Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2011, Article ID 365107, 9 pages
doi:10.1155/2011/365107

Research Article

Multipath Detection Using Boolean Satisfiability Techniques

Fadi A. Aloul[1] and Mohamed El-Tarhuni[2]

[1] *Department of Computer Science and Engineering, American University of Sharjah, P.O. Box 26666 Sharjah, United Arab Emirates*
[2] *Department of Electrical Engineering, American University of Sharjah, P.O. Box 26666 Sharjah, United Arab Emirates*

Correspondence should be addressed to Fadi A. Aloul, faloul@aus.edu

Received 13 March 2011; Revised 8 August 2011; Accepted 5 September 2011

Academic Editor: Daniele Tarchi

A new technique for multipath detection in wideband mobile radio systems is presented. The proposed scheme is based on an intelligent search algorithm using Boolean Satisfiability (SAT) techniques to search through the uncertainty region of the multipath delays. The SAT-based scheme utilizes the known structure of the transmitted wideband signal, for example, pseudo-random (PN) code, to effectively search through the entire space by eliminating subspaces that do not contain a possible solution. The paper presents a framework for modeling the multipath detection problem as a SAT application. It also provides simulation results that demonstrate the effectiveness of the proposed scheme in detecting the multipath components in frequency-selective Rayleigh fading channels.

1. Introduction

There has been a growing interest in developing high data rate mobile radio systems to support a wide range of applications such as real-time multimedia services and high-speed internet access. To achieve this goal, wide band transmission schemes are being investigated including single carrier and multicarrier spread spectrum techniques, Ultra-wideband systems, and OFDM-based schemes. Multipath propagation, caused by reflection, refraction, and scattering of radio waves as they pass through the wireless channel, is considered as one of the main challenges in wide band mobile radio communication systems. Multipath propagation results in receiving multiple copies of the transmitted signal. In narrow band transmission schemes, where the multipath components are very close and unresolved by the receiver, severe fading is observed in the received signal strength leading to significant degradation in the bit error rate (BER) performance of the system. On the other hand, in wide band signal transmission, where multipath components could be resolved by the receiver, multipath propagation can be exploited using a RAKE receiver to improve the system BER performance through the diversity gain from the different copies of the received signal. However, for full utilization of the multipath scenario, it is very important for the receiver to first detect the presence of these multipath components and identify their corresponding parameters (time delay, amplitude, and phase).

In spread spectrum systems, a pseudo-random (PN) code is used to spread the message spectrum over a wide bandwidth. At the receiving end, a time-synchronized version of the same PN code is used to despread the signal and recover the original message [1]. Synchronization is very crucial for the proper operation of the system. It can be accomplished by searching a range of delays for the correct multipath delays. The uncertainty range represents the possible delays that the signal may have and is related to the channel memory. The delay range is usually specified as cells that are one-chip or one-half of a chip apart, where a chip is the shortest element in the PN code. The search for the multipath components through these cells, that is, finding the cells that have strong energy and hence multipath components, can either be done in a serial or parallel fashion [2–5].

In serial search, one cell at a time is tested by measuring the signal energy at that cell using a single correlator circuit. If the energy exceeds a preset threshold, then the cell is declared as a multipath cell, either directly or after a verification stage, while if the energy is below the threshold, then it is declared as a no multipath cell. The search advances to

the next cell and the process is continued until all cells in the uncertainty range are tested. The other search strategy uses parallel search where the energies of all cells are calculated simultaneously using a bank of parallel correlators and cells with energy above the threshold are declared as multipath cells. Apparently, serial search is slower compared to parallel search as it takes longer time to search all the cells and find the delays. On the other hand, serial search has a much lower reduced complexity (both hardware and processing).

A common drawback of existing schemes is that in searching for the correct cells they do not utilize the inherent structure of the PN code. In the worst case or in a low SNR environment, these schemes need to search all possible cells in the search window, which could be as large as the length of the PN code, in order to find the correct cells. For example, for a PN code with a length of 2047 chips (generated by an 11-stage shift register) the serial and parallel search schemes need to test 2047 cells if the search step is one chip or twice of that if the search step is one-half of a chip. This testing may need to be repeated many times if the multipath components were not detected at the first trial due to noise and fading. In this paper, we propose a PN code acquisition scheme that exploits the structure of the PN code to reduce the number of decisions needed to find correct cells. The proposed scheme is based on using advanced Boolean Satisfiability (SAT) techniques to perform intelligent search of the uncertainty region and hence reduce the number of decisions needed to find the correct cells significantly. This is done by searching only PN code phases that result in minimum difference (minimum distance) between the PN code in the received signal and a locally generated PN code.

Recently, Boolean Satisfiability (SAT) has been shown to be very successful in solving complex problems in various Engineering and Computer Science applications. Such applications include Formal Verification [6], FPGA routing [7], Power Optimization [8, 9], Fault Tolerance [10], and Microprocessor Verification [11]. SAT has also been extended to a variety of applications in Artificial Intelligence including other well-known NP-complete problems such as graph colorability, vertex cover, hamiltonian path, and independent sets [12]. Despite SAT being an NP-Complete problem [13], many researchers have developed powerful SAT solvers that are able of handling problems consisting of thousands of variables and millions of constraints [14–22]. Briefly defined, the SAT problem involves a set of Boolean variables and a set of constraints expressed in product-of-sum form. The goal is to identify an assignment to the variables that would satisfy all constraints or prove that no such assignment exists.

