Burst Size Matters: When Node Degree Helps Accelerate Opportunistic Dissemination

Matteo Sammarco⋆, Nadjet Belblidia⋆, Marcelo Dias de Amorim⋆,
Luis Henrique M. K. Costa†, and Vania Conan⋄

⋆LIP6/CNRS – UPMC Sorbonne Universités {sammarco,belblidia,amorim}@npa.lip6.fr
†Universidade Federal do Rio de Janeiro luish@gta.ufrj.br
⋄Thales Communications & Security vania.conan@fr.thalesgroup.com

***Abstract*—Disseminating large files in opportunistic networks requires splitting the content into smaller pieces in order to leverage short contacts between nodes on the move. A negative consequence of content chopping is that it may generate significant overhead, as nodes have to exchange more signaling information to determine which pieces the neighbor misses. In this paper, we investigate the convenience of exchanging a burst of pieces at once at the risk of sending redundant pieces. Although achieving a good tradeoff between signaling reduction and redundant transmissions is challenging, we found out that node degree is a good indicator to determine burst size. We propose a distributed multi-content dissemination protocol with an adaptive burst dimensioning based on the device neighborhood density. We score its performance using both synthetic mobility traces and a testbed composed of real mobile devices and finely monitor the behavior of the protocol by deploying passive monitors in the target area. Our experiments show that our proposal achieves much faster dissemination than related alternatives that employ a fixed burst size. As a matter of fact, our work provides insights into the necessity of adopting adaptive strategies in practical situations involving device-to-device content dissemination.**

I. Introduction

Opportunistic device-to-device communications are an ideal playground for collocated users to retrieve or share content with their social network followers and friends, anytime, anywhere. Mobile users, who used to be pure consumers, started exploring the features of handheld devices to produce and disseminate their own content while on the move.

Shifting content sharing to the opportunistic domain is not straightforward, though. One of the main challenges is the fleeting nature of contacts between users – a number of experimental initiatives have shown that most contact durations in human-driven opportunistic networks fall under the minute [1], [2], [3]. Thus, we must split large content into small pieces so that no short-lived contacts are wasted. This is illustrated in Fig. 1, where node n_1 meets n_2 and declares the pieces it misses. Node n_2, in turn, checks the list to determine which pieces it can send to n_1. By consecutively getting pieces through different contacts, n_1 eventually retrieves all the pieces that compose the content file.

The challenge when handling multiple content files composed of several pieces each is to limit the signaling overhead. Indeed, when two nodes meet, they must first handshake to determine which pieces one can obtain from the other – the larger the number of pieces, the higher the overhead, especially when handshaking after each piece exchange. Existing

978-1-4673-7306-7/15/$31.00 ©2015 IEEE

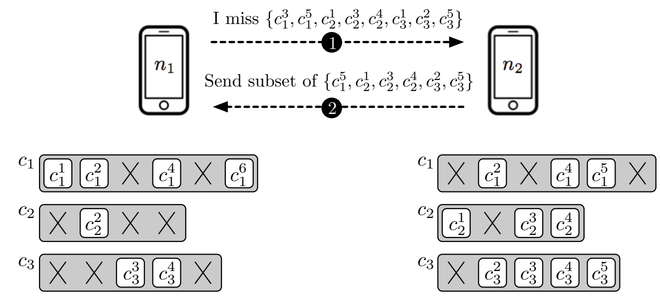


Fig. 1. In this example, three content files must be disseminated in the network. When node n_1 meets n_2, it sends the list of pieces it misses. Node n_2 checks the pieces it has and decides which one it can send.

algorithms in the literature focus on the prioritization of the missing pieces, but do not consider *how many* of them must be sent before handshaking again. It is intuitive to think that it is worth sending as many pieces as possible at once before the connection breaks. Although this holds true in theory and under some assumptions, the story is different in practice. In this paper, we show that sending a burst of pieces is indeed a good solution *only if the size of the burst is finely tuned*. In a nutshell, the main reason for such a statement is that, if the burst is too long, a node may, meanwhile, retrieve pieces from other nodes. This is due to the fact that, in practice, it is difficult to dequeue packets once they are sent to the output queue at the MAC interface. The consequence is the waste of communication resources with redundant data.

We find out that the node degree is a good metric to determine an appropriate size of the bursts. To this end, we adopt a fully experimental approach. We propose a protocol that dynamically adapts the burst size by referring only to a local measure of node degree. This metric turns out to be a very good indicator (during steady state) of the possibility for a node to obtain a given piece of a content from multiple neighboring nodes. To the best of our knowledge, no previous strategies tackled this specific problem. In order to build up our solution, we adopt a baseline strategy that determines the right pieces to transmit – but that uses *fixed burst sizes*.

