M.S. THESIS

Compressed Sensing Based Block Type Multi-User Detection for Sporadic IoT Communications

IoT 통신을 위한 압축 센싱 기반 Block Type Multi-User Detection 기법연구

BY

ZHANG GUIYONG

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Compressed Sensing Based Block Type Multi-User Detection for Sporadic IoT Communications

IoT 통신을 위한 압축 센싱 기반 Block Type Multi-User Detection 기법연구

BY

ZHANG GUIYONG

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Compressed Sensing Based Block Type Multi-User Detection for Sporadic IoT Communications

IoT 통신을 위한 압축 센싱 기반 Block Type Multi-User Detection 기법연구

지도교수 심병효

이 논문을 공학석사 학위논문으로 제출함

2016년 12월

서울대학교 대학원

전기정보 공학부

장계용

장계용의 공학석사 학위 논문을 인준함

2016년 12월

위 원 장:	김남수	(인)
부위원장:	심병효	(인)
위 원:	최성현	(인)

Abstract

Compressed Sensing Based Block Type Multi-User Detection for Sporadic IoT Communications

ZHANG GUIYONG
Department of Electrical and Computer Engineering
The Graduate School
Seoul National University

Over the last decade, a significant opportunity for wireless networks has been recognized as the Internet-of-Things (IoT). IoT is a massive device-interconnected platform that enables seamless communications among objects. Recently, a variety of diverse IoT applications have been developed to improve the human life. However, among those applications there are still many major challenges, such as the massive access issue (MAI) and coverage. In this dissertation, we study the MAI in massive IoT applications. By exploiting the block sparsity nature of sporadic IoT communications, we propose a

compressive sensing (CS) based block type multi-user detection (CS-BT-MUD) algorithm to address the challenge of MAI. In particular, our algorithm employs a decoupling operation to deal with complex-valued data, which enable to solve complex-valued problems via conventional CS algorithms. Based on two types of data traffic models, numerical evaluations demonstrate that our proposed CS-BT-MUD algorithm is very effective in addressing the MAI and thus offers benefits to block type multi-user detection in IoT communications.

Keywords: Compressive sensing (CS), Internet-of-things (IoT), wireless networks, massive access issue (MAI).

Student Number: 2015-22137

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Internet-of-Things (IoT): Concept and Current Issue

Recently, more and more attention has been attracted to enhance interconnections among a wide variety of objects. The concept of Internet of Things (IoT) is in the spotlight [1]. The IoT is a generalized platform for all physical objects, machines, and devices to exchange information. Accompany with the concept, emerging applications of IoT include remote metering, wearable devices, health-care, smart cities, and many more are coming to our sight. Due to the wide range of applications, over 50 billions of objects are expected to be connected by the year 2020 [2]-[3].

Typically, IoT applications are classified as *massive IoT* and *mission-critical IoT* (see Figure 1) [4]. Massive IoT refers to the circumstances where a huge amount of small data packets transmit infrequently on a regular basis. Hence, the requirements for massive IoT are characterized by a massive number of sensor nodes (e.g., 10^6 - 10^7 devices/km^2), wide coverage, low energy consumption (e.g., battery life of more than ten years), and cost-efficient devices (e.g., blow 10 dollars). Mission-critical IoT is known as the circumstances where data packets transmit without error and achieve a very low latency. For mission-critical IoT applications, due to the key requirement for reliability (e.g., 10^{-9} packet error rate) and latency (e.g., one msec end-to-end latency),

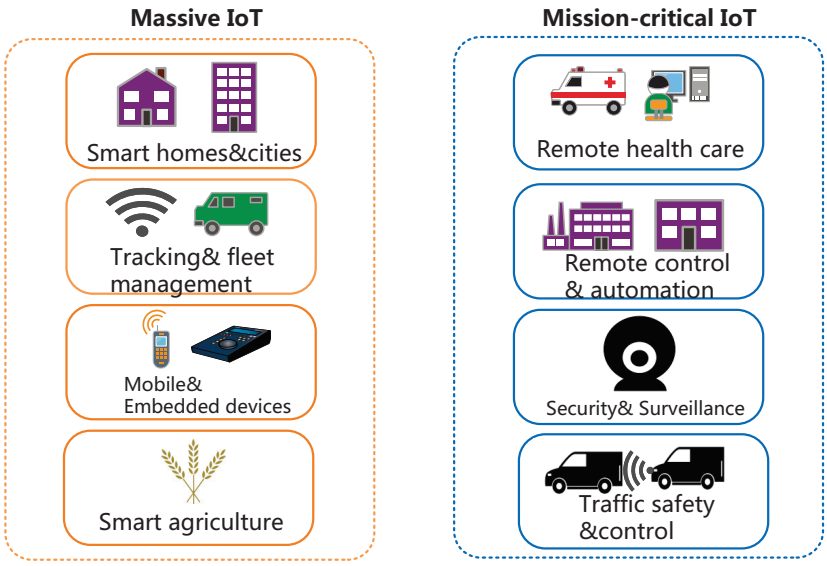


Figure 1.1: Massive IoT and misson-critical IoT

data rate and accuracy for information sending are much more important.

For practical massive IoT applications, a huge number of sensor nodes access base station simultaneously. Thus, an important issue is massive access issue (MAI). However, current LTE system is inefficient to tackle this issue, since connection-established approaches in LTE often lead to HARQ and signaling overhead. With increased retransmission times, the signaling overhead and delay will increase dramatically. In order to support massive access, physical layer technologies should be re-evaluated and redesigned. In fact, many recent efforts have been made to improve the early version of the IoT system. In particular, a variety of physical layer techniques relying on the unlicensed spectrum and the licensed spectrum respectively have been proposed. In the unlicensed spectrum, new proprietary radio technologies (e.g., SigFox and LoRa) have been developed [5]. The main purpose of these techniques is to meet the requirement of long range and the high reliability for low-end sensor segments. On the other hand, licensed band technologies have been developed under the umbrella of 3GPP standardized systems [6]-[9]. Evolutions in 3GPP, including MTC, enhanced MTC

(eMTC) and narrow-band IoT (NB-IoT), aim to provide enhancements in coverage, device cost and signaling overhead [10]-[11] .

1.2 Physical Layer Technologies for IoT

In this part, we will introduce the state of the art physical layer technologies deployed both in unlicensed and licensed band. For unlicensed band, Sigfox and Lora will be introduced with massive access issue (MAI). As for licensed band, we will give a brief introduction for recent 3GPP standardization and current status in IoT.

1.2.1 Ultra Narrow Band (UNB)

SigFox, a technology based on UNB and RF-TDMA, enables the support of massive number of devices and long range communications. The UNB allows the receiver to mitigate the effect of collision in conjugate with random access strategy. Using extremely narrow bandwidth of 100 Hz, data are transmitted with binary phase-shift keying (BPSK) with very low throughput for high reliability. Note that the power spectral density (PSD) of UNB is much higher than other transmission schemes (e.g., ultra wide band transmission and spread spectrum) . Thus, the SigFox can support very long communication range (e.g., 10 kilometers in urban area) in environments with severe signal attenuation [12].

Along with UNB, RF-TDMA is a random access scheme where carrier frequency and time slots are randomly selected in a continuous interval. For each message, we allow the transmissions up to three times with random frequency hopping (see Figure 1.2). Thus, complete collision probability dramatically decreases, although partial overlapping issue still exists with low probability.

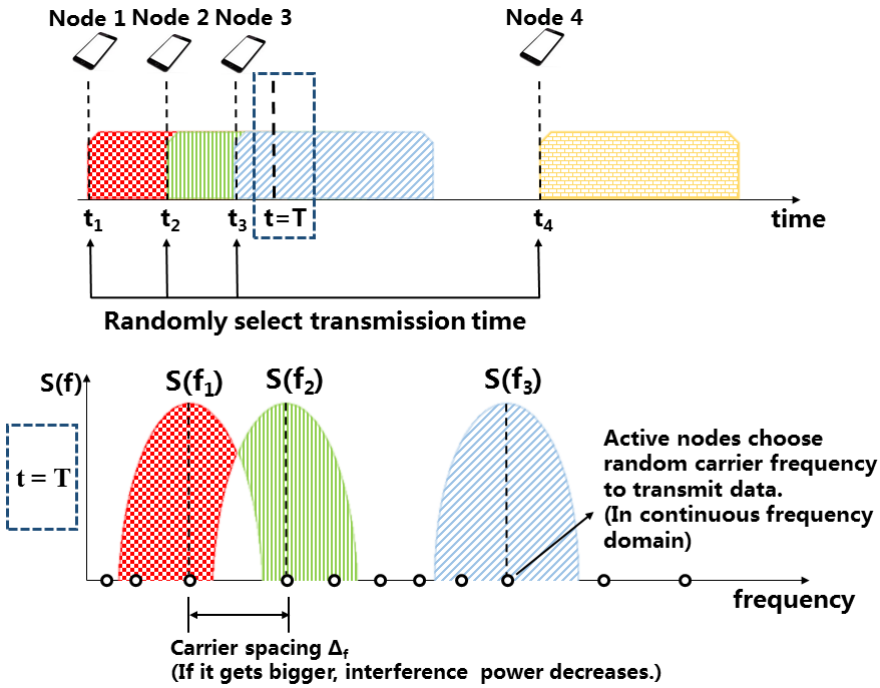


Figure 1.2: RFTDMA in time and frequency domain in SigFox

1.2.2 Chirp Spread Spectrum (CSS)

The CSS is a spread spectrum modulation technique that encodes information by using linear frequency chirp pulses. Since the chirp pulse is a signal that varies in frequency over time, the frequency of the chirp signal varies from low to high frequency (up-chirp) or from high to low frequency (down-chirp) (see Figure 1.3).

LoRa uses a CSS technology with a wideband spectrum equal to or more than 125 kHz. To reduce interference, LoRa exploits spreading effects through continuously varying the carrier frequency with multiple spreading factors, while direct-sequence spread spectrum (DSSS) takes advantages of pseudo-random sequence [5]. We note that two devices cannot use the same spreading factor at the same time, the interference issue can be solved.

