Towards Optimally Exploiting Physical Layer Information in OFDM Wireless Networks [*]

Invited paper

Zheng Zeng and P. R. Kumar
University of Illinois at Urbana Champaign
{zzeng2, prkumar}@illinois.edu

ABSTRACT

Wireless communication is inherently vulnerable in nature and packets can be corrupted due to various reasons. The network performance can potentially be improved if one is able to identify the reasons for packet corruption and react to them accordingly. However, none of the current wireless protocols do so. In this paper, we design a novel scheme to diagnose packet corruption in OFDM wireless networks by statistically analyzing certain available physical layer information, based on the observation that different causes of corruption result in different per-symbol-SINR patterns within a packet. Our approach introduces no additional traffic overhead. By running experiments on a GNU radio test bed in an electromagnetic anechoic chamber, we study and demonstrate that one can separate three reasons for packet corruption: weak signal, interference from transmitters using the same band, and interference from transmitters using an adjacent band. We study three statistical classification methods and compare their accuracy.

1. INTRODUCTION

The Open Systems Interconnection Basic Reference Model (OSI model) [16], which is used to guide the design of most network communication protocols, is a strictly layered architecture with each layer providing service to its upper layer but hiding as many details as possible to maintain isolation between layers. In wireless networks, for example, the MAC layer deals with addressing and multiplexing on multi-access media, while the physical layer handles only the issue of how to transmit/receive over the medium, i.e., how to communicate with a single device. Current physical layer receivers only provide a packet to the MAC layer if it is successfully received, or discard it otherwise. There is very little additional information provided by the physical layer to the upper layers in commodity wireless cards. Lately, however, the wireless networking community has evinced increasing interest in exporting information gathered by the physical layer to solve certain MAC level problems. Most current work is aimed at answering the question: Is it possible to obtain and export physical layer information from a packet that is *not* successfully received? If so, how do we do so?

This paper is addressed at answering to what extent available physical layer information can be exploited. If a packet is not received correctly, there are two most common reasons. The first cause is due to a weak received signal. This happens when the signal strength of the sender at the receiver side is not strong enough to overcome the noise caused by fading or processing circuits of the hardwire. The second cause is interference. Interference results from one or more concurrent transmissions from other devices, and it may vary in time even within the duration of a single packet transmission. Note that interference need not just come from other devices within the same wireless system, or even not spread over the same frequency spectrum. In this paper, however, by "interference" we will refer to transmissions over the same spectral band, unless specifically suggested otherwise. Separating this "same-band" interference from a weak signal is our primary interest. We will also study and discuss "overlapping-band" interference in Section 6 to complete our work. The specific question we want to address is: *By only analyzing certain available physical layer information in received data, can we separate the cause of packet corruption between weak signal and interference? Moreover, can we do so in a way that introduces no communication overhead and minimal computational overhead?* The answer to this question would depend on the specific physical layer design. In this paper we study the OFDM system.

Being able to identify the cause of packet corruption and exporting it to upper layers is of significant help in many ways. We identify two of them. First, the MAC layer can make more intelligent decisions when encountering corruptions. The best reaction that the MAC layer should take to different kinds of corruption differs. For example, if the cause is a weak signal, then the sender could use a lower data rate in the retransmission without backoff; on the other hand, if the cause of packet corruption is interference from transmitters within the same wireless system, then the sender could backoff but retransmit at the same rate; Finally, if it is interference from other systems (say, ultra-wide-band devices) whose spectrum overlaps with the transmitter-receiver pair, but only interferes over a few sub-carriers, then the retransmission would better be done without backoff, at the same rate, but avoiding using those interfered sub-carriers. Therefore, by allowing the link to always respond to packet corruption in the best way, and always operate at the "best" setting in different situations, the network performance can potentially be improved.

A second major benefit of such packet classification is that it can be used to provide a more accurate time-varying network topol-

[*]This material is based upon work partially supported by USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287, and NSF under Contract Nos. IIS-0720482, CNS-0423431, ECCS-0701604, CNS-07-21992, CNS-0626584, and CNS-05-19535, and CCR-0325716.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WICON'08 November 17-19, 2008, Maui, Hawaii, USA
Copyright 2008 ACM ICST 978-963-9799-36-3 ...$5.00.

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

ogy/conflict graph on the fly. Scheduling problems in wireless network [3][6] have been studied for a long time. Many of the solutions are based on a *given* conflict graph which defines the relation between any two links, whether they interfere with each other or not, and in practice such conflict graphs are usually gained through one-time measurement only, as suggested in [10]. However, the conflict graph may be time-varying since the channel condition between nodes is time-varying. It is possible that two links that do not interfere in the morning do interfere with each other in the afternoon in a medical environment (which is one of the motivations for our study). Therefore a static conflict graph does not suffice. However, with proper coordination, feedback and diagnosis on when and which packets get corrupted due to interference, we can modify the interference graph in real-time, while introducing no overhead traffic when the conflict graph remains static. This can potentially result in better performance.

The key contribution of this paper is the design of a classifier called *Packet Corruption Classifier (PCC)*, which immediately identifies the cause of a packet corruption, by analyzing the pattern of certain physical layer information in OFDM system. The analysis can be done in parallel with the normal packet de-modulation process at the receiver side. We study and implement three algorithms in *PCC* to separate the causes of corruption, and compare their accuracy based on experiments held in an electromagnetic anechoic chamber using a GNU radio test bed.

The rest of this paper is organized as follows. After briefly introducing OFDM in Section 2, we formulate the corruption diagnosis problem as a change point problem, and present *PCC* in Section 3. Section 4 describes the set up of the experiment. The performance of *PCC* is evaluated in Section 5, followed by a discussion on how to choose the proper classification algorithm. Section 6 expands *PCC* to identify "overlapping-band" interference. In Section 7 we discuss the related work, and finally conclude in Section 8.

2. BACKGROUND MATERIAL

In this section, we first present a brief description of the OFDM physical layer, and define notation that will be used throughout the paper. Then we identify and explain the key observation that leads to the design of PCC.