We conduct experiments using real devices embedded in an uncontrolled environment. To support performance data analysis, we also rely on wireless traffic traces captured by passive monitors during experiments. We also develop a simulator based on a pedestrian mobility model, to have the possibility of studying the impact of our solution in synthetic traces beyond real mobility traces.

We focus our investigation on the convenience of exchanging bursts in terms of dissemination latency. We clearly observe that sending a large burst of pieces can help or worsen the dissemination, depending on the size of the burst. Thanks to its adaptive behavior, our solution allows much faster dissemination than solutions using fixed burst sizes.

In summary, our contributions are the following:

- We experimentally evaluate the interest of sending a burst of pieces after each handshake. We show that there is a correlation between burst size and density of collocated nodes.
- We propose a protocol that relies on node degree to determine, on the fly, the burst size. Our proposal relies on local information only and does not generate any undesirable signaling overhead.
- We evaluate our solution using real experiments as well as simulations relying on both synthetic and measured mobility datasets. Our analyses confirm the necessity to fine-tune the burst size at each encounter.

The rest of this paper is organized as follows. We first describe the motivations and context behind this work in Section II. We show then in Section III the experiments that reveal the importance of the burst size when transmitting content files composed of multiple pieces. We propose the adaptive, degree-based solution in Section IV and evaluate its performance in Section V. We present related work in Section VI and conclude the paper in Section VII.

II. Rationale and problem statement

The motivation behind this work appeared with the implementation of a distributed network protocol (namely EPICS) to disseminate as quickly as possible multiple content files in opportunistic networks [4]. To achieve this goal, it chops the files into pieces in order to leverage short contacts (frequent in real-world situations). In the general case, a user has multiple content files, each one divided in several pieces.

In a nutshell, EPICS relies on a local measure of "prevalence" to decide which piece to transmit first. Upon a contact, a node decides to first send pieces that, from its local point of view, are less prevalent in the network. The prevalence is computed based on a counter that indicates how many times the node has detected a copy of a given piece at its neighbors. It involves a handshaking phase where a node sends to its neighbor the list of pieces it misses (as shown in Fig. 1). For the purposes of this paper, this description should suffice but, for the sake of completeness, a description of how nodes handshake is presented in Appendix A.

Because nodes are likely to experience contacts with multiple neighbors at the same time, the baseline functioning of EPICS performs handshaking after the transmission of each piece. During the experiments, we noticed that performance changed significantly depending on the network conditions. In particular, we figured out that handshaking after each transmission was the best configuration in dense networks but did not perform well in sparse setups. We decided to investigate this issue further and the results are reported in this paper.

Fig. 2. At the bottom, off-the-shelf Android devices running our protocol. At the top, two passive monitors with three antennas capturing environmental wireless traffic.

In the rest of this write-up, we will also call EPICS as "fixed strategy" and our proposal "adaptive strategy". We refer to the "baseline" approach as the case of a burst of size = 1.

III. Experimental assessment of the impact of burst size

Our work adopts a fully experimental approach – as we will discuss later, the performance degradation we observed in our tests are due to practical aspects that are not captured by existing models. Before investigating the impact of the burst size on the behavior of the system, let us first describe the testbed we considered in our experiments.

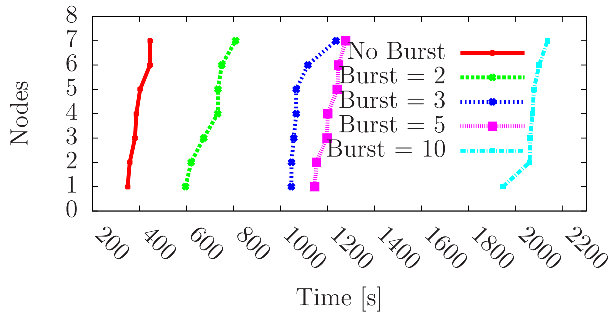
A. Testbed

We conduct experiments in a real, uncontrolled environment of smartphones running Android and configured in IEEE 802.11 ad hoc mode [5]. The system can be easily scaled up since it is also available for Android-x86 running as a virtual machine [6]. In our experiments, we use eight Samsung Galaxy SII smartphones. We setup the dissemination protocol to chop content in pieces of 25 Kbytes. During a calibration phase, we found out that 25 Kbytes is an appropriate piece size, in the range [5-64] Kbytes, in order to obtain fast dissemination [7].

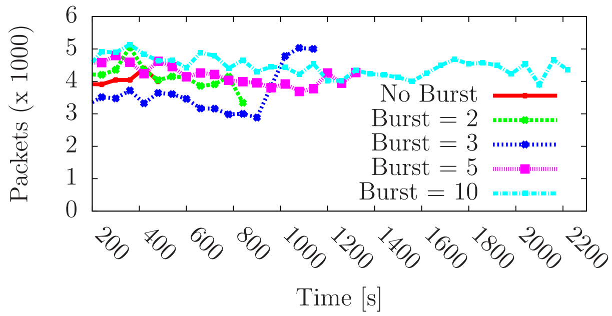
Given that we use real devices in a real environment, we decided to run a passive wireless monitoring system to capture events and traffic generated by surrounding devices, helping us to understand the experiments in detail. We use WiPal both as wireless capturing and trace merging software [8], [9].

