On the Use of Manchester Violation Test in Detecting Collision

Yee-Loo Foo and Hiroyuki Morikawa

Dept. of Info. and Comm. Eng., The University of Tokyo, Tokyo 113-8656, Japan
{ylfoo,mori}@mlab.t.u-tokyo.ac.jp

Abstract. Smart devices in the ubiquitous computing environment implement service/device discovery protocol that helps discovering each other and the services provided. As client device may receive multiple service description messages, it implements at its MAC layer a collision resolution mechanism to resolve the collision of messages. Effectiveness of collision resolution relies on the accuracy of detecting collision at the PHY layer. In this paper, we question the reliability of the conventional collision detection technique, which inaccuracy will affect the completeness of service/device discovery. Our analysis shows that capture effect and packet reception failure can cause failure in collision detection when the conventional technique is used. We suggest a detection technique that makes use of Manchester violation test. Implementation and evaluation of the proposed technique on some smart devices show its superiority over the conventional approach.

Keywords: Wireless radio, service/device discovery, concurrent transmissions, collision detection and resolution, capture effect, Manchester coding and violation.

1 Introduction

Ubiquitous computing envisions a world where various computing services run on a wide range of devices in our surroundings. Automated discovery of these devices and the services they offer will certainly enhance our quality of life and change the conventional concept of how service is found and delivered to us. This service/device discovery mechanism automates the process of identifying a device and describing the service it provides, and if necessary, setting up a connection with it. It involves passing *discovery* message that contains the device identity (ID), description of the service it provides, device configuration and connection setup information.

We are particularly interested in the possible implementation of discovery protocol on low-power 'smart' devices, which have only limited computing resources, and use only low-power radio to communicate over short distance at low data rate. Running service/device discovery protocol on these resource-limited devices is challenging. Nevertheless, these smart devices are 'powerful' in the sense that they are cheap and tiny, and thus can be easily tagged with other devices or even daily objects. The presence of a large number of smart devices in the surroundings provides us an unimaginably intelligent environment which we will benefit from.

T. Vazão, M.M. Freire, and I. Chong (Eds.): ICOIN 2007, LNCS 5200, pp. 729–740, 2008.

As a device may receive multiple discovery messages simultaneously, it implements at its MAC layer a collision resolution mechanism to resolve the collision of messages. Effectiveness of collision resolution relies on the accuracy of detecting collision at the PHY layer. In this paper, we raise a question on the reliability of the conventional collision detection technique, which inaccuracy will affect the completeness of service/device discovery. Our analysis shows that capture effect and packet reception failure can cause failure in collision detection when the conventional technique is used. We thus suggest a collision detection technique that makes use of Manchester violation test, which is different from the conventional one. Implementation and evaluation of the proposed technique on some smart devices show its superiority over the conventional approach.

The rest of this paper is organized as follows. Section 2 provides the problem analysis which also includes the background study. Section 3 describes how we approach the problems. Section 4 describes the implementation of our algorithm, experiment setup and procedures, followed by evaluation results. Related works are given in Section 5. Finally Section 6 concludes this paper.

2 Problem Analysis

2.1 Background

2.1.1 Service/Device Discovery Methods

There are two types of discovery method: *push*-type (or announcement-based) and *pull*-type (or on-demand based) [1]. Following the push-type approach, a service provisioning device advertises its service by repeatedly sending out discovery message through broadcast announcement. Interested parties gather all kinds of advertisements, picking up those that provide desired services for further actions, while filtering the unwanted ones. It is the service provider that bears the responsibility of getting its message delivered to its targets. A targeted device does not put effort in making sure that it receives the discovery message directed to it. On the other hand, in a pull-type discovery model, a client device proactively queries for desired services. Prospective service providers respond with their respective discovery message. However, it is the client's responsibility to make sure that it correctly receives all discovery messages targeted to it. A discovery protocol can be designed to support both discovery methods.