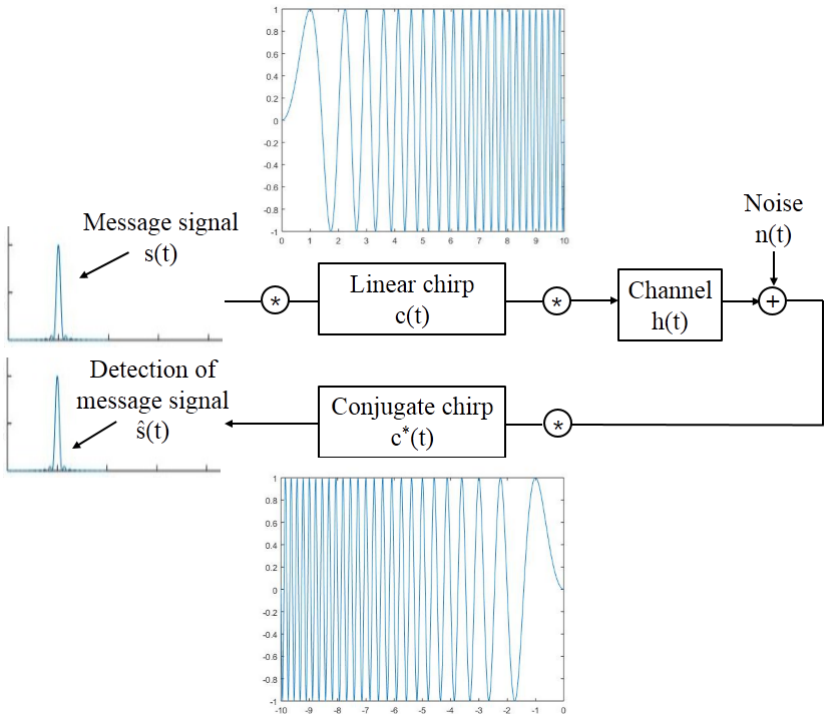


Figure 1.3: Overview of chirp spread spectrum technology of LoRa

1.2.3 Machine-type Communications (MTC)

Machine-type Communications (MTC) is the first trial to support "light version" of the LTE system which has been released with 3GPP Rel. 8. Accompany with concept, the least capable device category called Cat. 1 is introduced. The device can provide a peak data rate up to 10 Mbps for downlink and 5 Mbps for uplink with 20 MHz system bandwidth. But for stringent IoT requirements, such as long battery life, low cost, and extended coverage, another enhanced device category eMTC has been released (called Cat. *M*) in Rel. 13 [13]-[14]. eMTC has been made with the aim of achieving even lower complexity. Some key distinct features of Cat. M devices are as follows:

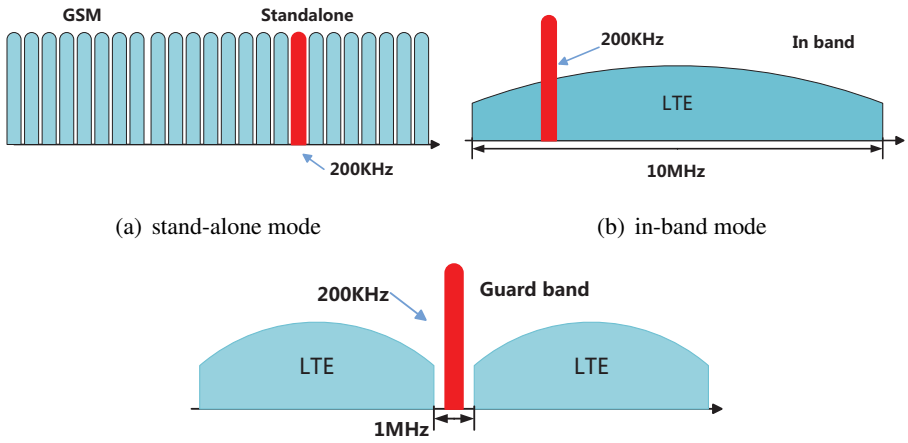
- System bandwidth is reduced from 20 MHz to 1.4 MHz for both uplink and downlink, which helps achieve a low-cost target.

- Maximum transmit power will be reduced to 20 dBm, which is less than the transmission power of Cat. 0 device (i.e., 23 dBm).
- Enhanced PSM is introduced to improve the battery life of the device with a much longer sleep duration.

However, reducing the maximum transmit power will lead to coverage shortage in the uplink. Also, with 1.4 MHz bandwidth transmission, frequency diversity gain cannot be used for both downlink and uplink transmissions (e.g., 1-2 dB degradation). To overcome these degradations, coverage enhancement techniques, such as repetition, power boosting, and retransmission, have been employed for various physical channels. While such coverage enhancement techniques require additional resources (e.g. time subframe), which are not feasible for mission-critical IoT applications. Furthermore, for dense deployment of MTC devices, both the system bandwidth and the cost are still needed to reduced substantially.

1.2.4 Narrow Band-IoT (NB-IoT)

NB-IoT has been initiated by 3GPP with new features to specify the physical layer for cellular IoT. Based on a great extension on a non-backward-compatible variant of LTE, NB-IoT has improved latency and provided support for massive devices. In order to achieve backward compatibility, NB-IoT has been designed to use 180 kHz RF bandwidth. For downlink transmission, OFDMA with either 15 kHz or 3.75 kHz subcarrier spacing will be supported for narrower bandwidth transmission. For the uplink, two options are suggested: One is the frequency division multiple access (FDMA) with Gaussian minimum shift keying (GMSK) modulation, while the other is single-carrier frequency division multiple access (SC-FDMA) that includes single-tone transmission as a special case of SC-FDMA for enhanced coverage [11]. NB-IoT is designed on purpose to co-exist and interwork with LTE, which provides great deployment flexibility in the following three scenarios (see Figure 1.4):



(a) stand-alone mode

(b) in-band mode

(c) guard-band mode

Figure 1.4: NB-IoT carrier deployment scenarios

i) Standalone mode: The spectrum currently used by GERAN[1] systems is utilized as a replacement of one or more GSM carriers.

ii) In-band mode: Resources can be deployed within a normal LTE carrier.

iii) Guard-band mode: Resources also can be deployed in a LTE carrier's guard-band.

Note that, in the guard-band mode, the carrier will be placed in the guard-band between two LTE carriers. Assume that the LTE system bandwidth is 10 MHz. Then the guard-band between two LTE carriers should be given at least 1 MHz without affecting LTE carriers.

1.3 Compressive Sensing (CS) Recovery

In this part, we will first introduce two classical compressive sensing based algorithms: the orthogonal matching pursuit (OMP) algorithm and the simultaneous or-

[1]GERAN is an abbreviation for Global System for Mobile Communications (GSM) Enhanced Data rates for GSM Evolution (EDGE) radio access network. The standards for GERAN are maintained by the 3GPP [11].

thogonal matching pursuit (SOMP) algorithm. Afterwards, we will show how these algorithms can be employed to address the MAI in IoT communications in Chapter 2.

Before preceding, we give a brief introduction for common notations which are used in this thesis. Matrices and vectors are denoted by using upper-case and lower-case boldface letters respectively. Also, $(\cdot)^T$, $(\cdot)^{-1}$ and $(\cdot)^+$ are defined as transpose, matrix inversion, Moore-Penrose matrix inversion. $|T|$ denotes total number of elements in a set T.

1.3.1 Single Vector Model

In signal processing literature, recovery of high-dimensional sparse signal from a small number of linear measurements is a fundamental problem. We consider the system model

$$\mathbf{y} = \Phi\mathbf{x} + \mathbf{n}, \tag{1.1}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the measurement vector, $\Phi \in \mathbb{R}^{m\times K}$ is the measurement/sampling matrix and $\mathbf{n} \in \mathbb{R}^m$ is the measurement error. The support of vector vector $\mathbf{x} = (x_1, ..., x_m) \in \mathbb{R}^m$ is defined to be the set $supp(x) = \{l : x_l \neq 0\}$. In particular, x is said to be K-sparse if $|supp(x)| \leq K$.

The OMP algorithm can be used to recover the support of K-sparse signal $\mathbf{x}$ under the system model in (1.1). Recovery of $\mathbf{x}$ is also called single measurement vector (SMV) recovery in CS literature. OMP is an iterative greedy algorithm that selects at each step the column of Φ which is most correlated with the current residuals. After that, this column is added into the set of selected columns. Then, the residuals will be updated by projecting the observations $\mathbf{y}$ onto the linear space spanned by the columns that have already been selected. The detailed steps of the OMP algorithm are given in Algorithm 1. Assume the columns of Φ are normalized so that $||\Phi_i||_2 = 1$ for $i = 1, 2, \cdots, K$. For any subset $D \subseteq \{1, 2, ..., K\}$, denote by $\Phi(D)$ the submatrix of Φ consisting of the columns Φ_i with $i \in D$. A key component of OMP is the stopping rule which depends on the noise structure.

Algorithm 1 Original Orthogonal Matching Pursuit (OMP) [15]

Input: measurement $\mathbf{y} \in \mathbb{R}^m$, measurement matrix $\Phi \in \mathbb{R}^{m \times K}$, Sparsity level s

Output: Index set D

Initialization: $\mathbf{r} \leftarrow y$, $\Phi(d_0) \leftarrow \emptyset$, $i = 1$.

1: Find the variable Φ_{t_i}, that solves the maximization problem:

$t_i \leftarrow \underset{t}{\operatorname{argmax}} |\Phi_t^{'} r_{i-1}|$

2: Update index set $d_i = d_{i-1} \cup t_i$

3: Projection onto span $(\Phi(d_i))$:

$P_i \leftarrow \Phi(d_i)(\Phi(d_i)^{'}\Phi(d_i))^{-1}\Phi(d_i)^{'}$

4: Update $r_i = (I - P_i)y$

5: If $r_i = 0$, **stop**. Otherwise, $i \leftarrow i + 1$

6: **return** index set D.

1.3.2 Multiple Vector Model

Since the original OMP algorithm has been introduced as a solution for single measurement vector (SMV) recovery problems. For multiple measurement vector (MMV) problems, we introduce an algorithm called the Simultaneous OMP (SOMP) algorithm. We define a similar system model as

$$\mathbf{Y} = \Phi\mathbf{X} + \mathbf{N}. \tag{1.2}$$

where the measurement matrix $\mathbf{Y} \in \mathbb{R}^{m \times K}$ with $\mathbf{Y} = [\mathbf{y_1}, \cdots, \mathbf{y_K}]$, the matrix $\mathbf{X} \in \mathbb{R}^{K \times K}$ with $\mathbf{X} = [\mathbf{x_1}, \cdots, \mathbf{x_K}]$ and the measurement error matrix $\mathbf{N} \in \mathbb{R}^{K \times K}$. Since the measurements are $m \times K$ matrix, this system is no longer a SMV problem and called MMV problem. To distinguish with OMP system model, Φ is seen as a dictionary matrix whose columns $\{\phi_j\}_{j \in K}$ are the atoms of the associated dictionary.

SOMP is a greedy algorithm that provides approximate solutions to the joint support recovery problem by successively picking atoms from Φ to simultaneously approximate the K measurement vectors $\mathbf{y} \in \mathbb{R}^m$.

Algorithm 2 Simultaneous Orthogonal Matching Pursuit (SOMP) [16]

Input: $\mathbf{Y} \in \mathbb{R}^{m\times K}$,$\Phi \in \mathbb{R}^{m\times K}$, Sparsity $s \geq 1$.