Even though in recent years we have seen a surge in the application of SAT techniques to assist in finding solutions to various Engineering problems, very few researchers reported on the use of SAT-based techniques in mobile communication-related research. In this paper, we propose the formulation of the PN acquisition problem as a SAT instance and use intelligent SAT search engines for multipath detection.

The reminder of this paper is organized as follows. Sections 2 and 3 present the signal model and an overview of SAT, respectively. Section 4 describes the proposed scheme and shows how to formulate the PN code acquisition problem as a SAT instance. Simulation results are presented and discussed in Section 5. Finally, the paper is concluded in Section 6.

2. Signal Model

A direct-sequence spread spectrum system is investigated in this paper. The signal model assumes that a separate pilot signal is transmitted along with the data channel to allow for PN code acquisition and tracking as well as channel estimation. The transmitted signal is given by

$$s(t) = \sqrt{P}\sum_{i=0}^{M-1}\left(d_i W + \sqrt{G_p}V\right)\sum_{k=0}^{N-1} c_k g(t - iT_b - kT_c), \quad (1)$$

where P is the transmitted power, d_i is a random sequence of information data with $d_i \in \pm 1$, W and V are orthogonal codes with length N (i.e., Walsh codes) used to separate the pilot channel from the data channel, G_p is the pilot channel power gain relative to the data channel, $c_k \in \pm 1$ is the spreading pseudo-random (PN) code, N is the PN code length which is the same as the number of chips per bit, that is, $N = T_b/T_c$, T_b is the bit duration, T_c is the chip duration, and $g(t)$ is the chip pulse shape. M is the number of data bits.

The radio channel is modeled as a frequency-selective Rayleigh fading channel, which is a common model for mobile radio systems, using narrow-band transmission. The received signal is given by

$$u(t) = \sum_{l=1}^{L}\beta_l s(t - \tau_l) + n(t), \quad (2)$$

where L is the number of paths, β_l is the lth path complex coefficient with Rayleigh amplitude and uniform phase distribution over $[0, 2\pi)$, τ_l is the lth path delay that we would like to estimate, and $n(t)$ is an additive white Gaussian noise (AWGN) with zero mean and two-sided power spectral density $N_0/2$ that models the effect of the receiver noise.

To maximize the signal-to-noise ratio, the received baseband signal is first applied to a chip-matched filter to produce the following signal samples at the chip rate:

$$z[k] = \int_{(k-1)T_c}^{kT_c} u(t)g(t)dt. \quad (3)$$

In conventional PN code acquisition schemes, the output of the chip-matched filter is correlated with a locally generated PN code with different offsets that cover the delay uncertainty region (possibly the whole PN code period) as follows:

$$y[i] = \sum_{k=0}^{N-1} z[k - i]c_k; \quad i = 0, 1, \ldots, N-1, \quad (4)$$

where the index i indicates the delay offset under test. The correlation results in (4) are used to estimate the energy at different delay offsets and a decision is made on the existence of the multipath delays based on the highest energy values. It

is also common to use a preset threshold where only energy values that exceed the threshold are declared as potentially correct multipath components while others are ignored. Note that in some cases, especially for very long PN codes, it is possible to perform the correlation over a fraction of the code length and the upper limit in (4) will be less than $N-1$.

The main objective of the acquisition system is to maximize the probability of detection while minimizing the probability of false alarm. Based on the outcome of the decision process, we can have one of the following events.

(i) Detection. This event occurs when the energy value exceeds the threshold and the estimated delay matches one of the actual delays of the multipath components in the received signal. We would like to maximize the detection probability to improve the performance of the RAKE receiver in detecting the transmitted data.

(ii) False Alarm. This event occurs when the energy value exceeds the threshold but the estimated delay did not match any of the actual delays of the multipath components. We would like to minimize the false alarm probability since the RAKE receiver would be using a signal that has no useful energy to detect the data.

(iii) Miss. This event occurs when the energy value is below the threshold but the delay offset has a correct multipath component. We would like to minimize such event since the RAKE receiver will not get all useful energy in detecting the data.

It is also noted that there are other performance criteria for evaluating code acquisition schemes, such as the mean acquisition time and the probability of achieving correct acquisition within a specified period of time [23].

3. Boolean Satisfiability

The last few years have seen significant advances in Boolean satisfiability (SAT) solving. These advances have lead to a successful deployment of SAT solvers in a wide range of problems in Engineering and Computer Science. Given a set of Boolean variables and a set of constraints expressed in product-of-sum form, the goal of SAT solver is to find a variable assignment that satisfies all constraints or prove that no such assignment exists. The term "Satisfiability" emerges from that fact that we are asked to find a satisfying assignment, while the term "Boolean" comes from the fact that such assignment consists of only *true* or *false* variable states.

The SAT problem is usually expressed in conjunctive normal form (CNF). A CNF formula φ on n binary variables $x_1, \ldots, x_n$ is the conjunction (AND) of m clauses $\omega_1, \ldots, \omega_m$ each of which is a disjunction (OR) of one or more literals, where a literal is the occurrence of a variable or its complement. A formula φ maps to a unique n-variable Boolean function $f(x_n, \ldots, x_n)$ [24]. Clearly, a function f can be represented by many equivalent CNF formulas. We will refer to a CNF formula as a *clause database* and use "formula" and "CNF formula" interchangeably.