2.1.2 Collision Resolution

The choice of discovery method affects the design of Medium Access Control (MAC) protocol that runs at the lower layer, which governs the access to a shared channel from multiple devices. When push-type method is chosen, the MAC protocol implemented at the service provisioning devices is mainly a *collision avoidance* mechanism that aims to avoid simultaneous sending of discovery messages which will cause message collision at a listening device. Collision causes loss of discovery messages, implicating incomplete service/device discovery. When the pull-type discovery model is to be followed, in addition to the implementation of collision avoidance mechanism at service providers, clients should implement *collision* *resolution* mechanism that acts to resolve collision when it occurs. The resolution mechanism aims to recover the information lost due to collision, by principally requesting those involved in the collision to retransmit their messages. The ultimate goal of collision resolution is to allow every conflicting sender to successfully deliver its message to its target receiver. Collision resolution scheme, for example, is deployed in Radio Frequency Identification (RFID) readers (where it is called *anti-collision* scheme) to resolve collisions arise from RFID tags identification process [2, 3].

2.1.3 Collision Detection

Collision resolution mechanism starts operating whenever collision is detected, and it ends after resolving all collisions and no new collision is detected. The activation and deactivation of collision resolution mechanism closely relies on the detection of collision at the physical (PHY) layer. PHY layer must have function that is capable of detecting collision and provides the correct information to the collision resolution mechanism. Accuracy of collision detection is very important and becomes our concern here.

How does a low-power radio receiver detect collision? Collision refers to the situation where a receiver fails to receive the packet (that contains a message) sent by a transmitter due to interference from other simultaneous packet transmissions. The receiver can identify such situation (i.e. collision) when it senses the presence of a packet transmission, but detects incompleteness in the information that it receives (e.g. incorrect packet checksum). In short, conventionally the presence of collision is identified when erroneous packet is received. However, the completeness of this logic has not been previously questioned. The assumption is that concurrent transmissions always result in reception of erroneous packet (at a listening radio). In this paper, we reveal that this is not true.

2.2 Issues

2.2.1 Capture Effect

Instead of collision, concurrent transmissions can result in *capture*, i.e. successful reception of the packet that is the strongest among all transmitted packets. The ‘captured’ packet is received with full integrity i.e. correct checksum.

While collision is the result of interference induced by other concurrent transmissions, capture is the result of receiver’s tolerance against the interference. When capture effect prevails, collision goes undetected. Based on this wrong indication of the absence of collision, the collision resolution mechanism fails to take action to resolve the occurred collision, resulting in incomplete service/device discovery.

Some early works that described about capture effect are [4, 5]. Commonly used in the literature to quantify capture effect [4 - 8], *capture ratio* refers to the minimum required signal-to-interference ratio (*SIR*) for a signal to be successfully received despite the presence of other transmissions. When capture ratio is low (i.e. close to 0 dB), concurrent transmissions result in collision more likely than capture. As the capture ratio increases, the trend is reversed that it is more likely to have capture.

2.2.2 Packet Reception Failure

There are collision conditions where receiver fails to receive a packet, not even an erroneous one. As a result, it does not sense the collision or even the presence of transmissions itself. When this happens, collision resolution mechanism does not activate, resulting in incomplete discovery.

A generic packet reception process is explained as follows. A radio packet is preceded by preamble, followed by synchronization (SYNC) bytes, and then data [9]. Preamble is usually a series of alternating bit 0 and 1. It informs the presence of a packet and allows a listening receiver to achieve bit synchronization with this packet transmission. Without achieving bit synchronization, the receiver will not be able to correctly identify the first bit of the SYNC byte that follows. Hence, a receiver that is yet to receive preamble would keep itself busy in searching for one. After successfully receiving preamble, the receiver starts tracing for SYNC bytes in order to reach byte synchronization with the packet transmission. After byte synchronization is achieved, the receiver is able to correctly identify the first byte of data that follows. Without receiving preamble and SYNC bytes, the receiver assumes that whatever that has been received is noise. Only after receiving SYNC bytes, the receiver starts buffering data bits that follow. There is no point of start buffering without first receiving both preamble and SYNC bytes, because synchronization would not have been achieved, or there simply is not any packet to be received. After all data bits have been buffered, the receiver examines the checksum. Incorrect checksum implies collision.