Output: $\mathbf{S}$.

Initialization: $\mathbf{R}^{(0)} \leftarrow \mathbf{Y}$ and $S \leftarrow \emptyset$.

1: $t \leftarrow 0$

2: **while** $t < s$ **do**

3: Determine the atom of Φ to be included in the support:

$j_t \leftarrow \text{argmax}_j(||\mathbf{R}^{(t)}\phi_j||_1)$

4: Update the support: $S_{t+1} \leftarrow S_t \cup \{j_t\}$

5: Projection of each measurement vector onto span $(\Phi_{S_{t+1}})$:

$\mathbf{Y}^{(t+1)} \leftarrow \Phi_{S_{t+1}}\Phi^{+}_{S_{t+1}}\mathbf{Y}$

6: Projection of each measurement vector onto span $(\Phi_{S_{t+1}})^{\perp}$:

$\mathbf{R}^{(t+1)} \leftarrow \mathbf{Y} - \mathbf{Y}^{(t+1)}$

7: $t \leftarrow t+1$

8: **end while**

9: **return** S_s {Recovered support}

Now, we give a detailed explanation for the procedures of the SOMP algorithm. We denote the t-th residual as $\mathbf{R}^{(t)}$ and $\mathbf{P}^{(t)}$ is the projector onto the orthogonal complement of span (Φ_{S_t}). Based on the maximum correlation computation (Step 3), we choose the atom to update residual (Step 4). Where $\mathbf{r}_k^{(t)}$ refers to the k-th column of $\mathbf{R}^{(t)}$) since

$$||(\mathbf{R}^t)^T\phi_j||_1 = \sum_{k=1}^{K}|\langle\phi_j, \mathbf{r}_k^{(t)}\rangle|. \tag{1.3}$$

Then the original signal $\mathbf{Y}$ is projected onto the orthogonal component of span $(\Phi_{S_{t+1}})$ (Steps 5 and 6) [16].

1.4 Dissertation Outline

In this paper, we aim at investigating the nature of signal sparsity caused by sporadic IoT communications and apply modified compressive sensing based algorithm to solve corresponding block type MUD problem at BS side. Our contributions are proposing a CS based algorithm for addressing the MAI and investigating practical data traffic models for massive IoT applications. Using numerical evaluation results, we also provide potential solutions to massive access in IoT communications.

The rest of the dissertation is divided into four major parts. In Chapter 2, we will introduce the concept and system model in for massive access in IoT communication, and investigate single type and block type CS-MUD problem in Section 2.1 and 2.2 respectively. Along with problem statement, we also present the how these problems could be solved by our proposed CS based algorithm. Then, we give our proposed CS-BT-MUD algorithm in Section 2.3 and some analysis in Section 2.4.

In Chapter 3, numerical evaluations are presented with detailed analyses. We introduce the simulation parameters in Section 3.1, The BER performance for two data traffic models are given in Section 3.2 and 3.3. System load ratio discussion is given in Section 3.4. Then, we defined a new concept as reconstruction error for further analysis in Section 3.5.

Chapter 4 gives our final conclusion.

Chapter 2

SYSTEM MODEL AND ALGORITHMS

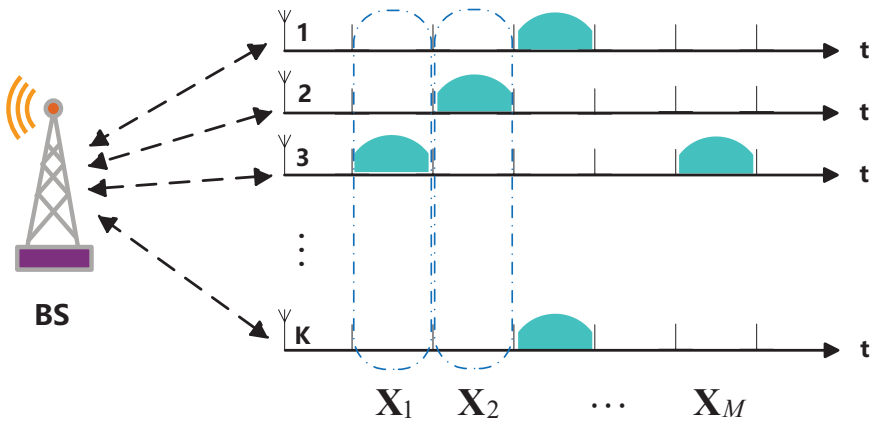


Figure 2.1: System model for massive access

In massive IoT application scenarios, such as smart home and smart agriculture, a huge number of sensor nodes are deployed in a fixed area. These sensor nodes will monitor, store and transmit the records of temperature and humidity. For some practical cases, the sensor node only needs to send the data when the temperature or humidity changes a lot. This means that not all the sensor nodes will transmit data all the time. In a certain period, only a few sensor nodes will be activated. Then, the status of sensor node will keep a approximate stable trend day by day. For convenience, we assume the

activity of the sensor nodes as a sporadic process. Throughout the thesis, the assumption is based on massive access system model in Figure 2.1. A set of K sensor nodes sporadically send data to a base station (BS). The features of sporadic transmission is described by using a statistic data traffic model with the activity probability p_a. We also assume p_a is same for all nodes at same time point t. Consequently, the probability of the sensor to keep silent is $1-p_a$. In practical massive IoT application, the value of p_a is assumed to be extremely small with $p_a \ll 1$.

Among massive IoT applications, the data traffic source can be described by different statistic traffic models. For sporadic IoT communication scenario, the Bernoulli traffic model and Poisson-Zeta (PZ) traffic model will be considered. For detailed discussions on statistic data traffic model will be introduced in the next part.

Simply speaking, Bernoulli traffic model is a well-known model which follows bernoulli distribution at time point t with activity probability p_a for each sensor node. While Poisson-Zeta traffic model $PZ[\lambda, g_l]$ is a discrete time ON/OFF process, where the number of activated sensors at each time point t is given by a poisson distribution with mean λ. For time axis, the duration l of each activity just follows Zeta distribution with probability g_l.

Generally, we formulate the system model as (2.3). In order to simplified the formulation, the sensor data will be expressed as vector form. Assume $x_1, x_2, \cdots, x_{K1}$ are defined as data generated by sensor node 1, 2 and K at time point t_1. Then, the corresponding channel state can be denoted as h_1, h_2 and h_K. We also take the spreading factor into consideration by using a chip sequence s_k with length N_s, where k denotes the corresponding user with $k = 1, 2, \cdots, K$. This means that after spreading by using this sequence we could map the modulated signal into N_s subcarriers. Then, the received signal at the BS side can be simply expressed as

$$\begin{cases} y_1 = s_{11}h_1x_1 + s_{21}h_2x_2 + \cdots + s_{K1}h_Kx_K, \\ y_2 = s_{12}h_1x_1 + s_{22}h_2x_2 + \cdots + s_{K2}h_Kx_K, \\ \vdots \\ y_{N_s} = s_{1N_s}h_1x_1 + s_{2N_s}h_2x_2 + \cdots + s_{KN_s}h_Kx_K. \end{cases} \tag{2.1}$$

where

$$\begin{bmatrix} y_1 \\ \vdots \\ y_{N_s} \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1N_s} \\ \vdots & \ddots & \vdots \\ s_{K1} & \cdots & s_{KN_s} \end{bmatrix}^T \cdot \begin{bmatrix} h_1 & 0 & 0 & 0 \\ 0 & h_2 & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & h_K \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix}. \tag{2.2}$$

Based on the above equations, the formulation can be arranged in a general form

$$\mathbf{y} = \mathbf{SHx}. \tag{2.3}$$

where $\mathbf{y} \in \mathbb{C}^{N_s}$ is the received signal, $\mathbf{S} \in \mathbb{C}^{N_s \times K}$ is the spreading code matrix, $\mathbf{H} \in \mathbb{C}^{K \times K}$ is the channel matrix and $\mathbf{X} \in \mathbb{C}^K$ is the sensor data. Next, we will show how this formulation could be extended to single type CS-MUD and block type CS-MUD problem .

2.1 Single Type CS-Multi-User Detection (MUD) Problem

Recall the system model in Figure 2.1, where a set of K sensor nodes sporadically transmitting data to a BS. Based on statistic data traffic model, the transmitted data from sensor nodes can be expressed by using a matrix $\mathbf{X}$ with dimension $K \times M$. Among them, the element in each row denotes transmitted data from the k-th sensor node, where $k = 1, 2, \cdots, K$. While for the element in each column denotes the transmitted data at consecutive time symbol m, where $m = 1, 2, \cdots, M$.

Mathematically, we represent the status of the sensor data by using two kinds of elements that zero and one. One denotes the sensor node is activated, zero is defined

for silent sensor node. Then, the transmitted sensor data matrix can be converted into a 0-1 matrix. Since we assume the activity probability for each sensor is extremely small, the sparsity is guaranteed. For sensor activity detection, we can use CS recovery algorithms at BS side.

For subcarrier deployment, we consider the NB-IoT scenario. The subcarriers are placed within the coherence bandwidth of the channel for each sensor. So that each sensor experiences flat fading. Based on this assumption, it is reasonable to avoid channel estimation.

Another advantage for allocating subcarriers within the coherence bandwidth is that it will save spectrum resource. As we introduced in NB-IoT part, NB-IoT application will be deployed in ultra narrow band with low data rate.

2.1.1 Procedures for Sensor Data Transmission

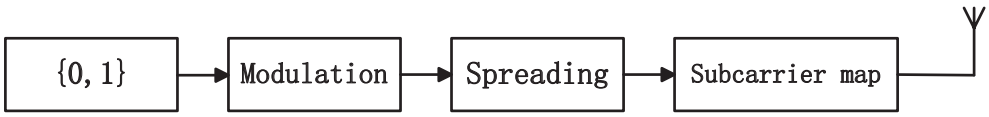


Figure 2.2: Data processing at sensor side

The transmitted signal for a sensor node is denoted as $x_k(t)$ at time slot t , where k denotes the sensor node with $k = 1, 2, \cdots, K$. And K is total number of sensor nodes. For modulation step, we use DPSK to modulate the data from active sensor nodes. Then, these modulated symbols from active sensor nodes can be spread by using a chip sequence $s_k \in \mathbb{C}^{N_s}$ with length N_s. This allows one to one mapping between spread codes and the number of subcarriers. We can see the total procedures in Figure 2.2. Then, it is more practical to allocate more than K sensor nodes to N_s subcarriers. Thus, we define system load ratio as $\theta = \frac{K}{N_s}$. The definition of system load ratio reflects the mapping degree between the number of input signal and the number of subcarriers. Also, we find that the value of system load ratio should maintain in a reasonable range.