B. Impact of burst size

First, we investigate is if it is worth sending a burst of *least prevalent* pieces at each contact. We set up our testbed with one source node which has five contents of 3 Mbytes each and seven nodes, each requesting all the contents. As soon as a node gets a piece, it also acts as a source in a P2P fashion.



(a) Dissemination latency of five contents on seven nodes modulating the burst.



(b) External traffic observed during the experiments.

Fig. 3. Investigation of dissemination time in function of the burst size. To be sure the results are not influenced by external surrounding wireless traffic, we also monitor the environment with a passive monitoring system.

The burst size is modulated in the set of {2, 3, 5, 10} pieces. The basic design of EPICS uses bursts of size = 1.

Performance, in terms of dissemination latency, becomes worse and worse as the burst size increases. Fig. 3(a) shows, for each burst size, the elapsed time to achieve a complete dissemination of the five contents to one node, two nodes, until all the seven nodes which requested the contents. In many cases, with a burst size = n the dissemination on the seven nodes is faster than the same dissemination on only one node with a burst size = $n + x$. These results are counter-intuitive, as they are completely independent from the external wireless traffic listened by the monitors (Fig. 3(b)). For example, when burst size = 3, we record less external traffic than in the case we do not use any burst; it takes, however, exactly three times more to disseminate the contents.

Fig. 4 shows the piece transmission efficiency by tuning the burst size. The efficiency is calculated as the percentage of the total amount of pieces required by all the nodes over total number of pieces placed in the transmission queue by all the nodes. As an example, let us consider that 7 nodes require 5 contents of 126 pieces each. Thus, in order to complete the dissemination, $7 \times 5 \times 126 = 4410$ right pieces must be received among all nodes. If 8820 pieces have been sent instead, the efficiency is equal to 50%. With no burst, the efficiency is around 70%. The remaining 30% efficiency decrease is due to pieces lost during transmission and then retransmitted, and by duplicate pieces, that are pieces well received but useless, since already received from other nodes before. In the analysis of wireless traces, we have not found many retransmitted pieces, so the efficiency is mostly affected by duplicate pieces. With a burst size = 2, the efficiency drastically decreases to almost 20%, and continues to decrease with larger burst sizes. Moreover, pieces experience longer queue delays, enlarging the burst.

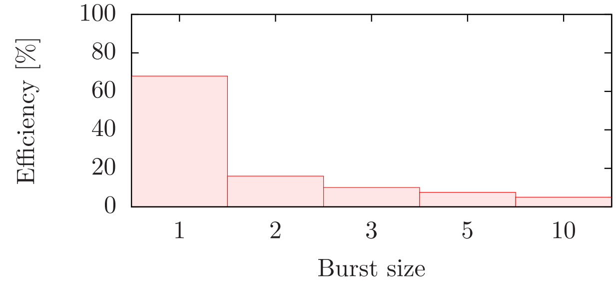


Fig. 4. Piece transmission efficiency tuning the burst size.

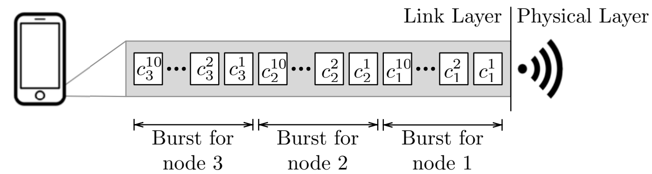


Fig. 5. View of a node's transmission queue using a burst size = 10. Note that pieces experience a long queue delay becoming outdated once transmitted.

Let us assume a burst size = 10 as shown in Fig. 5. At each contact with another node, at most ten pieces are placed in the transmission queue. The queue grows ten times faster than the basic solution without burst. These ten pieces are chosen based on a local and contemporary view. The transmission queue is FIFO, without preemption. For each piece included in a data-link frame, a node must gain access to the wireless medium waiting to be idle or reserving a slot with the RTS–CTS mechanism. Thus, when pieces in the tail of the queue (e.g., pieces to node 3 in the figure) eventually reach the head of the queue, they are likely to be obsolete, wasting transmission slots (i.e., other neighbors may have already sent that piece to the node).