The problem is that preamble and SYNC bytes can be corrupted because of collision or noise. If we follow the conventional collision detection principle, we find that only collision that corrupts packet data but not preamble and SYNC bytes can be detected. Collision that corrupts preamble and SYNC bytes cannot be detected because the receiver cannot even confirm the presence of transmission.

Noise is another cause of corruption, although it is not related to collision or capture. In the case of single packet transmission, even when there are no concurrent transmissions that can possibly cause collision, the packet preamble and SYNC bytes can be corrupted by noise. As a result, the transmitted packet will not be received, giving the same impact of incomplete service/device discovery.

3 Approach

This section describes how we approach the presented challenges. An effective approach should be able to track down capture and packet reception failure, which cause the problems. When either capture or packet reception failure is identified, from the perspective of collision resolution scheme it should be counted as a collision. In fact, the conventional view on ‘collision detection’ and ‘collision resolution’ should be renewed to as follows: *Instead of detecting collision, we should make effort to detect the presence of concurrent transmissions. Instead of resolving collision, we should resolve the conflict arises from concurrent transmissions.* Here we introduce the use of Manchester violation (MV) test as our approach.

Manchester coding embeds transmitter’s clock information into the bit stream, making synchronous transmission possible [10]. Manchester code requires a signal level transition in the middle of a bit. If such transition is not observed, the Manchester coding format is said to be violated. Violation detection is determined by how 'balanced' a bit looks. A distorted bit is likely to cause a violation. And bit distortion is usually a result of noise or interference contributed by other transmission. Therefore a violation could serve as a good indication of the presence of concurrent transmissions.

We suggest the following methodology when applying MV test for the purpose of detecting concurrent transmissions that result in collision or capture. When there is a positive detection, the collision resolution scheme is notified and it will activate. The algorithm suggested in the following is to be implemented on the radio receivers of client devices that are in search of services following the pull-type discovery approach.

A client broadcasts a query, and allocates a fixed duration to collect the discovery messages responded by potential service provisioning devices. Within the duration, if preamble and SYNC bytes have been received, all the bits that are received after them will be buffered and examined for data integrity through checksum function e.g. Cyclic Redundancy Code (CRC) check. Following the conventional collision detection technique, if checksum is incorrect, collision is assumed to be the cause and the collision resolution scheme will be alerted and activated. However, if the checksum is correct, collision is assumed to be absent because a packet has been received with complete integrity. No consideration is given to the possibility of capture.

Our approach is different, and it is capable of detecting both collision and capture. It requires a listening radio to record the MV status of every data bit (i.e. all bits that are received after the preamble and SYNC bytes). Let say the number of violation bits in the data part of a packet is N_{data}. $N_{data} = 0$ means the received data has not experienced distortion i.e. collision is regarded to be absent. On the other hand, $N_{data} > 0$ means certain degree of distortion has been experienced due to either interference resulted from concurrent transmissions or noise. Regardless of the cause, the correctness of the received data is now in doubt. The radio receiver simply assumes that collision is present and thus activates the collision resolution scheme.

As we will show in the Results subsection (Fig. 3), our technique is capable of detecting all collisions that are detectable through conventional detection technique. On top of that, it can also detect to some extent the capture cases that are not detectable through conventional technique. This is one of the reasons the suggested technique is considered more superior to the conventional one that relies on checksum algorithm.

