BLESSED with Opportunistic Beacons: A Lightweight Data Dissemination Model for Smart Mobile Ad-Hoc Networks

Okan Turkes, Hans Scholten, and Paul J. M. Havinga
Pervasive Systems, EEMCS, University of Twente, Enschede, 7500AE NL
{o.turkes,j.scholten,p.j.m.havinga}@utwente.nl

ABSTRACT

This paper introduces BLESSED, a universal opportunistic ad hoc networking model intended for smart mobile devices. It enables fast and lightweight data dissemination in wireless community networks through the complementary utilization of the IEEE 802.11 and Bluetooth Low Energy standards. As a ubiquitous alternative to the publicly-limited ad hoc networking interfaces, it resolves many of the peer-to-peer data forwarding issues with smart beacon advertisements. Opportunistic beacons of BLESSED require neither association nor connection for data sharing, instead exploit the network identifiers as message carriers. Our applicability tests on a real-life setup indicate the soundness of the model. Providing a high data dissemination performance, BLESSED can be applied to numerous daily application scenarios.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communications

General Terms

Design, Performance

1. INTRODUCTION

Today's smart mobile devices are onerous to automatically establish mobile ad hoc connections with our physical circle of friends and between occasional peer-to-peer (P2P) contacts. In fact, they accommodate the commonly-accepted IEEE 802.11 and Bluetooth interfaces with ad hoc and P2P connectivity support. Based on the IEEE 802.11 standard, Wi-Fi Ad Hoc mode, Wi-Fi Direct, and Wi-Fi Hotspot are the available interfaces. Besides, Bluetooth is a long-running standard for wireless personal area networking. Nevertheless, these interfaces bring along significant limitations for efficient and reliable mobile networking:

i) They lack of self-organization. Prior to a connection, Classic Bluetooth and Wi-Fi Direct enforce secure pairing to unfamiliar devices [1]. Connections formed over Wi-Fi Ad Hoc mode require a data forwarding protocol that maintains dynamic topologies [2]. Connecting multiple clients via Wi-Fi Hotspot devices also requires a self-organizing data routing protocol [3].

ii) Mobility causes frequent disconnections/reconnections that increase routing overhead. Devices consume scarce bandwidth and energy for route rearrangements [4]. Besides, high density causes bandwidth-related issues [5].

iii) The commonly-used mobile operating systems such as *Android*, *iOS*, and *Windows Phone* restrict several functions of these interfaces. Wi-Fi Ad Hoc mode can be activated only under root access privileges on *Android* and is natively unsupported by *iOS* and *Windows Phone*. Wi-Fi Direct is unavailable on *iOS* and *Windows Phone*. Wi-Fi Hotspot has restricted bandwidth, hence is allowed with limited number of connections—at most 10 clients on *Android*; 5 clients on *iOS* and *Windows Phone*. For Bluetooth piconets, this maximum is 7 by default.

Above-mentioned facts are a disincentive to development of opportunistic network (OppNet) applications for general public use. OppNets provide data sharing among scattered wireless communities with the exploitation of mobile devices often intermittently connected to each other [6]. Such mobile environments necessitate rapid neighbor discovery and association as well as reliable packet switching between devices. Motivated by these limitations and challenges, we present a lightweight opportunistic ad-hoc data dissemination model for OppNets of smart mobile devices. We offer our model above the widely-adopted IEEE 802.11 and Bluetooth Low Energy (BLE, or Bluetooth 4.0+) standards. We provide packet switching and coordination between devices over the built-in advertisement beacons, i.e. human-readable wireless network identifiers of the BLE and Wi-Fi Hotspot interfaces.

Named BLESSED (BLE profile and IEEE 802.11 Service Set Encoding-based Dissemination), our model regulates simple message forwarding during Wi-Fi and BLE device discovery phases. Eliminating connection establishment phase, it exploits the wireless broadcast advantage to provide instantaneous and bandwidth-free transmissions even at short periods of inter-contact durations. It operates in OppNets of heterogeneous devices without violating the standards; thus represents high availability and applicability for general public use according to our real-life applicability tests. In comparison to the OppNet proposals presented for today's smart mobile platforms, it provides low-throughput

CHANTS'15, September 11, 2015, Paris, France.
© 2015 ACM. ISBN 978-1-4503-3543-0/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2799371.2799382.

but highly-scalable dissemination even under harsh environments according to our realistic simulation tests.