A variable x is said to be *assigned* when its logical value is set to 0 or 1 and *unassigned* otherwise. A literal l is a *true* (*false*) literal if it evaluates to 1 (0) under the current assignment to its associated variable, and a *free literal* if its associated variable is *unassigned*. A clause is said to be *satisfied* if at least one of its literals is true, *unsatisfied* if all of its literals are false, *unit* if all but a single literal are set to false, and *unresolved* in the remaining cases. A formula is said to be satisfied if all its clauses are satisfied, and unsatisfied if at least one of its clauses is unsatisfied. In summary, the SAT problem is defined as follows. Given a Boolean formula in CNF, find an assignment of variables that satisfies the formula or prove that no such assignment exists.

In the following example, the CNF formula

$$\varphi = (a \vee b) \cdot \left(\overline{b} \vee c\right) \cdot (\overline{a} \vee c) \tag{5}$$

consists of 3 variables, 3 clauses, and 6 literals. The assignment $\{a = 1, b = 0, c = 0\}$ violates the third clause and unsatisfies φ, whereas the assignment $\{a = 1, b = 0, c = 1\}$ satisfies φ. Note that a problem with n variables will have 2^N possible assignments for the variables. The above example with 3 variables has 8 possible assignments.

Despite the SAT problem being NP-Complete [13], there have been dramatic improvements in SAT solver technology over the past decade. This has lead to the development of several powerful SAT algorithms that are capable of solving problems consisting of thousands of variables and millions of constraints. Such solvers include GRASP [18], zChaff [17], Berkmin [20], MiniSAT [16], and RSat [21]. In the next three subsections, we describe the basic SAT search algorithm, recent extensions to the SAT solver input, and the use of hardware with SAT.

3.1. Backtrack Search. Most modern complete SAT algorithms can be classified as enhancements to the basic Davis-Logemann-Loveland (DLL) backtrack search approach [25]. The DLL procedure performs a search process that traverses the space of 2^N variable assignments until a satisfying assignment is found (the formula is satisfiable), or all combinations have been exhausted (the formula is unsatisfiable). It maintains a *decision tree* to keep track of variable assignments and can be viewed as consisting of three main engines: (1) *Decision* engine that makes *elective* assignments to the variables, (2) *Deduction* engine that determines the consequences of these assignments, typically yielding additional *forced* assignments to, that is, implications of, other variables, and (3) *Diagnosis* engine that handles the occurrence of conflicts, that is, assignments that cause the formula to become unsatisfiable, and backtracks appropriately. An example of a decision tree is shown in Figure 1.

Recent studies have proposed the use of the *conflict analysis* procedure in the diagnosis engine [18]. The idea is whenever a conflict is detected, the procedure analyzes the variable assignments that cause one or more clauses to become unsatisfied. Such analysis can identify a small subset of variables whose current assignments can be blamed for

$$f(a,b,c,d) = (a \vee b \vee c) \cdot (a \vee b \vee \bar{c}) \cdot (\bar{a} \vee c \vee d) \cdot (\bar{a} \vee c \vee \bar{d}) \cdot (\bar{b} \vee \bar{c} \vee d) \cdot (\bar{b} \vee \bar{c} \vee d) \cdot$$

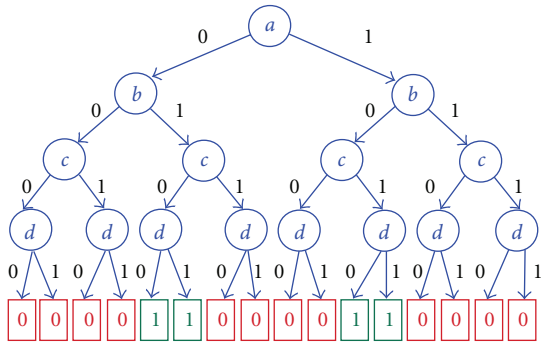

Figure 1: An example of a satisfiable SAT instance showing its corresponding decision tree.

the conflict. These assignments are turned into a *conflict-induced clause* and augmented with the clause database to avoid regenerating the same conflict in future parts of the search process. In essence, the procedure performs a form of learning from the encountered conflicts. Today, conflict analysis is implemented in almost all SAT solvers [16–18, 20, 21].

3.2. More Expressive Input. Restricting the input of SAT solvers to CNF formulas can restrict their usage in various domains. Therefore, researchers have focused on extending SAT solvers to handle stronger input representations. Specifically, SAT solvers [14–16, 19, 22] have recently been extended to handle pseudo-Boolean (PB) constraints which are linear inequalities with integer coefficients that can be expressed in the normalized form [14] of

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b, \tag{6}$$

where $a_i, b \in Z^+$ and x_i are literals of Boolean variables. Note that any CNF clause can be viewed as a PB constraint; for example, clause $(a \vee b \vee c)$ is equivalent to $(a + b + c \geq 1)$.

