Interactive Code to Correct and Detect Omniscient Byzantine Adversaries

Zhiqing Xiao, Yunzhou Li, Ming Zhao, and Jing Wang
Tsinghua University, Beijing 100084, China
Email: xzq.xiaozhiqing@gmail.com

***Abstract*—This paper considers interactive transmissions in the presence of omniscient Byzantine attacks. Unlike prior papers, it is assumed that the number of transmissions, the number of erroneous transmissions therein, and the direction of each transmission are predetermined. Besides, the size of the alphabet in each transmission is unequal and predefined. Using these transmissions, two nodes communicate interactively to send a message. In this model, both attack strategies and coding bounds are considered. Although the codebook can not fully describe the interactive code, we still assert the existence of successful attack strategies according to the relations between codewords in the codebook. Furthermore, to ensure that the code is able to detect or correct a given number of transmission errors, upper bounds on the size of code are derived. Finally, the tightness of the bounds is discussed.**

I. Introduction

Consider the interactive transmissions in the presence of omniscient Byzantine adversaries (depicted in Fig. 1). Node A tries to send a message to node B via n transmissions. Each transmission is either from A to B or from B to A, and one letter in the alphabet of the transmission is noiselessly transmitted when the transmission is not attacked. The direction and the alphabet of each transmission are predefined. The adversary knows the message *a priori*, and maliciously modifies letters in at most z transmissions, where the resulting letters are still in the alphabets of the corresponding transmissions. At the same time, A and B cooperate to execute an interactive code to either correct or detect the adversary.

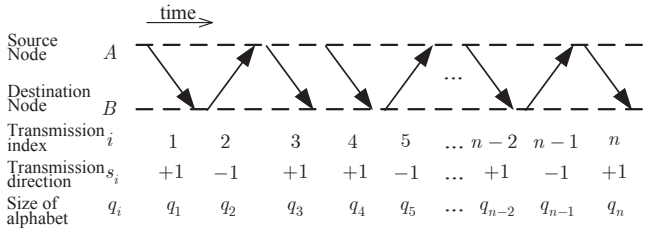


Fig. 1. Interactive transmissions between A and B. Transmissions are denoted by indexes $1, 2, \ldots, n$. The alphabet of transmission i is $\{0, 1, \ldots, q_i - 1\}$. Other notations are defined in Section II.

Interactive coding can be used to combat transmission errors. In [2]–[4], two nodes communicate interactively to perform a task, and adversaries maliciously corrupt a constant faction of transmissions. In these papers, the alphabet of transmissions is constant-sized, but the direction of each transmission is designable. They derived a lower bound on the proportion of erroneous transmissions that the code can correct and an upper bound on the total number of transmissions to guarantee the accomplishment of the task. Unlike these works, the model in our paper predetermines all transmissions' parameters, including the total number of transmissions, the directions of each transmission, and the alphabets of each transmission. We try to figure out how much information can be sent via these transmissions.

The idea that uses redundant transmissions to detect or correct errors is also called *network error correction* [5], [6]. Previous works showed that, linear network error correction codes can attain the capacity of directed acyclic network of unit-capacity links when repeated channel uses are allowed [7]–[11]. However, [12]–[14] shows that linear code does not suffice to achieve the capacity in general. In [12], a kind of networks, called zigzag networks, were considered. The topology of zigzag networks is similar to that in this paper, since the communications in zigzag networks are essentially interactive. Additionally, [12] showed that, for a network error correction code, if two codewords match in a particular way, the adversary can confuse the legitimate parties accordingly. Furthermore, upper bounds on the capacity were derived using contradiction: If the size of a network error correction code is larger than an upper bound, according to the Pigeonhole Principle, there exist two codewords such that the letters are identical in some links, which results in the existence of uncorrectable attacking scheme. In some zigzag networks, the capacity can be attained by network error correction codes such as "Guess and Forward" code.

In this paper, the transmitting and defending strategies of the legitimate parties, henceforth simply called "*code*," consist of:

- *Encoders:* For any transmission, the encoder needs to decide what to transmit according to the received letters. For A, the message to send is also taken into consideration.
- *Decoder:* After the transmissions, the decoder in B maps the letter sequence that B received to a message in the message set, or reports error when it encounters errors that are unable to be corrected.

