QuadScatter for Simultaneous Transmissions in a Large-Scale Backscatter Network

Ousmane Zeba, Shunsuke Saruwatari and Takashi Watanabe
Graduate School of Information Science and Technology
Osaka University, Japan

***Abstract*—Backscatter communication is attracting attention for its ultra-low-power consumption ability. The said capability will enhance IoT which aims to enable a large number of novel applications for object-to-object communication. In such a network, where numerous tags may transmit simultaneously, it is important to make sure that the number of simultaneous transmissions allows reliable decoding operations on a large scale. Here we introduce QuadScatter, a series of algorithms that make profit of the maximum number of simultaneous backscatter transmissions to enable a network of a significant number of transmitters, tags, and readers. The maximum number of simultaneous transmissions is verified by simulations and the overall capacity of the network allows successful communications above 3 dB of signal-to-noise ratio (SNR). QuadScatter shows agreeable results compared to the exhaustive search algorithm. The simulation results highlight computational time and simultaneous transmission improvements of 250x and 2x. Furthermore, while the exhaustive search is limited to a few nodes(< 20), our proposal uses a larger number of nodes.**

***Keywords*—** Simultaneous backscatter transmissions, quadtree, quadtree quadrant, nearest neighbor search, exhaustive search, IoT

I. Introduction

The backscatter device is attracting attention because of its ultra-low-power wireless communication capabilities. The ability of power consumption below the milliwatt is significant for the upcoming IoT technology. IoT aims to enable a large number of novel applications, where objects will be capable of responding to the presence of people and other objects.

In a typical bistatic backscatter system design [1], the backscatter tag receives an excitation signal from the transmitter and reflects it to the receiver or reader, without generating or amplifying any signal. Numerous studies such as [2] and [3], showing significant results, have been conducted in the case of concurrent transmissions from several backscatter tags to the same reader.

The impending IoT technology will need to make to benefit from the case of simultaneous backscatter transmissions where several backscatter devices transmit their data to different readers at the same time in the same environment. The reason behind that is if a large number of tags are deployed in the presence of an equivalent number of transmitters and receivers, simultaneous transmissions will occur inevitably. In addition, selecting the best threesome source-tag-reader can agreeably improve the capacity of the network.

In [4] the detection problem of multiple passive and semi-passive RFID tags with impulsive backscattered signals is addressed in an ultrawideband (UWB) technology. The goal was to improve the reader capability and enable robust tag detection, even in the presence of multi-tag interference and clock drift effects.

Choir is presented in [2] to overcome the challenges related to density and transmission range in urban Low-Power Wide Area Networks (LP-WANs). Choir allows a maximum of 10 concurrent transmissions and is capable of retrieving data from devices located as far as 2.65km. This performance is achieved by applying chirp spread spectrum modulation [5].

In a recent work [3], a large scale backscatter network named NetScatter is presented. NetScatter is a wireless protocol that enables a network of 256 backscatters for concurrent transmissions on 500kHz channel bandwidth. The number of concurrent transmissions increases according to the bandwidth, reaching a thousand of concurrent transmissions for only 2MHz of bandwidth. All this is achieved by using a distributed chirp spread spectrum coding based on chirp spread spectrum modulation and ON-OFF keying. Furthermore, NetScatter is capable of operating with weak backscatter signals around noise floor. However, time synchronization is compulsory for NetScatter. The backscatter devices must be synchronized when sending to the access point. Besides, hardware delays such as clock frequency variations, inherent to backscatter devices, and propagation delay need to be corrected. The above-mentioned timing mismatch is difficult to avoid with a growing number of backscatter devices.

The above-cited publications analyze the multi-tag to one reader scenario. In other words, they are dealing with a network of many backscatter tags and only one receiver to which all the backscatter signals are directed. In contrast, we are investigating in this study the case of a network of numerous backscatter devices and an equivalent number of transmitters and receivers. The main constraint of multi-tag to reader transmissions is to increase the number of concurrent transmissions and communication range while keeping reliable decoding operations on the reader side. As for our case, dealing with numerous backscatters and numerous APs is challenging for two reasons.

1) Computational complexity. Finding the best AP for each tag among numerous devices can entail extremely high computational cost. To solve this issue we apply quadtree, a 2-D nearest neighbor search algorithm that reduces significantly the computational cost.

