Efficient Neighbor Discovery in RFID Based Devices Over Resource-Constrained DTN Networks

Danilo Amendola
DIET
Univ. of Rome "La Sapienza"
Rome, Italy
danilo.amendola@uniroma1.it

Floriano De Rango
DIMES
University of Calabria
Rende (CS), Italy
derango@dimes.unical.it

Khalil Massri
DIAG
Univ. of Rome "La Sapienza"
Rome, Italy
massri@dis.uniroma1.it

Andrea Vitaletti
DIAG
Univ. of Rome "La Sapienza"
Rome, Italy
vitaletti@dis.uniroma1.it

***Abstract*—In this paper we consider Delay Tolerant Network (DTN) as a technology to implement a future network in a People Centric Networking paradigm, using Active RFID carried by people that exchange information with each other. We propose a novel and real Neighbor Discovery (ND) phase on active RFID based DTN using Open Beacon devices. In particular, we propose a solution using the Sift distribution on a probabilistic persistent approach called *Sift-Persistent*. We simulated *P-Persistent*, *Aloha* and our solution using our customized Java simulator. We implemented *Sift-Persistent* and *P-Persistent* on Open-Beacon devices, comparing the simulation results and test-beds. Moreover, simulations and real testbed show a coherent behavior validating our proposal in the RFID context. Performance evaluations have been tested in terms of discovered neighbors.**

Keywords: DTN, RFID, Ad Hoc, Neighbour Discovery, IoT.

I. Introduction

In recent years research activity on ad-hoc routing and the increasing use of wireless protocols, witch evolved in the 1990s, when *Mobile Ad-hoc NETworking (MANET)* became the focus of increasing attention, has resulted in more advanced communications protocols and environments, such as routing protocols for mobile scenarios [1] and VANETs [2]. Due to the distributed and mobile nature of MANET, frequent disconnections in topology can be experienced leading to low performing traditional protocol developed for ad hoc networks. With this aim, a more recent paradigm that allows a hop-by-hop control, tolerable to disruption in network topology has been considered. This technology is called Delay Tolerant Network (DTN) and was initially designed for very long delay and intermittent network such InterPlanetary Networks (IPN) [3]. Later on, introducing many optimizations in terms of Quality of Service (QoS) and dissemination techniques DTN were also extended to classic MANET and VANET [4] becoming an attractive technology for both scientists and industry.

Delay-Tolerant Network (DTN) is a wireless network paradigm allowing communications among intermittently connected (and/or subject to packet disruptions) networks. In [5] the author proposed a reference architecture for DTN routing protocols and a thorough quantitative evaluation and a new taxonomy of many protocols proposed in the literature.

In DTN there are no end-to-end paths, and we do not know how many times the message needs to be delivered to the destination node. The messages will be routed through the neighbor nodes to the destination. A continuous communication in wireless systems can not always be available nor reliable. Also, devices composing a network might move and consequently, the paths connecting the nodes continuously change. Due to all these issues, the ability of ensuring message delivery in networks characterized by probable lacks of continuous communications, is becoming more and more relevant. The recent work [6] attempts to afford the problem from an interesting structured point of view, taking into account also for the need of network coding and distributed source coding strategies. However, in order to overcome this inherent difficulty of opportunistic networking, intermediate nodes, apart from relaying messages, are also willing to locally store the messages and forward them at a later opportunity.

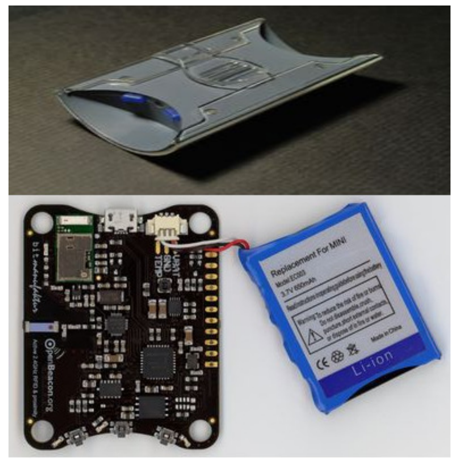


Fig. 1: The OpenBeacon Active RFID device.

