國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master's Thesis

大規模非同步多用戶之通道使用的延遲和活動偵測

Joint Delay and User Activity Detection in Asynchronous Massive MAC

于士程

Shih-Cheng Yu

指導教授：林士駿 博士

Advisor: Shih-Chun Lin, Ph.D.

中華民國 113 年 8 月

August, 2024

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

致謝

謝謝林士駿教授的指導。

謝謝我家人資助我生活費。

謝謝我實驗室同學: 鍾軒、信樺、浩鈞、智堯、展翌的陪伴。

謝謝口委撥冗參加我的口試，並且給予我寶貴的意見。

謝謝實驗室學弟: 峻儀、羿賢、育林、彥承幫忙處理實驗室的事物。

謝謝學長回答我的各種疑難雜症，讓我對未來有更清楚的認識。

謝謝實驗室舒適的環境讓我可以在裡面做事。

謝謝碩士期間的一切。

碩士生涯說長不長，說短也不短，但絕對值得未來的我去回憶。

2024/07/10 于士程

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

摘要

在大規模物聯網（IoT）中，為了省電，許多設備處於不傳送數據的狀態，每段時間中僅有少數用戶處於活動狀態。為了保證可靠的數據傳輸，檢測這些物聯網設備中的活動用戶為哪些設備及其路徑延遲非常重要。為了應對這種情況，我們利用熱帶演算法來幫助我們以非常少的正交資源達成目標，這與系統中的總用戶數相比是非常少的。此外，為了減少延遲，我們採用長度有限的碼，可以使得接收端在不必等待所有碼到達的情況下知道活動用戶及其路徑延遲。我們還發現了接收碼的有限時間期限，以及其與碼和路徑延遲之間的數學關係。最後，即使物聯網系統中的總用戶數大幅增加，我們仍能輕鬆地推導出我們所要用到的碼字，同時保持數據傳輸的可靠性。根據我們在無噪聲通道下的模擬結果，隨著正交資源數量的二次方增長，我們的設計在每個總用戶數下均無檢測錯誤。

關鍵字：物聯網、熱帶演算法、低延遲

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Abstract

In large IOT devices, in order to reduce power, most devices are silent and only few users are active for each time frame. It is important to detect who are the active users in our IOT system and their path delays to guarantee reliable data transmission. We leverage tropical-arithmetic to help us reach our goal with very less orthogonal resources comparing to the number of total users in our system. Moreover, to reduce latency, the codeword we adopt is finite length and the receiver obtains message without having to wait for all codewords to arrive. We also found the relationship between finite receiving deadline and codewords' length and path delay. Lastly, we are able to derive our codebook easily when the number of total users increase dramatically in our IOT system, while still remaining reliable data transmission. According to our simulation results for noiseless channel, our design has zero detection errors for every number of total users, as the number of total users increases quadratically according to the number of orthogonal resources.

Keywords: IOT, tropical arithmetic, low latency

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Contents

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

List of Figures

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Chapter 1 Introduction

Establishing 6G networks with scalability and sustainability are crucial to meet the demands of the United Nations sustainable development goals. Specifically, internet-of-things (IoT), sensor-networks, and massive machine-type communications have attracted significant attention for 6G communications [13][10]. Classical information theory uses multiple-access channel (MAC) to analyze these systems. However, due to the dramatically increasing number of devices (transmitters) several interesting and new aspects arise. One of them is how to build an efficient communication where only a small number of users are active in a massive MAC [12]. Furthermore, the classical MAC considers individual codebooks for all devices which may be prohibitive in a massive MAC. A new promising technique unsourced MAC was proposed in [12] where all transmitters share an identical codebook and the amount of data to be transmitted at each transmitter is identical. Authors in [3] further investigate the asymptotic capacity when the number of users grows as the blocklength increases, and users apply individual codebooks for identifications and one single codebook for transmitting information.

Asynchrony among users occurs naturally in massive access. For example, authors in [1, 9] utilized the orthogonal frequency-division multiplexing (OFDM) with cyclic-prefix to transfer the time-asynchronous communication to a frequency-shift one. Similarly, authors in [4] used a sparse orthogonal frequency-division multiple access and compressed

doi:10.6342/NTU202402984

sensing-based algorithm to reliably identify arbitrarily asynchronous devices and decode messages. Recently and surprisingly, [6] revealed the advantages of asynchrony in the unsourced MAC. Tradeoff between the energy-per-bit and the number of active users in frame asynchronous unsourced multiple access channel (AUMAC) was later explored in [16].

For synchronous massive MAC [6] and even the AUMAC in [6][16], the delay profile of active users is needed at the receiver. To find this delay profile, in this paper we formulate a joint delay and user activity detection problem in terms of tropical linear algebra, which can support massive users with a few orthogonal wireless resources. Note that most previous works focused on only user activity detection [2] or jointly estimating channel coefficients instead of path delays with activity [11]. Also another line of group testing, combinatorial group testing [8], was applied for decoding packet in slot synchronous random access channel [7] where each user only has one packet. These works are fundamentally different to our problem.

Tropical linear algebra was applied to the group testing problem for Covid19 testing in the pandemic [15]. In tropical group testing, only a few samples contain Covid19 which is akin to the fact that in massive MAC typically only a few users are active. Compared with the tropical-arithmetic in [15], the new design issue is that each user must have a finite path delay bounded by the maximum one. A necessary condition for the maximum path delay to ensure zero-error detection is then identified. Simulation results also verify our claims.

For synchronous massive MAC [6] and even the AUMAC in [6][16], the delay profile of active users is needed at the receiver. To find this delay profile, in this paper we

doi:10.6342/NTU202402984

formulate a joint delay and user activity detection problem in terms of tropical linear algebra, which can support massive users with a few orthogonal wireless resources. Note that most previous works focused on only user activity detection [2] or jointly estimating channel coefficients instead of path delays with activity [11]. Also another line of group testing, combinatorial group testing [8], was applied for decoding packet in slot synchronous random access channel [7] where each user only has one packet. These works are fundamentally different to our problem.

Tropical linear algebra was applied to the group testing problem for Covid19 testing in the pandemic [15]. In tropical group testing, only a few samples contain Covid19 which is akin to the fact that in massive MAC typically only a few users are active. Compared with the tropical-arithmetic in [15], the new design issue is that each user must have a finite path delay bounded by the maximum one. A necessary condition for the maximum path delay to ensure zero-error detection is then identified. Simulation results also verify our claims.

In [11], they aim to do channel estimation for massive access, whereas we aim to do user activity detection and delay detection.

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Chapter 2 Problem Formulation

2.1 Our Scheme: a Few Active Users in Massive MAC

Let's say there are K users in our IOT system, and for every time frame there are only K_a active users, where K_a is very small comparing to K. There exist a deadline of data-transmission called ℓ_{in}. If a user's path delay is smaller than ℓ_{in}, that user is called an active user. Our goal is to find who are the active users and their corresponding path delays. Our scheme that $\frac{K}{K_a}$ is very large is quite common in nowadays IOT system, where most of the time devices are silent to reduce power. We can use few orthogonal resources to be able to detect vert small number of active users comparing the total number of users.

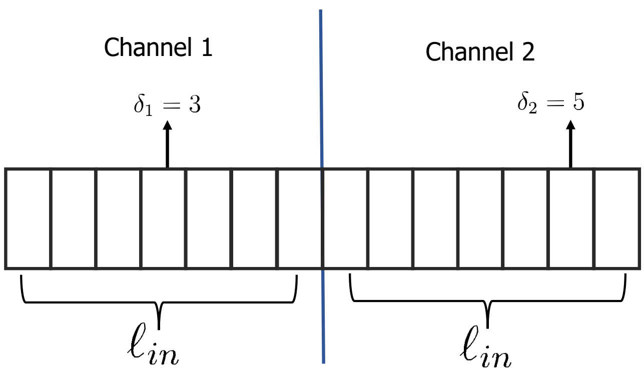

Figure 2.1: TDMA is quite time consuming

.

doi:10.6342/NTU202402984

2.2 Traditional MAC Scheme

In the traditional MAC scheme, to be able to detect K users path delays and identities we would need K resources, which is quite resource consuming. We first demonstrate two common traditional MAC methods, TDMA and CDMA.

2.2.1 Traditional TDMA Scheme

For the traditional TDMA scheme, if the deadline ℓ_{in} units of time and there are 2 users in total. In figure 2.1 ℓ_{in} is 7. The receiver would have to use $2\ell_{in}$ time units to detect users' identities and path delays, which is very time consuming and inefficient. User 1 sends a packet at time-slot 0, and it will be received at time-slot 3, since $\delta_1 = 3$. In order to prevent data sent from user 1 being received by user 2, user 2 sends its data after ℓ_{in} of time, which is at time-slot 7. Data sent by user 2 will be received at time-slot 12, since $\delta_2 = 5$. Notice that, we have to use 14 time-units for reliable data transmission. User 1 uses time-slot 0 to 6, which is channel 1. User 2 uses time-slot 7 to 13, which is channel 2.

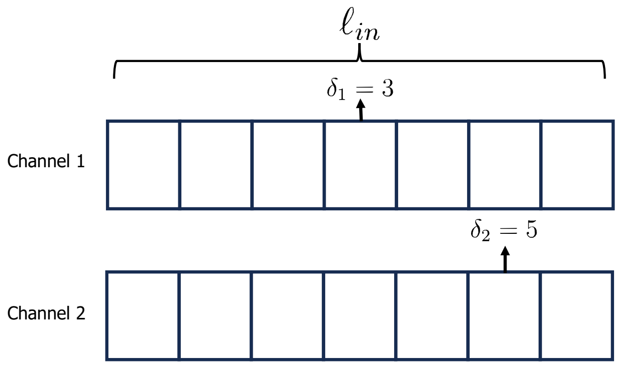

Figure 2.2: Use 2 mutual complementary orthogonal sequences to perform 2 orthogonal channels

.

doi:10.6342/NTU202402984

2.2.2 Traditional CDMA Scheme

For the traditional CDMA scheme, to be able to use less time than TDMA, which is ℓ_{in} units of time. We send 2 users' data at the same time. 2 users would have to use 2 corresponding sets that are in a mutually orthogonal complementary sets(MOCS) to create 2 orthogonal channels. From [17] we know the definition of (M,N,L)-MOCS is that the correlation between two sets is the following.

$$\rho(C^{t1}, C^{t2}; u) = \sum_{n=0}^{N-1} \rho(\mathbf{c_n^{t1}}, \mathbf{c_n^{t2}}; u) = \begin{cases} NL, & u = 0 \text{ and } t_1 = t_2 \\ 0, & 0 < |u| < L \text{ and } t_1 = t_2 \\ 0, & |u| < L \text{ and } t_1 \neq t_2 \end{cases} \tag{2.1}$$

M is number of sets composed of sequences, N is the number of sequences in a set, and L is the length of every sequence. Different users use different sets of sequences. u is the delay between two receiving sequences, namely the path delay's difference between two sequences. t_1 and t_2 are the identity for two users. Also from [17], we know that the correlation between two sequences is the following

$$\rho(\mathbf{c_n^{t1}}, \mathbf{c_n^{t2}}; u) = \begin{cases} \sum_{k=0}^{L-1-u} (\mathbf{c_n^{t1}}[k+u]) * (\mathbf{c_n^{t2}}[k])^*, & 0 \leq u \leq L-1 \\ \sum_{k=0}^{L-1+u} (\mathbf{c_n^{t1}}[k]) * (\mathbf{c_n^{t2}}[k-u])^*, & -L+1 \leq u < 0 \end{cases} \tag{2.2}$$

From [14] we gain a (2,2,4)-MOCS, which is the following.

$$\begin{aligned} \mathbf{c_1^{t1}} &= [1{-}1{-}1{-}1]; \\ \mathbf{c_2^{t1}} &= [-1{-}11{-}1]; \\ \mathbf{c_1^{t2}} &= [-11{-}1{-}1]; \\ \mathbf{c_2^{t2}} &= [111{-}1]; \end{aligned} \tag{2.3}$$

doi:10.6342/NTU202402984

$\mathbf{c}_1^{\mathbf{t1}}[\delta_1]$				1	-1	-1	-1
$+$							
$\mathbf{c}_1^{\mathbf{t2}}[\delta_2]$		-1	-1	1	-1		
Time-shift i	-1	0	1	2	3	4	5 ···
correlate with $\mathbf{c}_1^{\mathbf{t1}}[i]$	0	0	0	4	0	0	0 ···
Time-shift j	-3	-2	-1	0	1	2	3 ···
correlate with $\mathbf{c}_1^{\mathbf{t2}}[j]$	0	4	0	0	0	0	0 ···

Figure 2.3: The correlation output for the first combined sequences, according to different time-shifts

.

The first set is composed of $\mathbf{c}_1^{\mathbf{t1}}$ and $\mathbf{c}_2^{\mathbf{t1}}$, while the second set is composed of $\mathbf{c}_1^{\mathbf{t2}}$ and $\mathbf{c}_2^{\mathbf{t2}}$ We have verified it by simulation so that the above sequence satisfies equation (2.1).