The remainder of the paper is organized as follows: Section 2 presents BLESSED. Section 3 explains our performance analysis. Section 4 discusses the related works. Section 5 gives the conclusion and our future research directions.

2. DISSEMINATION MODEL: *BLESSED*

BLESSED is based on the complementary use of BLE profiles and IEEE 802.11 SSID fields. Data exchange between participating devices is enabled with the following wireless access interfaces: Wi-Fi Hotspot mode, Wi-Fi Intrastructure mode, BLE Peripheral mode, and BLE Central mode.

In Wi-Fi Hotspot mode, devices advertise a previously encoded packet on their SSID fields. In Wi-Fi Infrastructure mode, devices perform scanning to discover the encoded packets within their wireless range. Since Wi-Fi Hotspot and Infrastructure modes are mutually exclusive, i.e. SSID advertising and scanning cannot run simultaneously, devices periodically switch between these modes. For coordination of SSID advertisements and switch times, devices further use BLE profiles, i.e. Universally Unique Identifier (UUID) fields. Figure 1 shows our approach of how packets are produced, disseminated, and interpreted on a single device. The BLESSED data exchange automaton has 3 superstates:

I) **Opportunistic Beacon** (OB): Wi-Fi Hotspot mode and BLE Central mode are run simultaneously in order to propagate a pre-encoded network packet through SSID advertisements and listen for coordination messages, i.e. encoded UUIDs, respectively.

II) **Beacon Observer** (BO): Wi-Fi Infrastructure mode and BLE Peripheral mode are run simultaneously in order to listen for network packets, i.e. SSID advertisements, and announce pre-encoded coordination messages through UUID advertisements, respectively.

III) **Update & Switch** (U&S): Wi-Fi Hotspot mode and BLE Peripheral mode need reactivation to modify their network identifiers. Therefore, either SSID or UUID is re-encoded prior to becoming an OB or a BO.

Running the automaton with random initialization times, devices in an OppNet can be in either of these three superstates at a specific time. During the course of their OB services, denoted as t_{OB}, devices can only perceive BOs within their communication range. Likewise, during the course of their BO services, denoted as t_{BO}, devices can only perceive OBs within their communication range. On the other hand, during U&S, denoted as t_{US}, devices are unable to perceive messages over both communication channels. BLESSED utilizes a dynamic deadline optimization in order to reduce the number of same service conflicts (OB-OB and BO-BO).

2.1 Service Scheduling

Figure 2 demonstrates the concurrent operations during an OB or a BO service. A device constantly announces a packet (P) with a fixed beacon interval (t_{BI}). For OB, P is a pre-encoded SSID whereas for BO, a pre-encoded UUID. Both SSID and UUID conform specific encoding types consisting of three fields: `Deadline`: The designated end time of current service. `Request`: The flags to inform application-specific requests. `Data`: The contents of a message. At each scan interval (t_{SI}), a device decodes perceived Ps, i.e. announced deadlines, application-specific requests, and data of the nearby devices which run the opposite service. It stores the received data as well as the requests for further analysis at the next U&S. Besides, it compares the decoded deadlines (D_i) with its current deadline (D_{curr}). As shown in Algorithm 1, it adjusts its timer according to the device(s) having the earliest D_i. This adjustment ensures two main advantages: First, it schedules its upcoming period with at least one device running in the opposite service. Second, it reduces redundant wait times for the current service.

During decoding, a device uses a mode flag for the next U&S. Once it detects at least one P that has a period congruent to its period, `mode` is set to *frequent*. It is set to *discovery* if no P is found. Algorithm 2 shows the use of `mode` for the determination of the next service and its corresponding deadline (D_{next}) during U&S. *Frequent* mode enables a device to periodically switch between the OB and BO services with a duration relatively shorter, one-half shorter in our design, than the actual service time. Here, the aim is to increase the number of different P advertisements and discoveries between the devices running the opposite services with the assumption that they will remain within range of each other. Besides, *discovery* mode assigns a random deadline, ranging from one-half shorter to one-half longer than the actual service time in our design. Here, the aim is to avoid OB-OB and BO-BO conflicts. Devices which cannot encounter any other device within their communication range extend their service times with the randomized deadlines.