2) Simultaneous communications. In the current scenario we do not have concurrent transmissions like in [2] and [3]. However, multiple backscatter tag-AP pairs may transmit simultaneously. In that case, knowing the upper bound of the number of simultaneous communications helps in increasing the overall capacity of the whole network by allowing the maximum simultaneous transmissions. To this end, and for this first study, we numerically determined the upper limit of simultaneous transmissions for a given signal-to-noise-ratio threshold of 3dB.
3) Supported number of network nodes. As for now, only the exhaustive search algorithm can be used in such situation. However, it has a serious handicap regarding the network size. The exhaustive search algorithm deals only with a network of very few number of nodes.

The main contribution of this work is a series of algorithms that enable simultaneous transmissions in a large scale backscatter network resulting in a reduced computational time and cost, and an increased network size compared to the exhaustive search algorithm. Simulation results that verify our conclusions are also presented. The paper is organized as follows: Section II states the problem of simultaneous transmissions in a large scale backscatter network; Section III describes the algorithms designed for this study; Section IV focuses on the numerical analysis of simultaneous transmissions. Section V presents simulation results while the conclusion follows in Section VI.

II. Problem Statement

Imagine a scenario where n backscatter tags and m APs are deployed in a given area. Backscatter tags are passive devices that only reflect [1] incident signals from their surrounding environment. APs can generate and transmit their signals and receive them from other transmitting devices. In our case, the tags will use the received signal from APs to transmit their own collected data. Backscatter tags have only one task: receive signals from sources and reflect them to the desired destinations. The task of APs is different when we have a big number of tags deployed with a limited number of APs. In addition to generating, transmitting and receiving signals, each AP has to handle many tags. It may alternately be transmitter and receiver for many tags or receiver for many tags. There is also the possibility of being a transmitter for many tags. However, that case does not demand too much to the AP, as being a transmitter for one or many makes no difference in terms of power.

One of the most important parameters in a network is its transmission capacity. A major factor of the capacity according to Shannon's law is the signal-to-interference-plus-noise ratio (SINR) [6] [7], which is equivalently replaced in this paper by the signal-to-interference ratio (SIR), neglecting the ambient noise for the sake of simplicity. The better the SIR is, the better the capacity will be. Unfortunately, the SIR is subject to the distance between the transmitter and receiver and the interference occurring at the receiver. Obtaining a high SIR, which gives, in turn, an interesting and reliable transmission capacity, requires to avoid or mitigate the interfering signals and reduce the distance between transceivers.

In a network formed by backscatter tags and APs, only the APs are capable of receiving signals. In reality, those APs receive more signals than the signal intended to them. An AP in such a case, in addition to the inbound signal, receives interfering signals from other tags in its immediate neighborhood and the transmitter AP as illustrated. Assuming that the signal reflected by the backscatter has a slightly different frequency due to the impedance of the backscatter device, we can consider the signals from other backscatters as the only interference in addition to the ambient noise. Figure 1 gives an illustration of the devices and the different signals.

According to [8], considering only the signal from the tag to the reader we can write

$$SIR_{y|n} = \frac{\tau P_r d_{(z,y)}^{-\beta}}{\sum_i^n \tau P_{\dot{r}} d_{(\dot{z}_i,y)}^{-\beta}}, \tag{1}$$

where P_r is the received power at the backscatter tag and τ is the factor introduced by the backscatter tag ($0 < \tau < 1$) and $\beta = 2$ to 4 is a propagation constant. $P_{\dot{r}}$ and $d_{(\dot{z}_i,y)}$ are related to other tags that are source of interference. P_E being the transmit power from the source transmitter and $d_{(x,z)}$ the distance between source and tag, we have

$$P_r = \frac{P_E}{d_{(x,z)}^{\beta}}. \tag{2}$$

Hence eq. 1 becomes

$$SIR_{y|n} = \frac{\tau P_E d_{(x,z)}^{-\beta} d_{(z,y)}^{-\beta}}{\sum_i^n \tau P_{\dot{E}_i} d_{(\dot{x}_i,\dot{z}_i)}^{-\beta} d_{(\dot{z}_i,y)}^{-\beta}}. \tag{3}$$

Assuming that all the sources are equal in power leads to writing eq. 3 as

$$SIR_{y|n} = \frac{d_{(x,z)}^{-\beta} d_{(z,y)}^{-\beta}}{\sum_i^n d_{(\dot{x}_i,\dot{z}_i)}^{-\beta} d_{(\dot{z}_i,y)}^{-\beta}}, \tag{4}$$

which shows that the SIR depends only on the distances between the nodes involved in the transmission and interfering signal.