2.1 An overview of OFDM physical layer

OFDM, short for *Orthogonal Frequency Division Multiplexing*[4], is an FDM modulation technique for transmitting large amounts of digital data over radio. OFDM works by splitting the radio signal into multiple smaller sub-signals that are then transmitted simultaneously at different frequencies, which are defined as *sub-carriers*, to the receiver.

Figure 1 gives a (partial) block diagram of an OFDM receiver. The input is a sequence of sampled signals after the signal has been converted to baseband, while the output is a packet queue, or, say, a bit stream. One key element in the receiver is the *slicer*, which quantizes the sampled signal to one of a few symbols $\{\hat{a}_k\}$ in the form of complex numbers. The set $\{\hat{a}_k\}$ is determined by modulation, and usually we have $|\hat{a}_k| = 1$. In current off-the-shelf devices, the physical layer exports only the final bit stream, which results in the isolation of the upper layers from implementation details of the physical layer. In this paper, we seek to identify additional information related the physical layer implementation that breaks this isolation, but helps estimate channel condition and interference. It turns out that such infomation can be gained from the input and output stream of the *slicer*.

Figure 2 illustrates the composition of a packet in an OFDM receiver. The *x*-axis is the time-line while the *y*-axis is the *sub-carrier* index. We use m to denote the number of *sub-carriers*. A *symbol* refers to a collection of signals modulated in single unit of time synchronously across all m *sub-carriers*, i.e., a column in Figure 2. We use S_{ij} to represent the content at the j-th sub-carrier in the i-th symbol of a packet before it enters the *slicer*, and Y_{ij} after that. It follows that $S_{ij} \in R^2$ and $Y_{ij} \in \{\hat{a}_k\}$. The distance between them is defined as

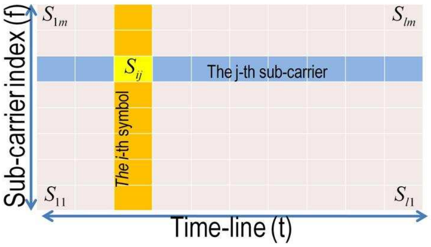


Figure 2: An OFDM packet composition

$$X_{ij} := Y_{ij} - S_{ij}. \tag{1}$$

Note that $1/|X_{ij}|$ is a good approximation of SINR (Signal to Interference plus Noise Ratio).

2.2 Hint from constellation graph

The following three assumptions guide the design of PCC.

(A1) When there is no interference, the channel condition remains invariant throughout a packet reception.

(A2) All sub-carriers share the same channel condition.

(A3) When the cause of corruption is interference, we only consider the situations when interference starts or ends somewhere during the packet reception.

(A1) implies that we don't consider fast-fading, which is reasonable since our target scenario is an indoor environment without high speed mobility. **(A2)** implies that we don't consider the corruptions caused by interference whose duration entirely *covers* the duration of packet reception. **(A2)** does not hold when de-modulation requires higher SINR than the OFDM synchronization procedure, and we are considering relaxing **(A2)** in future work.

Given the three assumptions, we hypothesize the following conjecture. If a packet corruption is caused by a weak signal, $\{X_{ij}\}$ conforms to one single distribution. On the other hand, if a packet corruption is caused by interference, then $\{X_{ij}\}$ from the non-interfered part conforms to one distribution, while $\{X_{ij}\}$ from the interfered part conforms to another. We have obtained experimental results from the GNU radio test bed that support this conjecture. Figure 3 shows the constellation graph of a packet that is ten symbols long transmitting on 128 sub-carriers. The interference starts in the middle of the reception. The left figure plots $\{S_{ij}|0 < i < 3\}$ while the right one plots $\{S_{ij}|7 < i < 11\}$ and the difference can be seen to be dramatic. The important conclusion is that by following this conjecture, the original problem can be formulated as a *change point problem*.

3. PROBLEM FORMULATION AND PCC DESIGN

3.1 The change point problem

We start with the formal statement of a general change point problem. Let $S = \{x_1, x_2, ..., x_n\}$ be a sequence of independent

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

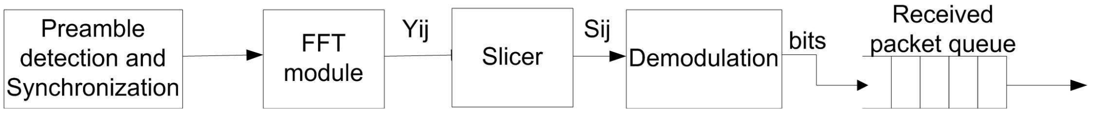
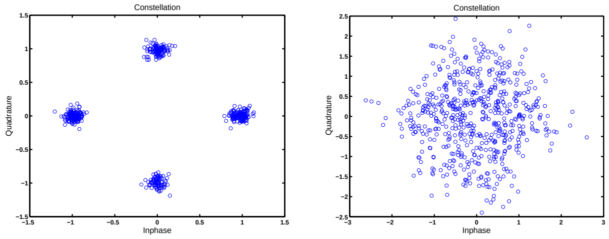


Figure 1: Block diagram of an OFDM receiver



Before Interference

During Interference

Figure 3: Consellation graph of an interfered packet

random variables, with probability distributions $F_1, F_2, ..., F_n$, respectively. Then the problem is to test the following null hypothesis:

$$H_0 : F_1 = F_2 = ... = F_n, \tag{2}$$

versus the alternative

$$H_q : F_1 = ... = F_{k_1} \neq F_{k_1+1} = ... = F_{k_q} \neq F_{k_q+1} ... = F_n, \tag{3}$$

where $1 < k_1 < k_2 < ... < k_q < n$, q is the unknown number of change points, and $k_1, k_2, ..., k_q$ are the respective unknown positions that have to be estimated.

For our problem, S consist of $|X_{ij}|$, where $n = m \times l$, and $x_{(i-1)\times m+j} = |X_{ij}|$. There are three hypothesis to be tested. First, if the packet is corrupted due to weak signal, H_0 defined in (2) holds. Second, if the packet is corrupted due to interference that either (i) starts before the packet reception and ends during the packet reception or (ii) starts during the packet reception and lasts till the end of reception, then there is only one change point and H_1 holds, with

$$H_1 : F_1 = ... = F_{k_1} \neq F_{k_1+1} = ... = F_n. \tag{4}$$

Last, if the packet is corrupted due to interference that both starts and ends during the packet reception, then there are two change points and H_2 holds, with

$$H_2 : F_1 = ... = F_{k_1} \neq F_{k_1+1} = ... = F_{k_2} \neq F_{k_2+1} = ... = F_n. \tag{5}$$

The goal is to test H_0 against H_1 plus H_2. We note that it is not needed to separate H_1 from H_2, i.e., we only need to test *whether* there exists a change point(s) or not.