We deeper investigate this point varying the quantity of nodes involved in the content exchange. We make an experiment sharing one content of 3 Mbytes with only one source and one node requiring the content (node degree = 1). Then, we make the same with one source and two other nodes (node degree = 2), until there are eight nodes in total (node degree = 7). For each experiment, we tune the burst size from one to ten and we repeat it three times in different hours of days. Fig. 6 shows the average dissemination latency in each case. We note that dissemination is faster using a large burst when only few nodes are employed. On the other hand, with more nodes in contact, dissemination is faster by reducing or disabling the burst. As we can see, with node degree = 1, the minimum latency is achieved with the maximum burst size = 10. With a node degree = 2, the minimum values of dissemination delay are expected with a burst size = 4. With larger degrees, it is worth using smaller burst sizes, until the node degree = 5, where the minimum latency is achieved without bursts.

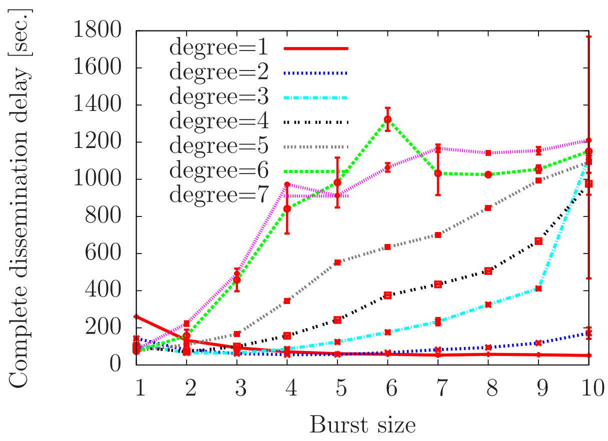


Fig. 6. Complete dissemination delay average and standard deviation, tuning the burst size and the nodes degree.

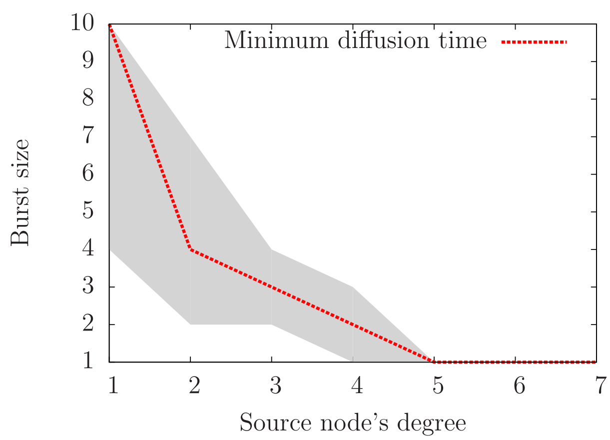


Fig. 7. Operation area for an adaptive solution. Line connects burst size values for minimum diffusion time. Gray area includes dissemination delay values 30 seconds longer than the minimum.

IV. Fine Tuning Burst Sizes for Fast Opportunistic Dissemination of Large Files

In Fig. 7 we connect burst size values, per source node's degree, achieving the minimum dissemination delay in Fig. 6. The gray area includes values of burst for a diffusion time at most 30 seconds longer than the minimum. Note that this area becomes narrower and narrower as the number of nodes in contact grows. In this plot, the fixed burst size solution moves on the bottom, meaning that it can be improved, up to a node degree of four. From a node degree of five up, it is worth sharing only one piece per contact handshake.

As a consequence, we advocate for the inclusion of some flexibility in the choice of the burst size. Based on the observations reported in the previous sections, we propose a degree-based adaptive dissemination solution that modulates the burst size according to the number of neighbors, always following the minimum diffusion time line of Fig. 7 and shown in the following abacus:

Abacus

node degree	1	2	3	4	≥5
burst size	10	4	3	2	1

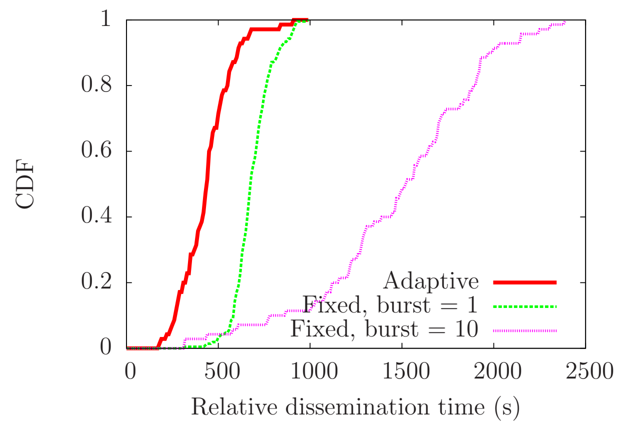


Fig. 8. Adaptive vs. fixed with burst sizes of sizes 1 and 10. The y-axis shows the CDF of relative dissemination time per content and per node.