The simplest DTN mechanism works by storing messages and flooding them in the network when nodes opportunistically meet [7]. More sophisticated routing protocols provide more efficient algorithms to avoid or limit the redundancy in terms of message copies in the network and optimize metrics such as the *delivery delay* and *delivery ratio*.

In [8] the authors also presented a full implementation schema on OpenBeacon RFID tags [9] of the reference architecture for DTN routing proposed in [5]. In particular, all the three techniques in [5], namely forwarding (FW), replication (R) and queue management (QM) were implemented on OpenBeacon-usb2 devices (see Fig 1) that are equipped with 32KB of flash for code and 8KB SRAM for data memory. *Queue management* orders and manages the messages in the node's buffer, *forwarding* selects the messages to be delivered when there is a contact and, finally, *replication* bounds the number of message replicas in the network.

The most fundamental primitive to implement any DTN routing protocol is the Neighbor Discovery process (ND). This primitive is fundamental for two main reasons: 1) it allows selection of nodes to forward the current message to, 2) it allows for the acquisition and exchange of the necessary knowledge to support the QM, FW and R techniques. However, in their implementation of the ND phase they used the traditional *P-Persistent* CSMA algorithm (discussed in section III-B), where they empirically adjusted its parameters.

For our purpose we identified the OpenBeacon project [9]. The OpenBeacon project was founded in 2006 as an open platform for active RFID applications operating in the license free 2.4GHz ISM band. OpenBeacon is based on Open Source software and a very flexible and re-programmable low cost Open Source RF-module. The firmware sources and hardware schemas are available under GPL license. We developed our ideas for OpenBeacon platform.

In this paper we aim to improve the implementation of the ND phase using OpenBeacon devices. In particular, we propose a solution using the Sift distribution [10] on probabilistic persistent approach called *Sift-persistent*. We simulated *P-Persistent*, *Aloha* and our solution using our customized Java simulator. We implemented *Sift-Persistent* and *P-Persistent* on OpenBeacon [9] devices, comparing the simulation results and testbeds. We envisioned the DTNs useful to implement a future network in a People Centric Networking paradigm, using Active RFID carried by people that exchange information with each other.

This document is organized as follows: in section II we present the identification problem on RFID Tag; in section III we briefly present related work and other solution for Neighbor Discovery (ND) on RFID; in section IV we present the neighbor discovery strategies on resource constraint devices under DTNs, and present our proposal: *Sift-Persistent*; in section V we present a performance evaluation on the simulation scenario and testbeds with the OpenBeacon devices.

II. RFID Tag identification problem

The RFID tags may be one of two types: Active or Passive. Active RFID tags have their own power source and our work was carried out on these kind of devices; the advantage of these tags is that the reader can be much farther away and still receive the signal. Even though some of these devices are built to have up to a 10 year life span, they do have limited life spans. Passive RFID tags, however, do not require batteries, and can be much smaller and have a virtually unlimited life span.

A. Reader-Tag networks

In normal *reader-tags* configuration, RFID applications require reliable, secure and fast communications. These requirements demand a tag identification process which is also reliable and fast for multiple objects, assuming that the exact amount of tags is unknown.

During a typical identification process, the reader broadcasts a message requesting the tags IDs or their stored data. Following the reception of this message, the tags send their response to the reader. On one hand, if only one tag responds, the reader will receive just one message which can be correctly decoded. On the other hand, if two or more tags respond simultaneously such responses will collide on the radio frequency (RF) channel [8]. This problem, known as tag collision or tag-tag collision, is one of the main research directions on RFID networks. While solving this problem, the goal is to optimize bandwidth, power consumption in the tags as well as the identification delay, the latter being one of the most important performance measures on RFID networks for reader-tags.