From the perspective of the transmitter, user 1 sends two flocks $\mathbf{c}_1^{\mathbf{t1}}[0]$ and $\mathbf{c}_2^{\mathbf{t1}}[0]$. We can say that user 1 sends a MOCS set made up of two flocks, $\mathbf{c}_1^{\mathbf{t1}}[0]$ and $\mathbf{c}_2^{\mathbf{t1}}[0]$. User 2 sends two flocks as well, $\mathbf{c}_1^{\mathbf{t2}}[0]$ and $\mathbf{c}_2^{\mathbf{t2}}[0]$. The receiver receives two combined flocks, because there are two flocks for both users. Namely, $\mathbf{c}_1^{\mathbf{t1}}[\delta_1] + \mathbf{c}_1^{\mathbf{t2}}[\delta_2]$ and $\mathbf{c}_2^{\mathbf{t1}}[\delta_1] + \mathbf{c}_2^{\mathbf{t2}}[\delta_2]$, δ_1 and δ_2 are the path delays of users. Path delay is determined by the distance between user and the base station. To perform correct de-spreading for CDMA sequences, we need to do correlation to the received combined sequences.

We give an example in figure 2.3 for the first combined flocks, with $\delta_1 = 3$ and $\delta_2 = 1$. When we want to create the first channel, we correlate $\mathbf{c}_1^{\mathbf{t1}}[\delta_1] + \mathbf{c}_1^{\mathbf{t2}}[\delta_2]$ with $\mathbf{c}_1^{\mathbf{t1}}[i]$ under different time-shifts i. Only when time-shift $i = 2$ the correlation output is going to be $L = 4$. From figure 2.3, we can observe that u in equation (2.1) is 2 because $\delta_1 - \delta_2 = 2$. The reason for the 0 correlation output for different time-shifts except for $i = 2$ is that $t_1 \neq t_2$ and $i \neq \delta_1 - \delta_2$. Similarly, when we want to correlate with $\mathbf{c}_1^{\mathbf{t2}}[j]$, the only non-zero correlation output occurs when $j = \delta_1 - \delta_2 = -2$. Since when $j = -2$, the correlation output of $\mathbf{c}_2^{\mathbf{t1}}[-2]$ and $\mathbf{c}_2^{\mathbf{t1}}[-2]$ equals to $L = 4$. Notice that there are two combined flocks,

doi:10.6342/NTU202402984

so when time-shift equals to u the correlation outputs for both C^{t1} and C^{t2} are going to be $N * L = 8$, which is equivalent to equation (2.1). We also know that if we want to correlate with $\mathbf{c_1^{t1}}$, $u = \delta_2 - \delta_1$. If we want to correlate with $\mathbf{c_1^{t2}}$, $u = \delta_1 - \delta_2$. This is consistent to u in equation (2.2).

2.3 Pre-shifts

In reality, it is quite difficult to find many sets of MOCS sequences. That is, orthogonality still exists between sequences after arbitrary shifts is quite hard. Even if that many sequences can be generated, IoT devices are not capable of storing that many sequences. To increase the number of supporting users, we simply pre-shift existing sequences before transmission without finding new sequences. We are able to support much more users than the number of existing sequences, where tropical arithmetic is going to be introduced in the next chapter. In traditional MAC, the number of users is going to be consistent to the number of resources.

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Chapter 3 Tropical Arithmetic

3.1 System Model

In the scenario of massive MAC where user activities and path delays are unknown, it is critical to detect and estimate them in order to ensure reliable data transmission.

If only a few users are active during one period of communication time, we can formulate the problem in terms of tropical linear algebra in [15]. As in AUMAC and [7], we assume that every user can transmit symbols at discrete time index $t = 0, \ldots \infty$. However, because of different path delays for distinct links, the transmitted symbol will arrive with an uncertain path delay. Therefore, there are $K > \mathsf{K}_a$ users and user $j \in [K]$ encounters an unknown path delay, determined by the distance between user and the base-station.

$$\delta_j \in \{0, 1, \ldots, \ell_u, \ell_{in}\},$$

where $\delta_j = \ell_{in}$ if user j is inactive during training. For simplicity, delays are assumed to be discrete integers. Note that $\ell_{in} = \infty$ in [15] while is finite here. Traditionally during the training, one needs to create K orthogonal resources such as non-overlapping time frames in TDMA, to identify active users and their delays $\delta_j \neq \ell_{in}, j \in [K]$. In CDMA, K orthogonal PN sequences with perfect auto- and cross-correlations for arbitrarily starting time indexes are also needed [18]. We investigate whether we can use $T < K$ orthog-

doi:10.6342/NTU202402984

onal resources to reach the same goal. Using TDMA as an example and by treating T non-overlapping time frames as T orthogonal channels in Fig. 3.1, assume user j has a sequence $[s_{1j}, s_{2j}, \ldots, s_{Tj}]^T$ where at the $t-$th channel it transmits a training symbol at discrete symbol time index s_{tj} which will arrive at time slot $s_{tj} + \delta_j$. For a CDMA receiver, it will shift the $t-$th pilot in time to perform correlation, and record the correlation at each time shift [19]. Let user j transmits the t-th pilot by pre-shifting it with s_{tj}, the receiver correlation is expected to be large enough at time shift $s_{tj} + \delta_j$. Then equivalently one can also treat that there is a reception on time $s_{tj} + \delta_j$ on $t-$th virtual channel in Fig. 3.1. To reduce the processing complexity, the receiver will record only at the first time it observes a transmitted symbol from tth (virtual) channel, that is $\min_{j\in[K]} s_{tj} + \delta_j$, where $[K] := \{1, \ldots K\}$. To simplify the notation, we define a $K \times 1$ unknown delay vector

$$\boldsymbol{\delta} := [\delta_1, \delta_2, \ldots \delta_K]^\top,$$

the asynchrony-detection matrix $\mathbf{S}$ of which s_{tj} is the $(t, j)-$th element of it, and the $T \times 1$ observation vector from T channels is

$$\boldsymbol{S} \odot \boldsymbol{\delta} := \begin{bmatrix} \min \ell_{in, \tilde{j}\in[K]}^{min} (s_{1\tilde{j}} + \delta_{\tilde{j}}) \\ \vdots \\ \min \ell_{in, \tilde{j}\in[K]}^{min} (s_{T\tilde{j}} + \delta_{\tilde{j}}) \end{bmatrix}. \tag{3.1}$$

In [5], we have discussed that our aim is to design $\boldsymbol{S}$ such that there is a one-to-one correspondence between $\boldsymbol{S} \odot \boldsymbol{\delta}$ and $\boldsymbol{\delta}$. It is important to note that if (3.1) were substituted with matrix multiplication in traditional linear algebra, a one-to-one mapping would not be feasible because a $T \times K$ matrix does not have an inverse if $T < K$. In practical applications, if the receiver can observe the tropical multiplication $\boldsymbol{S} \odot \boldsymbol{\delta}$ using T orthogonal

doi:10.6342/NTU202402984

resources, the one-to-one mapping guarantees zero error probability in identifying active users and their delays $\delta_j \neq \ell_{in}, j \in [K]$ without requiring K orthogonal resources, where $[K] := 1, \ldots, K$. Below is an example demonstrating how $\boldsymbol{S} \odot \boldsymbol{\delta}$ can be obtained in practice. This example is also presented in [5].

Example: We can further employ K overlapping yet orthogonal pilot sequences with perfect auto- and cross-correlations for any starting time [18] to acquire $\boldsymbol{S} \odot \boldsymbol{\delta}$. To explain this concept, we first review the conventional scheme in [18] where $T = K$. Let the t-th set of PN sequences has N non-overlapping equal-length sequence flocks as $\boldsymbol{c}_0^t, \cdots, \boldsymbol{c}_n^t, \cdots, \boldsymbol{c}_{N-1}^t$ [18]. At the nth flock, the receiver receives:

$$\sum_{t=1}^{K} \boldsymbol{c}_n^t(\delta_t) \tag{3.2}$$

where $\boldsymbol{c}_n^t(\delta_t)$ represents $\boldsymbol{c}_n^t$ time-shifted by δ_t. The tth correlator performs correlation using time-shifted $\boldsymbol{c}_n^t(u), u = 0, 1, \ldots$ and records the values of corresponding aperiodic cross-correlation functions over N flocks. If PN sequences form a mutually orthogonal complementary set, from [18, (1)], only when $\delta_t - u = 0$ does the correlator output a non-zero value due to the non-zero autocorrelation of $\boldsymbol{c}_n^t(u)$ and $\boldsymbol{c}_n^t(\delta_t)$. We set the time-shift $u \leq \ell_u$ so that when user t is inactive with $\delta_t = \ell_{in}$, the tth correlator always outputs zero for $t = 1, \ldots, T = K$ and $\boldsymbol{\delta}$ can be determined.

From equation (3.2), we know the received sequence of one flock. However, each MOCS set contains N flocks. We need to summerize over all N flocks. Therefore, the received sequence os then the following.

$$\sum_{n=1}^{N} \sum_{t=1}^{K} \boldsymbol{c}_n^t(\delta_t) \tag{3.3}$$

doi:10.6342/NTU202402984

Using only $T < K$ orthogonal PN sequences, we let user j transmit the t-th pilot sequence pre-shifted by s_{tj} in $\boldsymbol{S}$. The tth correlator output is expected to be non-zero at time shift $u = s_{tj} + \delta_j$, as illustrated in Fig.3.1. Specifically, at the n-th flock, the receiver has:

$$\sum_{t=1}^{T}\sum_{j=1}^{K} \boldsymbol{c}_n^t(s_{tj} + \delta_j), \tag{3.4}$$

At the tth correlator in Fig.3.1 , (3.4) is correlated with time-shifted $\boldsymbol{c}_n^t(u)$. Only when $s_{tj} + \delta_j - u = 0$ does this correlator yield a non-zero value. The receiver observes tropical multiplication $\boldsymbol{S} \odot \boldsymbol{\delta}$ in (3.1) by using T correlators on (3.4) and recording the first non-zero output for each correlator (shown as a black box in Fig.3.1). For (3.4), if the time-shift u of each correlator does not exceed ℓ_{in}, then non-zero correlator output will not occur at $u = s_{tj} + \delta_j$ if $s_{tj} + \delta_j \geq \ell_{in}$. For example, $s_{t3} + \delta_3$ and $s_{t4} + \delta_4$ from inactive users 3 and 4 in Fig.3.1. Although user 2 is active in Fig. 3.1, $s_{22} + \delta_2 > \ell_{in}$ because we set $s_{22} = \ell_{in}$. Mathematically, it can be treated that $\boldsymbol{c}_n^t(s_{tj} + \delta_j)$ will not be included in (3.4) if $s_{tj} + \delta_j \geq \ell_{in}$. By setting $T = K$ and s_{tj} to 0 when $t = j$ and ℓ_{in} otherwise, (3.4) reverts to the conventional scheme (3.2) in [18][2]..

From (3.4), we know the received sequence of one flock. However, each MOCS set has N flocks. We need to summerize all N flocks. The received sequence is then the following.

$$\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{K} \boldsymbol{c}_n^t(s_{tj} + \delta_j), \tag{3.5}$$

If we define the set of user's path delay as $\{0, \ldots, \ell_u, \ell_{in}\}$ and the set of pre-shift as $\{0, \ldots, \ell, \ell_{in}\}$. From (3.3) we know that the total amount of time we need to observe is $(\ell_u + L) * N$. From (3.5) we know that the total amount of time we need to observe is $(\ell + \ell_u + L) * N$. To reach the goal of massive users, we sacrifice the total amount of time we need to observe, which is determined by pre-shift ℓ. In our following work we aim to

doi:10.6342/NTU202402984

minimize ℓ delay for one and two active users in massive system.

Note that indeed δ_j in $\boldsymbol{\delta}$ is the delay of the user which choose the jth column of $\boldsymbol{S}$ as training sequence. Define the support size of delay vector $\boldsymbol{\delta}$ to be the number of users who have a delay value less than ℓ_{in} in the delay vector, which equals to K_a. Here indeed we allow there is less than K_a users to be active. For inactive user j, we propose to set δ_j to be a large finite number ℓ_{in} instead of ∞ in [15]. For example, for the $t-$th channel in Fig. 3.1, the arrival time $s_{t3} + \delta_3$ and $s_{t4} + \delta_4$ for training symbols respectively from inactive users 3 and 4 will mathematically be longer than the observation deadline ℓ_{in} and never be received. In practice, this also captures the fact that these two users never transmit. The definition for a finite maximum delay tropical code is then

Definition 1. A (T, K, K_a)-tropical code is an asynchrony-detection matrix with maximum delay ℓ

$$\mathbf{S} \in \{0, 1, 2, \ldots, \ell, \ell_{in}\}^{T \times K}$$

such that $\mathbf{S} \odot \boldsymbol{\delta}_1 \neq \mathbf{S} \odot \boldsymbol{\delta}_2$ for every distinct

$$\boldsymbol{\delta}_1, \boldsymbol{\delta}_2 \in \{0, 1, 2, ..., \ell_u, \ell_{in}\}^{K \times 1}, \tag{3.6}$$

where $\ell_{in} - \ell > 0$, $\ell_{in} - \ell_u > 0$, and tropical matrix-vector multiplication is modified from [15] as (3.1). Also the support size K_a of delay vector $\boldsymbol{\delta}$ is the number of entries that are smaller than ℓ_{in}.