With the above-mentioned issues and the given parameters, how can we maximize the SIR of the receiver to improve the transmission capacity of the network? Some works have already addressed the issue of interference mitigation such as [9]–[11]. We opted to treat that issue in a simple but different manner as follows:

1) First, we built an assignment or selection algorithm to optimize the received power on the reader's side. The received power depends on the distance between the sender and receiver, assuming that all the carrier signals are emitted with the same power. Thus, if the nearest transmitter and the nearest in the vicinity of the tag can be selected it would be easier to obtain a

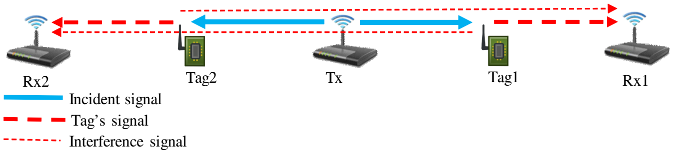

Fig. 1. Simultaneous backscatter transmissions: Signals and interferences.

maximized received power. One has to bear in mind that the backscatter tag is not able to select a carrier signal nor to decide which reader should receive the reflected signal. For that reason, there is a controller that controls all the Network components.

2) Second, we tried to find the maximum number of simultaneous backscatter transmissions that can be tolerated by the reader. By simultaneous transmissions, we mean many tags reflecting signals to different readers at the same time. The signal from other tags will create interference to the inbound signal to each tag. According to the SIR threshold for successful decoding of the signal, there will be a limit to the acceptable interference. Which corresponds to a certain number of interfering signals and backscatter devices.

III. Proposed Algorithm

This section describes our proposed algorithm to maximize the transmission capacity in a backscatter network. Two cases have been distinguished in tackling this issue: before the deployment of the network and after the network is deployed. In this work, we only focus on an already deployed network.

In an already deployed network, it is hard to change the topology to a suitable one without redeploying the whole network, which would cost much effort and time. To avoid redeployment while trying to improve the transmission capacity of the network, we just apply the AP selection technique and limit the number of simultaneous communications to the maximum allowed number.

The AP selection algorithm looks for APs (transmitter and receiver) that suit the most to each tag in terms of distance. It just picks the two closest APs to the tag. For a huge number of tags and APs, finding the nearest neighbor can become NP-hard due to the computational complexity. For n number of tags and m, finding two nearest neighbors of each tag gives the complexity of $\mathcal{O}(2^n)$ by making a naive comparison with the exhaustive search algorithm. The high computational cost is attributable to the process of finding the most appropriate AP. Therefore, a nearest neighbor search algorithm will considerably reduce the computing cost.

1) Nearest Neighbor Search: There exist many algorithms for nearest neighbor search. Among them are Locality Sensitive Hashing (LSH) and quadtree.

LSH is an algorithmic technique that creates a bucket of input items with a high probability of similarity using hash functions [12]. Since it separates similar items from dissimilar ones, LSH can be used for the nearest neighbor search. However, LSH is more convenient in high dimensional spaces, which is different from our two-dimensional space. For example, the Python implementation of LSH in [13] is only suitable for data samples with high dimensionality.

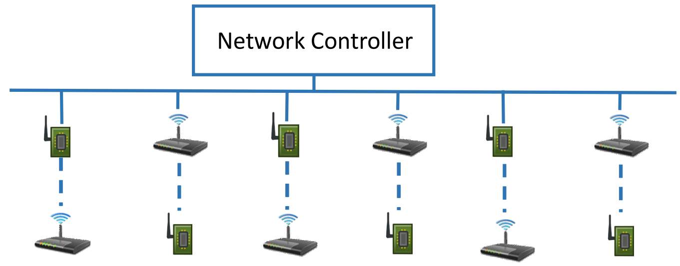

Fig. 2. Backscatter controlled network.

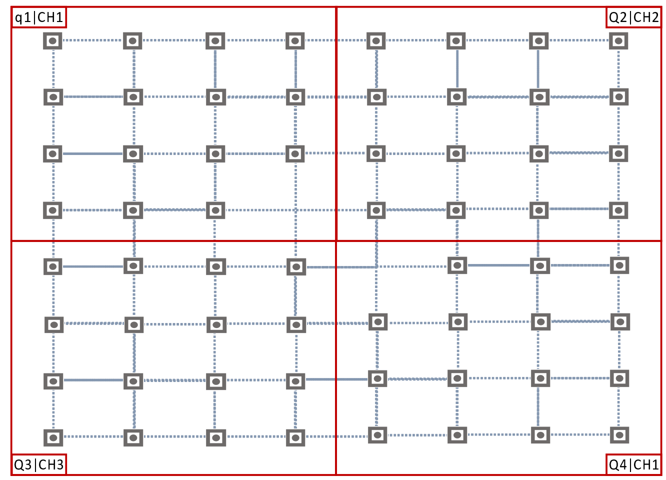

Fig. 3. Backscatter network and quad tree quadrants: Each quadrant Q_n is assigned a channel CH_k orthogonal to its immediate neighbors' channels.