Besides tag collisions, there are also *reader-reader* collisions and *reader-tag* collisions. The former occurs when there is interference between the signals of two or more readers. The latter occurs when two or more readers want to communicate with the same tag.

In our case we configure a communication without any reader, we setup a network using only active tags, those tags exchange messages between them with a DTN communication mode. In some cases they could collect information and send it to a Sink node.

B. Tag-to-Tag networks

In our network topology, the nodes are strict resource constraint devices. Therefore, we take in consideration that the classic neighbor discovery protocol is not optimized for our topology network.

Tag-to-tag Neighbor Discovery: Our goal is represented by the implementation of a distributed network of strict resource constraint based devices (RFID) on Delay Tolerant Networking paradigm. There are no other works or implementations on RFID over DTNs, therefore the cited related works regarded classical *reader-to-tag* networks.

A node in our network (DTNs) consists of an OpenBeacon [9] device equipped with an embedded processor, a small amount of memory and transmitter/receiver circuitry. These resource constraint nodes are normally battery powered and they should coordinate among them selves to perform a common task. Instead the RFID, like the OpenBeacon, normally was used in a client-server network topology.

Our test is the first experiment that attempts to configure an RFID network without a reader (coordinator), to exchange information, self-organize the node itself, and capture the neighbor's information. This is the initial implementation of a DTN and ND protocol on RFID using OpenBeacon platform. All actual RFID neighbor discovery protocols are designed for a *reader-to-tag* topology, in our case we intend to implement a *tag-to-tag* network.

III. LITERATURE

There are many anti-collision solutions based on the ALOHA protocol, [11], for *reader-to-tag* RFID network. Here we present a technique for Neighbor Discovery in RFID, this solution is always based on a *reader-to-tag* network.

A. ALOHA-based protocols

ALOHA is the pioneer protocol for medium access control. There are currently several RFID protocols based on ALOHA, e.g., Pure ALOHA (PA), Slotted ALOHA (SA), Frame Slotted ALOHA (FSA), Basic Framed Slotted Aloha (BFSA), and Dynamic Framed Slotted Aloha (DFSA), Advanced Framed Slotted Aloha (AFSA). ALOHA protocols are widely used on RFID because most of the RFID standards are based on a modified version of FSA; some examples are EPC-Gen [12] and ISO 18000-7 [13].

a) Slotted Aloha (SA): The Slotted ALOHA is one of the simplest MAC protocol in literature, which introduced discrete time slots and increased the maximum throughput. A station can only send at the beginning of a time slot, and thus collisions are reduced. In this case, we only need to worry about the transmission-attempts within 1 frame-time and not 2 consecutive frame-times, since collisions can only occur during each time slot.

b) Framed Slotted ALOHA (FSA): The FSA is an improved version of the SA protocol which assumes that time is slotted and grouped in frames. A slot is a time interval where RFID tags may transmit their ID [14]. FSA makes several identification (read) cycles in order to identify all the tags inside the coverage range of a reader; each identification cycle lasts one frame. A frame is the time interval elapsed between reader requests; it is composed of a given number of slots. Tags are limited to one transmission per frame since in another case there would be several tag collisions. There are two different classes of FSA: *Basic Framed Slotted Aloha (BFSA)* where the frame size is fixed during the identification cycle. In BFSA the reader passes information to the tags about the identification cycle length. Each tag selects a time slot in a random fashion [15]. *Dynamic Framed Slotted Aloha (DFSA)* is where the frame size varies. DFSA implements an estimation function at the end of each identification cycle in order to adjust the size of subsequent frames. In this way, by using the feedback of the read cycle of the current frame, the number of tags that have not been identified yet is estimated [16].

c) Advanced Framed Slotted Aloha (AFSA): The AFSA algorithm estimates the number of tags and determines a proper frame size for the estimated number of tags and identifies tags using the determined frame size [17], [18]. AFSA algorithm uses an estimation function of the number of tags as a Chebyshev's inequality Equation. The number of tags is estimated using both the frame size used in the read cycle and the results of the previous read cycle.