2.1.2 Processing for Received Sensor Data

Above, we assume that all subcarriers are deployed within the coherence bandwidth of the channel of each sensor node. After receiving the uplink signal, the BS needs some operations such as CP removing and Discrete Fourier Transform (DFT). Based on (2.1) and (2.2), we make an extension with the as follows:

$$\begin{cases} y_{11} = s_{11}h_{11}x_{11} + s_{21}h_{22}x_{21} + ... + s_{K1}h_{KK}x_{K1}, \\ y_{21} = s_{12}h_{11}x_{11} + s_{22}h_{22}x_{21} + ... + s_{K2}h_{KK}x_{K1}, \\ \vdots \\ y_{N_s1} = s_{1N_s}h_1x_{11} + s_{2N_s}h_2x_{21} + ... + s_{KN_s}h_{KK}x_{K1}. \end{cases} \tag{2.4}$$

Then, the received signal $\mathbf{Y} \in \mathbb{C}^{N_s \times M}$ in frequency domain can be written as

$$\mathbf{Y} = \mathbf{SHX} + \mathbf{N}. \tag{2.5}$$

Specifically

$$\mathbf{Y} = \overbrace{\begin{bmatrix} s_{11} & \cdots & s_{1N_s} \\ \vdots & \ddots & \vdots \\ s_{K1} & \cdots & s_{KN_s} \end{bmatrix}^T}^{\mathbf{S}} \cdot \overbrace{\begin{bmatrix} h_{11} & 0 & 0 & 0 \\ 0 & h_{22} & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & h_{KK} \end{bmatrix}}^{\mathbf{H}} \cdot \overbrace{\begin{bmatrix} x_{11} & \cdots & x_{1M} \\ x_{21} & \cdots & x_{2M} \\ \vdots & \ddots & \vdots \\ x_{K1} & \cdots & x_{KM} \end{bmatrix}}^{\mathbf{X}} + \mathbf{N}. \tag{2.6}$$

To make the formulation clear, we have a detailed description in (2.6). Here the matrix $\mathbf{X} \in \mathbb{R}^{K \times M}$ is the sensor data matrix. From the dimension, we note that each column carries data from K sensor nodes. Each row denotes the sensor data from k-th node. Based on previous assumption that activity probability for sensor node $p_a \ll 1$, only few sensor data will transmit data at time point t. Mathematically, we can formulate activity status of the sensor nodes by using 0-1 matrix $\mathbf{X}$. Thus, there are only a few non-zero elements in each row and other elements are zeros. This supports the sparsity of $\mathbf{X}$.

The matrix $\mathbf{H} \in \mathbb{C}^{K\times K}$ is a diagonal matrix containing the corresponding channel conditions,

$$\mathbf{H} = diag\{\mathbf{h}\}. \tag{2.7}$$

where $\mathbf{h} = [h_1, \ldots, h_k, \ldots, h_K]^T$ represents fading conditions for each sensor node k. Here the matrix $\mathbf{S} \in \mathbb{C}^{N_s\times K}$ is spreading factor which reflects the mapping relations for subcarriers. $\mathbf{N}$ denotes the additive white Gaussian noise (AWGN) matrix. From (2.5), we need recover $K \times M$ unknown symbols by using $N_s \times M$ measurements. Since $N_s \leq K$, this is a kind of underdetermined system. Then, the CS algorithm can applied to solve the problem if the sparsity of $\mathbf{X}$ can be guaranteed.

2.2 Block Type CS-MUD Problem

In this part, we give detailed analysis for block type CS-MUD problem. Based on previous single type CS-MUD case, this problem is an extension case. To distinguish with single type CS-MUD case, the system model formulation for block type CS-MUD problem could be expressed as

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{S}}\tilde{\mathbf{H}}\tilde{\mathbf{X}} + \tilde{\mathbf{N}}. \tag{2.8}$$

We will discuss the dimension of each matrix in next part. In practical massive IoT application, we always take the bandwidth efficiency into consideration. In order to improve the spectrum efficiency, we allow an overlapped manner for subcarrier spacing. Thus, the discussion for block type CS-MUD will be given with non-overlapped and overlapped cases.

2.2.1 Non-overlapped Case

We assume the modulated data are spaced within subcarriers in a non-overlapped manner (see Figure 2.3). The transmitted data matrix is enlarged with $\tilde{\mathbf{X}} \in \mathbb{R}^{(N_w\cdot K_w)\times M}$. While K_w is the number of sensor nodes in block w and w is the total number of blocks.

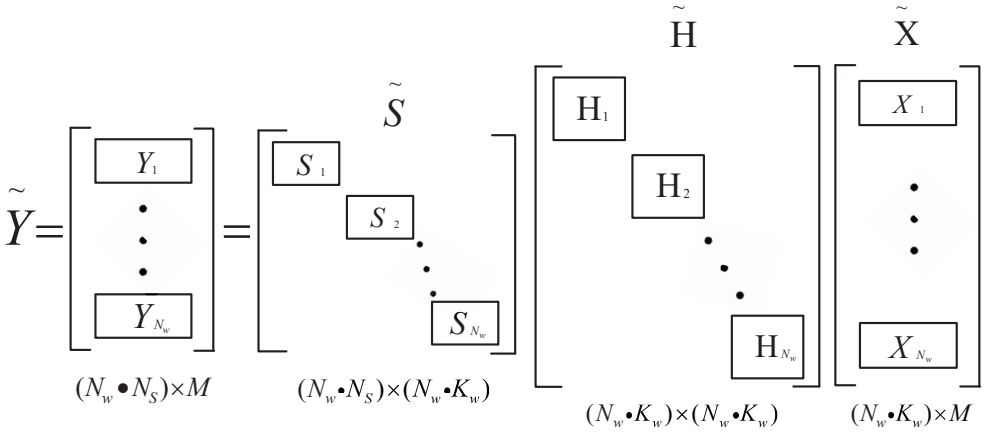


Figure 2.3: Block type CS-MUD with non-overlapped manner

The k_w^{th} row also denotes the sensor data from the k_w^{th} node. The channel matrix $\tilde{\mathbf{H}} \in \mathbb{C}^{(N_w \cdot K_w)\times(N_w \cdot K_w)}$ is extended with corresponding channel conditions for each sensor node.

$$\tilde{\mathbf{H}} = diag\{[\mathbf{h}_1^T, \ldots, \mathbf{h}_w^T, \ldots, \mathbf{h}_{N_w}^T]^T\}. \tag{2.9}$$

$\tilde{\mathbf{N}}$ is the extended AWGN matrix.

Then, we note matrix $\tilde{\mathbf{S}} \in \mathbb{C}^{N_w \cdot N_s \times N_w \cdot K_w}$ reflects the spreading and overlapped patterns in subcarriers. It is interesting to find a proper overlapped pattern value for this system. From Figure 2.3, $\tilde{\mathbf{S}}$ is approximate block diagonal matrix. Among them, we can decouple this system as multiple single type CS-MUD system given above.

Obviously, block type CS-MUD problem can be seen as Multiple Measurement Vector (MMV) problem in CS literature. We can apply CS algorithms to recover the block type sensor data matrix $\tilde{\mathbf{X}}$.

2.2.2 Overlapped Case

We have the same assumptions as in Section 2.1. In contrast to non-overlapped case, we allow the appearance of overlapped pattern in spreading matrix $\tilde{\mathbf{S}}$ and N_q is the overlapped pattern value. That means a pair of adjacent sub-blocks in spreading matrix

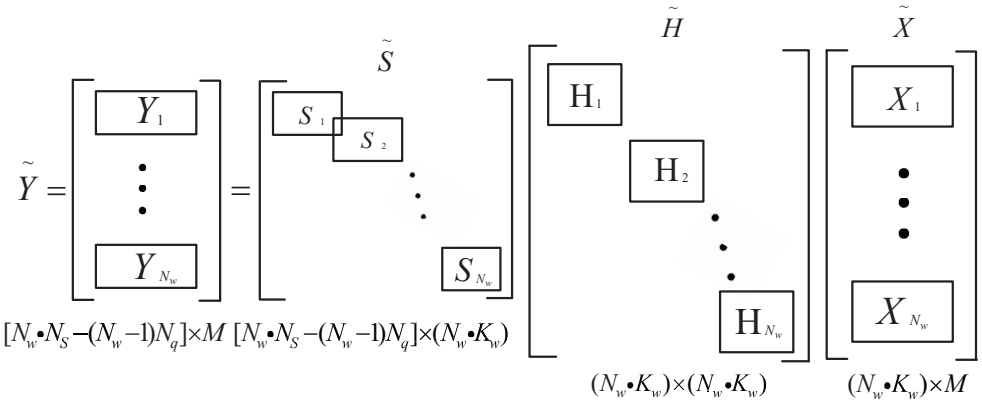


Figure 2.4: Block type CS-MUD with overlapped manner

can share N_q subcarriers for each block. To make it clear, we can see in Figure 2.4. Then, the dimension of spreading matrix is $\tilde{\mathbf{S}} \in \mathbb{C}^{[N_w \cdot N_s - (N_w - 1)N_q] \times N_w \cdot K_w}$ which allowing the overlapped pattern in each sub-block. We denote each sub-block as $\mathbf{S}_w$, where $1 \leqslant w \leqslant N_w$. The dimension of received signal becomes $\tilde{\mathbf{Y}} \in \mathbb{C}^{[N_w \cdot N_s - (N_w - 1)N_q] \times M}$. Finally, we achieve frequency saving with N_q in each block.

Because the overlapped case is a special extension on non-overlapped case, they will share the same property for matrix $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{H}}$. Then, the problem can also be converted into MMV problem in CS literature.

Generally speaking, both non-overlapped and overlapped cases are all block (pattern) reconstruction problems. In [17], Group-OMP based algorithm has been investigated for pattern reconstruction problem. While we will give a different view on the block type activity detection problem by using modified CS based algorithm.

2.3 The CS-BT-MUD Algorithm

As we know, either the OMP or SOMP algorithm is often used in the real-number field. In our massive IoT scenario, however, the original signals are expressed in complex-valued form. Before performing the conventional OMP or SOMP algorithm,

we need an appropriate pre-processing part to deal with the complex-valued data. The key idea is to use a reshape operation, which leads to a modified SOMP algorithm. See Algorithm 3.

Algorithm 3 CS-BT-MUD Algorithm

Input: measurement matrix $\mathbf{Y} \in \mathbb{C}^{m\times K}$, Dictionary $\Phi \in \mathbb{C}^{m\times n}$, Sparsity $s \geq 1$.

Output: $\tilde{\mathbf{X}}$,S.

Initialization: $\mathbf{R}^{(0)} \leftarrow \mathbf{Y}$ and $S \leftarrow \emptyset$.