The *codeword* is the transmitting letter sequence in all transmissions when no transmissions are attacked. For a code,

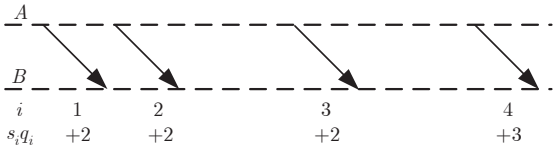
Due to space limitations, the details of the proofs and related discussions are presented in the extended version of this paper [1].

978-1-4799-5999-0/14/$31.00 ©2014 IEEE

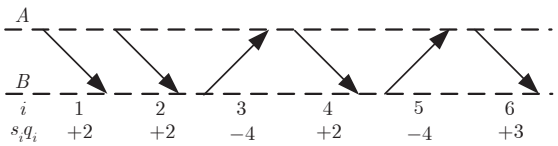
the set of all codewords is called *codebook*. In the cases that all the transmissions are from A to B, the correction/detection capability of a code is merely decided by the codebook. Specifically, if the minimum Hamming distance of any two distinct codewords in the codebook is $d_{\min}$, the code can correct arbitrary $\lfloor (d_{\min}-1)/2 \rfloor$ errors, or detect arbitrary $(d_{\min}-1)$ errors. However, this property generally no longer holds when there exist transmissions from B to A, since the codebook can not fully describe the code. The codebook can only determine (1) what A is to transmit when all its previous received letters are correct, and (2) what B is to transmit when the previous received letter sequence matches some sequence in the codebook. But in other cases, the letter sequence the node receives may not match the entry/entries in the codebook, so the codebook can not decide how to encode or decode afterward. However, subsequent encoding and decoding operations do affect the error correction/detection capability. Therefore, the codebook of an interactive code can not fully determine the correction/detection capability.

Fortunately, it is also possible to identify some limitations on the error correction/detection capability of a code merely through the codebook. For example, if two codewords differ only on the entry in the last transmission, which is a transmission from A to B, the code can neither detect nor correct one error. The reason is, when the message associated with one of the aforementioned codewords is being sent, the adversary can maliciously change the letter in the last transmission to what would be sent when the message associated with the other codeword is sent. In this case, B is unable to detect this modification. Therefore, it is also possible to assert that a code is unable to correct (or detect) a given number of errors merely by the codebook.

Let the term "the *size* of the code" denote the cardinality of the message set of an interactive code. Obviously, the size of the code is equal to the number of different codewords in the codebook. As is known to all, feedback can increase the capacity of memory channels. Similarly, the existence of transmissions from B to A can enlarge the message set. For example, in Fig. 2(a), there are four transmissions with alphabets $\{0,1\}$, $\{0,1\}$, $\{0,1\}$, and $\{0,1,2\}$ respectively. The maximum size of the code to correct one arbitrary error is three. That is



(a) Four feedforward transmissions (size of 1-correctable code is 3)



(b) Six interactive transmissions (size of 1-correctable code is 4)

Fig. 2. Example of the benefit of feedback transmissions

because, in order to correct one error, the minimum distance of any two distinct codewords in the codebook should be ≥ 3. In [15, Proposition 4.1(v)], it is shown that, in such mixed binary/ternary case, the maximum size of codes of minimum distance ≥ 3 is three exactly. (One possible codebook is $\{0000, 0111, 1012\}$.) Compared to Fig. 2(a), Fig. 2(b) has two more transmissions from B to A, of which each transmits one letter in $\{0,1,2,3\}$. In Section IV-C of this paper, it is proved that the maximum size of such 1-correctable code is four. Therefore, the existence of transmissions from B to A can increase the size of the code.