B. CSMA-based protocols

CSMA based protocols are anti-collision systems that efficiently manage medium access control in RFID networks. Some CSMA-based mechanisms have been proposed in the literature for RFID identification in reader-tag configuration, such as Non-persistent CSMA [19], CSMA Multiphase approach, p-persistent CSMA [20].

d) Non-persistent CSMA: In this proposal the authors of [19] propose non-persistent CSMA as a collision resolution protocol for RFID networks. This proposal uses the distribution function proposed in [10], so that every tag chooses a time slot in a random fashion in a contention window in order to transmit its ID or its corresponding information.

A CSMA contention window is equivalent to an FSA frame on this latter scheme. The CSMA contention window size is kept constant. There are other works that implement a CSMA multi-phase approach, [21]. We consider this too expensive to implement in the OpenBeacon devices. We concentrate our work on a single-phase approach solution.

e) P-Persistent CSMA: P-Persistent CSMA is a slotted scheme; a station ready to transmit first senses the channel. If the channel is idle, the station transmits with probability p and delays its transmission until the next slot with probability $q = 1-p$. If the next slot is free, a transmission occurs or it is delayed with probabilities p and q, respectively. This process is repeated until the end of the contention window, or until the beginning of a new transmission by another station. In the last case, the protocol behaves as if a collision has occurred. If at the beginning of a transmission a station detects a busy channel, it waits until the next time slot and then it follows the algorithm just described [20].

IV. NEIGHBOR DISCOVERY STRATEGIES ON OPENBEACON RESOURCE CONSTRAINT DEVICES

The most fundamental primitive to implement any DTNs routing protocol is the Neighbor Discovery (ND). In this paper we focus our attention on this problem. We could consider the implementation based on [5] and [8], but, in this paper we did not implement a complete Delay Tolerant Network protocol, concentrating just on the Neighbor Discovery phase. The ND phase is fundamental for two main reasons: 1) it allows to select one of the neighbors to forward the messages to, 2) it allows to acquire and exchange knowledge to support the DTN protocol.

We designed and implemented a novel probabilistic neighbor discovery protocol based on *Sift distribution* [10] and called it *Sift-Persistent*. We assume that each tag is located in the communication range of all its neighbors, and that all other tags are in the same communication range.

To perform ND, a node goes through two phases, the ND Request(NDReq) phase and the ND Response(NDRes) phase, as described below:

ND-Req phase.: Periodically, node turns on its radio and listens for the channel for a randomly selected period of time T.

ND-Res phase.: When a node receives an NDReq packet, it starts the *NDRes phase* that lasts for W time-slots. In our experiments, we implemented the well known *P-Persistent*

CSMA protocol, according to [20], and compared it with *Sift-Persistent*.

Our intention is to obtain an effective mechanism for collision resolution for active RFID environments in which a number of tags has to reply to a common trigger from a RFID neighbor (i.e. an event like the NDReq packet) using a fixed window size.

P-Persistent: The probability of successfully completing the ND phase depends on: the selected probability p, the size of W and the number of neighbors answering the NDReq message N. Clearly, a larger W implies more probability of success, but also a higher latency to complete the procedure. Similarly, a larger P would, in principle, allow more nodes to answer, but it also increases the probability of collisions in a slot. The proper selection of those parameters, is an open research area [20], [8].

Sift-Persistent: Our proposal is a probabilistic persistent MAC protocol in which the probability threshold changes following the Sift Distribution function, [10]. We consider a simple network topology with a node requester in the center encircled by some responding nodes, the responding nodes number is initially between two and five, and then ten. The Requester starts an ND-Request phase therefore all neighbors begin the ND-Response phase and reply to the requester. This behavior is applied to *P-Persistent* and also to *Sift-Persistent*.