1: Decouple $\mathbf{Y}$ and Φ:

$\mathbf{Y_d} \leftarrow Decouple(\mathbf{Y})$, $\Phi_d \leftarrow Decouple(\Phi)$

2: Reshape Y_d and Φ_d:

$\mathbf{Y_r} \leftarrow Reshape(\mathbf{Y_d})$, $\Phi_r \leftarrow Reshape(\Phi_d)$

3: $\mathbf{Q} \leftarrow diag(q_1, ..., q_K)$

4: The columns of $\mathbf{Y_r}$ are weighted beforehand:

$\mathbf{Y_r} \leftarrow \mathbf{Y_r Q}$

5: Apply the regular SOMP algorithm:

$S \leftarrow \mathbf{SOMP}(\mathbf{Y_r}, \Phi, s)$

6: **return** S,$\mathbf{X_{est}}$

7: Transform $\mathbf{X_{est}}$:

$\tilde{\mathbf{X}} \leftarrow \mathbf{X_{est}}$

8: return $\tilde{\mathbf{X}}$

end

We give a detailed analysis for the procedures of our proposed algorithm. Generally, we divide the procedure into three parts, which are called decouple, reshape, and transform, respectively. In the decouple part, the general purpose is to enable conventional CS algorithms to process complex-valued signals. Specifically, the measurements $\mathbf{Y}$ and dictionary Φ are decoupled into their real and imaginary parts as follows:

$$\mathbf{Y_d} = \begin{bmatrix} \mathscr{R}\{\mathbf{Y}\} \\ \mathscr{I}\{\mathbf{Y}\} \end{bmatrix} \text{ and } \Phi_d = \begin{bmatrix} \mathscr{R}\{\Phi\} & -\mathscr{I}\{\Phi\} \\ \mathscr{I}\{\Phi\} & \mathscr{R}\{\Phi\} \end{bmatrix} \tag{2.10}$$

where $\mathscr{R}\{\cdot\}$ and $\mathscr{I}\{\cdot\}$ are the real and imaginary operators, respectively. We note that the structure of the Φ_d is slightly different from $\mathbf{Y}$. The reason for this structure is to match the dimension with $\mathbf{Y_d}$. Thus, by operation SOMP algorithm, we could achieve the recovered signal $\mathbf{X_{est}}$ with real part $\mathbf{X}^r$ and imaginary part $\mathbf{X}^i$.

$$\mathbf{X_{est}} = \begin{bmatrix} \mathbf{X}^r \\ \mathbf{X}^i \end{bmatrix}. \tag{2.11}$$

In the reshape part, the main target is converted sensor data matrix from sporadic IoT communication into a row shared common support structure. This operation will make it possible for the SOMP algorithm to process complex data. Let $\mathbf{x_i}$ denote the i-th column of $\mathbf{X}$ with $i = 1, 2, \cdots, 2n$. For sporadic IoT communication, if the duration $t = n$, the reshape operations is done as follows:

$$\mathbf{X_r} = \begin{bmatrix} \mathbf{x_1} & \cdots & \mathbf{x_n} \\ \mathbf{x_{n+1}} & \cdots & \mathbf{x_{2n}} \end{bmatrix}, \tag{2.12}$$

which, together with decouple and reshape operation, allows us to recover the expected signal matrix by using (2.10), (2.11), (2.12).

Finally, in the transform part, we will transform the results given by SOMP algorithms. The first step is to reshape the results by using the same method in (2.12). Then, the second step is to combine the results. Totally, we achieve a complex-valued results. Moreover, we add a noise stabilization operation (Step 3,4). This stabilization is performed by multiplexing weighted coefficients

$$\mathbf{Q} = (\mathbf{q_1}, \cdots, \mathbf{q_K}).$$

2.4 Detection Criterion and Complexity Analysis

In CS literature, exact recovery criterion has been developed for OMP [18]. Along with practical scenarios, we also derive detection criterion for our proposed CS-BT-MUD algorithm. Before providing the details, we give some useful definitions as

follows. If $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k) \in \mathbb{R}^{m \times k}$ where $\mathbf{u}_l \in \mathbb{R}^m (1 \leq l \leq k)$, then $supp(\mathbf{U}) = \cup_{l \in [k]} supp(\mathbf{u}_l)$. This definition extends the original version of support to matrices. Then, $||\mathbf{U}||_0 = |supp(\mathbf{U})| = |\cup_{l \in [k]} supp(\mathbf{u}_l)|$ and $||\phi||_{\min} = \min_{i \in [m]} |\phi_i|$ which is not a norm.

We note that the maximum correlation in Step 5 of our proposed algorithm can be rewritten as

$$\max_{j \in [m]} (\sum_{k=1}^{K} |\langle \mathbf{r}_k^{(t)}, \phi_j \rangle| q_k) = ||\Phi_S^T \mathbf{R}^{(t)} \mathbf{Q}||_\infty, \tag{2.13}$$

where $\mathbf{R}^{(t)}$ is the residual at the t-th iteration, $\mathbf{SH} = \Phi$ may be viewed as a dictionary matrix whose columns (i.e., $\phi_j,\ j \in [m]$) are the atoms of the associated dictionary.

Let us assume that our proposed algorithm has made correct decisions before the t-th iteration. Then, the contribution of the noise $\mathbf{N}$ and that of the useful signal $\mathbf{X}$ to the residual can be expressed as:

$$\begin{aligned} \mathbf{R}^{(t)} &= (\mathbf{I} - \mathbf{P}^{(t)})\mathbf{Y} \\ &= (\mathbf{I} - \mathbf{P}^{(t)})(\Phi\mathbf{X} + \mathbf{N}) \\ &= (\mathbf{I} - \mathbf{P}^{(t)})\Phi\mathbf{X} + (\mathbf{I} - \mathbf{P}^{(t)})\mathbf{N}, \end{aligned} \tag{2.14}$$

where we give the new definition $\mathbf{Z}^{(t)} = (\mathbf{I} - \mathbf{P}^{(t)})\Phi\mathbf{X}$ and $\mathbf{N}^{(t)} = (\mathbf{I} - \mathbf{P}^{(t)})\mathbf{N}$. $\mathbf{P}^{(t)}$ is the projector onto the span of Φ_{S_t}, we have

$$\mathbf{R}^{(t)} = \mathbf{Z}^{(t)} + \mathbf{N}^{(t)}, \tag{2.15}$$

Thus, we have

$$||\Phi_S^T \mathbf{R}^{(t)} \mathbf{Q}||_\infty = ||\Phi_S^T (\mathbf{Z}^{(t)} + \mathbf{N}^{(t)})\mathbf{Q}||_\infty \tag{2.16}$$

To guarantee accuracy of the SOMP algorithm, we have

$$\begin{aligned} ||\Phi_S^T \mathbf{R}^{(t)} \mathbf{Q}||_\infty &\geq ||\Phi_{\bar{S}}^T \mathbf{R}^{(t)} \mathbf{Q}||_\infty \\ &= ||\Phi_{\bar{S}}^T (\mathbf{Z}^{(t)} + \mathbf{N}^{(t)})\mathbf{Q}||_\infty \end{aligned}. \tag{2.17}$$

Then, the SOMP algorithm with noise is lower bounded by (2.17) where $\bar{S}$ is the relative component of set S.

Table 2.1: Complexity of the CS-BT-MUD Algorithm (k-th iteration)

Operation	Complexity
Identification	$(2N_s-1)KM+K(M-1)$
Augmentation	$\frac{K(K+1)}{2}$
Estimation	$4kN_sM$
Residual Update	$2kN_sM$
Total	$2N_sKM+\frac{K(K-1)}{2}+6kN_sM=O(N_sMK)$

For CS algorithms, the computational complexity depends highly on the number of iterations. For the single type model with OMP algorithm, the computational complexity is dominated by maximum correlation calculation (**identification**) and new residual generation (**residual update**). For the k-th iteration, we need a matrix-vector multiplication with the number of floating point operations (flops) $(2m-1)K$. For residual update, we need compute the least squares of input with $4km$. Additional $2km$ is required for residual update. Generally, the total number of flops for OMP is about $2\kappa mK+3\kappa^2 m$ where κ is equal to the number of iterations.

Since our proposed CS-BT-MUD algorithm is based on the SOMP algorithm, the analysis procedure for computational complexity is somewhat similar to that for the OMP algorithm. We give the complexity analysis for the main steps of our algorithm as follows:

- **Identification**-We need compute the sum of correlation values between each atom and residual. The k-th computation cost for this step is $(2N_s-1)KM+K(M-1)$.

- **Augmentation**-Augmentation is a kind of data sorting, we only need to find out the maximum value for each iteration. The computational complexity is similar to bubble sorting algorithm with $\frac{K(K+1)}{2}$ [19].

The rest steps are similar to the OMP algorithm where the complexity of the estimate and residual update steps are specified in [20]. Since the dimension of input has been enlarged, the computation complexity for the rest part amounts to $6kN_sM$ at the k-th iteration. In summary, the total computational complexity for our proposed CS-BT-MUD algorithm is given by

$$\begin{aligned} C_{\text{CS-BT-MUD}} &= \sum_{k=1}^{\kappa} [2N_sKM + \frac{K(K-1)}{2} + 6N_sM]k \\ &= [N_sKM + \frac{K(K-1)}{4} + 3N_sM](\kappa^2 - 1) \\ &= O(\kappa^2 N_sMK) \end{aligned} \tag{2.18}$$

Chapter 3

SIMULATION RESULTS

In this part, we will provide simulation results for our proposed CS-BT-MUD algorithm. In our simulation, two practical data traffic models (i.e., the Bernoulli and Poisson-Zeta traffic models) are considered.

3.1 Parameter Design

We choose a practical NB-IoT scenario to begin our simulation parameter design. In this NB-IoT scenario, we assume the carriers are deployed in LTE network with guard-band mode. In order to evaluate the performance of our proposed algorithm, several system parameters are given as follows. The total number of sensor nodes is $K = 80$ in each block, and the time symbol number is $M = 80$. The activity probability for each sensor node in matrix $\mathbf{X}$ is defined as $p_a = 0.1$. This activity probability also follows the Bernoulli distribution. We assume a block fading channel, where the channel matrix $\mathbf{H}$ is generated by Rician channel function.

For spreading code design, we choose a PN-code with flexible length N_c. The length of the spreading code will depend on different data traffic model which will be discussed separately in data traffic model part. We define a new parameter called system load ratio factor with its definition as $\theta = \frac{K}{N_c}$. The system load ratio factor reflects

the overlapped degree and mapping between the sensor data matrix and corresponding subcarriers.

Basically, we assume the scenario with delay spread of $\tau_d = 16.67\ us$ in LTE system by using a long cyclic-prefix. Then, the coherence bandwidth can be computed as $B_{\text{coherence}} = \frac{1}{\tau_d} = 60\text{kHz}$. We also assume N_w consecutive subcarriers are deployed in the coherence bandwidth, the signals will suffer from flat fading channel within one block.