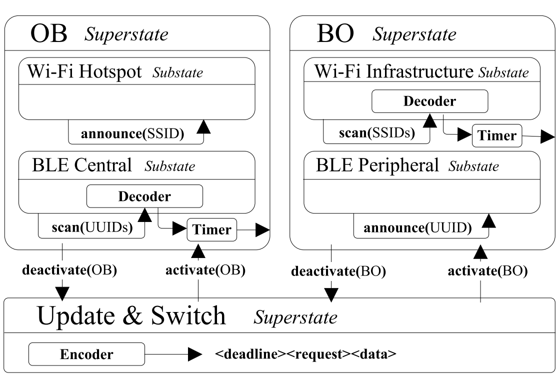


Figure 1: Finite state automaton

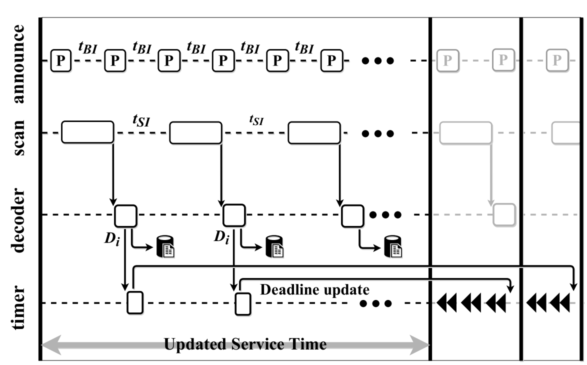


Figure 2: Sequence diagram of BLESSED operations

Algorithm 1: Dynamic Deadline Determination

require: **if** *Superstate=OB* **then** $t_{OB} > t_{SI}$ **else** $t_{BO} > t_{SI}$;
ensure: *List(P)=*$\emptyset$ **and** `mode` = empty;
repeat every *List(P)* $\leftarrow$ **scan()**
  for *each* D_i *in* *List(P)* **do**
    if $|D_i - D_{curr}| \leq t_{US}$ **then** `mode` = frequent;
    else if $D_i < D_{curr}$ **then** $D_{curr} = D_i$;
  end
until D_{curr};
if *List(P)=*$\emptyset$ **then** `mode` = discovery;

Algorithm 2: Next Service Determination

define: **if** *Next=OB* **then** $t_{Next} = t_{OB}$; **else** $t_{Next} = t_{BO}$;
if *mode* = *empty* **then**
  if *Prev* = *OB* **then** Next = BO; **else** Next = OB;
  $D_{next} = t_{now} + t_{Next}$;
else if *mode* = *frequent* **then**
  if *Prev* = *OB* **then** Next = BO; **else** Next = OB;
  $D_{next} = t_{now} + t_{Next}/2$;
else if *mode* = *discovery* **then**
  if *Prev* = *OB* **then** Next = OB; **else** Next = BO;
  $D_{next} = t_{now} + t_{Next} \pm$ **rand**$(t_{Next}/2)$;
end

2.2 Packet Structures

IEEE 802.11 SSID and BLE UUID have editable lengths of only 32 ASCII bytes and 128 bits, respectively. Therefore, Ps have to be encoded succinctly. In our design, all P structures consist of `Deadline`, `Request`, and `Data` fields.

Granted that all devices have the same current time, the last 3 digits of *UNIX time* are assigned to `Deadline` in our design. For SSID, `Deadline` is encoded to 2 ASCII bytes. For UUID, on the other hand, `Deadline` allocates 3 octets.

`Request` may involve a number of application-specific flags. Each mapped to a unique ASCII character, 94 different flags can be encoded on a single SSID byte. With a UUID bit, on the other hand, a single flag can be encoded. More detailed requests can be represented as the number of bytes/bits is increased. These flags are essential to increase dissemination efficiency. At every U&S, `Deadline` and `Data` can differently be encoded based on the received flags at the previous superstate. In our design, `Request` is a single flag with two types, **`repeat`** and **`shift`**, allocating 1 ASCII byte on a SSID encoding and 1 bit on a UUID encoding. This flag serves as a notification to other devices for packet distribution, and has no effect on the service scheduling presented in Section 2.1.

For the remainder bytes/bits, `Data` may involve a chunk of messages, or a summarized message of a sensed phenomena. In SSID encoding, `Data` may contain additional message information such as source ID, creation time, location, and intermediate router IDs. In UUID encoding, `Data` can only be used for primitive messages such as small integers and low precision values to represent a set of results or readings. To this respect, `Data` is used only on SSID encoding (OB), and is set to `null` on UUID encoding (BO) in our design.