By this definition, once the receiver gets $\boldsymbol{S} \odot \boldsymbol{\delta}$ in (3.1), it can decode δ without error and the activity and delay of K users are known. Note that from Definition (1), K_a is indeed the maximum number of active users which allows for less than K_a users to be active.

doi:10.6342/NTU202402984

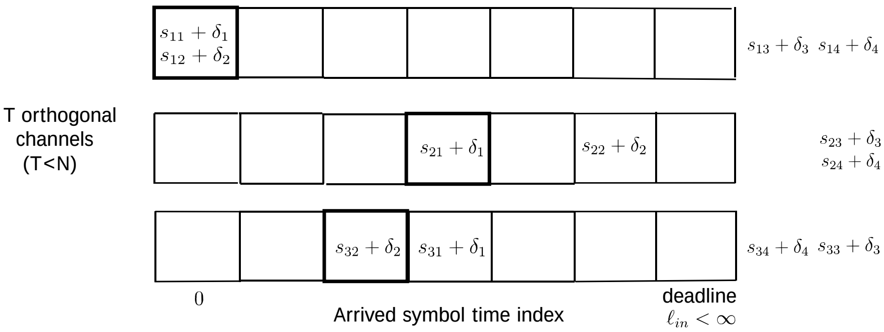

Figure 3.1: Example of using $T = 3$ orthogonal channels during the training to detect activity of $K = 4$ users, where $\mathsf{K}_a = 2$ of them is active. The black box indicates the time index recorded by the receiver from each channel.

3.2 One Active User

3.2.1 2 Orthogonal Resources

Lemma 1 (Converse for $(2, K, 1)$ case)**.** Assume that maximum delay is ℓ, under

$$\ell_{in} - \ell_u > \ell, \tag{3.7}$$

$$\ell_u > \ell, \tag{3.8}$$

it is impossible to find a $(2, K, 1)$ asynchrony-detection matrix $\boldsymbol{S}$ with maximum delay ℓ if $K > 2\ell + 3$

Proof. First, suppose the jth element δ_j is the only element in the delay vector $\boldsymbol{\delta}$ which is smaller than ℓ_{in}, and let $[a\ b]^T := \boldsymbol{S} \odot \boldsymbol{\delta}$. Note that the mapping from $[a\ b]^T$ to $[(a-b)\ b]^T$ is one-to-one and we can instead consider the latter. If neither s_{1j} or s_{2j} equals to ℓ_{in}, then neither a nor b equals to ℓ_{in} and

$$a - b = (s_{1j} + \delta_j) - (s_{2j} + \delta_j) = s_{1j} - s_{2j} \tag{3.9}$$

doi:10.6342/NTU202402984

from (3.1)(3.12). Also when $s_{1j} = \ell_{in} > s_{2j}$, from (3.1)

$$[a\ b]^T = [\ell_{in}\ s_{2j} + \delta_j]^T; \tag{3.10}$$

For this case, $[a\ \ b]^T$ will be different for different $\boldsymbol{\delta}$ if ℓ_{in} appears only once in the first row, otherwise there may exist δ_j and $\delta_{j'}$ such that $s_{2j} + \delta_j = s_{2j'} + \delta_{j'}$. Similarly if $s_{2j} = \ell_{in} > s_{1j}$,

$$[a\ b]^T = [s_{1j} + \delta_j\ \ell_{in}]^T \tag{3.11}$$

and will be different for different $\boldsymbol{\delta}$ if ℓ_{in} appears only once in the second row. Now we leverage (3.9) to prove that this is also true for the first case. Under (3.9), we show that two different $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ could result in the same $[(a-b)\ b]^T$ when $K > 2\ell+3$. Let δ_{j_1} and δ_{j_2} be the elements in the delay vectors $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ which are smaller than ℓ_{in}, respectively. Note that if $j_1 = j_2$, $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ will have different $[(a-b)\ b]^T$. If $j_1 \neq j_2$, size $|s_{1j} - s_{2j}|$ of possible $\{s_{1j} - s_{2j}\}$ is at most $2\ell + 1$ by definition of maximum delay and (3.9), and thus excluding the two cases resulting (3.10)(3.11) we should have $K - 2 \leq 2\ell + 1$. Now if $K - 2 > 2\ell + 1$ and assume $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ has same $a - b$ under $j_1 \neq j_2$, we argue it is impossible to make them have different b values, that is, $s_{2j_1} + \delta_{j_1} \neq s_{2j_2} + \delta_{j_2}$. The reason is $s_{2j} \neq \ell_{in}$, $|s_{2j_1} - s_{2j_2}| = 2\ell + 1 < |\delta_{j_2} - \delta_{j_1}| = 2\ell_u + 1$. Such code exists only if $K \leq 2\ell + 3$ under (3.13). □

Lemma 2 (Achievability for $(2, K, 1)$ case)**.** Assume that maximum delay is ℓ, under

$$\ell_{in} - \ell_u > \ell, \tag{3.12}$$

$$\ell_u > \ell, \tag{3.13}$$

and the following limitations on the synchronization matrix **S**

doi:10.6342/NTU202402984

1. ℓ_{in} appears only once in each row

2. no column is $[\ell_{in}, \ell_{in}]^T$,

if **S** satisfies

$$|\{s_{1j} - s_{2j}|\, j = 1, ..., K\}| = K \tag{3.14}$$

then $\boldsymbol{S}$ is a $(2, K, 1)$-tropical code within maximum delay ℓ when $K \leq 2\ell + 3$.

Proof. First, we show that if ℓ_{in} appears more than once in each row, $\boldsymbol{S}$ can not ensure one-to-one mapping after tropical multiplication. Without loss of generality, assume the second row of $\boldsymbol{S}$ has two ℓ_{in}s and the first two columns are $[s_{11}, \ell_{in}]^T$ and $[s_{12}, \ell_{in}]^T$ respectively. Also suppose the jth element δ_j is the only element in the delay vector $\boldsymbol{\delta}_j$ which is smaller than $\ell_{in}, j = 1, 2$. Then if $\delta_1 - \delta_2 = s_{12} - s_{11}$ then $\mathbf{S} \odot \boldsymbol{\delta}_1 = \mathbf{S} \odot \boldsymbol{\delta}_2$ after tropical multiplication, where $s_{11} + \delta_1 \leq \ell_{in}$ from (3.12). Next, if a column is $[\ell_{in}, \ell_{in}]^T$, basically this column is useless and can be removed. For the achievability, we first consider the situation where $K = 2\ell + 3$ and without loss of generality synchronization matrix meet (3.14) can be assumed as

$$\boldsymbol{S} = \begin{bmatrix} s_{11} & 0...0 & 0 & 1...\ell & \ell_{in} \\ \ell_{in} & \ell...1 & 0 & 0...0 & s_{2K} \end{bmatrix}. \tag{3.15}$$

where neither s_{11} or s_{2K} equals to ℓ_{in}, since all the other possible matrices meeting (3.14) will just be the column permutations of this $\boldsymbol{S}$. Again we let $[a\ b]^T := \mathbf{S} \odot \boldsymbol{\delta}$ and show that each distinct $\boldsymbol{\delta}$ will result in distinct $[a\ b]^T$. Specifically, if the support size of $\boldsymbol{\delta}$ is 0,

$$a = b = \ell_{in}, \tag{3.16}$$

doi:10.6342/NTU202402984

while if the support size of $\boldsymbol{\delta}$ is exactly 1, that is,

$$\boldsymbol{\delta} = [\ell_{in}, \ell_{in}...\ell_{in}, \delta_j, \ell_{in}...\ell_{in}, \ell_{in}]^T \tag{3.17}$$

where $\delta_j \in \{0, 1, 2, ..., \ell_u\}, 1 \leq j \leq K$. Under (3.12), we will prove the corresponding $a - b = s_{1j} - s_{2j}$ has K distinct values and a has $\ell_u + 1$ distinct values. Thus all possible $K(\ell_u + 1)$ distinct $\boldsymbol{\delta}$ are covered. b

To do this, we consider the following four cases.

Case 1: To prove (3.16) where $\boldsymbol{\delta}$'s entries are all ℓ_{in}. From (3.15)

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \min(\ell_{in}, \ell_{in} + s_{11}, \ell_{in} + 1, \ell_{in} + 2, ..., \ell_{in} + \ell, 2\ell_{in}) \\ \min(\ell_{in}, \ell_{in} + s_{2N}, \ell_{in} + 1, \ell_{in} + 2, ..., \ell_{in} + \ell, 2\ell_{in}) \end{bmatrix} = \begin{bmatrix} \ell_{in} \\ \ell_{in} \end{bmatrix}.$$

For (3.17), we have the following three cases.

Case 2: Assume $1 < j < K$ such that δ_j does not appear at the beginning and end of $\boldsymbol{\delta}$, there will be $(K-2)(\ell_u + 1)$ possible $\boldsymbol{\delta}$s. Then under (3.12), which ensures $\delta_j + \ell < \ell_{in}$

$$b = \begin{cases} \delta_j - (j - (\ell + 2)) & 2 \leq j \leq \ell + 2, \\ \delta_j & \ell + 3 \leq j \leq 2\ell + 2 \end{cases}$$

while all possible values of $a - b$ are $-\ell, -(\ell-1), ..., -1, 0, 1, ..., (\ell-1), \ell$ which is equal to the value of $s_{1j} - s_{2j}$ where $2 \leq j \leq K - 1$ from (3.9) except for the first and last column of **S**. The number of different values $a - b$ can have is $K - 2$, thus, the number of distinct $[a\ b]^T$ is $(K-2)(\ell_u + 1)$.

Case 3: If $\boldsymbol{\delta} = [\delta_1, \ell_{in}, ..., \ell_{in}, \ell_{in}, \ell_{in}]^T$ and $\delta_1 < \ell_{in}$, from (3.12)

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \min(\ell_{in}, s_{11} + \delta_1, \ell_{in}, \ell_{in}, \ell_{in} + 2, ..., \ell_{in} + \ell, 2\ell_{in}) \\ \min(\ell_{in}, \delta_1 + \ell_{in}, \ell_{in} + \ell, ..., \ell_{in} + 2, \ell_{in}, \ell_{in}, \ell_{in} + s_{2K}) \end{bmatrix} = \begin{bmatrix} s_{11}+\delta_1 \\ \ell_{in} \end{bmatrix}$$

doi:10.6342/NTU202402984

The total number of possible values for $\mathbf{S} \odot \boldsymbol{\delta}$ is $\ell_u + 1$.

Case 4: If $\boldsymbol{\delta} = [\ell_{in}, ..., \ell_{in}, \ell_{in}, \ell_{in}, \delta_K]^T$ and $\delta_K < \ell_{in}$, as in Case 3

$$\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] = \left[\begin{smallmatrix} \ell_{in} \\ s_{2K}+\delta_K \end{smallmatrix}\right]$$

from (3.12). Similarly, the total number of possible values for $\mathbf{S} \odot \boldsymbol{\delta}$ is $\ell_u + 1$.

Note that (3.15) meets (3.14) and indeed covers all possible $\mathbf{S}$ meeting (3.14) if $K = 2\ell + 3$. The proof can be simply applied to $K < 2\ell + 3$ by deleting columns in (3.15), since for delay vectors meeting (3.17) none of columns will have same results after tropical multiplication (3.1). From the achievability in (3.15), one can form a decoder which maps $\boldsymbol{S} \odot \boldsymbol{\delta}$ to an unique $\boldsymbol{\delta}$. This decoder is detailed in Appendix 5. □

Now we extend to the $(T, K, 1)$ case as follows.

3.2.2 T Orthogonal Resources

Lemma 3. Assume that maximum delay is ℓ, under

$$\ell_{in} - \ell_u > \ell, \tag{3.18}$$

$$\ell_u > \ell, \tag{3.19}$$

then a $(T, K, 1)$-tropical code $\mathbf{S}$ within maximum delay ℓ exists if and only if $K \leq (\ell + 2)^T - (\ell + 1)^T$.