Our proposal uses the distribution function proposed in [10], so that every tag chooses a time slot in a random fashion in a fixed contention window in order to transmit its ID or its corresponding information. Such a function is expressed by:

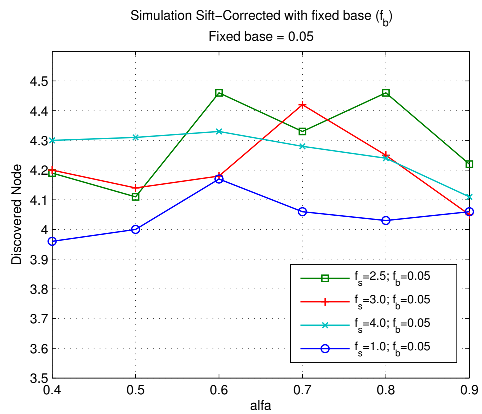
$$p_{sift-c} = \frac{(1-\alpha)\alpha^{cw}}{1-\alpha^{cw}} * \alpha^{slot} * fs + fb \qquad (1)$$

where the f_s is a spread factor and the f_b is a base factor to perform the Sift-Corrected equation 1 to our requirements. In Fig. 2(a) and Fig. 2(b) some simulation result for different value of f_b, f_s and $alpha$ are shown.

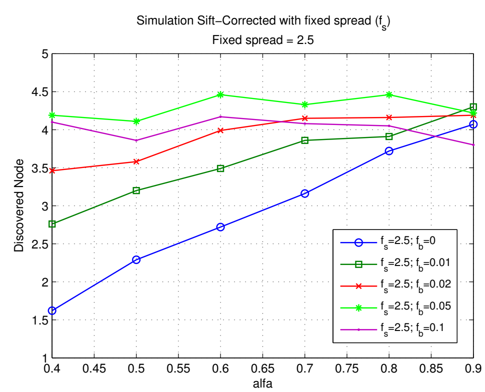
Algorithm 1 NDResponse in Sift-Persistent

while elapsed time-slots $\leq$ W **do**
  sense the carrier
  if channel is busy **then**
    wait until next time-slots
  else
    if rand() $\leq$ getSiftProbability(time-slot) **then**
      sendBackNDRes()
      break
    end if
  end if
end while

Using the original Sift [10] shape the probability threshold is too low for all slots. However, we would like to obtain a low threshold at the beginning and a relatively high threshold in the last slot. This is a suitable behavior if we have a certain number of nodes that, after transmitting, go to sleep. In our protocol the probability threshold is different for each time slot of the collision window. It begins at a low level (probability to transmit), in the first slot of the CW (at the beginning the number of neighbors is high). Therefore, at the end of the contention window the number of neighbor nodes decreases (after transmitting a node goes to sleep), and the probability of transmission increases. No neighbor should remain without transmitting. In order to obtain the best behavior we modified the Sift distribution using a *spread factor* and a *base factor*, (*Sift-corrected* Eq. 1). In Fig. 3 we show a graph in which we compare the simulations results versus the real experiment results on OpenBeacon RFID devices.



(a)



(b)

Fig. 2: Simulation using different values for f_b and f_s.

In order to validate our simulator we simulate the same real experiment using the same settings and from the figures it can be seen that our simulator provides very close results to real ones. Fig. 3 shows the results for *P-Persistent* using a $CW = 10$ and $p = 0.2$ both simulated and real experiment; and for *Sift-Persistent* using a $CW = 30$ both simulated and real experiment.

We configure the Sift, the *spread factor* and the *base factor*, making several simulations, significant simulations are shown in Fig. 2(a) and Fig. 2(b). In Sift-c 1 the $alpha$ is a shape factor, CW is the collision window dimension, $slot$ is the current time slot number ($slot\epsilon[1, cw]$), fs is the *spread factor* and fb is a *base factor* value.