3.2 The Packet Corruption Classifier (PCC) algorithm

The traditional change point solution is to go through the whole sequence S and test each variable x_i to determine whether it is the change point. This requires a lot of computation. Since our goal is to only search for the existence of one change point, we have designed the following simple sliding window algorithm to reduce the computational complexity. Let SW_1 and SW_2 represent two adjacent subsequences of S, each of which has l_w variables, i.e.,

$$SW_1 = \{x_{i+1}, x_{i+2}, ..., x_{i+l_w}\}, \tag{6}$$

$$SW_2 = \{x_{i+l_w+1}, x_{i+l_w+2}, ..., x_{i+2\times l_w}\}. \tag{7}$$

Then the hypotheses become

$$H'_0 : F_1 = ... = F_{i+2\times l_w}, \tag{8}$$

$$H'_1 : F_1 = ... = F_{i+l_w} \neq F_{i+l_w+1} = ... = F_{i+2\times l_w}. \tag{9}$$

We move SW_1 and SW_2 throughout S with step size t, and test H'_0 against H'_1 for each move. If H'_0 holds for every move, then we accept that the packet corruption is due to a weak signal. Otherwise, we say it is due to interference. The following pseudocode shows the details with return value true if it is interference, and false otherwise.

```
procedure $PCC(t, l_w)$
Interference=False
$SW_1 = x_{i+1}, x_{i+2}, ..., x_{i+l_w}$
$SW_2 = x_{i+l_w+1}, x_{i+l_w+2}, ..., x_{i+2\times l_w}$
while $SW_2$ stays in $S$ do
    test $H'_0$ against $H'_1$
    if $H'_1$ holds then
        Interference=True
        Jump out of the loop
    end if
    Move $SW_1$ and $SW_2$ $t$ variables toward the end of $S$
end while
Return Interference
```

Note that signals in one symbol are modulated in a single unit of time synchronously, so that we can assume the variables in one symbol conform to one distribution. Therefore the change point can only be located between symbols, i.e., between $X_{i,m}$ and $X_{i+1,1}$ for some i, so that it is safe to set the step size t to be m instead of 1, and set the window size l_w to be m or a multiple of m.

3.3 Evaluation Metrics

In the next section we discuss the tests that will be used to separate H'_0 from H'_1. Before that, it is necessary to first define the metrics that evaluate the test algorithms. If our goal is to detect interfered packets from among all corrupted ones, then there are two metrics:

False positive rate: This is the percentage of corrupted packets which the test identifies as interfered, from among all the packets whose real corruption reason is weak signal.

Miss Rate: This is the percentage of corrupted packets which the test identifies as weak signal, from among all the packets whose real corruption reason is interference.

3.4 Hypothesis tests

We apply and study three Hypothesis tests: (1) Change of Mean Test (CMT); (2) Chi-square Test of Independence (CST); and (3) Mann-Whitney-Wilcoxon (MWW). Although using different techniques, they share the same structure. The tests output an indicator result $I \in R$ for each sliding window operation, and compare it to

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

Table 1: Contingency table

Window	Cell 1	Cell 2	...	Cell k	Row
Index				Total)	
1	f_{11}	f_{12}	...	f_{1k}	l_w
2	f_{21}	f_{22}	...	f_{2k}	l_w
Column Average	E_1	E_2	...	E_k	

a pre-determined threshold I^{th}. If $I \leq I^{th}$, then H_0' is accepted; otherwise H_1' is accepted. We describe the three tests by defining their indicators I_{cmt}, I_{cst}, I_{mww}, respectively.

3.4.1 Change of Mean Test (CMT)

In the following discussion, F_i refers to the distribution of X_{ij} with interference which has mean μ_i and variance σ_i^2; while F_n refers to the distribution of X_{ij} without interference with mean μ_n and variance σ_n^2. Similarly, F_a refers to distribution of SW_1, while F_b refers to distribution of SW_2. *CMT* is motivated by the observation that $\mu_i \neq \mu_i$ when one window is interfered with while the other is not. Let $\bar{x}_a$ denote the mean value of variables in SW_1, and $\bar{x}_b$ denote the mean value of variables in SW_b. The indicator is defined as

$$I_{cmt} := |\bar{x}_a - \bar{x}_b|. \quad (10)$$

It is clear that I_{cmt} tends to be smaller when $F_a = F_b$, and larger when $F_a \neq F_b$. We further discuss *CMT* performance in Section 3.5.

3.4.2 The Chi-square Test of Independence (CST)

The chi-square test of independence is used to test whether two categories (each with many cells or groups) are related or not related (independent). In order to qualify for the test, we first need to constructed a *contingency table* based on SW_1 and SW_2, as in Table 1. A *cell* is a continuous range on R, f_{ij} is the number of variables from sliding window i whose value drops in *cell* j, and $Col_j := f_{1j} + f_{2j}$. k is a tuneable value whose setting determines the number of *cells*. In our experiment we set $k = 6$ where *cellj* $:= [(j-1)/5, j/5)$, for $j = 1, 2, ..., 5$ and *cell*$6 := [1, \infty)$. The indicator is defined as the χ^2 value

$$I_{cst} := \Sigma_{j=1}^{k} \frac{(f_{1j} - E_j)^2}{E_j} + \Sigma_{j=1}^{k} \frac{(f_{2j} - E_j)^2}{E_j}. \quad (11)$$

Again, the smaller I_{cst}, the more likely that $F_a = F_b$.

3.4.3 Mann-Whitney-Wilcoxon Test (MWW)

The Mann-Whitney-Wilcoxon test is a non-parametric test for assessing whether two samples of observations come from the same distribution. It is one of the best-known non-parametric significance tests. It works as follows.