Proof. We first consider the converse, which means that it is impossible for a synchronization matrix $\mathbf{S}$ which does not fulfill the constraints (3.18), or (3.19) , or has a column size $K > (\ell + 2)^T - (\ell + 1)^T$ to have a one-to-one correspondence between $\boldsymbol{\delta}$ and $\boldsymbol{S} \odot \boldsymbol{\delta}$.

doi:10.6342/NTU202402984

First, denote the set of column vectors that have at least one entry equal to zero by:

$$\mathbb{O} := \{0, 1, 2, \ldots, \ell, \ell_{in}\}^{T\times 1} \setminus \{1, 2, \ldots, \ell, \ell_{in}\}^{T\times 1}.$$

Initially, there might be some columns of $\boldsymbol{S}$ consisting of T all non-zero entries and, thus, not contained in $\mathbb{O}$. Without loss of generality, we may assume the first column $\mathbf{s}_1$ is one of those,

$$\mathbf{s}_1 \in \{1, 2, \ldots, \ell, \ell_{in}\}^{T\times 1},$$

and also the first element $0 \leq \delta_1 \leq \ell_u$ is the only element in a delay vector $\boldsymbol{\delta}$ which is smaller than ℓ_{in}. Next, define another column vector $\tilde{\mathbf{s}}_1$, of which the $k-$th entry being

$$s_{1k} - \min\{\mathbf{s}_1\}, \text{ if } s_{1k} \neq \ell_{in}, \tag{3.20}$$

and ℓ_{in} else. We know that $\tilde{\mathbf{s}}_1$ has at least one zero entry and no negative entries, and $\tilde{\mathbf{s}}_1 \in \mathbb{O}$. By definition,

$$\mathbf{s}_1 \odot \delta_1 = \tilde{\mathbf{s}}_1 \odot (\delta_1 + \min\{\mathbf{s}_1\}). \tag{3.21}$$

The above equation shows explicitly the way to shift the columns of $\boldsymbol{S}$ into the set $\mathbb{O}$ without changing the detection result. Therefore, the only way for another delay vector $\boldsymbol{\delta}_2$ which has only one entry, say, the second one, smaller than ℓ_{in}, to result in the same column vector as $\tilde{\mathbf{s}}_1 \odot (\delta_1 + \min\{\mathbf{s}_1\})$ is that $\tilde{\mathbf{s}}_1 \equiv \tilde{\mathbf{s}}_2$. However, this possibility has already been excluded by our selecting criterion of the columns of the code matrix. Note that if for a column vector $\mathbf{s}_1$ we have $\min\{\mathbf{s}_1\} = \ell_{in}$, then $\mathbf{s}_1$ consists solely of ℓ_{in} and should not be selected as a column of $\boldsymbol{S}$ for then no matter what the actual delay of the corresponding user has, it will be decimated after the tropical multiplication. Finally, if $K > (\ell + 2)^T - (\ell + 1)^T$ there must be at least two columns that have the same

doi:10.6342/NTU202402984

representative, and, thus can render the same output from two distinct vectors with support size 1. If the constraint $\ell_{in} > \ell + \ell_u$ in (3.18) is violated, there will exist two distinct delay vectors which induce the same results after tropical multiplication. For, two columns that differ at only one entry, where one of the columns has entry ℓ_{in} while the other has ℓ will then be mapped to the same vector by two different support 1 delay vectors when the delay values (at different corresponding positions) both equal ℓ_u. Therefore, One-to-one correspondence between delay vectors and results after tropical multiplication (3.1) can not be guaranteed by the matrix $\boldsymbol{S}$.

For the part of achievability, under the assumptions (3.18)(3.19) and $K \leq (\ell+2)^T - (\ell+1)^T$, the synchronization matrix $\boldsymbol{S}$ should further satisfy the following conditions:

1. a column must contain at least one 0 if it contains less than $T-1$ ℓ_{in}'s

2. no column can consist of all ℓ_{in}'s

Here we form an example of a $T \times (K = (\ell+2)^T - (\ell+1)^T)$) synchronization matrix as

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \end{bmatrix}, \tag{3.22}$$

where each column of $\boldsymbol{S}$ contains at least one-zero and matrices $\boldsymbol{B}, \boldsymbol{C}$ are systematically constructed as follows. First, each column of $\boldsymbol{B}$ contains $1 \leq N_{\ell in} \leq T-1$ ℓ_{in}s and that of $\boldsymbol{C}$ contains no ℓ_{in}. Matrix $\boldsymbol{B}$ is a $T \times ((\ell+2)^T + \ell^T - 2(\ell+1)^T)$ matrix which can be represented by

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{B}_{T-1} & \mathbf{B}_{T-2} & \cdots & \mathbf{B}_2 & \mathbf{B}_1 \end{bmatrix}. \tag{3.23}$$

where each column of $\mathbf{B}_{N_{\ell in}}$ has $N_{\ell in}$ elements being ℓ_{in}, $1 \leq N_{\ell in} \leq T-1$. For instance,

doi:10.6342/NTU202402984

$\mathbf{B}_{T-1}$ is a $T \times T$ matrix which can be represented as

$$\mathbf{B}_{T-1} = \begin{bmatrix} 0 & \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & \ell_{in} \\ \ell_{in} & 0 & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & \ell_{in} \\ \ell_{in} & \ell_{in} & 0 & \cdots & \ell_{in} & \ell_{in} & \ell_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ell_{in} & \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & 0 \end{bmatrix}. \tag{3.24}$$

From (3.1), $\mathbf{B}_{T-1} \odot \boldsymbol{\delta}$ has the physical meaning for the first T users in $\boldsymbol{\delta}$, that is, one adopt the traditional way to use T orthogonal channels to detect unknown activities of these T users. The construction of the rest matrices in (3.23) needs more attention. For example, $\mathbf{B}_{T-2}$ should be composed of columns containing $T-2$ ℓ_{in}s as

$$\begin{bmatrix} 0 & 0 & \cdots & \ell-1 & \ell & 0 & 0 & \cdots & \ell-1 & \ell \\ \cdots & \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & & & & \\ \ell & \ell-1 & \cdots & 0 & 0 & \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} \\ \cdots & \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & & & & \\ \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & \ell & \ell-1 & \cdots & 0 & 0 \\ \cdots & \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \cdots & \vdots & \vdots & \ddots & \vdots & \vdots & & & & \\ \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} \\ \cdots & 0 & 0 & \cdots & \ell-1 & \ell & & & & \\ \ell_{in} & \ell_{in} & \cdots & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} \\ \cdots & \ell & \ell-1 & \cdots & 0 & 0 & & & & \end{bmatrix}. \tag{3.25}$$

First, we form columns of which the last $N_{\ell in} = T-2$ elements are ℓ_{in}. By setting an

doi:10.6342/NTU202402984

element among the first $T - N_{\ell in} = 2$ locations to be zero, one then use a $(T - N_{\ell in} - 1)$-element variations from $\ell + 1$-element set $\{0, \ldots, \ell\}$, with repetition allowed, to form the rest $(T - N_{\ell in} - 1)$ elements. Totally we form $(T - N_{\ell in})(\ell + 1)^{(T - N_{\ell in} - 1)} = 2(\ell + 1)$ of these columns. After creating columns of which the last $N_{\ell in} = T - 2$ elements are ℓ_{in}, all T elements of these columns are further permutated to form rest columns in (3.25). By this construction, $\boldsymbol{B}$ has $\sum_{k=1}^{T-1} \binom{T}{T-k}((\ell+1)^{T-k} - \ell^{T-k}) = (\ell+2)^T + \ell^T - 2(\ell+1)^T$ columns. Matrix $\boldsymbol{C}$ has dimension $T \times ((\ell+1)^T - \ell^T)$ and can be constructed similar to $\boldsymbol{B}$ as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_{T-1} & \mathbf{C}_T \end{bmatrix}, \tag{3.26}$$

but now each column of $\mathbf{C}_{N_0}$ has N_0 elements being $0, 1 \leq N_\ell \leq T$. For instance, $\mathbf{C}_{T-1}$ can be shown as

$$\mathbf{C}_{T-1} = \begin{bmatrix} \ell & \ell-1 & \cdots & 2 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \cdots & 0 & 0 & \cdots & 0 & 0 & & & & \\ 0 & 0 & \cdots & 0 & 0 & \ell & \ell-1 & \cdots & 2 & 1 \\ \cdots & 0 & 0 & \cdots & 0 & 0 & & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & & & & \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \cdots & 0 & 0 & \cdots & 0 & 0 & & & & \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \cdots & \ell & \ell-1 & \cdots & 2 & 1 & & & & \end{bmatrix}. \tag{3.27}$$

Note that since each column of matrix $\boldsymbol{C}$ are chosen from $\{0, \ldots, \ell\}$ but must contain at least one zero, there are $(\ell+1)^T - \ell^T$ columns.

doi:10.6342/NTU202402984

Finally, assume the jth element δ_j of $\boldsymbol{\delta}$ does not equal ℓ_{in}, then for the above constructed $\boldsymbol{S}$

$$\boldsymbol{S} \odot \delta = \mathbf{s}_j \odot \delta_j$$

where $\mathbf{s}_j$ is the jth of column of $\boldsymbol{S}$ in (3.22), $1 \leq j \leq N$. However, unlike the case in (3.21), under the assumptions (3.18) and (3.19) , it is impossible to find

$$\mathbf{s}_j \odot \delta_j = \mathbf{s}_{j'} \odot \delta_{j'}, j' \neq j$$

even if $\delta_{j'} = \delta_j$. Therefore $\boldsymbol{S}$ must be one-to-one for tropical multiplication. □

This chapter we demonstrate how to create asynchrony-detection matrix of one active user. In next chapter we will introduce asynchrony-detection matrix that is able to detect two active users.

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Chapter 4   Two Active Users

4.1 Basic Building Block of Two Active Users

In the following, we discuss the $(T, K, 2)$ tropical code. With two active users, the design is much more complicated and we start from an toy $(4, 4, 2)$ example with asynchrony-detection matrix $\boldsymbol{S}$

$$\mathbf{S} = \begin{bmatrix} s_{11} & 0 & \ell_{in} & \ell_{in} \\ \ell_{in} & s_{22} & 0 & \ell_{in} \\ \ell_{in} & \ell_{in} & s_{33} & 0 \\ 0 & \ell_{in} & \ell_{in} & s_{44} \end{bmatrix}. \tag{4.1}$$

Though with $T = K = 4$ it is trivial to design a $(4, 4, 4)$ code with more active users, the $(4, 4, 2)$ code (4.1) is a basic building block for a more general $(T, \lfloor \frac{T}{2} \rfloor \lceil \frac{T}{2} \rceil, 2)$ code where K scale quadratically with T.

We leverage MOCS's sets in [14] to verify that (4.1) has the correlation as expected. Each set has 4 flocks, 16 flocks in total. MOCS sets given in below.

$$\begin{aligned}
C^{t1} &= \{\boldsymbol{c}_1^{t1} = [1\ 1\ 1\ 1], & \boldsymbol{c}_2^{t1} &= [-1\ -1\ 1\ 1], & \boldsymbol{c}_3^{t1} &= [-1\ 1\ -1\ 1], & \boldsymbol{c}_4^{t1} &= [1\ -1\ -1\ 1]\}; \\
C^{t2} &= \{\boldsymbol{c}_1^{t2} = [1\ 1\ -1\ -1], & \boldsymbol{c}_2^{t2} &= [-1 -1 -1 -1], & \boldsymbol{c}_3^{t2} &= [-1\ 1\ 1\ -1], & \boldsymbol{c}_4^{t2} &= [1\ -1\ 1\ -1]\}; \\
C^{t3} &= \{\boldsymbol{c}_1^{t3} = [-1\ 1\ -1\ 1], & \boldsymbol{c}_2^{t3} &= [1\ -1\ -1\ 1], & \boldsymbol{c}_3^{t3} &= [1\ 1\ 1\ 1], & \boldsymbol{c}_4^{t3} &= [-1\ -1\ 1\ 1]\}; \\
C^{t4} &= \{\boldsymbol{c}_1^{t4} = [-1\ 1\ 1\ -1], & \boldsymbol{c}_2^{t4} &= [1\ -1\ 1\ -1], & \boldsymbol{c}_3^{t4} &= [1\ 1\ -1\ -1], & \boldsymbol{c}_4^{t4} &= [-1 -1 -1 -1]\}
\end{aligned} \tag{4.2}$$

doi:10.6342/NTU202402984

Including $\{C^{t1}, C^{t2}, C^{t3}, C^{t4}\}$.

The following we give an example to demonstrate the correlation of $(4,4,2)$-tropical code, with $\ell = 2$, $\ell_u = 3$, $\ell_{in} = 8$.

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 8 & 8 \\ 8 & 2 & 0 & 8 \\ 8 & 8 & 1 & 0 \\ 0 & 8 & 8 & 1 \end{bmatrix}. \tag{4.3}$$

$$\delta^{\mathbf{T}} = \begin{bmatrix} 3 & 8 & 3 & 8 \end{bmatrix}. \tag{4.4}$$

$$\mathbf{S} \odot \delta = \begin{bmatrix} 4 \\ 3 \\ 4 \\ 3 \end{bmatrix}. \tag{4.5}$$

The receiver receives the following sequence.

$$\sum_{n=1}^{N=4} \sum_{t=1}^{T=4} \sum_{j=1}^{K=4} \boldsymbol{c}_n^t (s_{tj} + \delta_j), \tag{4.6}$$

We then correlate (4.6) with $\{C^{t1}, C^{t2}, C^{t3}, C^{t4}\}$. The corresponding output of four sets are $4, 3, 4, 3$, which is equivalent to the output of (4.5).

Lemma 4. The asynchrony-detection matrix in (4.1) is a (4,4,2)-tropical codes with maximum delay ℓ equals to max$\{s_{11}, s_{22}, s_{33}, s_{44}\}$, under

$$\ell_{in} - \ell_u > \ell, \tag{4.7}$$

$$\ell_{in} > \max\{s_{11}, s_{22}, s_{33}, s_{44}\} \neq 0. \tag{4.8}$$

doi:10.6342/NTU202402984

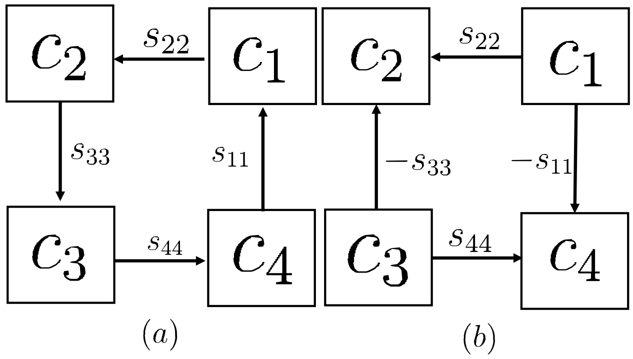

Figure 4.1: Bipartite graph from adjacency matrix (4.9) for $(4, 4, 2)$ (4.1), where (a) allows positive edge weights and (b) allows negative ones.