Quadtree is an algorithmic technique used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions [14]. The subdivided regions may have arbitrary shapes. Quadtree decomposes the space into buckets. Each bucket has a maximum capacity. When the maximum capacity is reached, the bucket divides into four quadrants. The quadtree algorithm is interesting for our situation in the sense that it is conceived for two-dimensional space. For that reason, quadtree drove our attention and convinced us to combine it with our assignment algorithm. Furthermore, it has a worse case complexity of $\mathcal{O}(n)$, which is significantly lower than the complexity of exhaustive search algorithm.

A. QuadScatter

QuadScatter is the algorithm we define for assigning APs to backscatter tags. Mainly four phases compose the *QuadScatter* algorithm: Apply quadtree to create quadrants, nearest neighbor selection phase, common transmitter search phase, and transmitter-receiver assignment phase.

1) Define and create an adequate number of quadrants: This phase is a crucial step in the whole process as it subdivides the target area into quadrants allowing an easier localization of nodes. An implementation of a quadtree allows deciding the maximum number of nodes in each quadrant after receiving the coordinates of the targeted area. It is also possible to limit the number of divisions by giving the maximum range of divisions. This phase of *QuadScatter* receives the area size and the coordinates of nodes deployed on it as an input, Algorithm 1 line 2. As an output, it gives the resulting quadrants and the points associated with each of them, Algorithm 1 line 3.

Algorithm 1: CREATEQUADRANT creates quadrants in area A

Input: A two-dimensional area A and the coordinates of its network nodes
Output: A finite set $Q = \{q_1, q_2, \ldots, q_n\}$ of quadrants

1 $Q \leftarrow \emptyset$
2 $Quadtree(A)$
3 $Q \leftarrow \{q_1, q_2, \ldots, q_n\}$
4 **return** Q

2) Nearest neighbor selection: Instead of randomly selecting the transmitters and readers, a thorough search of nearest APs would give a better result as the SIR and capacity depend on the distances separating the two nodes.

As already mentioned above in section III-1, we apply quadtree to the area to lower the computational complexity of our proposed algorithm. The quadrants or regions resulting from quadtree will contain a certain number of devices that we address with very little complexity. For a given quadrant, containing m tags and n APs, algorithm 2 compares the distances separating tags and APs, in lines 3 to 6. After that, it returns an array of the nearest APs to each tag, algorithm 2 line 7. In that array, each line corresponds to a tag in the quadrant. The same line is a sorted list of APs where the first AP is the nearest one.

Algorithm 2: FINDNEAREST finds the nearest APs in the quadrant

Input: A quadrant q of a given area and its points
Output: An array A of nearest APs to each tag

1 $A \leftarrow \emptyset$
2 $list \leftarrow \emptyset$
3 **for** $i \leftarrow 1$ ***to*** m **do**
4   **for** $j \leftarrow 1$ ***to*** n **do**
5   compute distance d_{ij}
6         $list \leftarrow d_{ij}$
7   sort $list$
8     $A \leftarrow list$
9 **return** A

3) Common transmitter search and assignment: The main objective of this stage is to avoid the usage of many transmitters while only one suffices for many backscatter tags. This phase named 3 works as described in the following lines. Algorithm 3 picks the first element of the array returned by algorithm 2 and compares it to the elements of the following line, lines 7 to 8 in algorithm 3. In case it matches any element in that line it jumps to the next line, line 9 of algorithm 3. In case no element of the current line matches, algorithm 3 goes back to the following element in the first line of the array, as described in lines 11 and 12. After any line is finished, algorithm 3 makes sure to always start with the first element of the next line as shown in line 14. After running through all the lines of the array, algorithm 3 stops looking for the common element, lines 15 and 16.