PB constraints can, in some cases, replace an exponential number of CNF constraints. They have been found to be very efficient in expressing "counting constraints" [14]. Furthermore, PB extends SAT solvers to handle *optimization* problems as opposed to only *decision* problems. Subject to a given set of CNF and PB constraints, one can request the minimization (or maximization) of an objective function which consists of a linear combination of the problem's variables:

$$\sum_{i=1}^{n} a_i x_i. \tag{7}$$

This feature has introduced many new applications to the SAT domain. Recent studies have also shown that SAT-based optimization solvers can in fact compete with the best generic integer linear programming (ILP) solvers [14, 15].

3.3. Hardware-Based SAT Solvers. Note that SAT solvers can be implemented in hardware. Several studies proposed the use of FPGA reconfigurable systems to solve SAT problems [26–29]. Hardware solvers could be a standalone or as an accelerator where the problem is partitioned between the hardware solver and the attached computer using software. Many different architectures were proposed to solve SAT problems in hardware. Linearly connected set of finite state machines, control unit, and deduction logic was proposed in [29]. The authors in [29] implemented their algorithm on Xilinx XC4028 FPGA. While in [26], the authors proposed a technique for modeling any Boolean expression. Their objective is to set the function output to 1. A backtrack algorithm is used to propagate the output back to the input and finding an assignment of the inputs to satisfy a logical 1 at the output.

The authors in [27] proposed an architecture for evaluating clauses in parallel. In their architecture, the clauses are separated into a number of groups and the deduction is performed in parallel. Then the results are merged together to allow the assignment to the variables.

A software/hardware solver for SAT was introduced in [28]. In their approach, they minimized the hardware compilation time which greatly reduced the total time to solve the problem. They also implemented their solver on an FPGA.

4. SAT Model for PN Code Acquisition

This section describes how to formulate the PN Code acquisition problem as a SAT instance to be able to process the received signal and find the delays of the L multipath components. As explained earlier, the received baseband signal is passed through a chip-matched filter to obtain the signal in (4). This signal contains delayed versions of the PN code (multipath components) plus a data part and noise. Since we are dealing with Boolean satisfiability (SAT), the first step is to convert the matched filter output to a binary sequence $z_b = \{z_b[0], z_b[1], \ldots, z_b[n-1]\}$ as follows:

$$z_b[i] = \begin{cases} 1, & z[i] \geq 0, \\ 0, & z[i] < 0. \end{cases} \tag{8}$$

Although hard decisions are in general not sufficient statistics for estimating the delay, but in the context of the developed SAT model for PN acquisition it would be enough to provide an estimate of the received PN code and hence allows for the SAT search to be implemented as will be discussed later.

The basic idea of the proposed algorithm is to locally generate a block of size n of the PN code using the known shift register (SR) structure with different initial states. A state is basically the content of the shift register at any instant of time. The SAT solver is used to find the initial state that would result in a PN sequence that is very close (ideally the same) to the received sequence z_b. Since an m-stage SR is used, then we will have $2^m - 1$ possible initial states to be tested. However, the SAT solver uses intelligent algorithms to efficiently traverse the decision tree and quickly find a valid solution as described in Section 3. Once a solution is

found, that is, finding an initial state of the SR that will result in the smallest difference (we also call it *distance*) between the locally generated PN code and the received sequence, the delay of the first multipath component is obtained from this initial state. The SAT solver then searches for the next initial state that would result in the next smallest distance to find the delay of the second multipath component. This process is repeated until all L multipath components are detected.

In order to illustrate how the state of the SR can be used to find the delay of a multipath component and without loss of generality, we assume a 2-stage SR used to generate a PN code of length $2^2 - 1 = 3$ chips as shown in Figure 2. Both stages are used to generate the feedback input to the SR through the XOR gate. Since we have two stages in the SR, there are 3 possible initial states, and once the SR is clocked at the chip rate, then the following states would be generated: $\{01, 10, 11\}$, $\{10, 11, 01\}$, or $\{11, 01, 10\}$ depending on which initial state was used. Suppose that the transmitter uses a PN code with initial state of 01 and the channel causes a delay of one chip, then the initial state of the PN code to be used by the receiver to match the received signal would be 10. On the other hand, if the channel causes a two-chip delay, then the solution for the initial state would be 11. Hence, we can estimate the channel delay based on the initial state of the SR that would result in best match with the received signal.

In order to use the advanced SAT solvers to find the L multipath delays in the received signal, the problem must be first expressed in the SAT solver input format as described in Section 3. To illustrate our approach, let us assume a system consisting of n received chips, and a Shift Register (SR) with m stages. The code length N is equal to $(2^m - 1)$ levels as shown in Figure 3.

Three sets of *variables* are defined for the problem as follow.

(i) A Boolean variable C_i is defined for each chip at the matched filter output at sample time i, that is, a total of n variables. A value of 1 or 0 for each variable indicates that the corresponding chip is a 1 or 0, respectively. Note that this variable is the same as the sequence z_b that was introduced in (8).

(ii) A Boolean variable Q_i is defined for each matched filter output as the difference between the C_i and the PN code chip, that is, a total of n variables.

(iii) A Boolean variable S_{ij} is defined for each SR stage i at each level j, that is, a total of $m \times (2^m - 1)$ variables.

Thus, the total number of needed Boolean variables is equal to $2n + m \times (2^m - 1)$.