The contribution of this paper is twofold:

(1) *Attacks to the Codebook:* Although the codebook can not fully determine the error detection/correction capability, when the codebook has some properties, there still exist ways to assert the existence of attack strategies to defeat the interactive code. First, the equivalent definitions of error-detectable codes and error-correctable codes are provided respectively. Second, we show that if the entries in two distinct codewords match in a particular way, there exists an attack strategy such that B can not detect or correct the modification.

(2) *Coding Bounds:* We derive the upper bounds on the size of interactive codes. Based on the existence of attack strategies, it is shown that, if an interactive code can detect or correct z malicious errors, the size of the code is not greater than the given bound. Moreover, an example is presented to discuss the tightness of the upper bounds.

Paper Outline: The remainder of the paper is organized as follows: Section II introduces the system model. Section III studies the existence of successful attack strategies. Section IV derives the upper bounds on the size of error-detecting codes and error-correcting codes. Section V concludes the paper.

Notation: Let $\mathbb{N}$ be the set of natural numbers. Some basic notations are defined in Table I. Other notations, such as $[i_1, i_2)$ and $X_-^{i_2}$, can be defined similarly. For any set $I \subseteq [1,n]$, let $|I|$ denote the number of elements therein.

TABLE I
SOME NOTATIONS
($i_1, i_2 \in \mathbb{N}$, $I \subseteq \mathbb{N}$, AND $\{X_1, \cdots, X_n\}$ IS A SEQUENCE.)

$[i_1, i_2]$	$= \{i \in \mathbb{N} : i_1 \leq i \leq i_2\}$
$(i_1, i_2]$	$= \{i \in \mathbb{N} : i_1 < i \leq i_2\}$
$[i_1, i_2]_+$	$= \{i \in [i_1, i_2] : s_i = +1\}$
I_+	$= \{i \in I : s_i = +1\}$
$X_{i_1}^{i_2}$	$= \{X_i : i \in [i_1, i_2]\}$
X^{i_2}	$= \{X_i : i \in [1, i_2]\}$
X_I	$= \{X_i : i \in I\}$
$X_{i_1+}^{i_2}$	$= \left\{X_i : i \in [i_1, i_2]_+\right\}$
$X_+^{i_2}$	$= \left\{X_i : i \in [1, i_2]_+\right\}$

II. PRELIMINARIES

There are n transmissions in total. All transmissions are indexed and denoted by $\{1, 2, \cdots, n\}$ in succession. For the ith transmission, its direction s_i is defined as

$$s_i = \begin{cases} +1, & \text{transmission } i \text{ is from } A \text{ to } B \\ -1, & \text{transmission } i \text{ is from } B \text{ to } A. \end{cases}$$

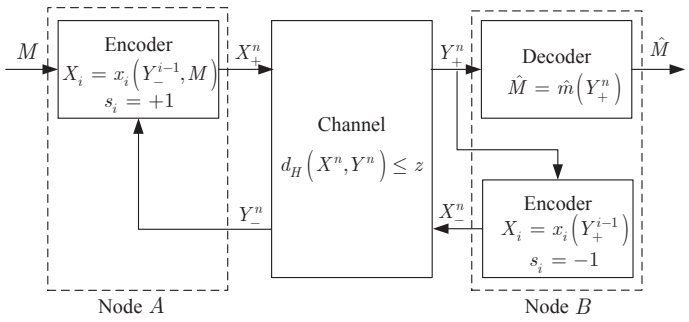


Fig. 3. Block diagram.

The alphabet of this transmission is $\{0, 1, \cdots, q_i - 1\}$. The corresponding transmitted letter and the received letter are X_i and Y_i respectively.

For two sequences X_I and Y_I, the Hamming distance of these two sequences is

$$d_H\left(X_I, Y_I\right) = \left|\{i \in I : X_i \neq Y_i\}\right|.$$

Let $\mathcal{M}$ be the message set. A message $M \in \mathcal{M}$ needs to be transmitted from A to B. A, which is on the left side of the channel in Fig. 3, knows the message M and consists of:

- *Encoder* for transmission i such that $s_i = +1$: $X_i = x_i\left(Y_-^{i-1}, M\right)$.