Algorithm 3: FINDCOMMON finds the common element to the lines of an array

Input: An array of n columns and m lines
Output: The common element to all the lines

1 $i \leftarrow 0$
2 $j \leftarrow 0$
3 $line \leftarrow 0$
4 $common \leftarrow \emptyset$
5 **for** $k \leftarrow 1$ ***to*** $n-1$ **do**
6   **while** $i \leq n$ ***and*** $j \leq m$ **do**
7         $common \leftarrow A_{0,k}$
8   **if** $common \neq A_{i,j}$ **then**
9             $j \leftarrow j+1$
10   **else**
11             $i \leftarrow i+1$
12             $j \leftarrow 0$
13             $line \leftarrow i$
14     $j \leftarrow 0$
15   **if** $line = m$ **then**
16   **break**
17 **return** $common$

4) Transmitter and receiver assignment: At this stage, the appropriate tags and APs are assigned to each other by the network controller. This process completes the backscatter network and enables communications between network nodes as described in algorithm 4. Algorithm 4 takes a set of APs and tags associated with one quadrant as an input. First of all, it makes sure that a common AP is already found or not. If a common AP exists, that AP will be set as the transmitter for all the tags in the quadrant. Once the common transmitter is designated, the nearest AP to each tag other than the common AP is set as receiver for the appropriate tag, as illustrated in lines 1 to 8 of Algorithm 4. In the case of no common AP, the nearest AP to each tag is taken as a receiver and the second one as a transmitter. The latter case is described in lines 9 to 14 of algorithm 4.

IV. MATHEMATICAL ANALYSIS

Equation 4 gives the SIR in the theoretical analysis of the capacity of backscatter communication as presented in [8]. Even though this formulation is the appropriate one for

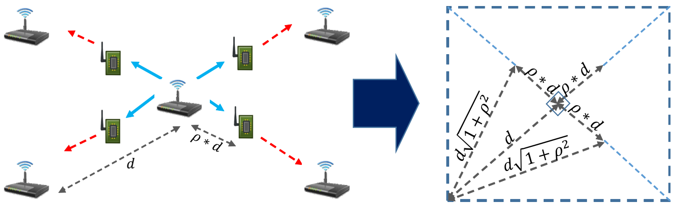

Fig. 4. Simultaneous transmissions and formulation of the distances: The distance is evaluated following the path of the signal from the transmitter to the receiver

the current situation, it might become arduous to manipulate when dealing with a large number of devices, due to the product of powered distances $d^{\beta}_{(x,z)}$ et $d^{\beta}_{(z,y)}$ and the sum $\sum_i d^{\beta}_{(\dot{x}_i,\dot{z}_i)} d^{\beta}_{(\dot{z}_i,y)}$. For the sake of simplicity and to reduce the heaver of mathematical manipulations, we consider a slightly different formulation. That is, as shown in Eq. 4, the factor τ induced by the backscatter disappeared as if the backscatter itself was no more involved in the transmission. Thus, we take into account the distance from the source x to the reader y as if it was a straight line instead of distances from the source to tag and from tag to the reader. As a result, we obtain a simpler and mathematically convenient to manipulate version of the SIR.

$$SIR_{y|n} = \frac{\left[d_{(x,z)} + d_{(z,y)}\right]^{-\beta}}{\sum_i \left[d_{(\dot{x}_i,\dot{z}_i)} + d_{(\dot{z}_i,y)}\right]^{-\beta}}. \quad (5)$$

Algorithm 4: ASSIGN assigns APs to tags

Input: A finite set A of n APs and m backscatter tags
Output: Selected and assigned APs and tags

```
if common ≠ ∅ then
    common ← transmitter
    for i ← 1 to m do
        for j ← 1 to n do
            if A_{i,j} ≠ transmitter then
                A_{i,j} ← receiver
                A_{i,j} ← tag_i
                break
else
    for i ← 1 to m do
        A_{i,0} ← receiver
        A_{i,1} ← transmitter
        A_{i,0} ← tag_i
        A_{i,1} ← tag_i
return 0
```

1) Simultaneous communications: According to Eq. 5 the SIR of 4 simultaneous backscatter communications as illustrated in Fig. 4 is given as follows, considering one source as transmitter and following the distances derived in Fig. 4.

$$SIR_{y|4} = \frac{d^{-\beta}}{2(\rho d + d\sqrt{1+\rho^2})^{-\beta} + (2\rho d + d)^{-\beta}}. \quad (6)$$

Since the propagation constant $\beta = 2$, obtain the following simplified version of Eq. 6

$$SIR_{y|4} = \frac{1}{\frac{1}{(1+2\rho)^2} + \frac{2}{(\sqrt{1+\rho^2}+\rho)^2}}. \quad (7)$$

Then while looking for the upper bound of $SIR_{y|4}$ we found the following. The details are given in the appendix section of this paper.

$$SIR_{y|4} < \frac{8\rho(3+2\sqrt{2})}{16\rho + (3+2\sqrt{2})}, \quad (8)$$

with $\rho_{y|4} > \frac{3+2\sqrt{2}}{8\sqrt{2}-3} \approx 0.7945$.