On the other hand, if preamble and SYNC bytes have not been received within the listening duration, no special action is taken in the conventional packet reception process. No consideration has been given to the possibility that collision has actually occurred and corrupted the preamble and SYNC bytes. Here we add a mechanism to help detecting this. When the radio receiver is listening within a specified duration, it examines each received bit for its MV status and records the result. Let say the number of violation bits observed within the duration is $N_{duration}$. Now when the duration is over, and if preamble and SYNC bytes have not been received, $N_{duration}$ is examined. If no transmission has been made within the listening range of a radio receiver, not a single Manchester encoded bit would have appeared at the input of

Manchester decoder in the receiver, and thus there should be a large number of violations (i.e. large $N_{duration}$) observed. A reasonable guess is that at least half of the total number of received bits could have committed violation. On the other hand, when at least one transmission has been made, a series of Manchester encoded bits (though some could have been distorted) must have been available at the decoder input, and thus a relatively smaller number of violations (i.e. smaller $N_{duration}$) are expected. If we can determine a threshold level, $N_{threshold}$, that can differentiate between $N_{duration}$ of these two conditions, the algorithm simply includes a comparison between $N_{duration}$ with $N_{threshold}$. If $N_{duration} > N_{threshold}$, it is assumed that there has been no transmission and only noise has been received. However, if $N_{duration} < N_{threshold}$, the algorithm regards the corruption of preamble and SYNC bytes has been due to collision or noise. Regardless of the cause, the collision resolution scheme is to be activated.

The following pseudo code represents the methodology we suggest to implement on the radio receiver of client device. The lines not in bold letter are the pseudo code of a generic packet reception mechanism that is constructed based on recommendations given in [11]. Conventional collision detection algorithm is also described in these lines. The lines in bold letter represent the algorithm we suggest.

```
  Initial state = PREAMBLE
  begin
  Receive 1 bit
  if (MV is present)
       N_duration ++
  Shift the bit value into a shift register
  goto current state

state: PREAMBLE
  if (shift register value = PREAMBLE)
       Preamble_count ++
  if (Preamble_count > Requirement)
       goto SYNC state

state: SYNC
  if (shift register value = SYNC)
       goto DATA state
  else
       Error ++
  if (Error > Tolerance)
       goto PREAMBLE state

state: DATA
  if (MV is present)
       N_data ++
  if (Packet length is reached)
       // Packet reception completes
       Examine N_data
       if (N_data > 0)
           Detect possible collision
           Report to collision resolution scheme
       else if (N_data = 0)
           No collision
       Perform CRC check
       if (checksum correct)
           No collision
```

```
      else if (checksum incorrect)
          Detect collision
          Report to collision resolution scheme
// end of all states

  if (End of listening duration is reached)
      if (Preamble & SYNC not received)
           Examine N_duration
           if (N_duration > N_threshold)
               No collision
           else if (N_duration < N_threshold)
               Detect possible collision
               Report to collision resolution scheme
```

4 Implementation and Evaluations

4.1 Implementation

For the purpose of evaluating our approach, we prepare three units of smart device. Each consists of the following main components: a Chipcon CC1000 low-power radio transceiver chip [12], which is controlled by PIC18LF4620 low-power microcontroller from Microchip [13]. In our following experiments, one of the devices plays the role of client which is in search of service, while the other two act as service provisioning devices that respond to the client. In order to replicate collision and capture effect, the latter two are made to reply simultaneously to every query issued by the client.

A generic packet transmission and reception mechanism is implemented in the form of software that runs on the microcontrollers of these devices. A CRC-16 checksum algorithm is also implemented in the software, following the guide given in [14]. On top of that, we implement our collision detection algorithm on the client device.

The CC1000 transceiver chip comes with MV test feature, where the violation status of each received bit is available at one of its pins. In receive mode, CC1000's demodulator passes each received bit and its accompanied MV status to the microcontroller for further processing. CC1000 use FSK modulation, and they are set to communicate at the rate of 4800 bps.