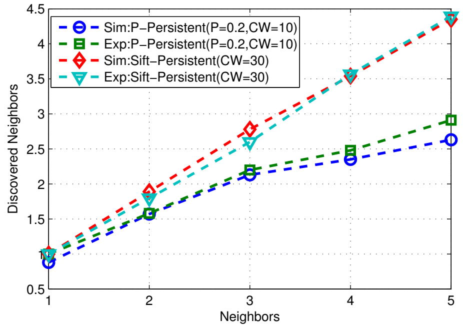


Fig. 3: Simulated vs real experiment results.

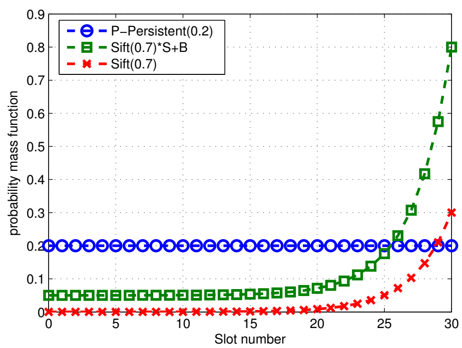


Fig. 4: The *Sift distribution*, the *corrected Sift distribution* with base $B = 0.05$ and spread $S = 2.5$ and the *P-Persistent* with $p = 0.2$

V. Performance Evaluation

To evaluate the performance we implemented a simple ND simulator in Java with a number of ND-Responding nodes that reply to an ND-Request. Therefore we simulate a Neighbor Discovery contention. In our Java class were simulated: *P-Persistent*, *Slotted Aloha* and our *Sift-Persistent* MAC protocols. Therefore we implemented the *P-Persistent* and *Sift-Persistent* protocols on OpenBeacon firmware, and results from this testbed are shown.

A. Simulated results

The validation results for our simulator comparing simulated and real experiments are shown in Fig. 3. A collision window length of both $CW = 30$ and $CW = 10$, a variable number of neighbor nodes between 1 and 30 and the *P-Persistent* protocol using threshold probability value as $p = 0.05$, $p = 0.1$, $p = 0.2$ and $p = 0.4$. In Figure 6 the results between *Sift-Persistent* and *P-Persistent* using a collision window $CW = 10$ with $p = 0.2$, $p = 0.1$ and $p = 0.4$ are shown. One advantage of our proposal is that in *Sift-Persistent* there are no parameters that need to be accorded with the collision window length. In Figure 5 the simulation results using $CW = 30$ are shown, the *Sift-Persistent* provides the best result (configured with a fixed $alfa = 0.7$, spread and base factor $fb = 0, 05$ e $fs = 2, 5$). We evaluate the neighbor discovered nodes for *P-Persistent* using different probability threshold value: $p = 0, 2$, $p = 0, 4$, $p = 0, 1$, $p = 0, 05$ to select the best configuration.

TABLE I: P-Persistent simulations settings.

Collision Windows (CW)	$300ms$
Slot time	$10ms$
Slot per CW	30
Responser selected probability	$0, 2; 0, 1; 0, 4$

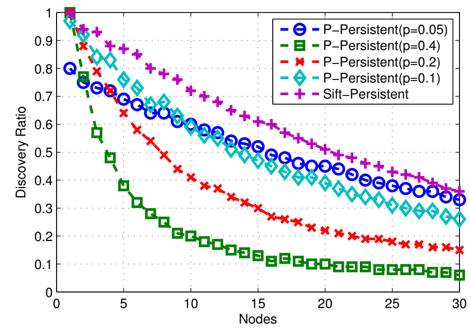


Fig. 5: Simulation. Neighbour Discovery ratio: $devices = 30$, $CW = 30$.