First, we arrange all the variables from SW_1 and SW_2 into a single ranked series. That is, we rank all the variables, regardless of which sliding window they are in. Then we add up the ranks for the variables which come from SW_1 and denote their sum by R_1; similarly, we add up the ranks for the variables which came from SW_2, and denote their sum by R_2. Define the indicator as

$$I_{mww} := \max(R_1, R_2). \quad (12)$$

It is known that if two populations share the same distribution, the probability of an observation from one population exceeding an observation from the second population is 0.5. Based on this fact, R_1 and R_2 tend to be closer, which means I_{mmw} is smaller when $F_a = F_b$.

3.4.4 Deciding the threshold I^{th}

The value of I^{th} directly affects the performance of PCC. Instead of choosing I^{th} from statistical tables, we set I^{th} through training sets obtained from controlled experiments. Given a group of corrupted packets for which we already know the reason of corruption, we specify a false positive rate threshold β, and then choose the corresponding threshold I_β^{th} to be the smallest real number such that the false positive rate of the training set is less than β. In the wireless protocol context, this design leaves the control to the actual link manager, so that it can choose the balance between false positive rate and hit rate, based on the application on its own preference.

3.5 Discussion on performance of the tests

The performance of all three statistical tests is affected by F_i and F_n, the parameters of PCC (such as sliding window size l_w), and even packet length. It will be helpful if we can analytically determine how well PCC works under certain scenarios. In other words: Given a certain threshold I_β^{th}, is it possible to approximately estimate the corresponding false positive rate and the miss rate mathematically? In this section, we will answer this question when PCC uses *CMT*. The analysis of *CST* and *MWW* can be done similarly.

3.5.1 The false positive rate

To facilitate the derivation of parameters of interest, for all corrupted packets due to weak signal, we assume that (i) the ground noise (relative to signal strength) for every packet conforms to Rayleigh distribution $Rayleigh(\sigma)$, and (ii) every such packet is l symbols long. Instead of determining the false positive rate for a given I^{th}, it is equivalent to find out what is the probability that PCC identifies one "weak signal" packet as interfered. Note that X_{ij} can be regarded as $N_{ij} + I_{ij}$, where N stands for noise and I stands for interference, with $I_{ij} = 0$ when there there is no interference. Therefore $F_n = Rayleigh(\sigma)$. According to the central limit theorem (CLT) which states that the sum of a sufficiently large number of identically distributed independent random variables each with finite mean and variance will be approximately normally distributed, we may approximate

$$\bar{x}_a, \bar{x}_b \sim N(\sigma\sqrt{\frac{\pi}{2}}, \frac{(4-\pi)\sigma^2}{2l_w}). \quad (13)$$

Then, because the difference between two normally distributed random variables is also normally distributed, it follows that

$$(\bar{x}_a - \bar{x}_b) \sim N(0, \frac{(4-\pi)\sigma^2}{l_w}). \quad (14)$$

For each run of CMT, the probability that H_0' is accepted is

$$p_1 = p(H_0') = p(|\bar{x}_a - \bar{x}_b| < I^{th}) = 2 \times \Phi(\frac{I_\beta^{th}}{\sqrt{(4-\pi)/l_w}\sigma}) - 1. \quad (15)$$

Since the step size of the sliding window is m, CMT is run $(l-1)$ times for each packet. Therefore the probability that PCC mistakes a "weak signal" packet as interfered is

$$p_f = 1 - p_1^{l-1}, \quad (16)$$

which is therefore the false positive rate. It easily follows from the derivation that given I^{th}, p_f is smaller with a lower ground noise level (smaller σ), shorter packets (smaller l), and larger sliding window size l_w.

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

3.5.2 The miss rate

Again, to facilitate the derivation of parameters of interest, for all corrupted packets due to interference, we assume that (i) the ground noise (relative to signal strength) for every packet conforms to Ray-leigh distribution $Rayleigh(\sigma_n)$, (ii) every packet is l symbols long, and (iii) $|X_{ij}|$ from the interfered part conforms to distribution F_i with mean μ_i and variance σ_i^2. When the sliding window moves to the change point, and assuming that SW_2 is interfered with, similar to the derivation in previous section, we obtain

$$\bar{x}_a \sim N(\sigma\sqrt{\frac{\pi}{2}}, \frac{(4-\pi)\sigma^2}{2l_w}), \tag{17}$$

$$\bar{x}_b \sim N(\mu_i, \frac{\sigma_i^2}{l_w}), \tag{18}$$

$$(\bar{x}_a - \bar{x}_b) \sim N(\mu_d, \sigma_d^2), \tag{19}$$

where $\mu_d = \mu_i + \sigma\sqrt{\frac{\pi}{2}}$ and $\sigma_d^2 = \frac{2\sigma_i^2+(4-\pi)\sigma^2}{2l_w}$. Therefore, the probability that PCC mistakes the packet as having a weak signal is

$$\begin{aligned} p_m &= 1 - p(H_1') = 1 - p(|\bar{x_a} - \bar{x_b}| < I^{th}) \\ &= 1 - \Phi(\frac{I_\beta^{th} - \mu_d}{\sigma_d}) + \Phi(\frac{-I_\beta^{th} - \mu_d}{\sigma_d}). \end{aligned} \tag{20}$$

It easily follows from the derivation that given I^{th}, p_m is smaller with lower ground noise level (smaller σ_n), stronger interference (larger μ_i), and larger sliding window size l_w.

4. EXPERIMENTAL SETUP

4.1 GNU testbed

We have evaluated PCC in a three-node GNURadio testbed. Each node is a commodity PC connected to a USRP GNU radio [1].

(a) Hardware and Software Environment: We use the Universal Software Radio Peripheral (USRP) for our RF sender/receiver, and the RFX2400 daughterboards which operate in the 2.4 GHz range. The OFDM software implementation for the signal processing blocks is from the open source GNU Radio project [2].

(b) Modulation. The OFDM implementation uses the modulation/demodulation module as a black-box, and works with a variety of modulation schemes. In our experiment, however, we only use differential quadrature phase-shift keying (DQPSK). The reason is that the PN synchronization implementation [13] of our software has no fine-synchronization stage. DPSK requires less accurate synchronization than non-differential modulation/demodulation schemes; therefore we choose DQPSK to minimize the effect of inaccurate synchronization.