Proof. By definition, the maximum delay equals that in the lemma statement. If no users are active, $\boldsymbol{S} \odot \boldsymbol{\delta}$ will result in a vector $\ell_{in}\mathbf{1}$. Furthermore if (4.7) is met $\boldsymbol{\delta} = \ell_{in}\mathbf{1}$ is the only $\boldsymbol{\delta}$ which results in $\boldsymbol{S}\odot\boldsymbol{\delta} = \ell_{in}\mathbf{1}$. When there is one active user in $\boldsymbol{\delta}$, only two elements in $\boldsymbol{S} \odot \boldsymbol{\delta}$ will be ℓ_{in} when (4.7) holds and the synchronization matrix is selected as (4.1). Moreover all four possible $\boldsymbol{S} \odot \boldsymbol{\delta}$s will be distinct, and the smallest element of $\boldsymbol{S} \odot \boldsymbol{\delta}$ is the delay of the corresponding active user. With two active users in $\boldsymbol{\delta}$ there are two different cases according whether two user indexes are modulo-4 adjacent or not. With adjacent user indexes, when (4.7) is true, only one element in $\boldsymbol{S} \odot \boldsymbol{\delta}$ will be ℓ_{in} while the other three will not. The output $\boldsymbol{S} \odot \boldsymbol{\delta}$ after tropical multiplication will be one-to-one, by checking the $\binom{4}{3}$ output for four possible $\boldsymbol{\delta}$s. When active user indexes are not adjacent, none of the elements of $\boldsymbol{S}\odot\boldsymbol{\delta}$ will be ℓ_{in}. Let $c_i, i = 1 \ldots 4$ be the ith element of tropical multiplication $\boldsymbol{S} \odot \boldsymbol{\delta}$, that is, the four first received packets among each channel. If the first and third users are active $\boldsymbol{\delta} = [\delta_1\ \ell_{in}\ \delta_3\ \ell_{in}]^T$ then $c_1-c_2+c_3-c_4 = (s_{11}+\delta_1)-\delta_3+(s_{33}+\delta_3)-\delta_1 = s_{11} + s_{33}$ under (4.7). Finally, if the second and fourth users are active $\boldsymbol{\delta} = [\ell_{in}\ \delta_2\ \ell_{in}\ \delta_4]^T$, then $c_1 - c_2 + c_3 - c_4 = \delta_2 - (s_{22} + \delta_2) + \delta_4 - (s_{44} + \delta_4) = -s_{22} - s_{44}$. Thus to achieve one-to-one correspondence $s_{11}+s_{33} \neq -s_{22}-s_{44}$, so (4.8) must be true since all elements in $\boldsymbol{S}$ are non-negative. $\square$

doi:10.6342/NTU202402984

$$
\begin{array}{cc} & \begin{array}{cc} c_2 & c_4 \end{array} \\ \begin{array}{c} c_1 \\ c_3 \end{array} & B = \begin{bmatrix} s_{22} & -s_{11} \\ -s_{33} & s_{44} \end{bmatrix} \end{array}
$$

Figure 4.2: Correspondence of the channel indexes after tropical multiplication to row and column indexes of bi-adjacency matrix (4.9).

For generalizing Lemma 4, note that in (4.1) each user can only participate two channels after tropical multiplication. In other words, only two elements in each column of (4.1) are not ℓ_{in}. By collecting such elements, (4.1) can also be represented by a 2×2 matrix

$$
\boldsymbol{B} = \begin{bmatrix} s_{22} & -s_{11} \\ -s_{33} & s_{44} \end{bmatrix} \tag{4.9}
$$

where the row are indexed according to channel indexes 1 and 3 while columns are indexed according to channel indexes 2 and 4. For example, the $(1, 2)$ element of $\boldsymbol{B}$ is the difference $c_4 - c_1 = s_{41} - s_{11} = 0 - s_{11}$ of the fourth and first elements of the first column of (4.1), which are not ℓ_{in}. Similarly, the $(2, 2)$ element of $\boldsymbol{B}$ is the difference $c_4 - c_3 = s_{44} - s_{34} = s_{44} - 0$ of the fourth and third elements of the fourth column of (4.1). Indeed $\boldsymbol{B}$ is the adjacency matrix of the bipartite graph in Figure 4.1(b) where the elements of tropical multiplication are grouped as two vertex parties $\{c_1, c_3\}$ and $\{c_2, c_4\}$. In the Figure 4.1(b), directed edges represent elements in $\boldsymbol{B}$ with the starting and ending vertices are respectively indexed by transforming the row and column indexes to corresponding channel indexes as in Figure 4.2. For example, the $(1, 2)$ element $-s_{11}$ of $\boldsymbol{B}$ corresponding to starting vertex as channel 1 and ending vertex as channel 4; while $-s_{11}$ is the difference of ending and starting channel values $c_4 - c_1$ where $c_1 = s_{11} + \delta_1$, $c_4 = \delta_1$

doi:10.6342/NTU202402984

are obtained from the tropical multiplication of (4.1) under (4.7) when only user 1 is active and has delay δ_1.

With the formulation of $\boldsymbol{B}$ in (4.9), we know that in order to distinguish the non-adjacent two active users pairs' case. We need $s_{22} - (-s_{11}) + s_{44} - (-s_{33}) \neq 0$, which is equivalent to constraint (4.8). We define $(s_{22}) - (-s_{11}) + (s_{44}) - (-s_{33})$ as cycle sum. For any 2×2 $\boldsymbol{B}$ matrix, $B_{11} - B_{12} + B_{22} - B_{21}$ is called cycle cum.

Note that Figure 4.2 also helps to transform the bi-adjacency matrix into a synchronization matrix. Since elements in the synchronization matrix are all non-negative, first we transform figure 4.1(b) into figure 4.1(a) where the edge weights are all positive, by reversing the edges and associated weights when the weights are negative. A column in the synchronization matrix will have one element indexed by the ending channel index being the edge weight and the other indexed by the starting channel index being zero. For example, for the edge from channel 4 to 1 in figure 4.1(a), the first and fourth element of the resulting column are s_{11} and 0 respectively.

From the above $(4, 4, 2)-$code example, we know that there is a one-to-one correspondence between $\boldsymbol{S}$ of (4.1) and $\boldsymbol{B}$ of (4.9). And thus, on the other hand, we know that we can obtain $\boldsymbol{S}$ from $\boldsymbol{B}$. That is, column j of $\boldsymbol{S}$ in (4.1) is formed by the $(i^{'}, j^{'})$-element of (4.9) by allocating the two non-ℓ_{in} elements on the corresponding $i^{'}$ and $j^{'}$ channel and fill in the remaining channels with ℓ_{in}, where $i' = 1 \ldots Nrow(\boldsymbol{B})$, $j' = 1 \ldots Ncol(\boldsymbol{B})$, and $j = 1 \ldots Nrow(\boldsymbol{B}) * Ncol(\boldsymbol{B})$. For example, the $(1, 2)-$element of $\boldsymbol{B}$ in (4.9) is $-s_{11}$. By the channel index given in (4.9) and $s_{11} > 0$, we know that for there exist a column in $\boldsymbol{S}$ that $c_1 = s_{11}$ and $c_4 = 0$. And we fill in the remaining channel index with ℓ_{in} and obtain a column $[s_{11}\ \ell_{in}\ \ell_{in}\ 0]^T$ in $\boldsymbol{S}$. As for the relationship between the index $(i^{'}, j^{'})$ of $\boldsymbol{B}$ and column index j of $\boldsymbol{S}$, we have the following transformation.

doi:10.6342/NTU202402984

$$j = (i' - 1) \times \text{Ncol}(B) + (\text{Ncol}(B) + 1 - j') \qquad \text{if } i' \text{ is odd} \tag{4.10a}$$

$$j = (i' - 1) \times \text{Ncol}(B) + j' \qquad \text{if } i' \text{ is even} \tag{4.10b}$$

When $(i', j') = (1, 2)$, i' is odd, so from the above formulation we know that the corresponding column index j of $\boldsymbol{S}$ is $(1 - 1) * (2) + (2 + 1 - 2) = 1$. Therefore, we know that the first column of $\boldsymbol{S}$ will be $[s_{11}\ \ell_{in}\ \ell_{in}\ 0]^T$, since $j = 1$. Another example is that when $(i', j') = (2, 1)$, $'i'$ is even, so we use (4.10b) to calculate j, and will get column index $j = 3$ for $\boldsymbol{S}$. By the above formulation we can know the connection of indecies between $\boldsymbol{B}$ and $\boldsymbol{S}$. And thus, from figure 4.2 and (4.10a),(4.10b) we can have the final general definition of $\boldsymbol{B}$ in (4.11), for a $(4, 4, 2)$-tropical code.

$$\boldsymbol{B} = \begin{bmatrix} s_{22} - s_{12} & s_{41} - s_{11} \\ s_{23} - s_{33} & s_{44} - s_{34} \end{bmatrix}. \tag{4.11}$$

Therefore, given $\boldsymbol{B}$ of (4.11) , we can transform $\boldsymbol{B}$ to $\boldsymbol{S}$. On the other hand, when given a synchronization matrix $\boldsymbol{S}$ we can represent it by a $\boldsymbol{B}$ matrix as well as a bipartite graph.

What's more, we know that the number of combinations of (i', j') in $\boldsymbol{B}$ is equal to the number of columns j in $\boldsymbol{S}$. Therefore, we gain the conclusion that a $\lceil T/2 \rceil * \lfloor T/2 \rfloor$ $\boldsymbol{B}$ matrix can map to a $T * (\lceil T/2 \rceil * \lfloor T/2 \rfloor)$ $\boldsymbol{S}$ matrix.

For a more general $(T, K, 2)$ case, we can obtain a tropical code from a given $\boldsymbol{B}_s$, where the larger $K - T$ the better. For the purpose of gaining a larger K from a given T and still map $\boldsymbol{B}_s$ and $\boldsymbol{S}$ to a bipartite graph we leverage Mantel's theorem. That is, a bipartite graph with T vertices have the most edges is when the vertices are separated into

doi:10.6342/NTU202402984

two near number parts, $\lfloor T/2 \rfloor$ and $\lceil T/2 \rceil$ vertices in each part, indicating two parts of channels.

Before discussing the general $(T, K, 2)$ case, we first define what is a 2x2 sub-matrix of $\boldsymbol{B}_s$. The intersection of any two rows and any two columns of $\boldsymbol{B}_s$ will have four elements, and these four elements will form a 2x2 sub-matrix of $\boldsymbol{B}_s$.

4.2 (T,K,2)-Tropical Code

Lemma 5. Let $T \geq 5$. Let $p \geq \frac{T}{2}$ be an odd prime, and under

$$\ell_{in} - \ell_u > \ell, \tag{4.12}$$

, and any 2x2 sub-matrix of the following $\boldsymbol{B}$ satisfies cycle sum $\neq 0$, then there exist a bipartite graph-based $(T, \lceil \frac{T}{2} \rceil \lfloor \frac{T}{2} \rfloor, 2)$-tropical code within maximum delay $\frac{p-1}{2}$. That is, given T, any $\lceil \frac{T}{2} \rceil$x$\lfloor \frac{T}{2} \rfloor$ biadjacency matrix $\boldsymbol{B}_s$ can be one-to-one mapped to a Tx$(\lceil \frac{T}{2} \rceil \lfloor \frac{T}{2} \rfloor)$ synchronization matrix $\boldsymbol{S}$. When $p = \frac{T}{2}$, we take the following $\boldsymbol{B}$ of (4.13) as our biadjacency matrix $\boldsymbol{B}_s$. When $p > \frac{T}{2}$, we delete the last $p - \lceil \frac{T}{2} \rceil$ rows and last $p - \lfloor \frac{T}{2} \rfloor$ columns of $\boldsymbol{B}$ and form $\boldsymbol{B}_s$. The formulation of the following $\boldsymbol{B}$ is adopted from [15].

$$\boldsymbol{B} = \begin{bmatrix} [1 \cdot 1]_p & [1 \cdot 2]_p & \dots & [1 \cdot p]_p \\ [2 \cdot 1]_p & [2 \cdot 2]_p & \dots & [2 \cdot p]_p \\ \vdots & \vdots & \ddots & \vdots \\ [p \cdot 1]_p & [p \cdot 2]_p & \dots & [p \cdot p]_p \end{bmatrix}. \tag{4.13}$$

doi:10.6342/NTU202402984

$[B_{ij}]_p$ can be defined as the following.

$$[B_{ij}]_p = \begin{cases} B_{ij}\ mod\ p & , B_{ij}\ mod\ p \leq \frac{p-1}{2} \quad (4.14a) \\ (B_{ij}\ mod\ p) - p & , B_{ij}\ mod\ p > \frac{p-1}{2} \quad (4.14b) \end{cases}$$

and $B_{ij}\ mod\ p$ is defined as

$$B_{ij}\ mod\ p = B_{ij} - p \cdot \lfloor \frac{B_{ij}}{p} \rfloor \tag{4.15}$$