We have summarized the system parameters in Table 3.1. Then, we consider the simulation results for the BER performance in two types of data traffic model. One is called Bernoulli traffic model and the other one is Poisson-Zeta traffic model.

Table 3.1: Simulation Parameters

Number of Nodes	$K = 80$
Spreading Code Length	$N_c = 40, 50, 60$
Number of Subcarriers	$N_s = 40, 50, 60$
Activity Probability	$p_a = 0.1$
System Load	$\theta = 2, 1.6, 1.33$
Symbol Frame Size	$M = 80$
Modulation	DPSK
Channel Model	Block fading
Power Control	Perfect
Delay Spread	16.7 *us*

3.2 Bernoulli Traffic Model

For some general massive IoT applications, the activity of sensor is triggered by certain events, such as temperature change, battery level change and emerging alarm. These applications have the feature that its event duration is flexible with very small activity probability. Consequently, the probability of activity event is unpredictable at a dedicated time point. While for a long period, we can assume the probability of event activity will follow a statistic distribution.

Generally, we use simple bernoulli distribution to represent the randomness of these IoT applications. Thus, we make the reasonable assumption for the activity probability p_a in massive IoT application, only a few sensor nodes are activated at time point n with $p_a \ll 1$. Obviously, the probability of a sensor node to keep silence is $1 - p_a$.

Mathematically, we formulate Bernoulli data traffic model by using a $K \times M$ sensor data matrix. In this sensor data matrix, the elements in each column are defined as the activity status of sensor nodes at time point n. For convenience, n is defined as an integer with $1 \leq n \leq M$ denotes the time symbol length. Thus, at the time point n, only a few sensor nodes will be activated among total K sensor nodes which follows bernoulli distribution. Also, the distribution for each column is identical independent distribution.

$$\mathbf{X} = \underbrace{\begin{bmatrix} \overbrace{x_{11}}^{X_1} & \overbrace{x_{12}}^{X_2} & \cdots & \overbrace{x_{1(M-1)}}^{X_{M-1}} & \overbrace{x_{1M}}^{X_M} \\ x_{21} & x_{22} & \cdots & x_{2(M-1)} & x_{2M} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{(K-1)1} & x_{(K-1)2} & \cdots & x_{(K-1)(M-1)} & x_{(K-1)M} \\ \underbrace{x_{K1}}_{\text{Each colunmn elements follows bernoulli distribution with probability} p} & x_{K2} & \cdots & x_{K(M-1)} & x_{KM} \end{bmatrix}}_{K \times M} \tag{3.1}$$

In (3.1), we defined the activity status as '1' and '0' respectively for each column

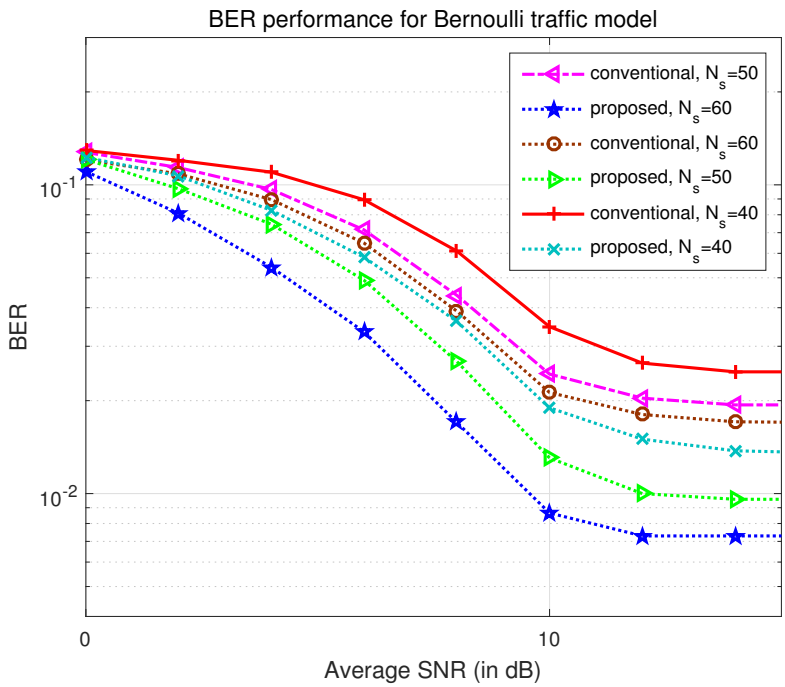


Figure 3.1: Bernoulli traffic model, $\theta = 2, 1.6, 1.33$

of signal matrix $\mathbf{X}$. When the sensor node is activated, "1" will be generated at corresponding point. For the silent sensor node, '0' will be given at the corresponding place. Then, only a few non-zero elements in each column. This can be approximately formulated as a sparse matrix.

As introduced in Section 1.3, CS algorithms are well-known for solving sparse signal reconstruction problems. We begin the simulation for Bernoulli traffic model by using the parameters in Table 1. The signal matrix $\mathbf{X}$ is generated by Bernoulli traffic model with activity probability $p_a = 0.1$ in each column. We perform each simulation for 100 times.

In the simulation, we compare the bit error rate (BER) performance of our proposed method with that of the conventional real-valued method. As can be seen from Figure 3.1, the proposed method provides better performance in relative high SNR

field for all tested θ (i. e, $\theta = 2, 1.6, 1.33$), while in low SNR field, there is no obvious improvement in BER performance. It can also be seen that the BER performance improves in Bernoulli traffic model as the system load ratio value decreases.

3.3 Poisson-Zeta Traffic model

Before we introduce Poisson-Zeta traffic model, an important concept of ON/OFF process will be discussed. An ON/OFF process is either in state ON or OFF which is from network traffic field. We follow the concept of Poisson-Zeta process in [21] and give a new definition for Poisson-Zeta traffic model as Figure 3.2.

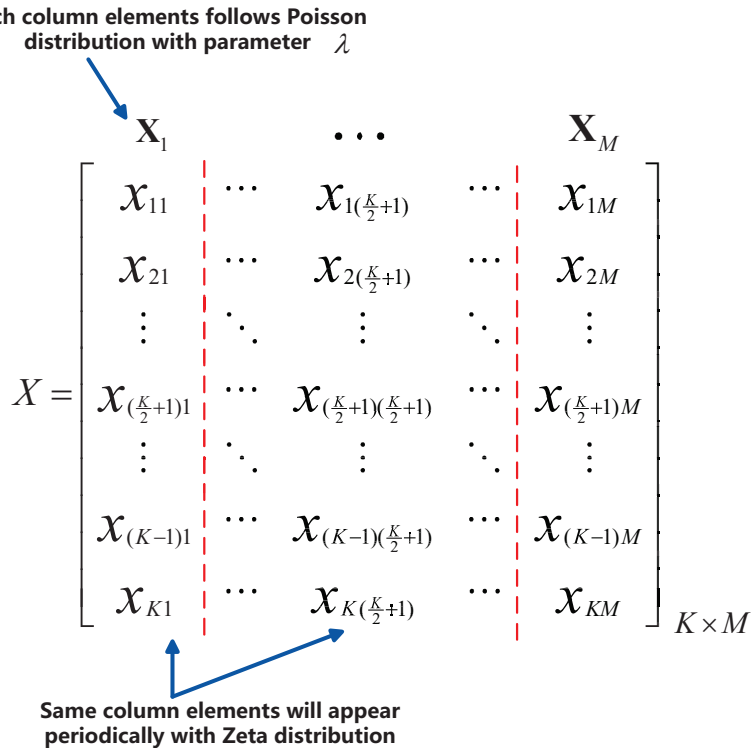


Figure 3.2: Poisson-Zeta traffic model

The activity of each sensor in each time point follows Poisson distribution with mean value of λ. We define l as the duration for each activated time point at the same

row. This duration will follow zeta distribution with probability g_l. We give a simplified expression as $\mathbf{X} \sim PZ[\lambda, g_l]$.

In some IoT applications, the activity data of the sensor nodes has the nature of periodicity. Especially for sporadic IoT communications, if there are enough time duration, a specific signal matrix will appear again after some durations. Based on previous assumption that '0' stands for the silence and '1' for the activity of sensor node, the formulated sensor data matrix only has two kinds of elements. Interestingly, the nature of periodicity in massive IoT applications can be approximately formulated by Poisson-Zeta traffic model.

In Figure 3.2, Poisson-Zeta traffic model is given by a $N \times M$ sensor data matrix $\mathbf{X}$. We assume the number of elements "1" in each column just follows poisson distribution with parameter λ. Each column of sensor data matrix $\mathbf{X}$ will appear periodically with duration l and $g_l = 1$. Based on the previous assumption on practical activity probability in Section 3.2, we have a very small value of λ (often less than 10 among total 60) which guarantees the sparsity of signal matrix.

For the Poisson-Zeta traffic model case, we make an approximation for generation of data traffic. We begin the simulation for Poisson-Zeta model with following parameters. The number of sensor nodes is $K = 80$, and the time symbol number is $M = 80$. $\mathbf{X}$ is generated by poisson distribution with mean value $\lambda = 4$. For fixed period $l = 10$, the same data will appear again. In Section 2.2, we have given analysis for block type CS-MUD problem with non-overlapped and overlapped manner. Then, the simulation for Posson-Zeta traffic model will be divide by the non-overlapped and overlapped cases.

3.3.1 Non-overlapped Case

In the simulation result for non-overlapped case (Figure 3.3), our proposed CS-BT-MUD algorithm provides better BER performance in $\theta = 2$, $\theta = 1.6$ and $\theta = 1.33$ cases. We note that different system load ratio also provides different performance. We

will give a performance comparison for different system load ratio later.