A packet is preceded by 4 bytes of preamble, followed by 2 bytes of SYNC, 28 bytes of data, and finally 2 bytes of CRC checksum, giving a total length of 36 bytes. When a packet is received perfectly without its bits experiencing distortion, all the received bits will be free from Manchester violation, except the initial few bits of the preamble. This is because the synchronization process takes place at preamble and before a receiver achieves synchronization with a transmission, the received bits most likely violate the Manchester coding format. We are interested in the part of a packet that can possibly give $N_{duration} = 0$ when an undistorted packet is received, since $N_{duration} = 0$ clearly indicates the absence of interference. As a result, we choose to skip monitoring the MV status of the preamble. In order to do this, we synchronize the transmitter and receiver to start transmitting and receiving at the same time. The receiver then skips recording the MV status of the first 4 received bytes (which are likely to be preamble if there is a transmission). Only the following 32 received bytes (or 256 bits) their MV status are recorded. From these 256 bits, the total number of Manchester violating bits gives $N_{duration}$.

After a receiver successfully receives preamble and SYNC bytes, it proceeds buffering the 28 bytes of data and 2 bytes of CRC checksum that follow. All these 30 bytes or 240 bits that are buffered will be examined for their MV status. The total number of Manchester violating bits gives N_{data}.

4.2 Experiments

To replicate collision and capture effect, we do the followings. Upon receiving a query message from the client device, the other two devices simultaneously send to the client a packet (which should contain the discovery message). Depending on the capture ratio, which describes the power relationships between these devices, the client device either captures one of the two packets, or the two collide and destroy each other and thus the client receives none of them. When capture ratio is high, capture effect becomes dominant, and vice versa.

To produce collision and capture effect in our experiments, we vary the capture ratio accordingly. This is done by first fixing the distance between client device and one of the other two devices. We then vary the distance of the remaining device from the client. As all devices transmit at equal and constant power, the client device receives different signal strength from the two devices that are located at different distance. To quantify capture ratio, it requires the client's radio to measure the received power from each device's transmission. The received power can be read out from the Received Signal Strength Indicator (RSSI) pin of the CC1000 radio. The ratio between the powers received from the two devices gives capture ratio. The received power values that have been taken into the calculation of capture ratio are the averages over a period of 1 second. Finally, we get a set of locations of all three devices, where each set of locations gives different capture ratio. Our experiment is repeated over these different sets of locations. For each set of locations, the client sends out 500 queries one after another, and thus triggering 500 concurrent transmissions from the other two devices, which end up in either collision or capture, allowing us to evaluate the effectiveness of our algorithm in detecting them. When the receiver in the client fails to receive any packet and $N_{duration} < N_{threshold}$, it is considered as a collision. And if the receiver receives a packet and $N_{data} > 0$, it is also counted as a collision.

4.3 Results

We are interested to find out a suitable value for $N_{threshold}$. We examine the $N_{duration}$ results of the cases where packet reception has failed due to corrupted preamble and SYNC bytes. We have observed 389 such cases for a particular set of devices locations that gives capture ratio of 0 dB. From the results we gather, we find that $N_{duration}$ is distributed between 10 and 160 (refer Fig. 1), and it gives a mean value of 58, out of 256 received bits which we have examined their MV status. Next, we examine the $N_{duration}$ results of the cases where no packet has been transmitted and all that the receiver receives is noise. We find that in this condition where only noise is received, $N_{duration}$ is distributed between 130 and 200 (refer Fig. 1), and it gives a

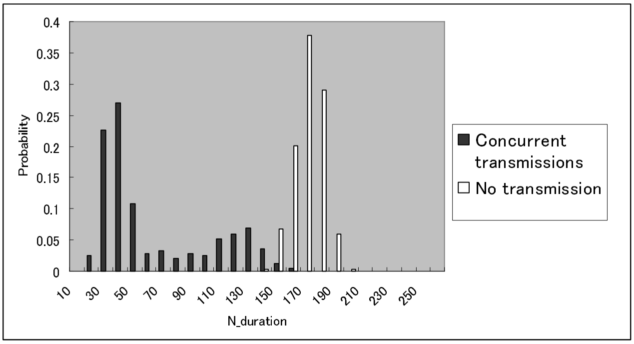

Fig. 1. Probability density function of $N_{duration}$ that belong to two different conditions