For five simultaneous communications, in the similar conditions as explained above, the SIR appears to be:

$$SIR_{y|5} = \frac{1}{\frac{2}{\left(\sqrt{1-\rho\frac{\sqrt{5}-1}{2}+\rho^2}+\rho\right)^2} + \frac{2}{\left(\sqrt{1+\rho\frac{\sqrt{10-2\sqrt{5}}}{2}+\rho^2}+\rho\right)^2}}. \quad (9)$$

For a better understanding of $SIR_{y|5}$ in Eq.5 we compute its upper bound:

$$SIR_{y|5} < \frac{(\rho+\sqrt{2})^2(4+2\sqrt{3})}{8+2\sqrt{2}+2(\rho+\sqrt{2})^2}. \quad (10)$$

It yields $\rho_{y|5} > \sqrt{\frac{6+8\sqrt{3}}{3}} - \sqrt{2} \approx 1.158$ for a minimum SIR of 3dB. Which clearly suggests the limit of simultaneous backscatter communication.

One must keep in mind that we are using an upper bound estimation ρ. Normally the real values are supposed to be inferior to the given values. Even though $\rho_{|5}$ appears to be greater than 1, at least five simultaneous communications can be considered possible, which is verified by the simulations results of *QuadScatter* in section V.

2) Quadrant channel assignment: One issue that might be encountered with *QuadScatter* is the inter-quadrant interferences. In fact, a node located in a given quadrant might suffer from the interferences from nodes located in its neighboring quadrants. To solve that issue we propose to alternately assign different channels to the quadrants as shown in Fig. 3. Considering the 2.4GHz band and its 3 orthogonal channels, the channels will be assigned so that a quadrant and its two immediate neighbors use the three orthogonal channels.

V. EVALUATION

This section gives the evaluation results of our proposed method. We compare the number of simultaneous transmissions and the computational time of our algorithm named *QuadScatter* to those of the exhaustive search or brute force algorithm. We simulated a network of randomly deployed tags

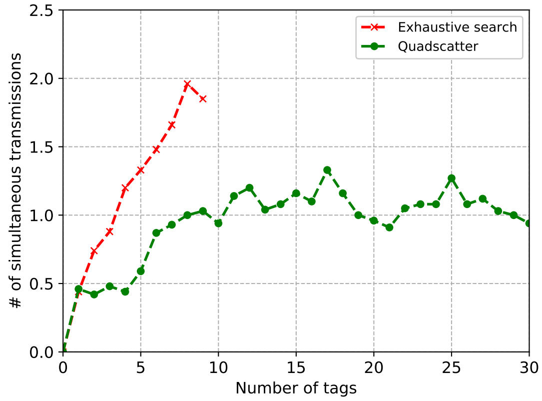

Fig. 5. Average number of simultaneous transmissions in random 30x30 area with 6 APs.

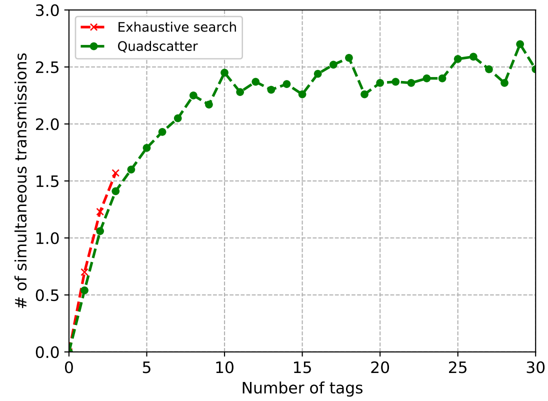

Fig. 7. Average number of simultaneous transmissions in random 30x30 area with 12 APs.

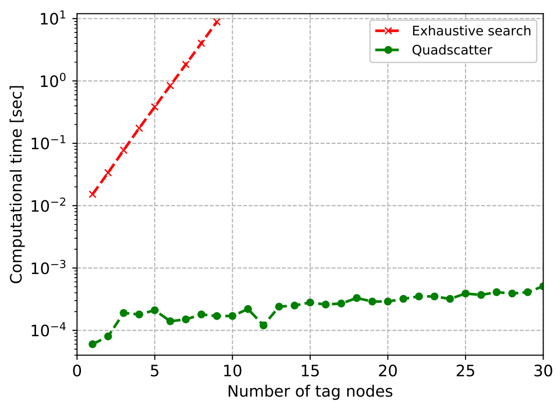

Fig. 6. Computational time of exhaustive search and *QuadScatter* in random 30x30 area with 6 APs.