2.3 Data Exchange

BLESSED represents two main restrictions for data exchange. First, `Data` forwarding is half-duplex, from an OB to BOs, at any given time. Second, network throughput is bound to the limited `Data` length that hinders sending all messages at once. Consequently, data dissemination is subject to potential delays even under long-lasting inter-contact durations. Nevertheless, BLESSED exploits the wireless broadcast advantage. At any given time, multiple OBs can be discovered by multiple BOs, and vice versa. This advantage provides `Deadline` and `Request` to better schedule the service deadlines and to better distribute several `Data` in an OppNet, respectively. As predefined variables in our design, t_{OB} and t_{BO} also play an important role on the data dissemination performance. Besides, `Request` is used to notify others of a current dissemination objective. Either serving as an OB or a BO, once a device receives a `Request` of **`repeat`**, it selects its last received `Data` for advertisement at its next OB service. Here, the aim is to disseminate recent messages in an OppNet. This type of data dissemination can be useful for participatory monitoring applications. On the other hand, once a device receives a `Request` of **`shift`**, it sequentially selects one from all of its received `Data` for advertisement at each OB service. Here, the aim is to disseminate all of the messages with distributed chances in an OppNet. This type of data dissemination can be employed in opportunistic Twitter-like applications that allow participators to share less-critical short messages with each other.

3. PERFORMANCE ANALYSIS

This section presents an analysis on the real applicability and networking performance of BLESSED. Our applicability study is carried out with 5 *Samsung S4 Mini* smart phones. The phones have run BLESSED as a proof-of-concept application which has been implemented on our *Android*-based mobile opportunistic networking platform, named *Cocoon* (Community-Oriented Context-aware OppNets) and introduced in [7]. The obtained real-life results are utilized under various simulated scenarios in order to evaluate dissemination performance with extended networking scales.

3.1 Applicability Study

The real potential of BLESSED is evaluated in terms of its wireless interfaces, wireless operations, and energy efficiency.

3.1.1 Wi-Fi and BLE Coexistence

Interference due to the collocation of 2.4GHz band Wi-Fi and Bluetooth is a fact for connection-oriented networks. However, BLESSED as a connection-free approach only uses advertisement beacon frames of these interfaces. BLE utilizes 3 advertisement channels (37, 38, 39), 2MHz-wide, and they are non-overlapping with the Wi-Fi channels 1, 6, 11. In theory, Wi-Fi channels 1,6, 11 do not create an in-band interference with the BLE advertisement channels. To test their coexistence, BLE Peripheral and Wi-Fi Hotspot modes are run 100 times with different OB/BO orientations. OBs utilized the Wi-Fi Hotspot Channel 6 whereas BOs utilized random BLE advertisement channels. Given in Table 1, our packet reception rate (PRR) results prove that BLESSED is not affected by the coexistence of Wi-Fi and BLE.

Table 1: SSID and UUID Reception Rates

Orientation	UUID PRR	SSID PRR
1 OB - 4 BO	377/400	98/100
2 OB - 3 BO	294/300	193/200
3 OB - 2 BO	191/200	286/300
4 OB - 1 BO	91/100	391/400

3.1.2 Device Density

A contention problem can arise when several devices simultaneously relay scan probe requests. Same problem can also occur when several advertisement beacons or scan probe responses are simultaneously relayed. In dense networks, beacon collisions dramatically decrease the neighbor discovery rates due to high number of probe requests/responses [8]. To avoid this, Wi-Fi and BLE employ specific clock synchronization algorithms. Wi-Fi has Carrier Sense Multiple Access with collision avoidance (CSMA/CA) whereas BLE has Clock Synchronization Protocol (CSP) under its multi-channel adaptation protocol. CSMA/CA and CSP utilize a special timing parameter called *contention window* to reduce the number of packet collisions. In case the shared channel is busy, each device backs off for a random period by increasing its *contention window*. Once the channel becomes idle, *contention window* is rest to its minimum value. This is handled at nanosecond level for both Wi-Fi and BLE, and significantly reduces the number of beacon collisions [8]. According to the IEEE 802.11 specification (Section 9.4.2) and Bluetooth v4.2 specification (Section 8.4.3), the contention-free period is ≈250ms in the worst case.

