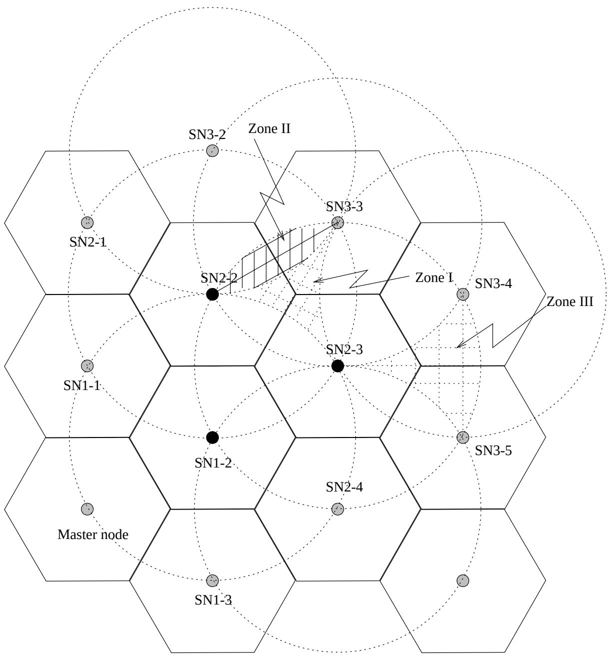

Fig. 4. Location zones

packets will be filtered by the master node. As we will see later, even if a large target area is crowded, the BLN can carry all location packets, valid or invalid, as far as the number of users and the number of badges are roughly similar. Moreover, note that if invalid badges correspond to static devices (such as printers), they will generate only *one* location packet, because their *new* column in SN location caches will be unmarked afterwards.

C. Location zones

The master node (or a service server attached to it) estimates that badge x is placed in a *location zone* that depends on the SNs that send location packets containing address x. Room-scale precision may be enough for many context-driven services in the scenarios in section I, while keeping SN complexity reasonably low. So far, we do not take signal strength nor signal delay into consideration.

Location precision depends on the number of SNs that detect a given badge. We define different location zone *classes*, for a given network topology. A badge is said to be located in a class I zone of a hexagonal tiling topology if *at most* three SNs detect the badge. For example, the class I zone depicted in figure 4 is defined by SN2-2, SN2-3 and SN3-3 detection. Note that neither SN1-2, SN3-2 nor SN3-4 detect the badges in that zone. A class I zone has an area of 16.12 m^2. A badge is said to be located in a class II zone if at most four SNs detect it. For example, SN2-2, SN2-3, SN3-2 and SN3-3 define the class II zone in figure 4. Class II zones have an area of 18.12 m^2.

Class III zones are special cases in the neighborhood of SN failures (or at the edges of the target area). The class III zone in figure 4 has an area of 34.24 m^2. In absence of SN failures, those zones can be avoided by placing extra SNs at the walls of the target area.

Finally, a class IV zone is a single SN. When a badge is placed exactly at a SN position, it is detected by seven SNs (the closest SN and all its slaves).

D. Scalability

As the BLN grows, the number of location packet hops may become too large, if originated at BLN edges. It is possible to avoid this issue by installing additional master nodes. Due to BLN symmetry, when the configuration protocol converges to a equilibrium point, the maximum distance to each of the master nodes is similar, if equally spaced. Figure 5 shows an example for four master nodes.

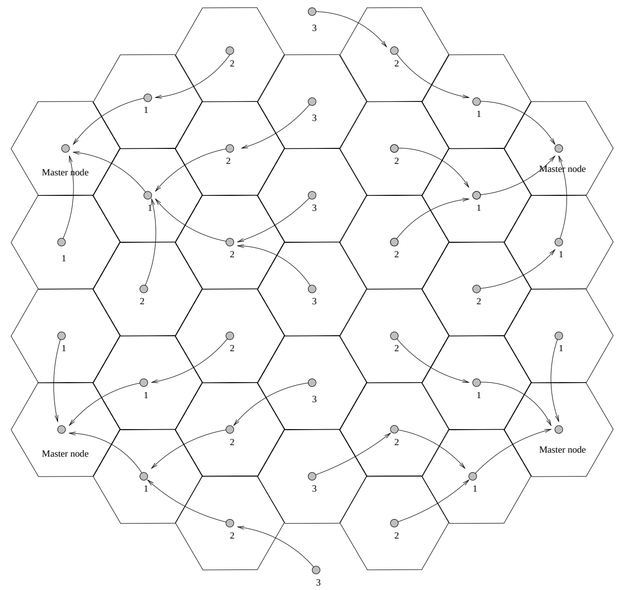

Fig. 5. Location packet routes (sample), four master nodes

MN	$\overline{SN}_{MN}$	$\overline{d}_{MN}$
1	36	2.3333
2	17.5	2.54 + (-0.06,0.09)
4	8.25	1.7134 + (-0.21,0.38)