(c) Configuration Parameters. We use the default GNU Radio configuration, i.e., on the transmitter side the DAC rate is 128e6 samples/ s, the interpolation rate is 128. On the receiver side, the ADC rate is 64e6 samples/s, and the decimation rate is 64. The number of sub-carriers is 102. Each packet consists of a PN preamble, a 300-byte (or 600-byte) payload, and 32-bit CRC. This implies that the body of each packet is 12 (or 24) symbols long.

4.2 Experimental environment

We have run the experiments in the Illinois Wireless Wind Tunnel (iWWT) [14], an electromagnetic anechoic chamber (Figure 4 shows the inside view of iWWT). An anechoic chamber is a shielded structure, with two important properties: (i) shielding prevents sources external to the chamber from interfering with reception at hosts within the chamber; and (ii) the anechoic chamber is lined internally with absorbing foam panels, which reflect minimal energy. Due to the second property, the walls of the chamber become essentially “invisible” to the devices inside the chamber. We have chosen iWWT as our experimental environment for property (i), because we must have full control of interference sources in order to establish the "ground truth" about the reason for a packet corruption. In iWWT, there are no interference sources except those that we deliberately input.

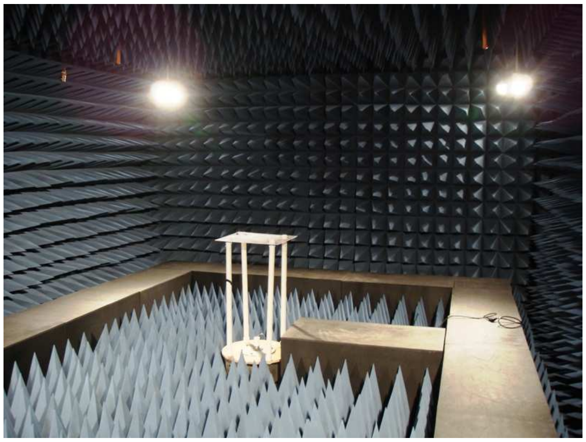


Figure 4: The inside of an anechoic chamber

4.3 Scenario design

All the experiments were conducted in iWWT, and the collected results are used as training sets for PCC. T_1 and T_2 are two transmitters and R is the receiver. T_1 and T_2 transmit over the same spectral band, but append different preambles to the packets they send, so that R can only detect the packets from T_1. We create two scenarios, with the second scenario divided into four categories:

Scenario A: Weak signal. T_2 is shut down in this scenario. We fix the distance between T_1 and R, and change the received signal power at R by tuning T_1's transmission power P_1. We gradually decrease P_1 to find (i) the threshold level (denoted by P_{no}) when R detects no packets from T_1, and (ii) the level (denoted by P_c) when almost all packets that R received are corrupted. We say a packet is corrupted if its bit error rate (BER) is higher than 1%. We assume a packet with (BER) lower than 1% can be recovered by forward error correction that is used by almost all current wireless protocols. Then we set P_1 to some value between P_{no} and P_c, and let T_1 transmit until R receives enough corrupted packets. All the packets are logged for further processing. We ran the experiments several times with different P_1 values within $[P_{no}, P_c]$. For each power level, we run the experiment twice, sending packets of size 300 bytes and 600 bytes respectively, and call them scenarios A_1 and A_2 respectively. In this way we obtain two training sets of packets corrupted due to weak signals: S_{A_1} and S_{A_2}.

Scenario B: Interference. In this scenario, T_1-R is the sender-receiver pair, while T_2 is the interference source. Both T_1 and T_2 send packets periodically, with the packet rate of T_2 being lower. By looking into the synchronized packet logs at the transmitters, we obtain the “ground truth” about which packets from T_1 have overlapping with packets from T_2, and those packets are then candidates for the training set of packets corrupted due to interference. T_1 and T_2 are placed the same distance away from R, so that we can assume that the ratio of their signal strengths at R is the same as the ratio of transmission powers they use. We repeat experiments by changing P_1 and P_2, and the interference scenario can

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

be further divided into four sub-scenarios based on the following combinations:

B_1**: Strong signal/strong interference** - We can determine the threshold level of P_1 (denoted by P_s) such that over 98% of received packets have zero BER. When $P_1 \geq P_s$, we say it is a *strong signal scenario*. When $P_2 \geq P_1$, we say the *interference is strong*. To summarize, we label the scenario as B_1 when $P_2 \geq P_1 \geq P_s$. Again, we repeat experiments several times using different power values, log packets received at R and label them with the appropriate (P_1, P_2) pair. Packets having the same label form a *group*.

B_2**: Strong signal/weak interference**. Interference is called *weak* when $P_2 \leq P_1$. The scenario is labeled as B_2 when $P_1 \geq P_s$ and $P_2 \leq P_1$.

B_3**: Borderline signal/strong interference** - We can determine the borderline value of P_1 so that for $P_1 \geq P_b$, 98% of the received packets are not corrupted. The scenario is labeled as B_3 when $P_2 \geq P_1 = P_b$.

B_4**: Borderline signal/weak interference**. The scenario is labeled B_4 when $P_1 = P_b$ and $P_2 \leq P_1$.

It is clear that $P_s > P_b > P_c > P_{no}$. When P_1 decreases from above P_s to below P_{no}, the link condition between T_1 and R downgrades from good, to fair, to noisy, and finally to no link at all.

5. PERFORMANCE

We assess the performance of PCC and the three classification algorithms by evaluating how accurately they can separate corruptions due to weak signals from those due to interference in scenarios B_1, B_2, B_3, and B_4.