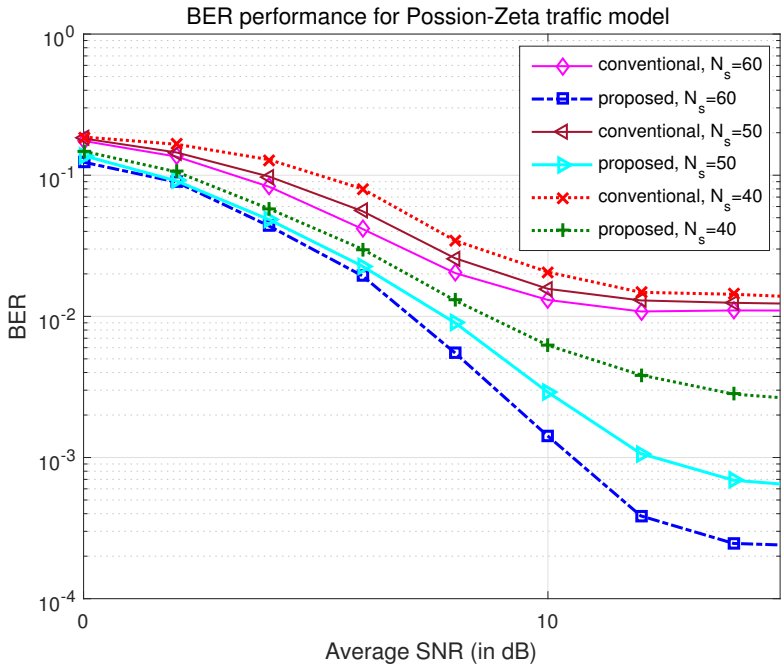


Figure 3.3: Poisson-Zeta traffic model with non-overlapped manner, $\theta = 2, 1.6, 1.33$

3.3.2 Overlapped Case

In Section 2.2.2, we have introduced an overlapped case for block type CS-MUD problem. Due to overlapping in block type spreading matrix, we can achieve improved spectrum efficiency. Based on previous setting parameters in Section 3.1, we investigate the relationship between the overlapped value and the BER performance. Specifically, we define a new parameter N_q to denote the overlapped pattern for mapping. Then, we give the definition for the saving in spectrum as

$$\Lambda_{\text{save}} = \frac{(N_w - 1)}{N_w N_s} \cdot N_q. \tag{3.2}$$

where N_w is the number of blocks with its corresponding number of sensor nodes K_w. N_q is the number of overlapped pattern, and N_s is the number of subcarriers.

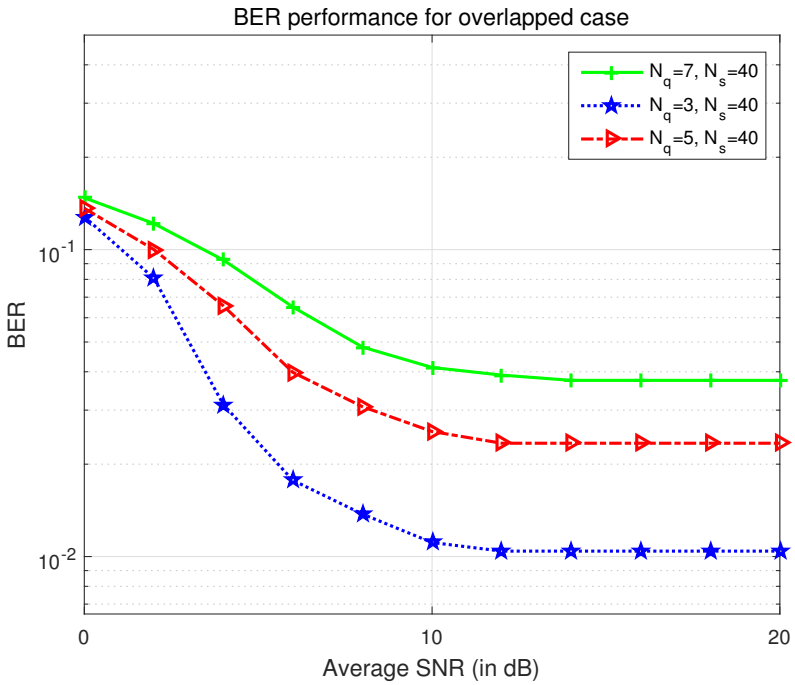


Figure 3.4: Poisson-Zeta traffic model with overlapped manner, $N_s = 40$

Then, the simulation setting is the same as that for the non-overlapped case. We let $N_s = 40, 60$ and $N_w = 6$ with $N_q = 3, 5, 7$, so that we achieve a saving in spectrum. For $N_s = 40$, a saving in spectrum can be achieved as $6.25\%, 10.4\%$ and 14.5%. Whereas for $N_s = 60$, we achieve the frequency saving as $4.16\%, 6.94\%$ and 9.7%.

From the simulation results in Figure 3.4 and 3.5, when increasing the total number of subcarriers N_s, the BER performance will be improved. This is caused by the decreasing in system load ratio value. While increasing the overlapped pattern value, the BER performance will be decreased. Thus, choose a proper value for overlapped patter value is very important for different IoT applications.

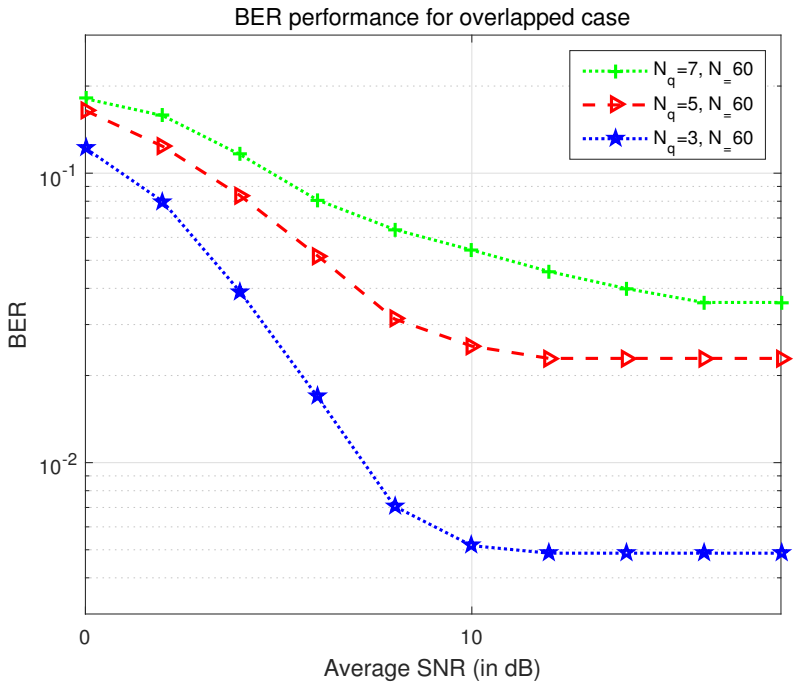


Figure 3.5: Poisson-Zeta traffic model with overlapped manner, $N_s = 60$

3.4 Discussions on System Load Ratio

As mentioned, the definition of system load ratio reflects the relationship between the number of sensor nodes and mapped frequency sources. It is important to find a suitable value for system load ratio in practical IoT communications. We begin the simulation with Poisson-Zeta traffic model and change a much smaller activity probability $p_a = 0.02$. Other parameters are just same with Section 3.1.

By changing the value of N_s, the sensor data will be mapped into N_s subcarriers. Then, for different N_s values we could achieve corresponding system load ratio θ. In this situation, the values of system load ratio are $\theta = 2$, $\theta = 1.6$, and $\theta = 1.33$. We can see the simulation results in Figure 3.6.

From simulation results, we can find that our proposed algorithm will achieve best performance with $\theta = 1.33$. But for other values, the performance is not good

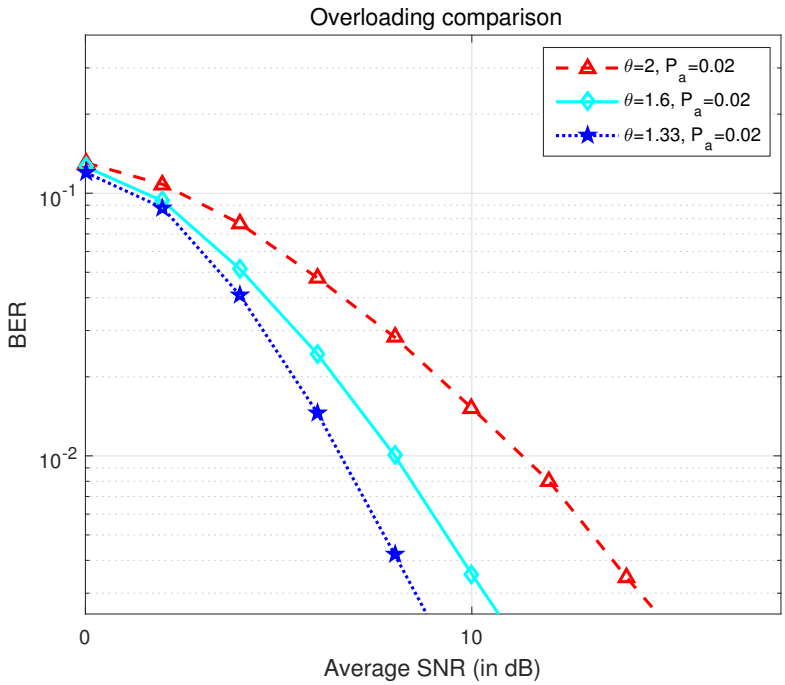


Figure 3.6: System load ratio discussions

as $\theta = 1.33$ case. For classical greedy algorithms, there is a sufficient condition for recovering sparsest representation of an input signal. By experiments, we find relationship between the sparsity of $\mathbf{X}$ and number of subcarriers N_s. If sparsity κ is much smaller than N_s, the BER performance will be guaranteed. Combined with practical requirements, this condition will provide a strategy for value selection for the number of sensor nodes in each block.

3.5 Reconstruction Error

In this part, we define a new parameter as reconstruction error which reflects the exact recovery degree of each algorithms. Then, the definition of reconstruction error

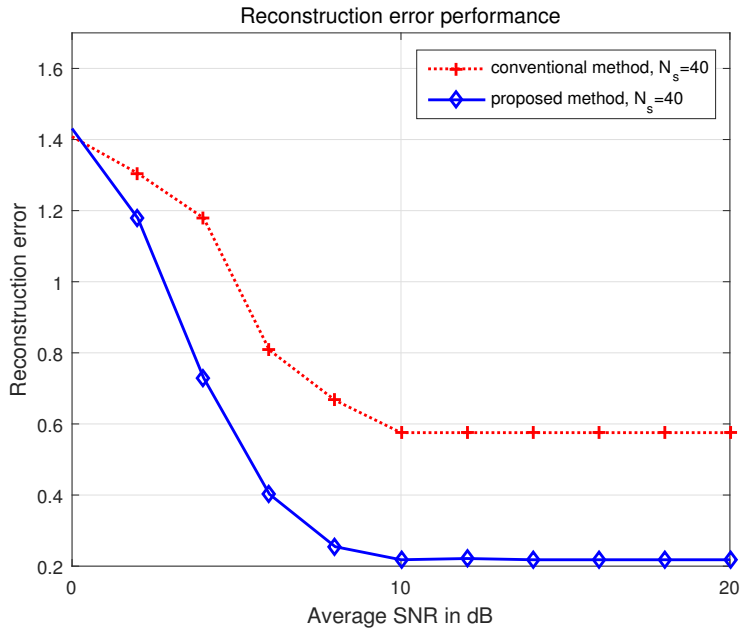


Figure 3.7: Reconstruction error with $\theta = 2$

is given as

$$C = \frac{\|X - X_{est}\|_2}{\|X\|_2}, \tag{3.3}$$

where X and X_{est} are the original signal and the estimated signal, respectively. By computing the normalized mean square error of estimated signal, we could achieve the performance of reconstruction error. The simulation is done based on same parameter setting in Section 3.4. As we can see in the simulation results (Figure 3.7, 3.8 and 3.9), our proposed method provides better performance in $\theta = 2$, $\theta = 1.6$ and $\theta = 1.33$ cases. When we increase the system load ratio value, the reconstruction error of the proposed method will increase. In contrast, for conventional methods, the performance dose not change a lot.