TABLE III

BLN SYMMETRY EVALUATION

Note that the maximum distance route to any master node has three hops. However, even for the same network, location routes could change at the next initialization, since all SNs evolve asynchronously. In order to evaluate BLN symmetry, we performed simulations of several three-layer ($\sim$ 4,000 m^2) BLN initializations. The results are summarized in table III (MN is the number of master nodes, $\overline{SN}_{MN}$ is the average number of SNs whose minimum distance routes lead to the same master node, and $\overline{d}_{MN}$ is the average maximum distance per master node).

Note that, spontaneously, the area managed by each master node is inversely proportional to the number of master nodes. For a given network, the maximum distance per master node is quite stable across different initializations.

III. Performance evaluation

In this section, we analyze BLN protocol timings. The inquiry cycle scans all frequencies, and it is recommended that it should last for at most 10.24 sec ([17], pag. 110).

We used the IBM BlueHoc simulator [22] to estimate the worst individual badge response time to SN inquiries, as the number of badges grow. Figure 6 shows the results. Ideally, all badges nearby must be detected during the inquiry cycles of the surrounding SNs. Therefore, the number of badges per SN should not exceed 50 (we see that some badges are not detected when we approach that limit). Since SN range is a circle of radius 10 m, there are only $\sim$ 6 m^2 per user, a crowded scenario. The average worst response time for 50 users is T_W =8 sec, with a global average of $\sim$ 6 sec, which is coherent with the results in [23]. Of course, we could detect even more badges if we allowed more than one inquiry cycle for badge detection, but it does not seem necessary, since we are already supporting a crowded area.

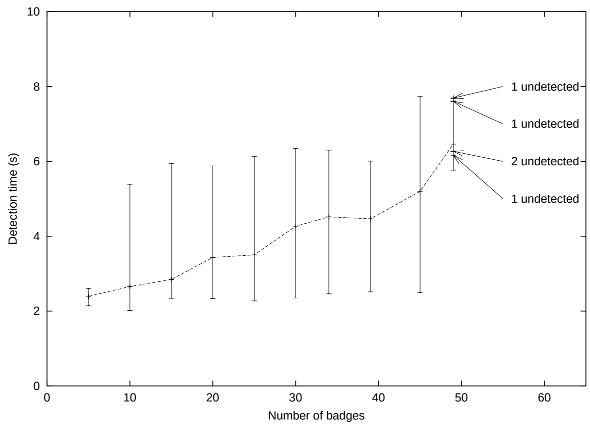

Fig. 6. Inquiry response time vs. number of badges

Next, and also using the BlueHoc simulator, we calculated location packet hop transfer timings. Instead of DM1 packets, we used 210-byte DM5 packets [17], which is the default for BlueHoc. Therefore, in a real implementation, hop transfer times would be much lower. Since SNs are in charge of location packet transmission, there may be up to six simultaneous location packet transmissions to the same SN destination (master node case). The resulting average transfer time was 4.32 msec, with a standard deviation of 1.34 msec, a maximum value of 9.87 msec and a minimum value of 1.02 msec.

If we consider the application scenarios in section I, we see that typical users are walking individuals that make frequent stops. If we set a goal of 30 sec for the service servers to update the position of each user, we can take the following facts into account:

- **The 30-sec window has an inquiry cycle interspersed with location packet processing, for which $30 - T_W = 22$ sec are available in the worst case.**
- **A SN collects location packets from up to six $\sim$ 300 m^2 neighbor cells. This means $\frac{6\ SNs \times 300 m^2}{6 m^2\ p.\ usr} \sim$ 300 user location packet transmissions, in the crowded scenario[1]. Again, considering $\sim$ 10 msec per transmission, all location transmissions in the target area require 3 sec multiplied by the number of hops in the largest route, in the crowded scenario. Thus, 22 sec allows something more than seven hops, which is the largest route in a circular target area of $\sim$ 18,000 m^2, for *a single master node*, and $\frac{18{,}000 m^2}{86.55 m^2\ p.\ SN\ cell} \sim 210$ SNs.**
- **We are not considering SN protocol tasks and alternative BLN data applications, which are certainly not negligible. However, the use of DM1 instead of DM5 packets, as explained in section II-B, would free most of the transmission time in the analysis above. Also, it could be possible to add more master nodes. As we saw in section II-D, master nodes would share the target area spontaneously.**
- **With additional master nodes, the BLN may cover larger target areas, for the same performance (or may improve performance, for the same target area size).**