We compare the adaptive solution against the baseline fixed version (i.e., with a burst size = 1) and against an extreme case with fixed bursts of ten pieces. We start this experiment with only two nodes: one source that has 10 contents of 3 Mbytes to share and another node. Then, every three minutes we add a new node requiring all of the contents, up to seven nodes. We gather relative completion times for each content from every node and we present the cumulative distribution function in Fig. 8. Even if both approaches take the same time to diffuse all the contents to all the nodes since from 5 nodes up in the network they have the same mode of operation, they present a considerable difference until the 97^{th} percentile. It means that the dynamic adaptation not only facilitates the diffusion when there are only a few nodes, but, since the content is almost fully received in many nodes, these nodes can better support the dissemination even when we insert more nodes in the network. On the other hand, when using fixed bursts with the maximum size, the dissemination is very fast for the first few contents (i.e., when there are only few nodes), and then it slows down taking five times longer.

We also compare the adaptive versus the fixed solutions with static node degrees (Fig. 9). In each case, at the beginning there is only one source node with a content of 3 Mbytes. A node degree of one reveals the largest difference (the adaptive approach is 5 times faster) between the two approaches. The difference becomes narrower as the node degree increases, but the adaptive solution still remains three and two times faster.

V. Using adaptive bursts in practice: Real and synthetic mobility traces

We showed in Section IV how tuning the burst size can either improve or worsen the dissemination performance depending on the number of nodes in contact. To check how important this profit margin is, we analyze in the following, some real and synthetic mobility traces. In particular, we examine the cumulative distribution function of nodes degree at every beaconing instant. We exclude from the distribution isolated nodes, as they cannot exchange contents with anyone.

We consider the following mobility traces:

- **Shopping Mall [10].** This is a six days dataset of real-world Bluetooth contact data collected from a mall

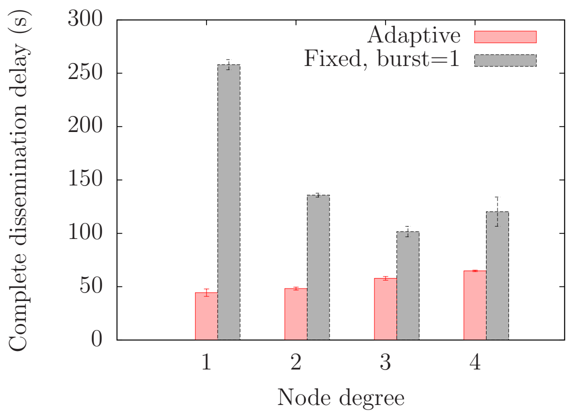


Fig. 9. Complete dissemination delay with respect to different node degrees.

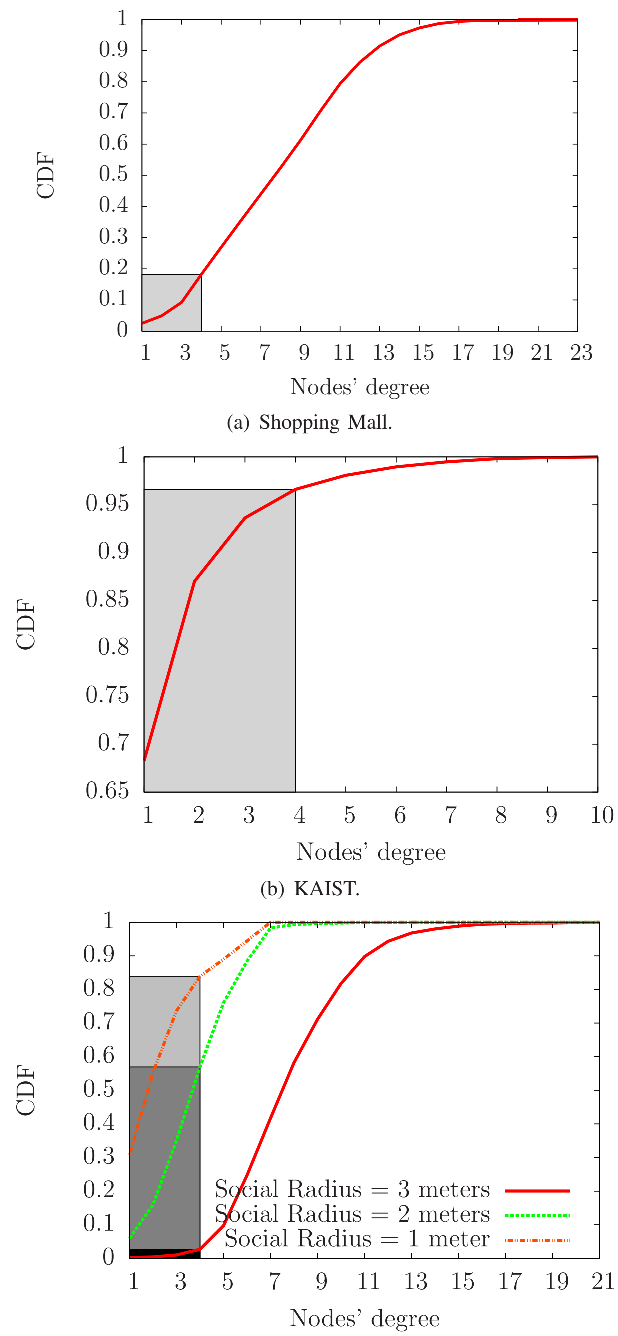
in Nottingham (UK). 25 devices captured Bluetooth contacts.