3.1.3 Wireless Operations

Table 2 shows the average (μ) and standard deviation (σ) values for t_{BI}, t_{SI}, and t_{US} obtained with 5 smart phones. In conformity with its standard value on *iOS*, *Android*, and *Windows Phone*, t_{BI} is ≈100ms for Wi-Fi Hotspot. In BLE, t_{BI} can be in a range of 20ms to 10,000ms on *iOS* and *Android*. Tested with 100ms on *Android*, the obtained t_{BI} values for BLE Peripheral mode are quite consistent as well. On the other hand, t_{SI} is non-adjustable. Wi-Fi Infrastructure mode may reflect varying t_{SI} values based on adapter capabilities. Including a complete scan for all 13 Wi-Fi channels, t_{SI} is expected to be 130ms at minimum according to the IEEE 802.11 specification. According to the BLE specification, t_{SI} ranges between 0.6ms and 1.2ms in BLE Central mode. In our tests, t_{SI} is set as 3000ms for both Wi-Fi Infrastructure and BLE Central modes. The obtained t_{SI} results show a slight variation in both modes.

Wireless operations of U&S are fourfold: *i)* switch from BO to OB, *ii)* switch from OB to BO. *iii)* reactivation of OB, and *iv)* reactivation of BO. Containing 815 repeated tests, our results indicate that *(i)*, *(ii)*, and *(iii)* which require activation and/or deactivation of Wi-Fi Hotspot mode takes considerably more time compared to *(iv)* which uses BLE activation and deactivation operations.

3.1.4 Power Consumption

Table 3 shows the average battery percentage usage (BPU) results obtained for several BLESSED automaton instances. The tests for each instance are carried out for ≈6 hours with a single type of smart phone. For both Wi-Fi and BLE, t_{BI} is set to 100ms and t_{SI} is set to 3000ms.

Unquestionably, the absolute BPU results can differ on different physical platforms. Nevertheless, the BPU results for different BLESSED instances relate with each other. Compared to connection-oriented interfaces, BLESSED demands considerably low energy. For instance, BPU of Wi-Fi Ad-Hoc mode is measured as 18.32%/h on a *Nexus 7* device. This is almost 3 times higher BPU than of BLESSED instances with 15s-long service times which are the maximum BPU results obtained in our tests.

Table 2: Wireless Operations

Operation	Time	Wi-Fi	BLE
Beacon Interval	t_{BI}	$\sigma = 0.014ms$	$\sigma = 0.022ms$
Scan Interval	t_{SI}	$\sigma = 247ms$	$\sigma = 12ms$
U&S (BO→OB)	t_{US} *(i)*	$\mu = 4302ms$ $\sigma = 524ms$	
U&S (OB→BO)	t_{US} *(ii)*	$\mu = 3407ms$ $\sigma = 327ms$	
U&S (OB→OB)	t_{US} *(iii)*	$\mu = 5910ms$ $\sigma = 641ms$	
U&S (BO→BO)	t_{US} *(iv)*	$\mu = 797ms$ $\sigma = 124ms$	

Table 3: Average BPU for BLESSED instances

Instance	Service time	BPU
Idle System Run	-	0,61%/h
Continuous BO	$t_{BO} = \infty$	3,58%/h
Periodic OB/BO switches	$t_{OB} = t_{BO}$ =45s	5,85%/h
Periodic OB/BO switches	$t_{OB} = t_{BO}$ =30s	5,98%/h
Periodic OB/BO switches	$t_{OB} = t_{BO}$ =15s	6,09%/h
Continuous OB	$t_{OB} = \infty$	5,52%/h

3.2 Networking Tests

The networking performance of BLESSED is tested on a realistic event-based simulator which is fed with our applicability study results. Table 4 summarizes the properties of the designated networks as well as expresses the simulated parameters. The network setups consist of three different group densities (N_1,N_2,N_3) each scattered over varying networking areas (M_1-M_5). Device movements are simulated with the Random Waypoint mobility model. Considering the fact that OB and BO discoveries may not be reciprocal due to fluctuating signal powers, at different times, devices are given random radio ranges uniformly ranging from 25m to 75m, and 20m to 60m for Wi-Fi and BLE interfaces, respectively. The results obtained in Section 3.1 are simulated for the wireless operations. The network setups are tested under different message frequencies and service operations.