mean value of 165, out of 256 received bits which we have examined their MV status. There is therefore an overlapping region (from 130 to 160) between the two distributions of $N_{duration}$. This implies that if a receiver gets a $N_{duration}$ value that falls within this region, it cannot conclude that whether there has been a collision or not. Fortunately, the likeliness of overlapping is actually very small. From Fig. 1, we find that a good choice for $N_{threshold}$ is 140, i.e. collision is assumed to be present when $N_{duration}$ < 140, and vice versa. By taking $N_{threshold}$ as 140, the probability of mistaking collision as being absent is only 1.80%, while the probability of mistaking collision as being present is as small as 0.26%.

Next, we are interested to find out the effectiveness of our technique in detecting concurrent transmissions that result in collision or capture. Fig. 2 shows the detection rate. The cases of packet reception failure with $N_{duration} < N_{threshold}$, and successful packet reception with $N_{data} > 0$, are both counted as 'Successful detection'. The cases of packet reception failure with $N_{duration} > N_{threshold}$, and successful packet reception with $N_{data} = 0$, are both counted as 'Unsuccessful detection'. With reference to Fig. 2, when the capture ratio is low (0 to 1.5 dB), the detection rate is rather high, ranging from 76% to 100% (giving an average of 84%). When the capture ratio is high (2 to 4 dB), the detection rate drops to near zero as the number of undetected captures grows. The explanation is that at low capture ratio, concurrent transmissions most likely

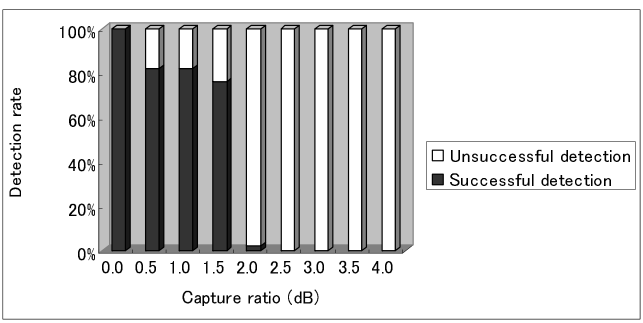

Fig. 2. Collision detection rate of proposed technique

result in collision. And we observe that our technique can detect all the collision cases. However, as the capture ratio increases, capture is more likely to become the end result of concurrent transmissions, and our technique can detect only cases of weak capture but not the severe ones (cf. the conventional technique can detect none of capture case). The above results show that capture effect gives a severe impact on collision detection, and remains a problem to be resolved.

Nevertheless, in Fig. 3 we show that our technique is still more superior to the conventional one. In the region where capture ratio is low (0 to 1.5 dB), our technique is at least twice more effective than the conventional one in terms of collision detection rate.

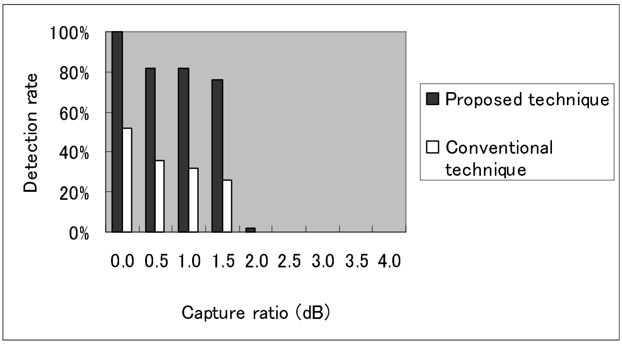

Fig. 3. Performance comparison between proposed technique and conventional technique

5 Related Works

In the literature, there has been no comprehensive investigation on the reliability of collision detection in a radio receiver. The possible problems posed by capture effect and packet reception failure have not been previously addressed. Nevertheless, in the context of RFID system, [15] did report that capture effect could actually influence the number of detectable RFID tags. On the other hand, we have not found any experimental work that evaluates the effectiveness of MV test as a means of collision detection.