- **KAIST [11].** Another real-world dataset, consisting of 92 daily GPS track logs collected from the KAIST university campus in South Korea. Traces have been overlapped in time to produce one single trace. We assume a contact range of 10 meters, compatible with the Bluetooth trace.
- **SIMPS (synthetic) [12].** We also consider a mobility model of human crowds with pedestrian motion called SIMPS and we implemented a simulator based on this model. We simulated a relatively dense toroidal space of 100×200 meters with 100 people moving around for one hour. This model is based, among other parameters, on a "social radius". Nodes take decisions about their movements according to the nodes they detect in that radius. In crowded environments, the social radius tends to shrink. Since we simulated a crowd model, we used a social radius of one, two, and three meters, varying the contact range accordingly.

We chose these traces because they are different in many aspects: nature (real and simulated), environment (mall, university campus, simulated toroidal plane), log collection (Bluetooth device, GPS, 2D position), plane size (medium, huge, small), and density (medium, low, high). Moreover, being pedestrian mobility traces, we believe that contact duration is enough to exchange the maximum burst of 10 pieces.

Fig. 10 shows the distribution of node degrees for the traces we consider. The probability to find a node with a degree, at most, equal to four, considerably varies from trace to trace. The Shopping Mall in Nottingham has an indoor surface area of 10,880 square meters. We can image it very crowded, especially during rush hours. Nonetheless, not everyone has a Bluetooth enabled device. Thus, we can detect nodes having a degree, at most, equal to 4, with about 20% probability.

The KAIST campus has a 1,432,882 square-meter area. This big scenario makes it possible to exhibit a very high probability (more than 95%) of nodes with a degree = 4 at most. For the sake of fairness, being a GPS trace, indoor places that should be the most crowded are not taken into account. In this case, using a large burst will largely improve the outdoor opportunistic exchange of content.



(a) Shopping Mall.

(b) KAIST.

(c) SIMPS (synthetic).

Fig. 10. Cumulative distribution function of the number of nodes that every node perceives in its neighborhood.

We have also simulated a very dense scenario using the SIMPS mobility model (with 100 people in a 100×200 m^2 plane). The improving margin considerably changes for slightly changes of social radius. In the case of three meters, there is only a 0.03% of improving margin. For a social radius of two and three meters, we get nodes with a degree of at most 4 with a probability of 55% and 85%, respectively.

As we can see, there is room for improvement in all cases. Furthermore, the gains can be significant. This means that the use of an adaptive burst size selection mechanism is recommended, as *the operation point observed in practical scenarios frequently falls in the* $[1-4]$*-degree range*.

VI. Related work

Several works dealing with opportunistic content dissemination, consider splitting contents in smaller pieces [13], [14], [15], [16]. They propose different piece selection strategies without framing the use of a burst. As an example, SPAWN is a cooperative proximity-driven rarest-first piece selection strategy for content downloading in vehicular networks [17]. The dynamic adaptation of dissemination protocol parameters, has been the subject of studies in sensor networks. Complementary to our performance objectives, these solutions are mainly designed to limit the energy consumption. Deluge dynamically adjusts the rate of advertisements to allow quick propagation when needed while consuming few resources in the steady state [18]. Data aggregation is a common mechanism for energy-efficient forwarding [19], [20].

Zyba et al. analyze users' social behavior aspects in real mobility traces to derive the proper conditions behind an efficient content propagation scheme [21]. They observe that under common real life circumstances, the effectiveness of dissemination mainly depends on the number of users in each social class rather than their social behavior. In particular, they find that many areas are dominated by "Vagabonds" class of nodes. "Vagabonds", despite the "Social" nodes, show up rarely and fleetingly. With regard to this classification, our adaptive protocol takes advantage in areas populated by "Vagabonds".

To have a more complete view on performance, we have deployed a passive wireless traffic capturing system. In large scale testbeds, with several mobile nodes moving in a large area, legacy wireless traffic capturing systems are unsuitable. They are based on the deployment of a very large array of monitors leading to high fixed costs (monitors purchase, installation and maintenance) and variable costs (wireless traces collect and analysis). Approaches based on monitors selection and on collaborative measurements limit the need of so many monitors keeping the good capture quality [22], [7]. For large scale testbeds, approaches based on monitors selection and on collaborative measurements, limit the need of a large set of monitors keeping the good capture quality [22], [7].

VII. Summary and outlook

Starting from an opportunistic multi-content dissemination protocol based on content chopping and piece selection, we investigate the convenience to transmit a burst of pieces at each opportunistic contact. To be sure not neglecting aspects related to wireless communication mechanisms, we simultaneously deploy a testbed with real off-the-shelf devices and passive monitors to capture the generated traffic. Results show the need for dynamic burst dimensioning based on the node degree.