The following set of CNF and PB constraints are generated.

(i) Received Chips Constraints. This constraint is used to set the input sequence utilized by the SAT solver to be compared with the locally generated PN code. The input sequence is obtained from (8). For each received chip i, its corresponding

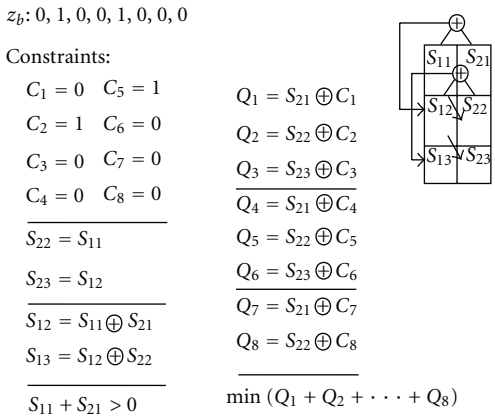

Figure 2: An example of a network with 8 data bits and 2 SR bits.

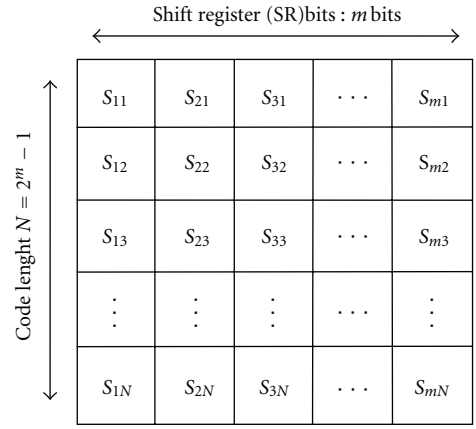

Figure 3: Sample layout of Shift Register bits.

C_i bit is set to 0 or 1 depending on the feed data. This can be expressed using a single PB constraint per chip as follows:

$$C_i = z_b[i]; \quad i = 0, 1, \ldots, n-1, \tag{9}$$

that is, a total of n PB constraints.

(ii) Initial State Constraints . This constraint is used to ensure that the initial SR state should have at least one bit assigned to 1 to avoid having an all-zero state for the SR. This can be expressed using a *single* PB constraint as follows:

$$\left(\sum_{i=1}^{m} S_{i1}^{k}\right) > 0; \quad k = 1, 2, \ldots, 2^m - 1. \tag{10}$$

(iii) Shifting Constraints. This constraint implements the shifting operation as the shift register is clocked; for example, $S_{22} = S_{11}, S_{32} = S_{21}, \ldots$, is expressed using the following equality constraint per SR stage:

$$(S_{il} = S_{(i-1)(l-1)}); \quad l = 2, \ldots, N; \; i = 2, \ldots, m. \tag{11}$$

This results in a total of $(m-1)$ $(2^m - 2)$ equality constraints. Each equality constraint of format $(x = y)$ can be expressed using two CNF constraints as shown in Table 1.

Table 1: Expressing logical constraints using CNF constraints.

Logical Constraint	CNF Constraint
$(x = y)$	$(\overline{x} \vee y) \cdot (x \vee \overline{y})$
$(x = y \bigoplus z)$	$(\overline{x} \vee y \vee z) \cdot (x \vee \overline{y} \vee z) \cdot (x \vee y \vee \overline{z}) \cdot (\overline{x} \vee \overline{y} \vee \overline{z})$

(iv) Feedback Constraints. This constraint ensures that the correct SR stages as used in the feedback part of the PN code generator. The PN code feedback relation is expressed using the following XOR constraint per initial SR content:

$$\left[S_{1l} = S_{p(l-1)} \bigoplus \ldots \bigoplus S_{q(l-1)}\right]; \quad l = 2, \ldots, N, \tag{12}$$

where $p, q \in \{1, \ldots, n\}$ is selected according to the feedback connection of the PN code generator. This results in a total of $(2^m - 2)$ XOR constraints. Each XOR constraint of format $(x = y \bigoplus z)$ is expressed using four CNF constraints as shown in Table 1.

(v) Difference Constraints. The mismatch between the received chip sequence and locally generated PN code, taken from the mth stage of the SR and for a given initial state k, is calculated as follows:

$$\left[Q_i^k = S_{mi}^k \bigoplus C_i\right]; \quad i = 1, \ldots, n;\ k = 1, 2, \ldots, 2^m - 1. \tag{13}$$

This results in n XOR constraints. As mentioned earlier each XOR constraint can be expressed using four CNF constraints.

(vi) Optimization Function. The objective of the SAT algorithm is to search through the possible initial SR states that results in minimizing the error (distance) between the received sequence and locally generated code. This is expressed using the following PB optimization objective:

$$D_k = \min\left(\sum_{i=1}^{n} Q_i\right); \quad k = 1, 2, \ldots, 2^m - 1. \tag{14}$$

The algorithm finds the smallest L values of the distance and the corresponding SR initial states. Then, the L multipath delays are estimated from the states as was explained earlier.

To further illustrate the formulation in SAT input, consider the example in Figure 2. The system consists of 8 data bits and 2 SR bits. Hence, the code length N is 3. The SAT problem generates a total of $2 \times 8 + 2(2^2 - 1) = 22$ Boolean variables. The figure displays the needed constraints.