All repeated 10 times, the tests are conducted regarding our model design presented in Section 2. The following results express the networking performance of the model in terms of data dissemination efficiency and point-to-point delivery efficiency.

The service scheduling (Section 2.1) and the packet coordination (Section 2.2) functions are separately evaluated through all network setups and for all $t_{OB} = t_{BO}$. Besides, message creation interval (t_{MI}) is set to 120s for these tests.

Contrary to the unscheduled runs, the runs with `Deadline` optimization show a slight improvement on the data dissemination performance (Figure 3). Especially in denser scenarios, the effect of the optimization is quite remarkable. In M_1, for instance, the data dissemination ratio is increased by ≈10% in average for all of the device groups. This improvement vanishes as the network setups get sparse.

Table 4: Simulation Properties & Parameters

Test Period: 2h for each combination of below parameters	
Number of devices: $N_1 = 50, N_2 = 100, N_3 = 150$	
Maps ($m \times m$): $M_1 = 500 \times 500, M_2 = 1000 \times 1000,$ $M_3 = 1500 \times 1500, M_4 = 2000 \times 2000, M_5 = 2500 \times 2500$	
Mobility: Random Waypoint	Movement pause (s): [0,300]
Wi-Fi range (m): $50 \pm [0, 25]$	BLE range (m): $40 \pm [0, 20]$
$t_{OB} = t_{BO}(s) = \{5, 15, 30\}$	t_{BI}, t_{SI}, t_{US} : Table 2
Message creation interval (s): $t_{MI} = \{30, 60, 90, 120, 150, 180\}$	

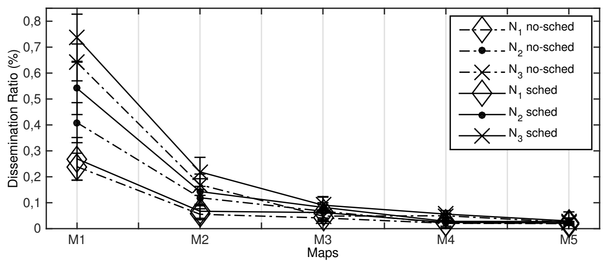


Figure 3: Impact of scheduling on dissemination performance under several device densities

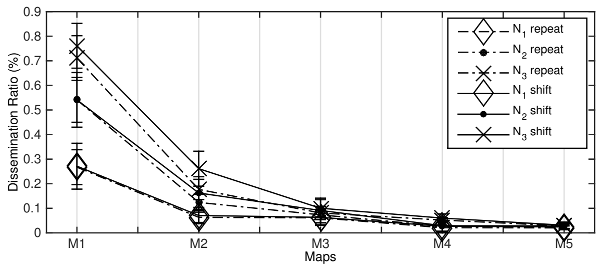
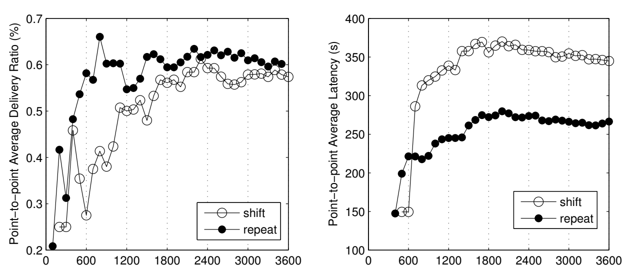


Figure 4: Impact of packet coordination on dissemination performance under several device densities

There is no significant difference between `Request:repeat` and `Request:shift` in terms of data dissemination performance (Figure 4). Nevertheless, `Request:repeat` performs much better than `Request:shift` in terms of point-to-point data delivery (Figure 5). At early times of the networking, it is evident that messages which are repeatedly advertised have higher delivery rates than alternatingly advertised messages. For both of these advertisement types, data delivery ratios run parallel as the network saturates in terms of number of messages (Figure 5(a)). But, repeatedly advertised messages are delivered almost twice earlier than alternatingly advertised messages (Figure 5(b)).

Regardless of network density or network size, too frequent U&S initiations may negatively affect the networking performance, which is caused by the short OB and BO service time assignments (t_{OB} and t_{BO}) (Figure 6).