From (4.10a),(4.10b) and figure 4.3, we know there is a general representation of $\boldsymbol{B}$. And thus, we have the following $\boldsymbol{B}$. (4.10a) and (4.10b) guides the transformation of element index of $\boldsymbol{B}$: $B_{i'j'}$ to column index of $\boldsymbol{S}$: S_j. Once the column index of $\boldsymbol{S}$ is known, we use figure 4.3 to know the two non-ℓ_{in} elements' location and value for every columnn.

$$\boldsymbol{B} = \begin{bmatrix} s_{2p} - s_{1p} & s_{4,p-1} - s_{1,p-1} & \cdots & s_{2p,1} - s_{11} \\ s_{2,p+1} - s_{3,p+1} & s_{4,p+2} - s_{3,p+2} & \cdots & s_{2p,2p} - s_{3,2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{2,2(p-2)+1} - s_{p-1,2(p-2)+1} & s_{4,2p-2} - s_{p-1,2p-2} & \cdots & s_{2p,3p-4} - s_{p-1,3p-4} \\ s_{2,p^2} - s_{p,p^2} & s_{4,p^2-1} - s_{p,p^2-1} & \cdots & s_{2p,p^2-(p-1)} - s_{p,p^2-(p-1)} \end{bmatrix} \tag{4.16}$$

Proof. We first proof that the above lemma holds when $p = \frac{T}{2}$. When $p = \frac{T}{2}$, we take $\boldsymbol{B}$ of (4.16) as our biadjacency matrix $\boldsymbol{B}_s$. Notice that any 2x2 sub-matrix $\boldsymbol{B}_{22}$ in (4.13) has the following structure

$$\boldsymbol{B}_{22} = \begin{bmatrix} tb & te \\ ub & ue \end{bmatrix} \tag{4.17}$$

t and u represent the row factor, while b and e represent the column factor. Every $\boldsymbol{B}_{22}$

doi:10.6342/NTU202402984

$$
\begin{array}{cc}
 & \begin{array}{cccc} c_2 & c_4 & & c_{2p} \end{array} \\
B = \begin{array}{c} c_1 \\ c_3 \\ \\ c_{2p-1} \end{array} & \begin{bmatrix} [1\cdot 1]_p & [1\cdot 2]_p & \dots & [1\cdot p]_p \\ [2\cdot 1]_p & [2\cdot 2]_p & \dots & [2\cdot p]_p \\ \vdots & \vdots & \ddots & \vdots \\ [p\cdot 1]_p & [p\cdot 2]_p & \dots & [p\cdot p]_p \end{bmatrix}
\end{array}
$$

Figure 4.3: Correspondence of the channel indexes after tropical multiplication to row and column indexes of bi-adjacency matrix (4.13).

satisfies $[tb - te + ue - ub]_3 \equiv [(t-u)(b-e)]_3 \not\equiv 0$. Therefore, the corresponding $\boldsymbol{S}$ of $\boldsymbol{B}_s$ is capable of detecting two active users, which is congruent to cycle sum $\neq\ 0$ of (4.9).

From $\boldsymbol{B}_s$ in (4.16), we know that $\boldsymbol{S}$ is a Tx($\lceil\frac{T}{2}\rceil\lfloor\frac{T}{2}\rfloor$) synchronization matrix directly mapped from $\boldsymbol{B}_s$ in (4.16). Since each element of $\boldsymbol{B}_s$ will directly map to a column of $\boldsymbol{S}$. For example, B_{1p} is 0, according to figure 4.3 we know that $c_{2p} - c_1 = 0$. Hence, the first column of $\boldsymbol{S}$ is $[0\ \ell_{in}\ \dots\ \ell_{in}\ 0]^T$. From (4.16) we can obtain the only $\boldsymbol{S}$ when p is determined, because all the elements in $\boldsymbol{S}$ must be positive and each column of $\boldsymbol{S}$ must contain at least one zero. Since the final $\boldsymbol{B}_s$'s value B_{ij} are all $\leq \frac{p-1}{2}$ and B_{ij} maps to the values in $\boldsymbol{S}$ other than ℓ_{in} or 0, the maximum delay $\ell = \frac{p-1}{2}$. The reason of doing the procedure in (4.14b) and (4.14b) is to restrict the maximum delay in $\boldsymbol{S}$ to $\frac{p-1}{2}$.

From $\boldsymbol{B}_s$ in (4.16), we can get a corresponding $\boldsymbol{S}$ in figure (4.4). Each element of $\boldsymbol{B}_s$ in (4.16) maps to a column in $\boldsymbol{S}$ in figure (4.4). For example, $B_{ij} = s_{2p} - s_{1p} = 1$ maps to the p^{th} column of $\boldsymbol{S}$. Since $B_{ij} = 1 > 0$ and every column of $\boldsymbol{S}$ must contain a zero, we know that $S_p = [0\ 1\ \ell_{in}\ \dots\ \ell_{in}]^T$. Other columns of $\boldsymbol{S}$ can be mapped by this way too.

doi:10.6342/NTU202402984

$$
S = \begin{bmatrix}
s_{11} & \dots & s_{1,p-1} & s_{1p} & \ell_{in} & \ell_{in} & \dots & \ell_{in} & \dots & \ell_{in} \\
\ell_{in} & \dots & \ell_{in} & s_{2p} & s_{2,p+1} & \ell_{in} & \dots & \ell_{in} & \dots & s_{2,p^2} \\
\vdots & \dots & \ell_{in} & \ell_{in} & s_{3,p+1} & s_{3,p+2} & \dots & s_{3,2p} & \dots & \ell_{in} \\
\vdots & \dots & s_{4,p-1} & \vdots & \ell_{in} & s_{4,p+2} & \dots & \ell_{in} & \dots & \vdots \\
\vdots & \dots & \ell_{in} & \vdots & \vdots & \ell_{in} & \dots & \vdots & \dots & \vdots \\
\vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \dots & \ell_{in} \\
\vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \dots & s_{p,p^2} \\
\ell_{in} & \dots & \vdots & \vdots & \vdots & \ell_{in} & \dots & \ell_{in} & \dots & \ell_{in} \\
s_{2p,1} & \dots & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & \dots & s_{2p,2p} & \vdots & \ell_{in}
\end{bmatrix}
$$

Figure 4.4: Synchronization matrix of a (T,K,2)-tropical code.

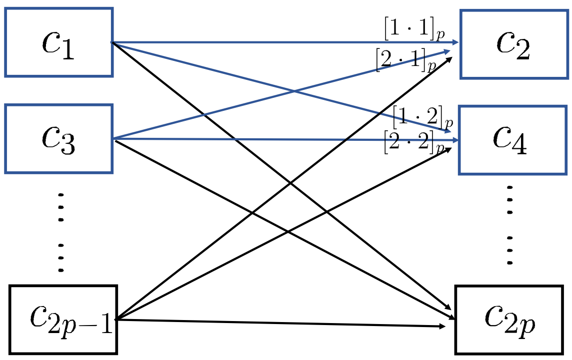

Figure 4.5: Bipartite of a (T,K,2)-tropical code.

Similar to above $(4, 4, 2)$ proof, we discuss the four possible cases in below.

Case 1: If no users are active, $\boldsymbol{S} \odot \boldsymbol{\delta}$ will result in a vector $\ell_{in}\mathbf{1}$. Furthermore if (4.12) is met $\boldsymbol{\delta} = \ell_{in}\mathbf{1}$ is the only $\boldsymbol{\delta}$ which results in $\boldsymbol{S} \odot \boldsymbol{\delta} = \ell_{in}\mathbf{1}$.

Case 2: When there is one active user in $\boldsymbol{\delta}$, only 2 elements in $\boldsymbol{S} \odot \boldsymbol{\delta}$ will be ℓ_{in} when (4.12) holds, because each column of $\boldsymbol{S}$ contains two elements smaller than ℓ_{in}. Moreover, since every user has distinct starting channel and end channel, all N possible $\boldsymbol{S} \odot \boldsymbol{\delta}$s will be distinct. The smallest element of $\boldsymbol{S} \odot \boldsymbol{\delta}$ is the delay of the corresponding active user, since each column contains a zero .

Case 3: When two active users involves 3 different channels, which means two edges

doi:10.6342/NTU202402984

involves 3 vertices in figure (4.5). Under the constraint of (4.12), the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ will have 3 distinct non-ℓ_{in} location, so we can decode $\hat{\boldsymbol{\delta}}$ easily simply by observing the non-ℓ_{in}'s location. For example, when the two active users are $j = p - 1$ and $j = p$, from the definition of tropical arithmetic, we know that the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ will be $[c_1\, c_2\, \ell_{in}\, c_4\, \ell_{in}\, \ldots \ell_{in}]^T$. $[c_1\, c_2\, \ell_{in}\, c_4\, \ell_{in}\, \ldots \ell_{in}]^T$.will be distinct from other received $\boldsymbol{S} \odot \boldsymbol{\delta}$ so we can tell that the two active users are $j = p - 1$ and $j = p$. The above mentioned flow is equivalent to two adjacent-users case mentioned in the $(4, 4, 2)'s$ proof.

Case 4: This case shows that any 2x2 sub-matrix in (4.13) has the property of cycle sum $\neq 0$. When two active users involves 4 different channels, which means two edges involves 4 vertices in figure 4.5. For example, the blue part in figure 4.5 demonstrates users $j = p - 1$ and $j = p + 1$ active and $j = p$ and $j = p + 2$ active. Under the constraint of (4.12), when the active users are $j = p - 1$ and $j = p + 1$, the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ =$[c_1\ c_2\ c_3\ c_4\ \ell_{in} \ldots \ell_{in}]^T$. When the active users are $j = p$ and $j = p + 2$, the received $[c_1\ c_2\ c_3\ c_4\ c_5]^T$ also have the structure of $[c_1\ c_2\ c_3\ c_4\ \ell_{in} \ldots \ell_{in}]^T$. where c_1, c_2, c_3, c_4 are smaller than ℓ_{in}. We cannot distinguish these two case simply by observing the non-ℓ_{in}'s location. Therefore, we use $c_1 - c_2 + c_3 - c_4$ to distinguish between these two. $c_1 - c_2 + c_3 - c_4 = 2$ for $j = p - 1$ and $j = p + 1$-active and $c_1 - c_2 + c_3 - c_4 = -2$ for ($j = p$ and $j = p + 2$)-active. We can represent ($j = p - 1$ and $j = p + 1$)-active and ($j = p$ and $j = p + 2$)-active into a 2x2 sub-matrix by figure 4.6, and still we need cycle sum $\neq 0$. Note that cycle sum = $[1 \cdot 1]_p - [2 \cdot 1]_p + [2 \cdot 2]_p - [1 \cdot 2]_p \equiv [(2-1)(2-1)]_p \not\equiv 0$. For other two-active user's pairs that can result in a 2x2 sub-matrix can be examined by the same flow above, which is congurent to $[tb - te + ue - ub]_3 \equiv [(t-u)(b-e)]_3 \not\equiv 0$ for every (4.17) mentioned above. The above mentioned flow is equivalent to the two non-adjacent active users case mentioned in the $(4, 4, 2)$'s proof, where we need the constraint

doi:10.6342/NTU202402984

of cycle sum $\neq\ 0$.

We then proof that the above lemma holds when $p > \frac{T}{2}$. When $p > \frac{T}{2}$, we take part of $\boldsymbol{B}$ of (4.16) as our biadjacency matrix $\boldsymbol{B}_s$. That is, we delete the last $p - \lceil \frac{T}{2} \rceil$ rows and last $p - \lfloor \frac{T}{2} \rfloor$ columns of $\boldsymbol{B}$ and form $\boldsymbol{B}_s$. The number of elements deleted in $\boldsymbol{B}$, we result in same number of columns deleted in $\boldsymbol{S}$ of figure 4.4 mapped from $\boldsymbol{B}$. The deleted $p - \lceil \frac{T}{2} \rceil$ rows is congurent to deleting $p - \lceil \frac{T}{2} \rceil$ vertices of the left part of figure 4.3 as well as the edges connecting the deleted vertices. Also, The deleted $p - \lceil \frac{T}{2} \rceil$ columns is congurent to deleting $p - \lceil \frac{T}{2} \rceil$ vertices of the right part of figure 4.3 as well as the edges connecting the deleted vertices.

For case 1 and case 2 the new $\boldsymbol{S}$ mapped from the new $\boldsymbol{B}_s$ is still able to maintain one-to-one correspondence between $\boldsymbol{\delta}$ and $\boldsymbol{S} \odot \boldsymbol{\delta}$. When $\boldsymbol{\delta} = [\ell_{in} \ldots \ell_{in}]$, the result of $\boldsymbol{S} \odot \boldsymbol{\delta}$ is still $[\ell_{in} \ldots \ell_{in}]^T$ and is distinct from other received $\boldsymbol{S} \odot \boldsymbol{\delta}$. When $\boldsymbol{\delta}$ has exactly one element smaller than ℓ_{in}, the received $\boldsymbol{S} \odot \boldsymbol{\delta}$s still have 2 distinct smaller than ℓ_{in} locations, since we merely delete the columns from $\boldsymbol{S}$ of figure 4.4 mapped from the $\boldsymbol{B}$ of (4.13) and each column of the original $\boldsymbol{S}$ has 2 distinct non-ℓ_{in} locations.