5. Simulation Results

In this paper, we simulated a direct-sequence spread spectrum system with a PN code of length 2047 (11-stage shift register) operating over a frequency-selective Rayleigh fading channel with uniform power delay profile and a normalized Doppler of 10^{-3}. The Doppler frequency is normalized by the PN code length. Every simulation was repeated for 2000 independent trials. Although a square pulse shape was used for each chip, other pulse shaping methods may be used with no impact on the use of the proposed scheme. The number of paths is assumed to be three. The performance is measured by the probability of detecting at least one, two, or three multipath components as a function of the signal-to-noise ratio per chip (SNRc). This was done by finding the three minimum distances according to (14) and then checking if the initial shift register state corresponds to the correct delay or not. If the state matches the delay, then detection is declared; otherwise a miss is declared; Note that this is possible because we are performing simulation analysis, but in practice we expect to use a threshold to decide if a path exists or not. The effect of the duration of the correlation period used in calculating the difference between the locally generated PN code and the received data on the detection probability is also investigated. The performance is compared to that of a conventional energy-detector algorithm that measures the correlation at every possible offset and selects the energy of the three strongest paths. In these simulations, the Boolean Satisfiability (SAT) algorithm finds the delays of the three initial states of the SR that results in minimum error. All experiments were performed on an Intel Xeon 3.2 GHz workstation with 4 GB of RAM. We used the PBS 0-1 SAT-based ILP solver [14] for all experiments. Note that the above parameters were chosen for illustration purposes and are not expected to cause any restriction in the application of the proposed algorithm.

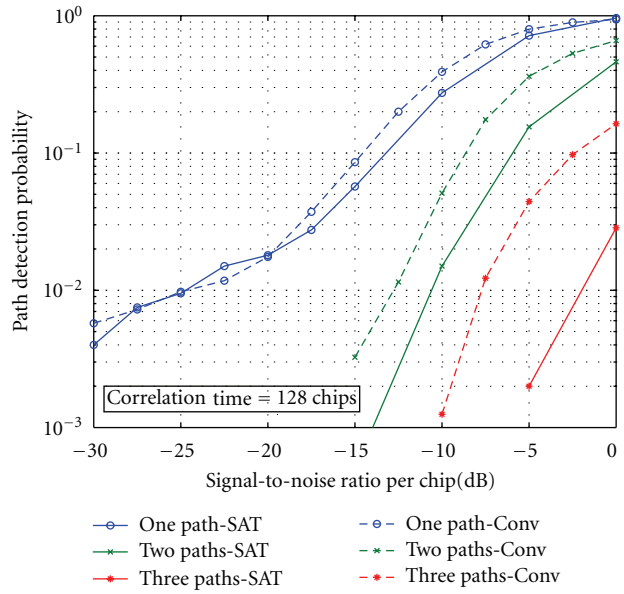

Figure 4: Probability of detection with 128 chips correlation.

Figure 4 shows the detection probabilities for a relatively short correlation period of 128 chips. It is clear that the multipath detection performance is relatively poor for both the SAT-based and conventional algorithms, although the latter shows better performance.

The multipath detection performance is shown in Figures 5, 6, and 7 for a correlation period of 256, 512, and 1024 chips, respectively. The results show that the performance

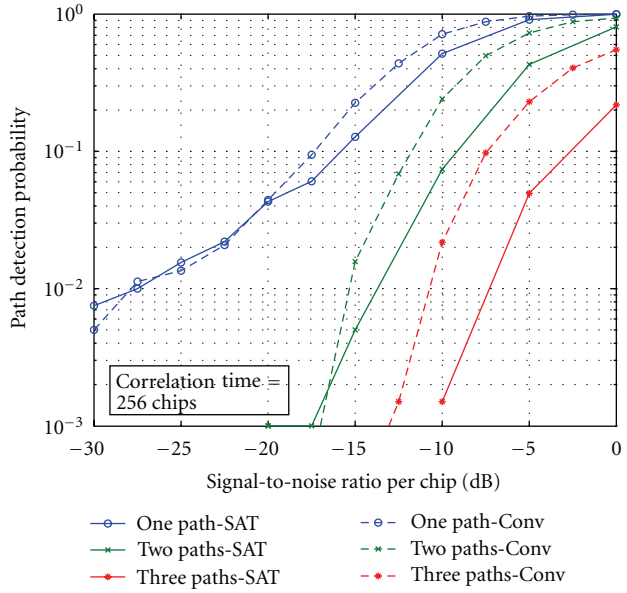

Figure 5: Probability of detection with 256 chips correlation.