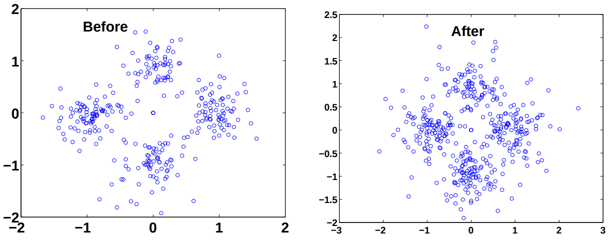
5.1 Performance comparison of algorithms

In order to compare the classification accuracies of CMT, CST and MWW, we first run PCC on S_{A_1}, pick I_β^{th} so that the false positive rate in S_{A_1} is β. Then we run PCC (with *indicator threshold* being I_β^{th}) on packets collected in scenarios B_1 to B_4, and compare the miss rate of each *group* between CMT, CST and MWW . Both the sliding window size and step size are 102, i.e., the number of *sub-carriers*. Figures 7, 8, 9 show the *classification* of packet corruptions gained by CMT, CST and MWW, respectively. Each histogram is from one *group* of the scenario. The (P_1, P_2) setting is (P_s, P_s) for Figures 7.b, 8.b, 9.b and $(P_b, 0.45 \times P_b)$ for Figures 7.c, 8.c, 9.c. Collectively the nine figures illustrate how well the classification is done by different algorithms under different scenarios. Note that the less two histograms overlap, the better the classification.

In scenarios B_1 and B_2, all three algorithms identify the cause of corruption quite accurately. For CST and MWW, the miss rate is less than 0.5%, with false positive rate 0.5% for any *group* of packets. For CST and MWW, the miss rate is less than 0.5% with false positive rate 3%. In scenarios B_3 and B_4, the accuracy of all three algorithms drops (especially in B_4), but CST drops faster than the other two. Figure 6 compares the performance of the algorithms under scenario B_4, and we explain the reason in the next section.

5.2 The impact of signal/interference strength

First we explain why classification accuracy drops in scenario B_4 so dramatically for CMT, as shown in Figure 6. Recall that as discussed in Section 3.5.2, given I^{th} in CMT, the miss rate decreases when $\mu_d = \mu_i + \sigma\sqrt{\frac{\pi}{2}}$ increases. A strong signal from the sender implies lower ground noise (smaller σ_n); interference stronger than the signal pulls S_{ij} far away from constellation points $\bar{a}_k$, hence yields larger μ_i. When either the signal or the interference is strong, we have μ_d large enough for CMT to perform well. In scenario B_4, however, when σ_n is not small enough and μ_i is not large enough, CMT barely works. Figure 5 illustrates this by showing the constellation graph of a weakly interfered packet with sender's signal strength at borderline. On the other hand, the classification accuracy of CST and WMM does not rely on μ_d but on how different the two distributions F_n and F_i are. Experimental results show that as long as (i) the sender's signal strength is at least fair (i.e., $P_1 \geq P_b$); and (ii) the interference is strong enough to cause packet corruption, then even in the worst case, F_n and F_i are different enough for CST and WMM to attain a miss rate lower than 10%, with the false positive rate threshold set to 5%. WMM has a slightly better performance than CST by achieving miss rate 6%, with the false positive rate threshold set to 5%.



Before Interference

During Interference

Figure 5: Consellation graph of an interfered packet with $P_1 = P_b$ and $P_2 = 0.45 \times P_1$

Table 2: The comparison of the false positive rate between A_1 and A_2. The packets corrupted due to interference are those collected with $P_1 = P_b = 2.2 \times P_2$. The miss rate is 8%.

	CMT	CST	MWW
A_1	0.2874	0.0784	0.0327
A_2	0.5014	0.1851	0.0399

5.3 The impact of packet size

When I^{th} is fixed, changing the packet size l does not affect miss rate very much. But a larger packet size l may increase the false positive rate for all three tests. This is because as suggested in (16), the longer the packet, the more likely that one of the $l - 1$ tests will return an indictor larger than I^{th}. Table 2 illustrates the impact of packet size and verify this. Here we choose the threshold I^{th} by fixing the miss rate, then we compare the false positive rate in S_{A_1} and S_{A_2}.

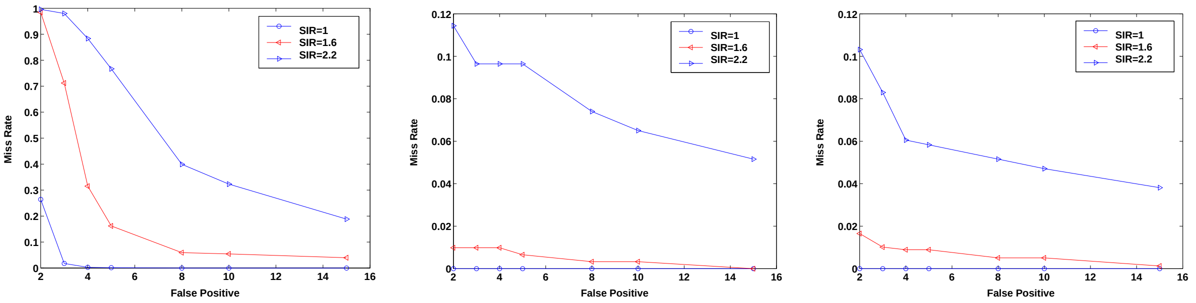
5.4 The impact of sliding window size

The sliding window size l_w is the size of samples to run the tests. Statistically, the larger l_w is, the better performance is, and experimental results support this. Table 3 assesses how well PCC-($l_w = 102$) and PCC-($l_w = 50$) separate packet corruptions in scenario A_1, from the corruptions in scenario B_4 with $P_1 = P_b = 2.2 \times P_2$. However, larger l_w costs more in computation. Moreover, if l_w is so large that it covers $w > 1$ symbols, then the interference that starts/terminates among the first (or last) w symbols may not be detected. Therefore, l_w must be tuned carefully.