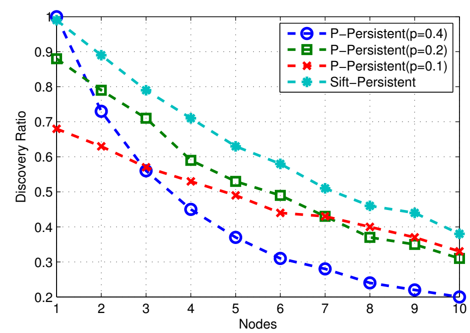


Fig. 6: Simulation. Neighbour Discovery ratio: $devices = 10$, $CW = 10$

B. Testbed results

In this section some results of the experiments on Open-Beacon [9] devices are presented. Each test had a duration of $4minutes$, with an identification cycle that repeated each $5seconds$ and each one test is repeated $5times$. Two different CW configurations were used in our test: using a CW of $30slots$ ($1slot = 10ms$) and the CW time duration is $300ms$, otherwise using a CW of $10slots$ (each $slot = 10ms$), CW time duration $100ms$.

TABLE II: Testbed configuration settings.

Single experiment duration	4 minutes
Transmission range	$\approx 5cm$
NDReq creation rate	$12NDReq/minute$
Number of Responser nodes	$5, 4, 3, 2$
Number of Requester nodes	1

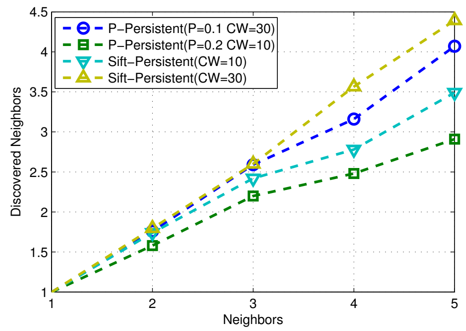


Fig. 7: OpenBeaonc Testbed *Sift-Persistent* vs *P-Persistent*.

The *NDRequester* sends a *NDReq* message every $5sec$; the *NDResponser* attend a *NDReq* and starts the *NDRes phase* when it receives it. The *NDRes phase* sends an *NDRes message* in one randomly selected time slot, using a specific protocol: *P-Persistent*, *Sift-Persistent*.

1) P-Persistent: The first version of Neighbor Discovery protocol is the *P-Persistent*. The threshold probability used in this testbed is $p = 0.2$, the contention window is $CW = 300ms$, $30slot$ each ones $10ms$. In our testbed we used a threshold $p = 0.1$, that is the best value obtained from the Java simulation using the contention window $CW = 30$, and $p = 0.2$, which is the best value for a contention window $CW = 10$. In Figure 7 the results from our testbed with OpenBeacon devices are shown, varying the number of neighbors between 1 to 5 using the *P-Persistent* protocol with two different collision window dimensions, $CW = 10$ and $CW = 30$.

2) Sift-Persistent: In *Sift-Persistent*, in one time slot each tag computes its probability to transmit and compare its value with a slot threshold. The slot threshold is a value that increases with the slot number. If a tag extracts a value of p which is less or equal than the *sift-threshold* (Figure 4) then it is allowed to transmit. The plot of some referred functions and our Equation 1 are shown in Figure 4. In Figure 7 the *P-Persistent* and *Sift-Persistent* experimental results are compared. Moreover the results between collision windows both of $10slots$ and either $30slots$ are shown.

VI. Conclusion

In this document we presented a probabilistic persistent protocol, implementing it on an RFID open platform, OpenBeacon [9]. We implemented the classical protocol *P-Persistent* and our solution called *Sift-Persistent* and compare the results of each protocol with different contention windows and using various setting values. Our solution to apply *Sift-Persistent* protocol over RFID-DTN is more suitable than the *P-Persistent* in terms of capacity to discover more neighbors and to support more signaling traffic. Moreover, the *P-Persistent* strategy needs configuration of threshold probability (p) according to the collision windows size and this can be critical under dynamic network conditions. The *Sift-Persistent*, instead, does not need to tune the collision window length becoming more robust and efficient in a context where a higher number of active RFID can be deployed. We envision the possibilities of this solution for an auto- configurable Neighbor Discovery protocol for strict resource constraints in Delay Tolerant Networks.