We found out that, the lower the neighborhood cardinality, the larger the burst size should be. In addition, when the degree is above 4, it is worth sending only one piece per handshake. Indeed, results show that using large bursts in crowded environments leads to a significant amount of redundant transmissions. Pieces experiencing long queuing delay are most likelihood to be sent earlier by other nodes, leading to duplicate exchange and loss of transmission slots.

After comparing our adaptive solution against a basic strategy that only sends one piece per contact and against a strategy that always uses a static large burst, we investigated its impact analyzing some real and simulated mobility traces. This analysis revealed that our proposal has enough potential to yield significant gains in multiple scenarios, and in particular in sparse environments.

Future works include the extension of our proposal by taking into account other parameters such as the piece size, the number content files, and lower level transmission rates.

Acknowledgment

Matteo Sammarco and Marcelo Dias de Amorim carried out part of the work at LINCS (http://www.lincs.fr). This work was partially funded by the European Community's Seventh Framework Programme under grant agreement no. FP7-317959 MOTO.

References

[1] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on DTN routing," Rio de Janeiro, 2009, pp. 1116–1124.

[2] S. Gaito, E. Pagani, and G. P. Rossi, "Opportunistic forwarding in workplaces," Barcelona, Spain, 2009, pp. 55–60.

[3] A. Barrat, C. Cattuto, V. Colizza, J.-F. Pinton, W. V. den Broeck, and A. Vespignani, "High resolution dynamical mapping of social interactions with active RFID," 2008, arXiv:0811.4170.

[4] N. Belblidia, M. Sammarco, L. Costa, and M. Dias de Amorim, "Epics: Fair opportunistic multi-content dissemination," *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[5] M. Sammarco, N. Belblidia, Y. Lopez, M. Dias de Amorim, L. H. M. K. Costa, and J. Leguay, "PePiT: Opportunistic Dissemination of Large Contents on Android Mobile Devices," Zurich, Switzerland, 2012.

[6] [Online]. Available: http://www.android-x86.org

[7] M. Sammarco, "Opportunistic Multi-Content Dissemination: Passive Monitoring and Adaptation to Network Conditions," Ph.D. dissertation, University Pierre and Marie Curie, May 2014.

[8] T. Claveirole and M. D. de Amorim, "Manipulating Wi-Fi packet traces with WiPal: design and experience," *Software Practice & Experience*, vol. 42, no. 5, pp. 585–599, May 2012.

[9] ——, "WiPal: IEEE 802.11 traces manipulation software," Jan. 2010. [Online]. Available: http://wipal.lip6.fr/

[10] A. Galati and C. Greenhalgh, "CRAWDAD data set nottingham/mall (v. 2013-02-05)," Downloaded from http://crawdad.cs.dartmouth.edu/nottingham/mall, Feb. 2013.

[11] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "CRAWDAD trace ncsu/mobilitymodels/gps/kaist (v. 2009-07-23)," Downloaded from http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels/GPS/KAIST, Jul. 2009.

[12] V. Borrel, F. Legendre, M. D. de Amorim, and S. Fdida, "Simps: using sociology for personal mobility." *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 831–842, 2009.

[13] M. Pitkänen, A. Keränen, and J. Ott, "Message fragmentation in opportunistic dtns," Newport Beach, CA, USA, 2008, pp. 1–7.

[14] S. Rajagpalan and C. C. Shen, "A cross-layer decentralized BitTorrent for mobile Ad Hoc networks," San Jose, USA, 2006, pp. 203–212.

[15] N. Gaddam and A. Potluri, "Study of BitTorrent for file sharing in ad hoc networks," Allahabad, India, 2009, pp. 1–6.

[16] M. Sbai, E. Salhi, and C. Barakat, "P2P content sharing in spontaneous multi-hop wireless networks," Bangalore, India, 2010, pp. 1–10.

[17] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Sanadidi, "Co-operative downloading in vehicular Ad Hoc wireless networks," St.Moritz, Switzerland, 2005, pp. 32–41.

[18] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 81–94. [Online]. Available: http://doi.acm.org/10.1145/1031495.1031506

[19] B. S. Mathapati, S. R. Patil, and V. D. Mytri, "Article: An adaptive energy efficient forwarding data aggregation and qos protocol for wireless sensor networks," *International Journal of Computer Applications*, vol. 46, no. 20, pp. 37–45, May 2012, published by Foundation of Computer Science, New York, USA.

[20] M. Busse, T. Haenselmann, and W. Effelsberg, "Energy-efficient aggregation forwarding for wireless sensor networks," in *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*, Oct 2007, pp. 584–589.