6. INTERFERENCE FROM AN ADJACENT BAND

This section discusses how PCC can be extended to detect interference from wireless devices transmitting on an adjacent spectral

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

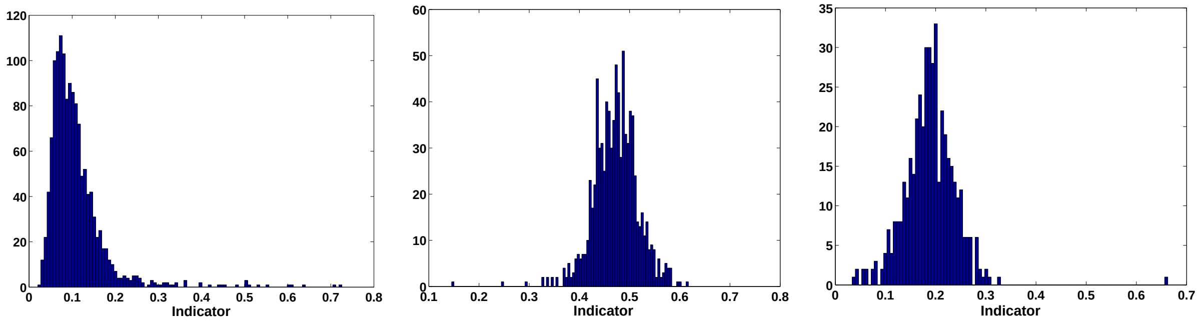


(a) mean value change

(b) chi-square test

(c) mean whitney U test

Figure 6: Miss rate comparison of three tests, under the scenario that sender's signal strength is around -80dbm at the receiver's side

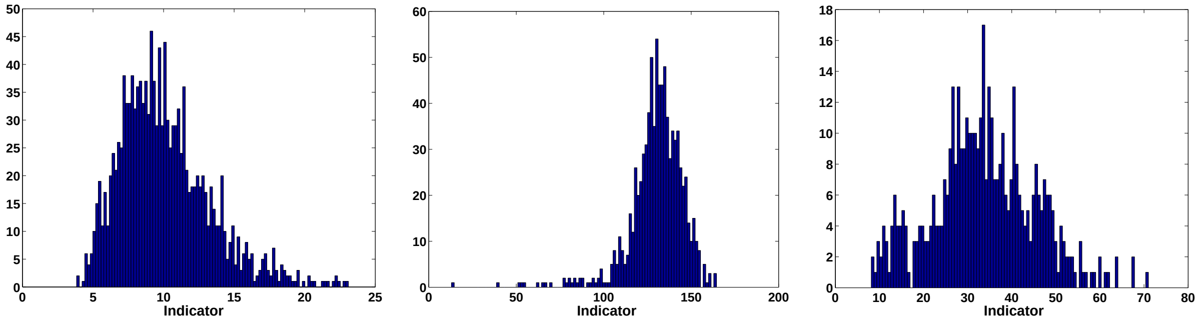


(a) Weak Signal

(b) Strong Signal/Strong Interference

(c) Weak Signal/Weak Interference

Figure 7: Histogram of Indicator output from CMT

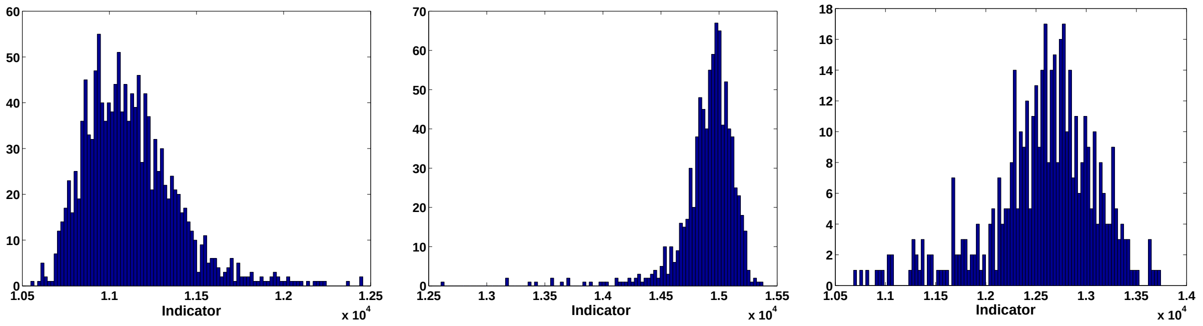


(a) Weak signal

(b) Strong Signal/Strong Interference

(c) Weak Signal/Weak Interference

Figure 8: Histogram of Indicator output from CST



(a) Weak Signal

(b) Strong Signal/Strong Interference

(c) Weak Signal/Weak Interference

Figure 9: Histogram of Indicator output from MWW

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

Table 3: The change of the false positive rate by changing the sliding window size l_w. The miss rate is 8%.

	CMT	CST	MWW
l_w=102	0.2874	0.0784	0.0327
l_w=50	0.4686	0.1251	0.0349

band, but "crossing the line". Recently there has been much cognitive radio/ultra-wideband usage aim at making use of as much of the unused spectrum as possible. Some aggressive protocols have been designed. For example, [11] proposes a wideband network protocol that keeps grabbing more and more spectrum till it learns from observed backoff activities of other narrowband devices that it hurts them, and then releases some of the spectrum. We expect that in the future when systems running different protocols operate together, interference from adjacent bands may be quite common.

When interference is from an adjacent band, only the *sub-carriers* on the edge where two bands overlap are affected. Therefore when such interference happens, we observe changes not only between symbols, but also between *sub-carriers*. However, if the overlapped band is too narrow, the interference cannot be well detected by PCC. The reason is that in order for PCC to work properly, the sliding window size cannot be too small. If the number of interfered *sub-carriers* is too small, PCC cannot get enough sample points to run the statistical tests.

Therefore we make the following assumption: we want to detect interference that affects at least k *sub-carriers*. Without loss of generality, in the following we only consider interference overlapping with the first to k-th *sub-carriers*. Then we execute the following steps to detect the interference: (i) Choose the sequence S to be $x_{(i-1)\times k+j} = |X_{ij}|$, for $1 \leq i \leq l$. This means that instead of processing the whole packet, we only process the contents of the first k *sub-carriers*; (ii) run PCC(k, l_w) on S to detect change point(s), noting that because k is small, one window may cover several symbols; (iii) if a change point is detected, then locate the duration of interference in the time-line. Suppose the interference starts at the s-th symbol and ends at the d-th, which covers l_i symbols; (iv) process the chunk composed by the symbols interfered. Choose S to be $x_{(i-s+1)+j\times l_s} = |X_{ij}|$, for $s \leq i \leq d$. Run PCC(l_w, l_w) to detect the change point along *sub-carriers*, but fix SW_1. We accept the hypothesis that there is adjacent band interference only when PCCs in (ii) and (iv) both find change points.