(a) (b)

Figure 5: Packet Coordination: `shift` and `repeat`; the averaged results of the tests with N_1, N_2, N_3 **under** M_1**.** t_{MI}=120s**,** t_{OB}=t_{BO} = 30s

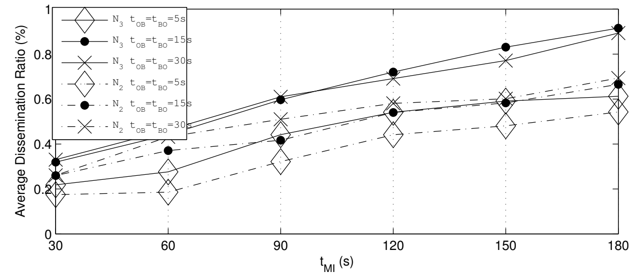


Figure 6: Impact of t_{OB}=t_{BO} **and** t_{MI} **on dissemination performance; the averaged test results of** M_1

The overall results indicate two prominent outcomes:

i) BLESSED achieves high networking performance in dense deployments. As the densest setup, N_3 under M_1 gives an average data dissemination ratio of 71%, with a standard deviation of 9%, when t_{MI} is 120s. For the same network setup, the average dissemination ratio decreases to 38%, with a standard deviation of 5%, when t_{MI} is 30s.

ii) BLESSED provides high suitability for low message frequency scenarios, or for the OppNets having single (or limited) message sources. Regardless of network density, the networking performance significantly increases as t_{MI} is increased (Figure 6). For t_{MI}=180s, our model has reached a data dissemination ratio of 58%, with a standard deviation of 7%, as the average test results with all network setups.

4. RELATED WORKS

Real-life OppNet implementations are emerging. As the pioneer works on community-oriented OppNets, the projects Serval and Haggle propose various OppNet applications on top of their framework implementations. Serval offers an *Android* application called The Serval Mesh [9] to sustain infrastructure-less mobile communications when outages occur in cellular infrastructures. The Serval Mesh creates a smart-phone mesh network by utilizing Wi-Fi Ad Hoc mode. Similarly, Haggle offers an OppNet paradigm called Pocket Switched Networks which envisions communications among people carrying mobile devices. Haggle aims at developing a cross-layer architecture which has support for the well-known desktop and mobile operating systems. According to the latest Haggle API [10], Haggle provides networking over Bluetooth and Wi-Fi Ad Hoc mode, in particular with support for *Android*, yet not for other platforms. Unless Wi-Fi Ad Hoc mode has built-in support on today's mobile operating systems, both The Serval Mesh and Haggle are available only on devices with root access privileges. Moreover, Classic Bluetooth also brings significant limiting factors for wide deployment of OppNets. Bluetooth discovery protocol is costly in terms of power [13]. Bluetooth networks mostly fit to link grouped devices under stable connectivities, and are not sufficient enough for OppNets [4].

As a recent project, Open Garden proposes mobile opportunistic platforms providing high applicability on today's smart mobile devices. Open Garden has an "off-the-grid" mobile messenger application, named Fire Chat [11], which uses Wi-Fi Direct and BLE and is available on *Android* and *iOS*. However, networking behind Fire Chat is not fully ad-hoc, and exploits fixed access points in case of necessity. As

Table 5: Smartphone-based OppNet applications & implementations

Implementation/ Application	WLAN/WPAN Interface(s)	Performance			Mobile O/S Availability		
		Scalability	Mobility	Throughput	Google Android	Apple iOS	Windows Phone
The Serval Mesh [9]	Wi-Fi Ad Hoc	++	++	+++	v2.2+ ★	v4.1+ ★	N/A
Haggle v0.4 [10]	Wi-Fi Ad Hoc & Bluetooth	++	++	+++	v2.2+ ★	v4.1+ ★	N/A
Fire Chat [11]	Wi-Fi Direct & BLE	++	+	++	v4.3+ ◆	v5.1+	v8.1+ ◆
WiFi-Opp [1]	Wi-Fi Hotspot	+++	++	++	v2.3+	v4.0+	v7.5+
PASA [12]	Wi-Fi Direct & Bluetooth	++	+++	+	v4.0+	v6.0+	v8.1+
Cocoon BLESSED	Wi-Fi Hotspot & BLE	+++	+++	+	v4.3+ ◆	v5.1+	v8.1+ ◆

★ Wi-Fi Ad Hoc mode requires root permission in Google Android; is not supported natively in Apple iOS and Windows Phone.