For case 3 and case 4 the new $\boldsymbol{S}$ mapped from the new $\boldsymbol{B}_s$ is still able to maintain one-to-one correspondence between $\boldsymbol{\delta}$ and $\boldsymbol{S} \odot \boldsymbol{\delta}$. Since, when two active users involves 3 different channels, similar to the previous proof, we can know the distinct $\boldsymbol{\delta}$ by observing the non-ℓ_{in} locations of the received $\boldsymbol{S} \odot \boldsymbol{\delta}$. For case 4, we can still leverage the distinct cycle sum when two different $\boldsymbol{\delta}$s have the same 4 location of non-ℓ_{in} elements of the received $\boldsymbol{S} \odot \boldsymbol{\delta}$.

Therefore, when $p > \frac{T}{2}$, under (4.12) and any 2x2 sub-matrix of $\boldsymbol{B}_s$ satisfies cycle sum $\neq 0$, then there exist a bipartite graph-based $(T, \lceil \frac{T}{2} \rceil \lfloor \frac{T}{2} \rfloor, 2)$-tropical code within maximum delay $\frac{p-1}{2}$. Note that deleting the elements of $\boldsymbol{B}$ in (4.13) can result in discarding

doi:10.6342/NTU202402984

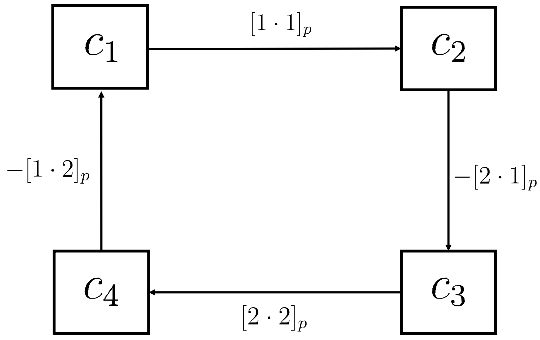

Figure 4.6: One sub 2x2 bipartite graph example of a (T,K,2)-tropical code, indicating cycle sum $\neq 0$.

the largest delay $\frac{p-1}{2}$, and thus the delay of the new $\boldsymbol{S}$ may be less than $\frac{p-1}{2}$.

4.2.1 Simple Example of (5,6,2)-Tropical Code

□

We now give an $(T, K, 2) = (5, 6, 2)$ example in below. Under

$$\ell_{in} - \ell_u > \ell, \tag{4.18}$$

and any 2x2 sub-matrix in the following $\boldsymbol{B}_s$ has the property of cycle sum $\neq 0$, which is equivalent to $s_{11} + s_{33} \neq -s_{22} - s_{44}$ in $(4, 4, 2)$-tropical code example. To be specific, any 2x2 sub-matrix in the following $\boldsymbol{B}_s$ can be denoted in the form of (4.9) and satisfies cycle sum $\neq 0$.

Then the following $\boldsymbol{B}_s$ in (4.19) can be one-to-one mapped to $\boldsymbol{S}$ in (4.25), which is a $(5, 6, 2)$-tropical code with maximum delay $\ell = \frac{p-1}{2}$, where $p = 3$. Also, $\boldsymbol{B}_s$ can be illustrated by a bipartite graph in figure (4.7).

doi:10.6342/NTU202402984

$$\boldsymbol{B}_s = \begin{bmatrix} s_{22} - s_{12} & s_{41} - s_{11} \\ s_{23} - s_{33} & s_{44} - s_{34} \\ s_{29} - s_{56} & s_{48} - s_{55} \end{bmatrix}.. \tag{4.19}$$

Also, by the more structural way to construct the values in $\boldsymbol{B}_s$ we mentioned above, that is the following.

$$\boldsymbol{B}_s = \begin{bmatrix} [1 \cdot 1]_p & [1 \cdot 2]_p \\ [2 \cdot 1]_p & [2 \cdot 2]_p \\ [3 \cdot 1]_p & [3 \cdot 2]_p \end{bmatrix}. \tag{4.20}$$

where $p = 3$ and $[B_{ij}]_p$ can be defined as the following.

$$[B_{ij}]_p = \begin{cases} B_{ij}\ mod\ p & , B_{ij}\ mod\ p \leq \frac{p-1}{2} \quad (4.21a) \\ (B_{ij}\ mod\ p) - p & , B_{ij}\ mod\ p > \frac{p-1}{2} \quad (4.21b) \end{cases}$$

and $B_{ij}\ mod\ p$ is defined as

$$B_{ij}\ mod\ p = B_{ij} - p \cdot \lfloor \frac{B_{ij}}{p} \rfloor \tag{4.22}$$

Proof. From (4.20),(4.21a), and (4.21b) we obtain the final $\boldsymbol{B}_s$

$$\boldsymbol{B}_s = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & 0 \end{bmatrix}. \tag{4.23}$$

Notice that any 2x2 sub-matrix in (4.20) has the following structure

$$\boldsymbol{B}_{22} = \begin{bmatrix} tb & te \\ ub & ue \end{bmatrix} \tag{4.24}$$

doi:10.6342/NTU202402984

t and u represent the row factor, while b and e represent the column factor. Every $\boldsymbol{B}_{22}$ satisfies $[tb - te + ue - ub]_3 \equiv [(t-u)(b-e)]_3 \not\equiv 0$. Therefore, the corresponding $\boldsymbol{S}$ of $\boldsymbol{B}_s$ is capable of detecting two active users, which is congurent to cycle sum $\neq 0$ of (4.9). The reason of doing the procedure in (4.21a) and (4.21b) is to restrict the maximum delay in the corresponding $\boldsymbol{S}$ to $\frac{p-1}{2}$, $p = 3$ in this example.

From (4.19),(4.23) we can obtain the only $\boldsymbol{S}$ in the following, because all the elements in $\boldsymbol{S}$ must be positive and each column of $\boldsymbol{S}$ must contain at least one zero. Since the final $\boldsymbol{B}_s$'s value B_{ij} are all $\leq \frac{p-1}{2}$ and B_{ij} maps to the values in $\boldsymbol{S}$ other than ℓ_{in} or 0, the maximum delay $\ell = \frac{p-1}{2}$, which can be observed in (4.25)'s $\boldsymbol{S}$.

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} \\ \ell_{in} & 1 & 0 & \ell_{in} & \ell_{in} & 0 \\ \ell_{in} & \ell_{in} & 1 & 0 & \ell_{in} & \ell_{in} \\ 0 & \ell_{in} & \ell_{in} & 1 & 0 & \ell_{in} \\ \ell_{in} & \ell_{in} & \ell_{in} & \ell_{in} & 0 & 0 \end{bmatrix} \tag{4.25}$$

We discuss the four possible cases in below.

Case 1: If no users are active, $\boldsymbol{S} \odot \boldsymbol{\delta}$ will result in a vector $\ell_{in}\mathbf{1}$. Furthermore if (4.18) is met $\boldsymbol{\delta} = \ell_{in}\mathbf{1}$ is the only $\boldsymbol{\delta}$ which results in $\boldsymbol{S} \odot \boldsymbol{\delta} = \ell_{in}\mathbf{1}$.

Case 2: When there is one active user in $\boldsymbol{\delta}$, only three elements in $\boldsymbol{S} \odot \boldsymbol{\delta}$ will be ℓ_{in} when (4.18) holds and the synchronization matrix is selected as (4.25). Moreover all five possible $\boldsymbol{S} \odot \boldsymbol{\delta}$s will be distinct, and the smallest element of $\boldsymbol{S} \odot \boldsymbol{\delta}$ is the delay of the corresponding active user. Case 3: When two active users involves 3 different channels, which means two edges involves 3 vertices in figure (4.7). Under the constraint of (4.18), the received $[c_1\ c_2\ c_3\ c_4\ c_5]^T$ will have 3 distinct non-ℓ_{in} location, so we can decode $\hat{\boldsymbol{\delta}}$ easily simply by observing the non-ℓ_{in}'s location. For example, when $j = 2$ and $j = 3$ are

doi:10.6342/NTU202402984

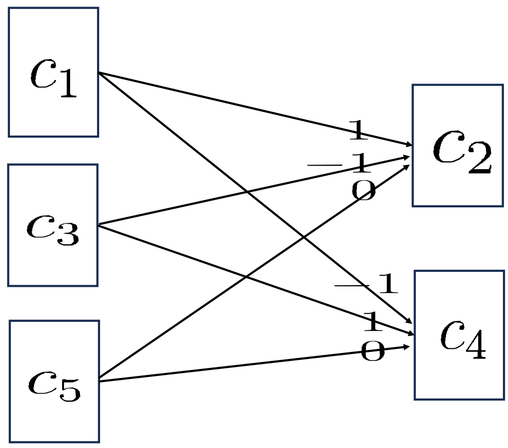

Figure 4.7: Bipartite of a (5,6,2)-tropical code.

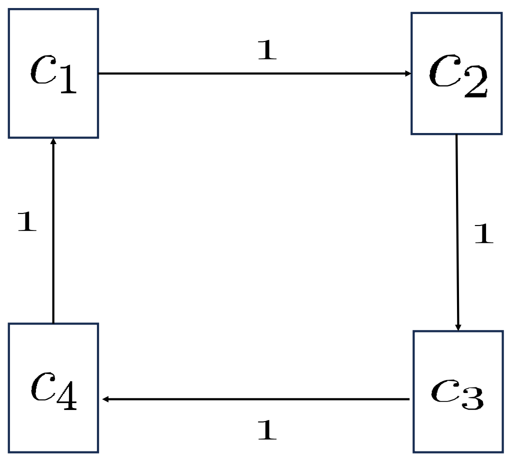

Figure 4.8: One sub2·2 bipartite graph example of a (5,6,2)-tropical code, indicating cycle sum $\neq 0$.

doi:10.6342/NTU202402984

the two active users, from the definition of tropical arithmetic, we know that the received $[c_1\ c_2\ c_3\ c_4\ c_5]^T$ will be $[c_1\ c_2\ c_3\ \ell_{in}\ \ell_{in}]^T$. $[c_1\ c_2\ c_3\ \ell_{in}\ \ell_{in}]^T$ will be distinct from other received $[c_1\ c_2\ c_3\ c_4\ c_5]^T$ so we can tell that the two active users are corresponding to the second and third columns of $\boldsymbol{S}$ in (4.25). The above mentioned flow is equivalent to two adjacent-users case mentioned in the $(4,4,2)'s$ proof.

Case 4: This case shows that any $2 \cdot 2$ sub-matrix in (4.19) has the property of cycle sum $\neq 0$. When two active users involves 4 different channels, which means two edges involves 4 vertices in figure 4.7. For example, under the constraint of (4.18), when the active users are $j = 1$ and $j = 3$, the received $[c_1\ c_2\ c_3\ c_4\ c_5]^T = [c_1\ c_2\ c_3\ c_4\ \ell_{in}]^T$. When the active users are $j = 2$ and $j = 4$, the received $[c_1\ c_2\ c_3\ c_4\ c_5]^T$ also have the structure of $[c_1\ c_2\ c_3\ c_4\ \ell_{in}]^T$, where c_1, c_2, c_3, c_4 are smaller than ℓ_{in}. We cannot distinguish these two case simply by observing the non-ℓ_{in}'s location. Therefore, we use $c_1 - c_2 + c_3 - c_4$ to distinguish between these two. $c_1 - c_2 + c_3 - c_4 = 2$ for ($j = 1$ and $j = 3$)-active and $c_1 - c_2 + c_3 - c_4 = -2$ for ($j = 2$ and $j = 4$)-active. We can represent ($j = 1$ and $j = 3$)-active and ($j = 2$ and $j = 4$)-active into a $2 \cdot 2$ sub-graph by figure 4.7, and still we need cycle sum $\neq 0$. For other two-active user's pairs that can result in a $2 \cdot 2$ sub-graph can be examined by the same flow above, which is congruent to $[tb - te + ue - ub]_3 \equiv [(t-u)(b-e)]_3 \not\equiv 0$ for every (4.17) mentioned above. The above mentioned flow is equivalent to the two non-adjacent active users case mentioned in the $(4,4,2)$'s proof, where we need the constraint of cycle sum $\neq 0$.

$\square$

doi:10.6342/NTU202402984

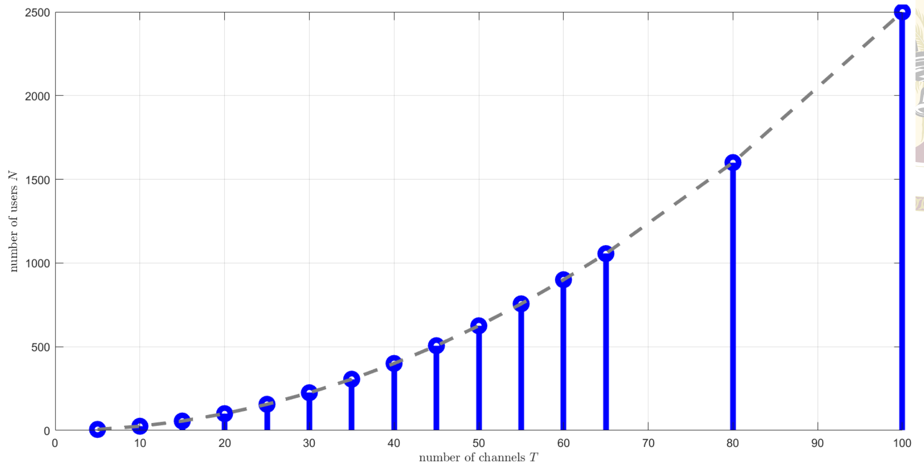

Figure 4.9: simulation results of (T,K,2)-tropical code.