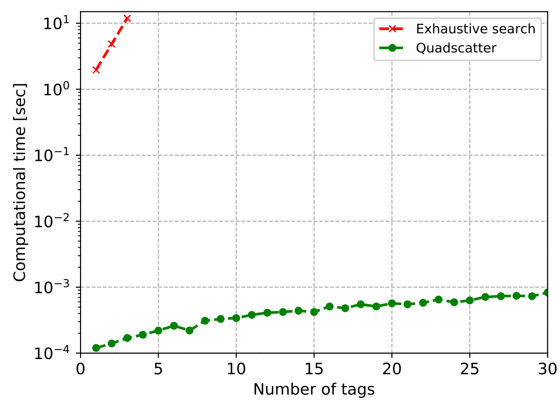

Fig. 8. Computational time of exhaustive search and *QuadScatter* in random 30x30 area with 12 APs.

and AP nodes on a 30x30 area. The data collected are the average values from one hundred different random topologies. Meaning that the program is executed one hundred times and at each time a different random topology of APs and tags is created. While the number of APs is kept unchanged, the number of tags is incremented by one after one hundred trials until the desired number of tags is reached. We conducted the simulations in three cases. Case 1 with only 6 APs, case 2 with 12 APs and case 3 with growing number of APs. 6 is the minimum number of APs that could give 5 simultaneous transmissions following our numerical study in section IV.

The simulation parameters are set as follows: Transmit power $= 20\text{dBm}$ equivalent to 100mW, sensitivity $= -94\text{dB}$, noise floor $= -97\text{dB}$, SNR threshold $= 3\text{dB}$, path loss exponent $\beta = 2$.

As for the results, we found that Exhaustive search reaches a maximum of 3 simultaneous transmissions whereas *QuadScatter* outputs a maximum of 7 simultaneous transmissions, in some occurrences, outperforming exhaustive search and confirming our theoretical analysis. Hence, if the suitable topology is found *QuadScatter* will deliver at least the double of the performance of exhaustive search. Note again that, the values shown by Fig.5, Fig.7 and Fig.9 are average values of 100 trials, which is more realistic for a random deployment scenario. Figure 5 shows a better performance of exhaustive search over *QuadScatter*. However, that performance is limited to a network of few nodes (< 20). As illustrated in Fig.6, the computational cost of exhaustive search does not comply with a large scale network. When the network size grows, Fig. 7, *QuadScatter* renders agreeable performance in terms of simultaneous transmission while keeping an ultra-low computational cost, Fig.8. In Fig. 9 and Fig.10 *QuadScatter* shows its ability of handling a network of a hundred of nodes, delivering around 5 simultaneous transmissions with a computational time lower than 0.01 sec. *QuadScatter* suitable to a network of numerous nodes while exhaustive search deals only with a network of a restrained number of nodes without time constraint. Naturally, exhaustive search does not go beyond 10 backscatter tags and its computational time extends considerably with an increased number of APs. For that reason, when the number of APs

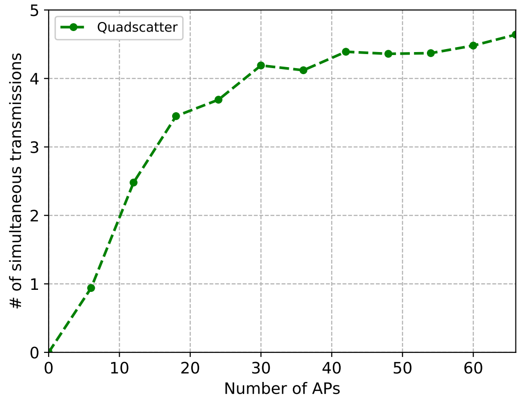

Fig. 9. Average number of simultaneous transmissions in random 30x30 area with 30 tags.

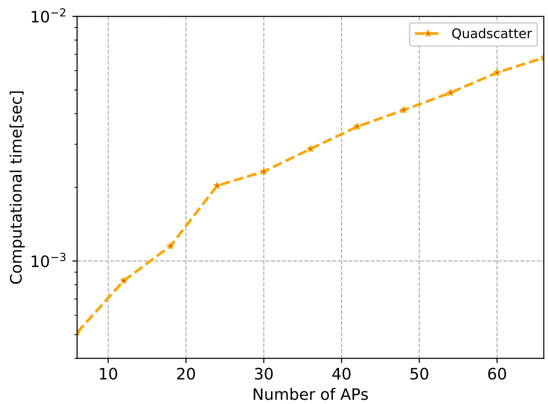

Fig. 10. Computational time of *QuadScatter* in random 30x30 area with 30 tags.

increased, case 2 and case 3, we deliberately restrained the number of tag nodes for exhaustive search to avoid running the experiment during many days.