[21] G. Zyba, G. M. Voelker, S. Ioannidis, and C. Diot, "Dissemination in opportunistic mobile ad-hoc networks: The power of the crowd." in *INFOCOM*. IEEE, 2011, pp. 1179–1187.

[22] M. Sammarco, M. E. M. Campista, and M. Dias de Amorim, "Trace selection for improved WLAN monitoring," in *Proceedings of the 5th ACM workshop on HotPlanet*, ser. HotPlanet '13. New York, New York, USA: ACM Press, 2013, pp. 9–14. [Online]. Available: http://dx.doi.org/10.1145/2491159.2491165

[23] N. Belblidia, M. Dias de Amorim, L. H. M. K. Costa, J. Leguay, and V. Conan, "Part-whole dissemination of large multimedia contents in opportunistic networks," Mar. 2012.

Appendix A
Opportunistic Piece-Selection Dissemination and Handshaking Mechanism

Let $\mathbf{N} = \{n_0, n_1, \ldots, n_{N-1}\}$ be the set of N mobile nodes in the network. We do not assume any knowledge of mobility patterns. We assume that all nodes in the network are interested in a set of contents $\mathbf{C} = \{c_0, c_2, \ldots, c_{C-1}\}$. Each content c_j is initially only available at a single data source. We do not make any assumption on the creation time of contents.

For each content c_j, the data source chops the content into K_j pieces of equal size (the piece size can be accurately determined to optimize communication opportunities [23]). Contents could have different sizes, hence, the number of pieces depends on the content size. Pieces are sequentially identified as $c_j = \{d_0, d_1, \ldots, d_{K_j-1}\}$. Nodes use their contact opportunities to get pieces, i.e., we assume that there is no infrastructure to help the dissemination process. Nodes can get pieces from the data source and from any other node in the network having it.

Each node n_i locally stores an *availability bitmap vector* $\mathbf{a}_{n_i,j} = \{a_0, \ldots, a_{K_j-1}\}$ and a *prevalence vector* $\mathbf{p}_{n_i,j} = \{p_0, \ldots, p_{K_j-1}\}$ both associated with every known content c_j. The availability bitmap vector $\mathbf{a}_{n_i,j}$ keeps track of c_j content pieces that the node n_i holds. It contains binary values associated to each piece, where $a_m = 1$ if the node n_i has piece d_m, and $a_m = 0$ otherwise. The goal of the prevalence vector is to give a local view of the prevalent pieces in the network. Initially, each node associates an empty prevalence vector to each content at each node.

For the sake of simplicity, assume that there is a single content c_0 to be disseminated in the network. Initially, all nodes in $\mathbf{N}$, except the one where c_0 was produced, have neither prevalence nor availability vectors associated to content c_0 because they are not aware of the presence of c_0 in the network. Nodes create these vectors as soon as they receive the availability vector relative to the new content c_0, $\mathbf{a}_{n_i,0}$, from a node n_i. When nodes n_i and n_j meet, they exchange their availability vectors $\mathbf{a}_{n_i,0}$ and $\mathbf{a}_{n_j,0}$. Node n_i (resp. n_j) computes $\mathbf{a}_{n_i,0} \wedge (\neg \mathbf{a}_{n_j,0})$ (resp. $\mathbf{a}_{n_j,0} \wedge (\neg \mathbf{a}_{n_i,0})$), which gives the candidate pieces to be transferred. They also update their prevalence vectors respectively as: $\mathbf{p}_{n_i,0} \leftarrow \mathbf{p}_{n_i,0} + \mathbf{a}_{n_j,0}$, and $\mathbf{p}_{n_j,0} \leftarrow \mathbf{p}_{n_j,0} + \mathbf{a}_{n_i,0}$. Among the candidate pieces to be transferred, nodes select the one with the lowest prevalence. In the case of a tie, a piece is chosen in a uniformly distributed random way. Let $d_{s_{i\to j}}$ be the piece sent by n_i to n_j and $d_{s_{j\to i}}$ be the piece sent by n_j to n_i. After one round of exchanges, nodes update their availability vectors as: $\mathbf{a}_{n_i,0} \leftarrow \mathbf{a}_{n_i,0} \vee \mathbf{i}_{d_{s_{j\to i}}}$, and $\mathbf{a}_{n_j,0} \leftarrow \mathbf{a}_{n_j,0} \vee \mathbf{i}_{d_{s_{i\to j}}}$, where $\mathbf{i}_{d_{s_{i\to j}}}$ and $\mathbf{i}_{d_{s_{j\to i}}}$ are K_0 element vectors with all positions set to 0 except the position relative to the piece just received, which is set to 1. Note that prevalence vectors have a limited influence at the beginning, but they gain importance as nodes move and exchange pieces. Availability vectors are broadcast while pieces transmissions are unicast. When n nodes meet, each one broadcasts an availability vector and $n - 1$ sessions per node are opened to exchange pieces.