4.3 Simulation Results

The points of picture in figure 4.9 shows the $(T, K, 2)$ combinations we have simulated by matlab. Any points on this figure has the property that $\boldsymbol{\delta}$ and $\boldsymbol{S} \odot \boldsymbol{\delta}$ have a one-to-one mapping relation. That is, from the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ we can decode a $\hat{\boldsymbol{\delta}}$ that is equal to $\boldsymbol{\delta}$. We define the error probability P_e as the probability of decoded $\hat{\boldsymbol{\delta}}$ from $\boldsymbol{S} \odot \boldsymbol{\delta}$ not equal to $\boldsymbol{\delta}$. We found that $P_e = 0$ for all the simulation points on figure 4.9. We run all the points in figure 4.9 with $\ell_{in} = 54$, because the largest T in this simulation = 100 thus $\ell = 26$ and we pick $\ell_u = 27$ for $\ell_u > \ell$, also under the constraint that $\ell + \ell_u < \ell_{in}$.

The decoder decodes $\boldsymbol{S} \odot \boldsymbol{\delta}$ and gets a $\hat{\boldsymbol{\delta}}$ which is the same as $\boldsymbol{\delta}$. The received $\boldsymbol{S} \odot \boldsymbol{\delta}$ can be divided into four cases, which is mentioned in *Lemma5*. When received $\boldsymbol{S} \odot \boldsymbol{\delta}$ is all ℓ_{in} we know that $\hat{\boldsymbol{\delta}}$ is all ℓ_{in} as well. When the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ contains two non-ℓ_{in} locations we know that there is only one active user. The active user's path delay δ_j is the smallest value in the received $\boldsymbol{S} \odot \boldsymbol{\delta}$, and the active user's index j is the column index of $\boldsymbol{S}$ that is equal to $\boldsymbol{S} \odot \boldsymbol{\delta}$ - $\delta_j \mathbf{1}$. When the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ contains three non-ℓ_{in} locations, we know that there are two active users and the two active users does not result in a bipartite

doi:10.6342/NTU202402984

graph. We find two columns of $\boldsymbol{S}$ that there union of non-ℓ_{in} locations are the same as the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ and mark them as S_{j1} and S_{j2}. Use $\boldsymbol{S} \odot \boldsymbol{\delta}$ to minus the non-ℓ_{in} values of S_{j1} and S_{j2} to get δ_{j1} and δ_{j2}, which indicies are not the intersection index of the non-ℓ_{in} index of S_{j1} and S_{j2}. When the received $\boldsymbol{S} \odot \boldsymbol{\delta}$ has 4 non-ℓ_{in} locations we do cycle sum of 4 values to determine which two pairs of columns that result in the received $\boldsymbol{S} \odot \boldsymbol{\delta}$. As we know the two columns we can decode the corresponding path delays by similar way mentioned above. We can refer to appendix 5 to see the complete decode flow.

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Chapter 5 Conclusion

In this paper, we leverage tropical arithmetic to propose a low latency massive MAC communication system. Specifically, our delay ℓ has two mathematical constraints, $\ell < \ell_{in} - \ell_u$ and $\ell < \ell_u$. Secondly, we wouldn't have to wait till infinity to be able to detect active user's activity and delay, the deadline ℓ_{in} is finite. Also, if we use CDMA our latency can be even lower comparing to TDMA. From given MOCS, this communication system is practical. For detecting two active users, from our proposed bipartite matrix we are able to instantly generate asynchrony-detection matrix, which can ensure zero error probability under noiseless channel. Lastly, if want to detect K users' identities and delays, we would need K MOCS sets. However, with tropical linear algebra, we only need $T < K$ MOCS sets and pre-shifts to detect K_a active users' delays and idetities.

For future work, we are dedicated to the following two points. If the delay vector's code book and asynchrony-detection matrix's code book are all composed of real numbers instead of integers. We doubt whether such asynchrony-detection matrix $\boldsymbol{S}$ still satisfy the definition of tropical code, even though [15] stated that tropical arithmetic still work with real number code book. We still need proper mathematical proof. Moreover, we want to expand our detection scheme to 3 or more active users in the future.

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

References

[1] K. Andreev, S. S. Kowshik, A. Frolov, and Y. Polyanskiy. Low complexity energy efficient random access scheme for the asynchronous fading MAC. In 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), pages 1–5, 2019.

[2] L. Applebaum, W. U. Bajwa, M. F. Duarte, and R. Calderbank. Asynchronous code-division random access using convex optimization. Physical Communication, 5(2):129–147, 2012.

[3] X. Chen, T.-Y. Chen, and D. Guo. Capacity of Gaussian many-access channels. IEEE Transactions on Information Theory, 63(6):3516–3539, 2017.

[4] X. Chen, L. Liu, D. Guo, and G. W. Wornell. Asynchronous massive access and neighbor discovery using OFDMA. IEEE Transactions on Information Theory, 69(4):2364–2384, 2023.

[5] S. Deryat, S.-C. Yu, H.-W. Young, E. A. Jorswieck, P.-H. Lin, and S.-C. Lin. Joint delay and user activity detection in asynchronous massive access. WOCC24, 2024.

[6] A. Fengler, A. Lancho, K. Narayanan, and Y. Polyanskiy. On the advantages of asynchrony in the unsourced MAC. In 2023 IEEE International Symposium on Information Theory (ISIT), pages 2523–2528, 2023.

doi:10.6342/NTU202402984

[7] H. A. Inan, S. Ahn, P. Kairouz, and A. Ozgur. A group testing approach to random access for short-packet communication. In 2019 IEEE International Symposium on Information Theory (ISIT), pages 96–100, 2019.

[8] H. A. Inan, P. Kairouz, and A. Ozgur. Sparse combinatorial group testing. IEEE Transactions on Information Theory, 66(5):2729–2742, 2020.

[9] S. S. Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy. Short-packet low-power coded access for massive MAC. In 2019 53rd Asilomar Conference on Signals, Systems, and Computers, pages 827–832, 2019.

[10] S.-C. Lin, T.-H. Chang, E. A. Jorswieck, and P.-H. Lin. Information theory, mathematical optimization, and their crossroads in 6G system design. Springer Series in Wireless Technology, 1st edition, 2023.

[11] K.-H. Liu, X. Li, H. Zhao, and G. Fan. Joint active user detection and channel estimation for massive grant-free access via difference of convex programming. In IEEE Global Communications Conference (GLOBECOM), Dec. 2023.

[12] Y. Polyanskiy. A perspective on massive random-access. In 2017 IEEE International Symposium on Information Theory (ISIT), pages 2523–2527, 2017.

[13] Qualcomm. 3GPP release 17: Completing the first phase of the 5G evolution, Mar. 2022.

[14] I. Trots, Y. Tasinkevych, and A. Nowicki. Orthogonal golay codes with local beam pattern correction in ultrasonic imaging. IEEE Signal Processing Letters, 22(10):1681–1684, 2015.

doi:10.6342/NTU202402984

[15] H.-P. Wang, R. Gabrys, and A. Vardy. Tropical group testing. IEEE Transactions on Information Theory, 69(9):6098–6120, 2023.

[16] J.-S. Wu, P.-H. Lin, M. A. Mross, and E. A. Jorswieck. Worst-case per-user error bound for asynchronous unsourced multiple access. In IEEE International Symposium on Information Theory (ISIT), 2024.

[17] S.-W. Wu, C.-Y. Chen, and Z. Liu. How to construct mutually orthogonal complementary sets with non-power-of-two lengths? IEEE Transactions on Information Theory, 67(6):3464–3470, 2021.

[18] S.-W. Wu, C.-Y. Chen, and Z. Liu. How to construct mutually orthogonal complementary sets with non-power-of-two lengths? IEEE Transactions on Information Theory, 67(6):3464–3472, 2021.

[19] C.-H. Yang, Y.-H. Lin, S.-C. Lin, and T.-D. Chiueh. Design of a low-complexity receiver for impulse-radio ultra-wideband communication systems. In 2004 IEEE International Symposium on Circuits and Systems (ISCAS), volume 4, pages IV–125, 2004.

doi:10.6342/NTU202402984

doi:10.6342/NTU202402984

Appendix A — Decoder for $(2, K, 1)$-tropical code of (3.15)

Require: $\mathbf{S} \odot \boldsymbol{\delta} = \begin{bmatrix} a & b \end{bmatrix}^T, \ell_{in} > \ell_u, \ell_{in} > \ell_u + \ell$

$result \leftarrow \begin{bmatrix} \ell_{in} & \ell_{in} & \dots & \ell_{in} \end{bmatrix}^T$

if $a == \ell_{in}$ and $b == \ell_{in}$ **then** ▷ Case 1

The delay matrix consists of l_{in}s

return $result$

else if $a\, != \ell_{in}$ and $b == \ell_{in}$ **then** ▷ Case 3

The index of the active user is 1

$\delta_u = a - s_{11}$

$result[1, 1] \leftarrow \delta_u$

return $result$

else if $a == \ell_{in}$ and $b\, != \ell_{in}$ **then** ▷ Case 4

The index of the active user is K

$\delta_u = b - S_{2K}$

$result[1, K] \leftarrow \delta_u$

return $result$

else ▷ Case 2

doi:10.6342/NTU202402984

$index_u \leftarrow a - b + l + 2$

if $a - b < 0$ **then**

$\delta_u \leftarrow a$

else

$\delta_u \leftarrow b$

end if

$result[1, index_u] \leftarrow \delta_u$

return $result$

end if

doi:10.6342/NTU202402984

Appendix B — Decoder for $(T, K, 2)$-tropical code of *Lemma* 5

Require: $T \geq 5, \ell_u > \ell, \ell_{in} > \ell_u + \ell$

input(T)

p = nextprime($\frac{T}{2}$), $\ell = \frac{p-1}{2}$

if $\boldsymbol{S} \odot \boldsymbol{\delta} == [\ell_{in} \ \ldots \ \ell_{in}]^T$ **then** ▷ Case 1

The delay matrix consists of l_{in}s

return $[\ell_{in} \ \ldots \ \ell_{in}]$

else if $\boldsymbol{S} \odot \boldsymbol{\delta}$ consists exactly 2 non-ℓ_{in} values **then** ▷ Case 2

c_1 = first smaller than ℓ_{in} index of $\boldsymbol{S} \odot \boldsymbol{\delta}$

c_2 = second smaller than ℓ_{in} index of $\boldsymbol{S} \odot \boldsymbol{\delta}$

Find the column of $\boldsymbol{S}$ that has the same index of c_1 and c_2, and name it j

δ_j = min($\boldsymbol{S} \odot \boldsymbol{\delta}$)

$result = [\ell_{in} \ldots \ell_{in}]$

$result[1, j] = \delta_j$

return $result$

else if $\boldsymbol{S} \odot \boldsymbol{\delta}$ consists exactly 3 non-ℓ_{in} values **then** ▷ Case 3

c_1 = first smaller than ℓ_{in} index of $\boldsymbol{S} \odot \boldsymbol{\delta}$

c_2 = second smaller than ℓ_{in} index of $\boldsymbol{S} \odot \boldsymbol{\delta}$

doi:10.6342/NTU202402984

c_3 = third smaller than ℓ_{in} index of $\boldsymbol{S} \odot \boldsymbol{\delta}$

Find two columns of $\boldsymbol{S}$ and find the uninon of their non-$\ell'_{in}s$ location, and name two columns j_1 and j_2

Find the intersection of the non-$\ell'_{in}s$ index for two columns S_{j_1} and S_{j_2}, and name it k

Find the non-k index for each column, and name them l and m

$result = [\ell_{in} \dots \ell_{in}]$

δ_1=$(\boldsymbol{S} \odot \boldsymbol{\delta})[l] - S_{j_1}[l]$

δ_2=$(\boldsymbol{S} \odot \boldsymbol{\delta})[m] - S_{j_2}[m]$

$result[1, j_1] = \delta_1$

$result[1, j_2] = \delta_2$

return $result$

else ▷ Case 4

The received $\boldsymbol{S} \odot \boldsymbol{\delta}$ has 4 exactly non-ℓ_{in} locations

We name the 4 non-ℓ_{in} values as c_1, c_2, c_3, c_4

There are going to be two columns pairs (S_{j_1}, S_{j_2}) and (S_{j_3}, S_{j_4}) that have the same 4 non-ℓ_{in} locations

two pairs all have distinct $c_1 - c_2 + c_3 - c_4$, so we know which pair, suppose j_1 and j_2 here

find the two 0 location for each column of the pair, and name them l and m

δ_1=$(\boldsymbol{S} \odot \boldsymbol{\delta})[l] - S_{j_1}[l]$

δ_2=$(\boldsymbol{S} \odot \boldsymbol{\delta})[m] - S_{j_2}[m]$

$result = [\ell_{in} \dots \ell_{in}]$

$result[1, j_1] = \delta_1$

doi:10.6342/NTU202402984

$result[1, j_2] = \delta_2$

return $result$

end if

doi:10.6342/NTU202402984