In conclusion, 3 major improvement points are brought by *QuadScatter* compared to exhaustive search. An increased number of simultaneous backscatter transmissions and backscatter nodes in a controlled network, and significantly faster computational time, *QuadScatter* is at least 250 times faster than exhaustive search.

VI. Conclusion

In this work, we analyzed computational cost and simultaneous transmissions in a large backscatter network. An algorithm was developed to improve the performance of the whole network. Our results show significant improvements compared to the exhaustive search algorithm. Future work will focus on increasing the number of simultaneous transmissions.

Appendix

Upper bound proof of $\rho_{y|4}$ and $\rho_{y|5}$ in section IV.

$0 < \rho < 1$ gives $1 + 4\rho < (1+2\rho)^2 < 8\rho$ which gives $\frac{1}{8\rho} < \frac{1}{(1+2\rho)^2}$. $0 < \rho < 1$ also gives $1 < (\sqrt{1+\rho^2}+\rho)^2 < (\sqrt{2}+\rho)^2$, which gives $\frac{2}{(\sqrt{2}+\rho)^2} < \frac{2}{(\sqrt{1+\rho^2}+\rho)^2}$ and $\frac{2}{3+2\sqrt{2}} < \frac{2}{(\sqrt{2}+\rho)^2}$. By adding $\frac{2}{3+2\sqrt{2}}$ to $\frac{1}{8\rho}$ we get the lower bound of $\frac{1}{\rho_{y|4}} = \frac{1}{(1+2\rho)^2} + \frac{2}{(\sqrt{1+\rho^2}+\rho)^2}$. Then by taking the inverse of that sum, we obtain the upper bound $\frac{8\rho(3+2\sqrt{2})}{16\rho+(3+2\sqrt{2})}$ of $\rho_{y|4}$. The upper bound of $\rho_{y|5}$ is computed in the same way.

References

[1] C. Xu, L. Yang, and P. Zhang, "Practical Backscatter Communication Systems for Battery-Free Internet of Things: A tutorial and survey of recent research," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 16–27, 2018.

[2] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering Low-Power Wide Area Networks in Urban Settings," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 309–321.

[3] M. Hessar, A. Najafi, and S. Gollakota, "NetScatter: Enabling Large-Scale Backscatter Networks." in *NSDI*, 2019, pp. 271–284.

[4] F. Guidi, N. Decarli, S. Bartoletti, A. Conti, and D. Dardari, "Detection of Multiple Tags Based on Impulsive Backscattered Signals," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 3918–3930, 2014.

[5] B. Reynders and S. Pollin, "Chirp Spread Spectrum as a Modulation Technique for Long Range Communication," in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*. IEEE, 2016, pp. 1–5.

[6] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[7] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[8] A. V. Padaki and M. J. Zawodniok, "Theoretical Capacity Analysis for Multi-hop Backscatter Communication Networks," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2011, pp. 1–6.

[9] W. Liu, K. Huang, X. Zhou, and S. Durrani, "Full-duplex Backscatter Interference Networks Based on Time-hopping Spread Spectrum," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4361–4377, 2017.

[10] M. R. Souryal, D. R. Novotny, D. G. Kuester, J. R. Guerrieri, and K. A. Remley, "Impact of RF Interference between a Passive RFID System and a Frequency Hopping Communications System in the 900 MHz ISM Band," *IEEE Electromagnetic Compatibility Magazine*, vol. 1, no. 3, pp. 97–102, 2012.

[11] A. Varshney, O. Harms, C. Pérez-Penichet, C. Rohner, F. Hermans, and T. Voigt, "LoRea: A Backscatter Architecture that Achieves a Long Communication Range," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM, 2017, p. 18.

[12] Wikipedia contributors, "Locality-sensitive hashing — Wikipedia, The Free Encyclopedia," https://en.wikipedia.org/w/index.php?title=Locality-sensitive_hashing&oldid=905340247, 2019, [Online; accessed 16-August-2019].

[13] Avinash Kak, "LocalitySensitiveHashing — LocalitySensitiveHashing.py version: 1.0.1," https://engineering.purdue.edu/kak/distLSH/LocalitySensitiveHashing-1.0.1.html, 2017-May-25, [Online; accessed 8-October-2019].

[14] Wikipedia contributors, "Quadtree — Wikipedia, The Free Encyclopedia," https://en.wikipedia.org/w/index.php?title=Quadtree&oldid=910044683, 2019, [Online; accessed 16-August-2019].