◆ BLE Peripheral mode is supported in Google Android v5.0+; is unavailable for Windows Phone yet.

a fully ad-hoc OppNet model, WiFi-Opp [1] presents a link layer model for smart mobile devices with the alternating use of Wi-Fi Hotspot and Infrastructure modes. Similar to our motivation, WiFi-Opp addresses the restricted ad-hoc networking availability on smart mobile platforms. Wifi-Opp provides self-organizing OppNets, nevertheless does not address bandwidth-related issues reflected by today's Wi-Fi Hotspot interfaces. Networks formed over Wi-Fi Hotspot connections are subject to low network throughput and scalability under high device density and high mobility [5], respectively. Similar to our approach, PASA in [12] uses wireless identifier fields to broadcast and intercept critical signals either over Wi-Fi Direct and Classic Bluetooth. Nevertheless, Wi-Fi Direct and Classic Bluetooth are costly interfaces in terms of device discovery [1, 13].

Table 5 itemizes the above-referred studies; summarizes their networking capabilities; and shows their usability on the widely-used mobile operating systems. In comparison to these studies, to the best of our knowledge, BLESSED provides higher network scalability, dissemination rate, energy efficiency, and public availability even under dynamic mobility and high nodal density.

5. CONCLUSION & FUTURE WORKS

In this paper, we have proposed BLESSED, a viable and extremely lightweight opportunistic communication model intended for ad hoc networking with smart mobile devices. We have presented the model as a ubiquitous alternative to the built-in wireless access interfaces which are onerous for opportunistic networking in practice. We have offered the model as a higher level connection-free protocol on top of the BLE and IEEE 802.11 Service Set standards which does not require special adaptations on the default platforms. We have presented the high availability and applicability of the model on today's smart mobile phones. Furthermore, we have analyzed the networking performance of the model through several data dissemination scenarios. According to our results, BLESSED presents a promising suitability for opportunistic short message dissemination applications.

Our future work is twofold. First, we will broaden our real-life experimental deployments with BLESSED for various opportunistic use cases. Second, we will investigate more adaptive schemes inside BLESSED to make it more congruent to the offered application scenarios.

Acknowledgments

This study is funded by SenSafety project under COMMIT, the Dutch National Program.

6. REFERENCES

[1] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "Wifi-opp: Ad-hoc-less opportunistic networking," in *ACM-CHANTS 2011*.

[2] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "Wi-fi in ad hoc mode: a measurement study," in *IEEE PerCom 2004*.

[3] D. Dubois, Y. Bando, K. Watanabe, and H. Holtzman, "Lightweight self-organizing reconfiguration of opportunistic infrastructure-mode wifi networks," in *IEEE SASO 2013*.

[4] D. Contreras and M. Castro, "Experimental assessment of the adequacy of bluetooth for opportunistic networks," *Ad Hoc Networks*, vol. 25, Part B, 2015.

[5] S. Sagari, A. Baid, I. Seskar, T. Murase, M. Oguchi, and D. Raychaudhuri, "Performance evaluation of mobile hotspots in densely deployed wlan environments," in *IEEE PIMRC 2013*.

[6] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, 2006.

[7] O. Turkes, H. Scholten, and P. J. M. Havinga, "Opportunistic beacon networks: Information dissemination via wireless network identifiers," in *review*.

[8] B. Dezfouli, M. Radi, S. A. Razak, K. Whitehouse, K. A. Bakar, and T. Hwee-Pink, "Improving broadcast reliability for neighbor discovery, link estimation and collection tree construction in wireless sensor networks," *Computer Networks*, vol. 62, no. 0, pp. 101 – 121, 2014.

[9] P. Gardner-Stephen and S. Palaniswamy, "Serval mesh software-wifi multi model management," in *ACM ACWR 2011*.

[10] "Haggle version 0.4." http://www.haggleproject.org. Accessed: 2015-05-15.

[11] "Fire Chat." http://opengarden.com/firechat. Accessed: 2015-01-30.

[12] Y. Mao, J. Wang, J. Cohen, and B. Sheng, "Pasa: Passive broadcast for smartphone ad-hoc networks," in *ICCCN 2014*, pp. 1–8, Aug 2014.

[13] C. Drula, C. Amza, F. Rousseau, and A. Duda, "Adaptive energy conserving algorithms for neighbor discovery in opportunistic bluetooth networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, pp. 96–107, Jan 2007.