References

[1] P. Fazio, M. Tropea, F. Veltri, and S. Marano, "A new routing protocol for interference and path-length minimization in vehicular networks," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*. IEEE, 2012, pp. 1–5.

[2] P. Fazio, F. De Rango, and C. Sottile, "An on demand interference aware routing protocol for vanets," *Journal of Networks*, vol. 7, no. 11, pp. 1728–1738, 2012.

[3] F. De Rango, M. Tropea, G. B. Laratta, and S. Marano, "Hop-by-hop local flow control over interplanetary networks based on dtn architecture," in *Communications, 2008. ICC'08. IEEE International Conference on*. IEEE, 2008, pp. 1920–1924.

[4] F. De Rango, S. Amelio, and P. Fazio, "Enhancements of epidemic routing in delay tolerant networks from an energy perspective," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*. IEEE, 2013, pp. 731–735.

[5] K. Massri, A. Vernata, and A. Vitaletti, "Routing protocols for delay tolerant networks: a quantitative evaluation," in *Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*. ACM, 2012, pp. 107–114.

[6] N. Cordeschi, V. Polli, and E. Baccarelli, "Interference management for multiple multicasts with joint distributed source/channel/network coding," *IEEE Transactions on Communications*, vol. 61, no. 12, pp. 5176–5183, 2013.

[7] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, Tech. Rep., 2000.

[8] K. Massri and A. Vitaletti, "DTN routing protocols on resource constrained devices: Design, implementation and first experiments"," in *The 21st International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2013)*, Split-Primosten, Croatia, sep 2013.

[9] (2006) Openbeacon active rfid project. [Online]. Available: http://www.openbeacon.org/

[10] K. Jamieson, H. Balakrishnan, and Y. Tay, "Sift: A mac protocol for event-driven wireless sensor networks," in *Wireless Sensor Networks*. Springer, 2006, pp. 260–275.

[11] X. Huang and S. Le, "Efficient dynamic framed slotted aloha for rfid passive tags," in *Advanced Communication Technology, The 9th International Conference on*, vol. 1. IEEE, 2007, pp. 94–97.

[12] "Epcglobal, class 1 generation-2 uhf rfid air interface protocol standard specifcations," 2004.

[13] "Iso/iec 18000-7.2004 information technology radio frequency identication for item management part 7. parameters for active air interface at 433 mhz," 2004.

[14] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Instant collision resolution for tag identification in rfid networks," *Ad Hoc Networks*, vol. 5, no. 8, pp. 1220–1232, 2007.

[15] P. Semiconductor, "I-code1 system design guide: Technical report," 2002.

[16] K. Finkenzeller, "Rfid handbook 2nd edition," 2003.

[17] H. Vogt, "Efficient object identification with passive rfid tags," in *Pervasive Computing*. Springer, 2002, pp. 98–113.

[18] ——, "Multiple object identification with passive rfid tags," in *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 6–pp.

[19] E. Egea-López, J. Vales-Alonso, A. S. Martínez-Sala, M. V. Bueno-Delgado, and J. García-Haro, "Performance evaluation of non-persistent csma as anti-collision protocol for active rfid tags," in *Wired/wireless internet communications*. Springer, 2007, pp. 279–289.

[20] D. Leonardo, M. Sanchez, M. Victor, and R. Ramos, "P-persistent csma as a collision resolution protocol for active rfid environments," in *Wireless and Optical Communications Networks (WOCN), 2011 Eighth International Conference on*. IEEE, 2011, pp. 1–5.

[21] A. Palomo-López, M. V. Bueno-Delgado, E. Egea-López, J. J. Alcaraz-Espín, and J. Vales-Alonso, "Csma multi-stage anti-collision protocol for active rfid systems." in *IWRT*, 2010, pp. 23–35.