The work by Whitehouse et al. [16] was related to capture effect and collision detection, which is close to ours. They suggest making use of capture effect to help detecting collision. Their technique requires the receiver to keep tracing for a second set of preamble even after receiving the first one. If a second set exists, the receiver drops the reception of the earlier packet, and resynchronizes to the later one. Instead of both packets collide and corrupt, at least the later one can be saved. This technique is effective if the later packet is relatively stronger than the earlier one, so that capture effect is present and can be exploited.

[17, 18] provided a signal processing viewpoint to the collision detection and resolution problem. Nevertheless, signal processing is usually prohibitively high cost to low-power radios.

6 Conclusion

In this paper, we raise a question on the reliability of the conventional collision detection technique that is deployed on radio receiver. We then elaborate on the issues that originate from capture effect and packet reception failure. We suggest a different detection technique that is based on the use of MV test. The performance of the proposed technique has been evaluated on some radio devices. The results are promising, and they show that the proposed technique outperforms the conventional one.

Acknowledgments. This work is supported by Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan.

References

1. Zhu, F., Mutka, M.W., Ni, L.M.: Service Discovery in Pervasive Computing Environments. IEEE Pervasive Computing, 81–90 (2005)
2. EPCglobal. 13.56 MHz ISM Band Class 1 Radio Frequency (RF) Identification Tag Interface Specification, `http://www.epcglobalinc.org/standards_technology/Secure/v1.0/HF-Class1.pdf`
3. EPCglobal. 900 MHz Class 0 Radio Frequency (RF) Identification Tag Specification, `http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class0.pdf`
4. Roberts, L.G.: ALOHA Packet System with and without Slots and Capture. ACM SIGCOMM Computer Comm. Review 5(2) (1975)
5. Leentvaar, K., Flint, J.: The Capture Effect in FM Receivers. IEEE Trans. Commun. 24(5), 531–539 (1976)
6. Krauss, H.L., Bostian, C.W.: Solid State Radio Engineering. John Wiley & Sons, New York (1980)
7. Ash, D.: A Comparison between OOK/ASK and FSK Modulation Techniques for Radio Links. Technical report, RF Monolithics Inc. (1992)
8. Nelson, R., Kleinrock, L.: The Spatial Capacity of a Slotted ALOHA Multihop Packet Radio Network with Capture. IEEE Trans. Commun. 32(6), 684–694 (1984)
9. Torvmark, K.H.: Short-Range Wireless Design, `http://www.embedded.com/story/OEG20020926S0055`
10. Halsall, F.: Data Communications, Computer Networks and Open Systems. Addison-Wesley, Reading (1996)
11. Torvmark, K.H.: Application Note AN009: CC1000/CC1050 Microcontroller Interfacing, `http://focus.ti.com/lit/an/swra082/swra082.pdf`
12. Chipcon. CC1000: Single Chip Very Low Power RF Transceiver, `http://www.chipcon.com/files/CC1000_Data_Sheet_2_3.pdf`
13. PIC18F2525/2620/4525/4620 Data Sheet, `http://ww1.microchip.com/downloads/en/DeviceDoc/39626b.pdf`
14. Williams, R.: A Painless Guide to CRC Error Detection Algorithms, `http://www.ross.net/crc/download/crc_v3.txt`
15. Floerkemeier, C., Lampe, M.: Issues with RFID Usage in Ubiquitous Computing Applications. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 188–193. Springer, Heidelberg (2004)

16. Whitehouse, K., et al.: Exploiting the Capture Effect for Collision Detection and Recovery. In: IEEE EmNetS-II, Sydney (2005)
17. Tsatsanis, M.K., Zhang, R., Banerjee, S.: Network-Assisted Diversity for Random Access Wireless Networks. IEEE Trans. Signal Processing 48(3), 702–711 (2000)
18. Zhang, R., Sidiropoulos, N.D., Tsatsanis, M.K.: Collision Resolution in Packet Radio Networks Using Rotational Invariance Techniques. IEEE Trans. Commun. 50(1), 146–155 (2002)