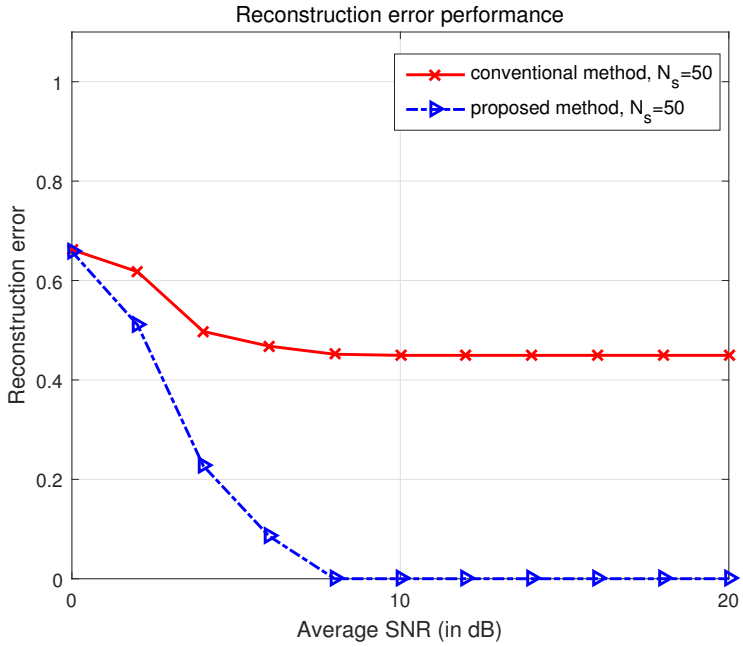


Figure 3.8: Reconstruction error with $\theta = 1.6$

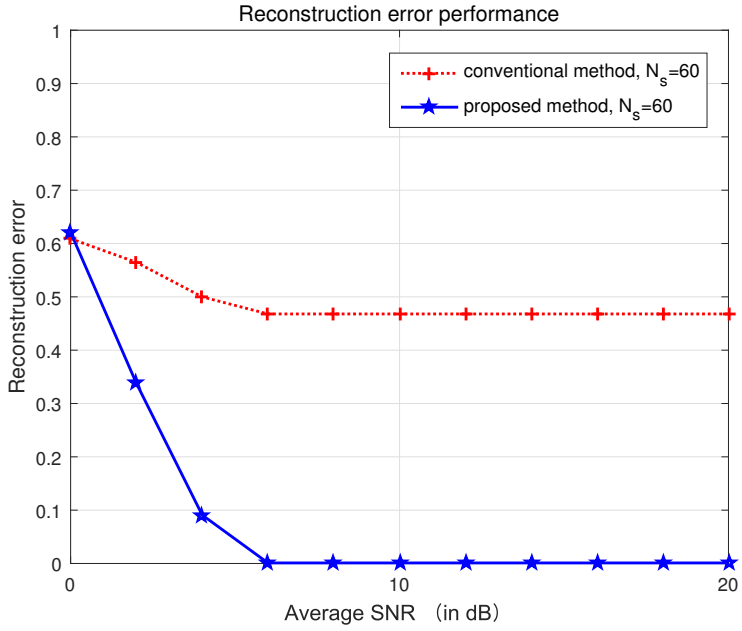


Figure 3.9: Reconstruction error with $\theta = 1.33$

Chapter 4

CONCLUSIONS

In this dissertation, we have studied the main challenges for IoT communications. Specifically, we have introduced an algorithm, termed as the CS-BT-MUD algorithm, for addressing the MAI in massive IoT applications. While previous researches on massive access have been focused on single type CS-MUD problem, we have given a new CS based algorithm for block type CS-MUD problem which exploits the block sparsity nature in the activity detection. Moreover, we have given simulation results under two types of data traffic models, instead of the simple Bernoulli data traffic model in previous works. The aim of this dissertation is to provide feasible solutions for addressing MAI in massive IoT applications and to apply the solutions to physical layer design in IoT communications.

The main contributions of this dissertation are summarized as follows.

- We have proposed CS-BT-MUD algorithm for block type CS-MUD problems and exploits block type activity detection problem.
- We have proposed complex-valued data preprocessing method which enables to process complex-valued data with conventional CS algorithms.
- We have given two practical data traffic models for numerical evaluations.

Essentially, various IoT applications have attracted much research interest in the past few years as these techniques will change the life style. For more practical challenges, we need more creative solutions. This dissertation is part of our efforts in this direction. For more critical MAI in physical layer, we remain it as our future work.

Bibliography

[1] ITU-T Technology Watch Report, “The Tactile Internet,” August 2014.

[2] Kwang-Cheng Chen and Shao-Yu Lien, “Machine-to-machine communications: technologies and challenges,” *Ad Hoc Networks.*, vol. 18, pp. 3-23, July 2014.

[3] Jeffrey G. Andrews; Stefano Buzzi; Wan Choi; Stephen V. Hanly; Angel Lozano; Anthony C. K. Soong; Jianzhong Charlie Zhang, “What will 5G be?,” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065-1082, June 2014.

[4] Ericsson White Paper, “Cellular networks for massive IoT,” Uen 284 23-3287, January 2016.

[5] LoRa alliance White Paper, “A technical overview of LoRa and LoRaWAN” , Technical Marketing Workgroup 1.0, November 2015.

[6] 3GPP, TR 37.869, “Study on enhancements to machine-type communications (MTC) and other mobile data Applications; Radio access network (RAN) aspects,” Rel.12, September 2013.

[7] Claire Goursaud and Jean-Marie Gorce, “Dedicated networks for IoT: PHY/MAC state of the art and challenges,” EAI endorsed transactions on Internet of Things, November 2015.

[8] Hamidreza Shariatmadari; Rapeepat Ratasuk; Sassan Iraji; Andrés Laya; Tarik Taleb; Riku Jäntti; Amitava Ghosh, “Machine-type communications: current sta-

tus and future perspectives toward 5G systems," *IEEE Communications Magazine.*, vol. 53, no. 9, pp. 10–17, September 2015.

[9] Rapeepat Ratasuk; Athul Prasad; Zexian Li; Amitava Ghosh; Mikko A. Uusitalo, "Recent advancements in M2M communications in 4G networks and evolution towards 5G," in *Proceedings of 18th International Conference on Intelligence in Next Generation*, Paris, France, February 2015. pp. 52–57.

[10] Guiyong Zhang; Guyoung Lim; Hyoungju Ji; Byonghyo Shim, "Recent physical layer technologies for IoT communications," in *Proceedings of IEEE 13th Vehicular Technology Society Asia Pacific Wireless Communications Symposium*, Tokyo, Japan, August 2016.

[11] 3GPP, TR 45.820, "Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT)," Rel.13, November 2015.

[12] Minh-Tien DO, "Ultra-narrowband wireless sensor networks modeling and optimization," Institut National des Sciences Appliqué*e*s de Lyon, doctoral thesis, July 2015.

[13] Kan Zheng; Suling Ou; Jesus Alonso-Zarate; Mischa Dohler; Fei Liu; Hua Zhu, "Challenges of massive access in highly dense LTE-advanced networks with machine-to-machine communications," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 12–18, June 2014.

[14] Ghasem Naddafzadeh-Shirazi; Lutz Lampe; Gustav Vos; Steve Bennett, "Coverage enhancement techniques for machine-to-machine communications over LTE," *IEEE Communications Magazine.*, vol. 53, no. 7, pp. 192–200, July 2015.

[15] Tony Tony Cai and Lie Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4680–4688, July 2011.

[16] Joel A. Tropp; Anna C. Gilbert; Martin J. Strauss, “Algorithms for simultaneous sparse approximation: part I: Greedy pursuit,” *Signal Processing*, vol. 86, issue. 3, March 2006.

[17] Fabian Monsees; Matthias Woltering; Carsten Bockelmann; Armin Dekorsy, “Compressive sensing multi-user detection for multi-carrier systems in sporadic machine type communication,” in *Proceedings of IEEE 81st Vehicular Technology Conference: VTC2015-Spring*, Glasgow, Scotland, May 2015.

[18] Jean-François Determe; Jérôme Louveaux; Laurent Jacques; François Horlin, “On The exact recovery condition of simultaneous orthogonal matching pursuit,” *IEEE Signal Processing Letters.*, vol. 23, no. 1, pp. 164–168, January 2016.

[19] Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein, “ *Introduction to Algorithms*”, Second Edition. MIT Press and McGraw-Hill, 2001.

[20] Jian Wang; Seokbeop Kwon; Byonghyo Shim, “Generalized orthogonal matching pursuit,” *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6202–6216, December 2012.

[21] Helmut Hlavacs; Gabriele Kotsis; Christine Steinkellner, “Traffic source modeling,” *Technical Report TR-99101*, Institute for Applied Computer Science and Information Systems, 1999.

초 록

지난 10년 동안 사물인터넷 (IoT) 기술이 무선 네트워크 분야에서 주목을 받아 왔다. 사물인터넷은 수많은 디바이스가 서로 연결하는 플랫폼으로 사물들 사이의 통신을 원활하게 한다. 사물인터넷 기술 기반으로 개발된 다양한 서비스들이 앞으로 여러 분야에서 사람들의 생활에 많은 영향을 줄것이다. 하지만 실제로 사물인터넷을 적용하기 위해서도 아직도 해결해야 할 많은 이슈들이 존재한다. 그중에서도 대규모 접속 이슈가 중요하다. 따라서, 본 논문은 사물인터넷을 위한 대규모 접속 기술에 대해서 연구를 진행했고 block sparsity 를 이용하여 대규모 접속 이슈를 해결하기 위한 압축 센싱 기반 block type multi-user detection (CS-BT-MUD) 알고리즘을 개발하였다. 특히, 복소수 형태의 데이터를 처리하는데 있어서 압축 센싱 기반 알고리즘을 응용 할수 있는 새로운 신호처리 기법도 함께 제시하였다. 다양한 데이터 트래픽 모델을 활용하여 CS-BT-MUD 알고리즘의 성능을 모의시험을 통해 검증을 진행하였으며 기존 알고리즘 대비 향상된 성능을 보여주었다.

주요어: 압축 센싱, 사물인터넷, 무선 네트워크, 대규모 접속

학번: 2015-22137