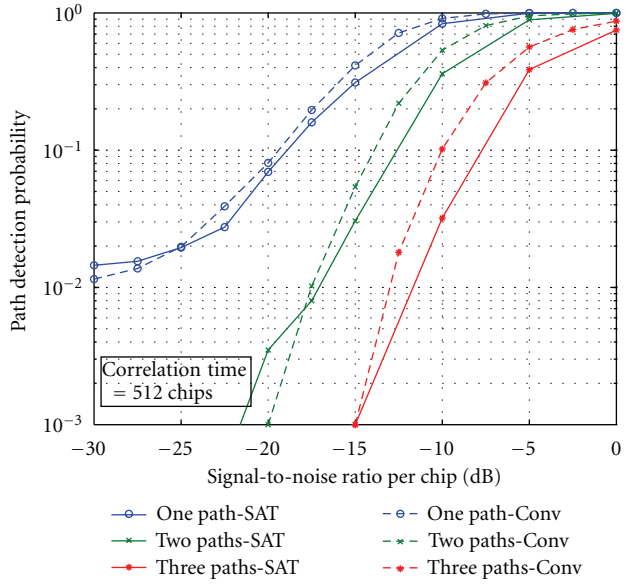

Figure 6: Probability of detection with 512 chips correlation.

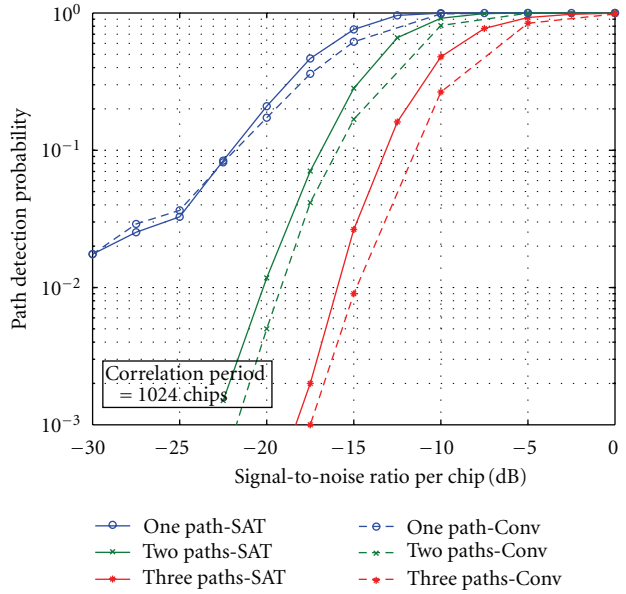

Figure 7: Probability of detection with 1024 chips correlation.

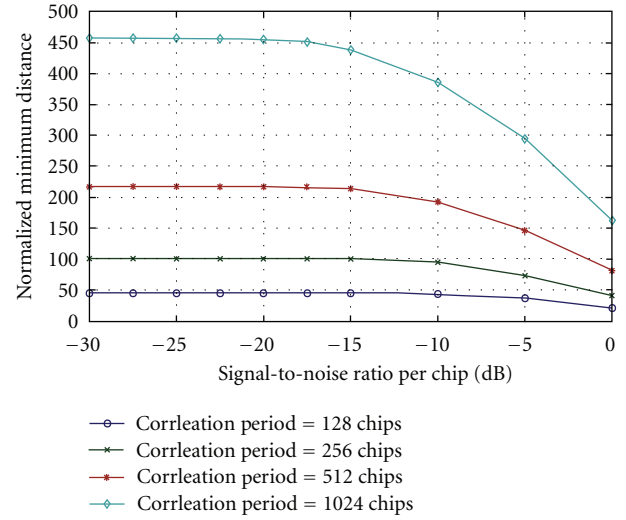

Figure 8: Minimum distance versus SNRc.

improved significantly to about 80% for detecting the three paths at an SNRc of zero dB. The detection of at least one or two paths is quite high indicating that the algorithm is successful in finding these delays. We also remark that as the correlation period increases, the SAT-based algorithm performance becomes closer to the conventional algorithm. Note that the SAT algorithm finds the correct delays by searching through the decision tree in an intelligent way and hence results in a reduced number of decisions compared to a brute force search strategy.

The SAT-based algorithm searched for the possible states that match the received signal with the PN code and the states that result in minimum difference, that is, the minimum distance between the received signal and locally generated sequence, is used to find the delay estimate. Figure 8 shows the minimum distance found at different values of the correlation period over an AWGN channel. It is observed that the difference tends to decrease as the SNRc increases because the SAT algorithm is supplied with more reliable data for the search.

Finally, we notice that it is difficult to make a direct comparison of the computational cost between the proposed SAT-based algorithm and the conventional correlation based since the metrics used by the algorithms are different. In particular, the conventional scheme uses the number of multiplications and additions needed to search for the multipath components and this is typically quantified as N^2 multiply-and-add operations where N is the PN code length

and assumed to be the search window. For the proposed SAT scheme, the complexity is measured by the number of decisions made while traversing the decision tree to look for a valid solution to the instance. The SAT solver uses advanced algorithms to intelligently traverse the decision tree and eliminate unsatisfiable paths. Depending on the instance's constraints, the SAT solver might be able to find a solution faster for some instances than others. In our simulations, most instances were solved after N decisions.

6. Conclusions

A new multipath detection algorithm using Boolean satisfiability (SAT) techniques has been presented. The SAT-based algorithm uses the deterministic structure of the PN spreading code to perform an intelligent search for the possible propagation delays. Simulation results showed that the proposed scheme was successful in providing correct delay estimates with high reliability over a multipath frequency-selective Rayleigh channel.

References

[1] M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread Spectrum Communications*, McGraw-Hill, New York, NY, USA, 1994.

[2] O. S. Shin and K. B. Lee, "Utilization of multipaths for spread-spectrum code acquisition in frequency-selective Rayleigh fading channels," *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 734–743, 2001.