IV. Conclusions

In this paper, we have proposed a Bluetooth location network for context-driven services, BLN. The location network has the following characteristics:

- **The BLN transmits position information to the service servers without user participation.**
- **The BLN is based on a general-purpose RF technology, available as an option in existing handhelds.**
- **The BLN can be used as a general-purpose data network.**
- **The BLN infrastructure consists of small, completely independent Bluetooth nodes (no wires).**
- **Spontaneous topology configuration allows scalability, by placing as many master nodes as necessary.**
- **The BLN can coexist with Bluetooth devices that are not part of the location system, such as printers or headphones.**
- **A single master node can cover a crowded area of 18,000 m^2 with 3,000 users, for a worst-case individual location update time of 30 sec.**

Note that the spontaneous BLN configuration allows SNs to find alternative routes to the master node in case of SN failures. Forthcoming work will study BLN survivability, as well as alternative BLN topologies: mesh and k-ary 3-cube BLNs.

[1] This is a rather pessimistic estimation, because motionless badges only generate one location packet. Thus, even in the crowded scenario, the number of location packets will be much lower. Also, six zones per SN is an upper bound (master node case). In a hexagonal tiling arrangement, a SN handles four zones or less typically.

References

[1] M-commerce world. WWW. http://www.m-commerceworld.com/.

[2] U. Varshney, R. J. Vetter, and R. Kalakota. Mobile commerce: A new frontier. *Computer*, 10:32–38, 2000.

[3] A. Darling. Waiting for the m-commerce explosion. *Telecommunication International*, 3:34–39, 2001.

[4] Datamonitor. M-commerce infraestructure in the US. WWW. http://www.datamonitor.com.

[5] Electronic Guidebook Project. WWW. http://www.exploratorium.edu/guidebook.

[6] Bluetooth iPAQ. WWW. http://www.compaq.com/products/handhelds/pocketpc/options/expansion_packs.html.

[7] Bluetooth Nokia. WWW. http://www.nokia.com/phones/6210/bluetooth.html.

[8] ETSI document TS 122 071 V3.2.0.

[9] A. Dorman. Can m-commerce find a place in your network? WWW. Network Magazine. http://www.networkmagazine.com/article/NMG20011101S0005/2.

[10] J. Werb and C. Lanzl. A positioning system for finding things indoors. *IEEE Spectrum*, 35(9):71–78, 1998.

[11] T. Kindberg and J. Barton. A web-based nomadic computing system. *Computer Networks*, 35(4):443–456, 2001.

[12] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: a mobile context-aware tour guide. *Wireless Netwoks*, 3(5):421–433, 1997.

[13] R. Want, W. N. Schilit, N. I. Adams, R. Gold, K. Peterson, D. Goldberg, J. R. Ellis, and M. Weiser. *The ParcTab Ubiquitous Computing Experience*. In *Mobile Computing*. Kluwer Academic Publishers, 1996.

[14] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proc. of the Sixth Annual ACM Internation Conference on Mobile Computing and Networking*, 2000.

[15] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proc. of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 59–68, 1999.

[16] T. Yamasaki, M. Kishimoto, N. Komoda and H. Oiso. An information offering system for exhibition explanation by Bluetooth Technology. In *SSGRR 2001, Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Italy, Aug. 2001.

[17] Bluetooth SIG. Specification of the Bluetooth system-core v1.0b. Technical report, December 1999.

[18] 10meters.com. WWW. http://www.10meters.com/blue_802.html.

[19] J. Lansford and P. Bahl. The design and implementation of HomeRF: A radio frequency wireless networking standard for the connected home. *Proceedings of the IEEE*, 88:1662–1676, 2000.

[20] W. Mao and D. M. Nicol. On k-ary n-cubes: theory and applications. NASA CR-194996 ICASE report # 94-88, 1994.

[21] A. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.

[22] BlueHoc. WWW. http://www-124.ibm.com/developerworks/opensource/bluehoc/.

[23] C. Law, A. K. Mehta, and K. Y. Siu. Performance of a new Bluetooth scatternet formation protocol. In *Proc. of the ACM Symposium on Mobile Ad Hoc Networking and Computing*, 2001.