We have also conducted experiments in iWWT to evaluate the above solution. The parameter settings are $l_w = 40$ and $k = 20$. In a borderline signal/strong interference scenario, when around 25 out 102 *sub-carriers* are interfered with, PCC-CST has a hit rate over 99%, with false positive rate less than 3%.

7. RELATED WORK

Paper [12] presents the first empirical study based on bit error patterns of received data for loss diagnosis in 802.11. The authors proposed several novel symbol level metrics, such as symbol error rate, error per symbol and S-score, to separate collisions (caused by interference from the same system) from weak signal. Those metrics motivated our design of PCC to some extent. However, their scheme, called COLLIE, requires the receiver to send back the whole corrupted packets back to the sender, which introduces a large reverse traffic overhead. Our PCC, on the other hand, only requires the receiver to send a few bits to the sender, but provides more accurate classification than COLLIE. Besides [12], there have been other research efforts aimed at diagnosing the packet loss problem. Yun and Seo [15] propose to detect collisions in 802.11 links by measuring the RF energy and its changes. However, this work was done through simulation only. The rate adaption mechanism CARA of [5] tries to detect collisions by using RTS-CTS. Their scheme fails in the presence of hidden terminals, which are actually major sources of collisions. None of the above approaches make use of *signal-level* physical layer information, but rather use bit level information, or the MAC layer logical information exported by commodity wireless devices.

In recent years, there has been a growing interest in the wireless network community at making use of more information from physical layer. We note a few works. Jamieson *et al.* [8] develop a partial packet recovery (PPR) system. PPR has a key component called *softPHY*, which is an expanded physical layer interface that provides hints to higher layers about the physical layer's confidence in each bit it decodes. The hint is based on the the Hamming distance in block decoding, which is a different approach from ours. Gollakota *et al.* [7] proposes Zigzag decoding to solve the hidden terminal problem by locating the start of the second packet when two packets overlap (thus collide). Zigzag's success is built upon the capability of the DSSS physical layer to detect preambles of packets at very low SINR, which does not apply in the OFDM system. Ziptx from Lin *et al.* [9] can improve system throughput by using pilot bits to detect the per-symbol BER, and using forward error correction to correct symbols with low BER while retransmitting symbols with high BER. Ziptx explores new ways of recovering from packet corruptions and also suggests different solutions to different corruptions.

8. CONCLUSIONS

In this paper, we have tried to determine if one can identify the cause of packet corruptions in the OFDM system. Unlike most of the previous approaches, our proposed scheme, PCC, looks directly into available physical layer information at *signal-level*, and uses three *statistical* tests to classify packets corrupted as being due to interference or weak signals. Through evaluations conducted on a GNU radio test bed in iWWT over a wide range of experiments, we identify the working ranges of the *statistical* tests, and compare their performances under different channel conditions. We also provide analytical explanations to the experimental results. When PCC uses the most accurate test WMM, it can achieve a low miss rate (of interference) of 6%, with the false positive rate threshold set to 5%, even under the scenario where interference is hardest to detect. Since all experimental results and conclusions given in this paper are based on experiments conducted on a real test bed, we expect that the implications of our results can be very useful in upper-layer problem domains, such as link scheduling, channel management, packet recovery design, etc., where a better understanding of the link behavior can help a lot.

9. ACKNOWLEDGMENTS

We thank Shanil Merchant, Prof. Jennifer T. Bernhard, Christopher D. Schmitz, Prof. Lui Sha, Lance Gorrell, Cheolgi Kim, Qixin Wang, Yan Gao, Yong Yang and all others that helped with the GNU radio test bed setup and the experiments in iWWT.

10. REFERENCES

[1] E. Inc. Universal software radio peripheral. *"http://ettus.com."*.

[2] GNUradio wiki. *"http://gnuradio.org/trac/"*.

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930

[3] G. Brar, D. M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *MobiCom '06*, pages 2–13, New York, NY, USA, 2006. ACM.

[4] R. W. Chang. Synthesis of band-limited orthogonal signals for multi-channel data transmission. *Bell System Technical Journal 46*, 1966.

[5] J. K. et al. Cara: Collision-aware rate adatpion for ieee 802.11 wlans. In *IEEE INFOCOM*, 2006.

[6] Y. Gao, J. C. Hou, and H. Nguyen. Topology control for maintaining network connectivity and maximizing network capacity under the physical model. In *INFOCOM 2008*, 2008.

[7] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. *SIGCOMM 08*, 2008.

[8] K. Jamieson and H. Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *ACM SIGCOMM*, Kyoto, Japan, August 2007.

[9] K. C. Lin, N. Kushman, and D. Katabi. Ziptx: Harnessing partial packets in 802.11 networks. In *Mobicom'08*, September 2008.

[10] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *MobiCom '07*, pages 171–182, New York, NY, USA, 2007. ACM.

[11] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat. Learning to share: narrowband-friendly wideband networks. *SIGCOMM Comput. Commun. Rev.*, 38(4):147–158, 2008.

[12] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee. Diagnosing Wireless Packet Losses in 802.11: Separating Collision from Weak Signal. In *IEEE INFOCOM*, Phoenix, AZ, April 2008.

[13] T. M. Schmidl and D. C. Cox. Robust frequency and timing synchronization for ofdm. *IEEE Trans. Communications*, 45(12):1613–1621, Dec. 1997.

[14] N. H. Vaidya, J. Bernhard, V. V. Veeravalli, P. R. Kumar, and R. K. Iyer. Illinois wireless wind tunnel: a testbed for experimental evaluation of wireless networks. In *E-WIND '05: Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*. ACM, 2005.

[15] J.-H. Yun and S.-W. Seo. Collision detection based on rf energy duration in ieee 802.11 wireless lan. In *Comsware*, 2006.

[16] H. Zimmermann. OSI reference model–the iso model of architecture for open systems interconnection. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 28(4):425–432, 1980.

Digital Object Identifier: 10.4108/ICST.WICON2008.4930
http://dx.doi.org/10.4108/ICST.WICON2008.4930