[3] W. Suwansantisuk and Z. Win, "Multipath aided rapid acquisition: optimal search strategies," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 174–193, 2007.

[4] W. Suwansantisuk, M. Z. Win, and L. A. Shepp, "On the performance of wide-bandwidth signal acquisition in dense multipath channels," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 5, pp. 1584–1594, 2005.

[5] L. L. Yang and L. Hanzo, "Serial acquisition performance of single-carrier and multicarrier DS-CDMA over nakagami-m fading channels," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 692–702, 2002.

[6] A. Biere, A. Cimatti, E. M. Clarke, M. Fujita, and Y. Zhu, "Symbolic model checking using SAT procedures instead of BDDs," in *Proceedings of the 36th Annual Design Automation Conference (DAC '99)*, pp. 317–320, June 1999.

[7] G. J. Nam, F. Aloul, K. Sakallah, and R. Rutenbar, "A comparative study of two boolean formulations of FPGA detailed routing constraints," in *Proceedings of the International Symposium on Physical Design (ISPD '01)*, pp. 222–227, April 2001.

[8] F. Aloul, S. Hassoun, K. Sakallah, and D. Blaauw, "Robust SAT-based search algorithm for leakage power reduction," in *Proceedings of the International Workshop on Power and Timing Modeling, Optimization, and Simulation (PATMOS '02)*, pp. 167–177, 2002.

[9] A. Sagahyroon and F. A. Aloul, "Using SAT-based techniques in power estimation," *Microelectronics Journal*, vol. 38, no. 6-7, pp. 706–715, 2007.

[10] F. A. Aloul and N. Kandasamy, "Sensor deployment for failure diagnosis in networked aerial robots: a satisfiability-based approach," in *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT '07)*, vol. 4501 of *Lecture Notes in Computer Science*, pp. 369–376, Springer, 2007.

[11] M. Mneimneh, F. Aloul, C. Weaver, S. Chatterjee, K. Sakallah, and T. Austin, "Scalable hybrid verification of complex microprocessors," in *Proceedings of the 38th Design Automation Conference (DAC '01)*, pp. 41–46, June 2001.

[12] N. Creignou, S. Kanna, and M. Sudan, *Complexity Classifications of Boolean Constraint Satisfaction Problems*, SIAM, 2001.

[13] S. A. Cook, "The complexity of theorem proving procedures," in *Proceedings of the Symposium on the Theory of Computing*, pp. 151–158, 2004.

[14] F. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah, "Generic ILP versus specialized 0-1 ILP: an update," in *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD '02)*, pp. 450–457, November 2002.

[15] D. Chai and A. Kuehlmann, "A fast pseudo-boolean constraint solver," in *Proceedings of the 40th Design Automation Conference (DAC '03)*, pp. 830–835, June 2003.

[16] N. Een and N. Sorensson, "An extensible SAT-solver," in *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT '03)*, pp. 502–508, 2003.

[17] K. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: engineering an efficient SAT solver," in *Proceedings of the 38th Design Automation Conference (DAC '01)*, pp. 530–535, June 2001.

[18] J. P. Marques-Silva and K. A. Sakallah, "GRASP: a search algorithm for propositional satisfiability," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 506–521, 1999.

[19] H. M. Sheini and K. A. Sakallah, "Pueblo: a modern Pseudo-Boolean SAT solver," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE '05)*, pp. 684–685, March 2005.

[20] E. Goldberg and Y. Novikov, "BerkMin: a fast and Robust SAT-solver," in *Proceedings of the Design Automation and Test in Europe Conference (DATE '02)*, pp. 142–149, 2002.

[21] K. Pipatsrisawat and A. Darwiche, "A new clause learning scheme for efficient unsatisfiability proofs," in *Proceedings of the Conference on Artificial Intelligence*, pp. 1481–1484, July 2008.

[22] H. E. Dixon and M. L. Ginsberg, "Inference methods for a pseudo-Boolean satisfiability solver," in *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI '02)*, pp. 635–640, August 2002.

[23] W. Suwansantisuk, M. Chiani, and M. Z. Win, "Frame synchronization for variable-length packets," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 52–69, 2008.

[24] J. Hayes, *Introduction to Digital Logic Design*, Addison-Wesley, 1993.

[25] M. Davis, G. Longman, and D. Loveland, "A machine program for theorem proving," *Journal of the ACM*, vol. 5, no. 7, pp. 394–397, 1962.

[26] M. Abramovici and D. Saab, "Satisfiability on reconfigurable hardware," in *Proceedings of the International Workshop on Field Programmable Logic and Application (FPL '97)*, pp. 448–456, 1997.

[27] A. Dandalis and V. K. Prasanna, "Run-time performance optimization of an FPGA-based deduction engine for SAT solvers," *ACM Transactions on Design Automation of Electronic Systems*, vol. 7, no. 4, pp. 547–562, 2002.

[28] I. Skliarova and A. B. Ferrari, "A software/reconfigurable hardware SAT solver," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 4, pp. 408–419, 2004.

[29] P. Zhong, M. Martonosi, P. Ashar, and S. Malik, "Using configurable computing to accelerate Boolean satisfiability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 6, pp. 861–868, 1999.