B, which is on the right side of the channel in Fig. 3, tries to recover the message and consists of:

- *Encoder* for transmission i such that $s_i = -1$: $X_i = x_i\left(Y_+^{i-1}\right)$, and
- *Decoder:* $\hat{M} = \hat{m}\left(Y_+^n\right)$, where $\hat{M} \in \mathcal{M} \bigcup \{\varepsilon\}$, and ε indicates an error symbol that does not belong to $\mathcal{M}$.

The adversary knows the code (x^n and $\hat{m}$) and the message (M) *a prior* and can maliciously change at most z letters:

$$d_H\left(X^n, Y^n\right) \le z.$$

Definition 1 (z-correctable): A code is z-correctable iff $\hat{M} = M$ always holds when $d_H\left(X^n, Y^n\right) \le z$.

Definition 2 (z-detectable): A code is z-detectable iff (1) when $d_H\left(X^n, Y^n\right) \le z$, $\hat{M} = M$ or $\hat{M} = \varepsilon$; (2) when $X^n = Y^n$, $\hat{M} = M$.

III. Attacks to Codewords

Recall that, the codeword of a message m is the letter sequence that all transmissions transmit when no errors occur (denoted as c^n). This section will show that, if two distinct codewords satisfy some conditions, we can constructively prove the existence of attack strategies to defeat the code.

A. Attack Error-Detecting Codes

To attack an error-detecting code, the adversary needs to modify the transmissions, so that the letter sequence received by B is equal to what B should receive when another message is to be sent. For example, if the adversary wants to fool B when A tries to transmit message m, the adversary needs to find another message $\bar{m}$, whose codeword is $\bar{c}^n$, and modifies some transmissions to let B receive $\bar{c}_+^n$. If he succeeds, B will decode $\bar{c}_+^n$ as $\bar{m}$ without detecting any modifications. Accordingly, we can reinterpret z-detectable codes as follows:

Proposition 1 (z-Detectable Code, Theorem 2 of [16]): A code is an z-detectable code only if for any two distinct messages $m, \bar{m} \in \mathcal{M}$ ($m \neq \bar{m}$), for any malicious modification of message m transmitting such that $d_H\left(y^n, x^n\right) \le z$, and any malicious modification of message $\bar{m}$ transmitting such that $\bar{y}^n = \bar{x}^n$, we have $y_+^n \neq \bar{y}_+^n$.

In Proposition 1, for the transmissions of message $\bar{m}$, the requirement that $\bar{y}^n = \bar{x}^n$ actually means no modifications. Using this proposition, we can derive the following theorem:

Theorem 1 (Not z-Detectable Code): For an interactive code, c^n and $\bar{c}^n$ are the codewords of two different messages. If there exists an integer $n' \in [0, n]$ such that

$$d_H\left(c^{n'}, \bar{c}^{n'}\right) + \left|(n', n]_+\right| \le z, \tag{1}$$

then the code is not z-detectable.

Proof Outline: Let $I_{a+} = \left\{i \in [1, n']_+ : c_i \neq \bar{c}_i\right\}$ and $I_{a-} = \left\{i \in [1, n']_- : c_i \neq \bar{c}_i\right\}$. (1) leads to

$$|I_{a+}| + |I_{a-}| + \left|(n', n]_+\right| \le z.$$

When A wants to send the message corresponding to c^n, the adversary changes the letters in I_{a+} from $c_{I_{a+}}$ to $\bar{c}_{I_{a+}}$, the letters in I_{a-} from $\bar{c}_{I_{a-}}$ to $c_{I_{a-}}$, and the letters in $(n', n]_+$ to $\bar{c}_{n'+1,+}^n$. Then, what B receives is exactly $\bar{c}_+^n$. Consequently, B decodes the received letters as $\bar{m}$ without detecting the modifications. ∎

In this theorem, all transmissions are divided into two phases (See Fig. 4). The comparisons of entries matter in the transmissions $[1, n']$, while the directions of transmissions matter in the transmissions $(n', n]$. This partition is according to whether A is allowed to detect the adversarial errors. In the first phase, A is unable to detect the errors, while in the second phase, he is. In this sense, such partition is without loss of generality.

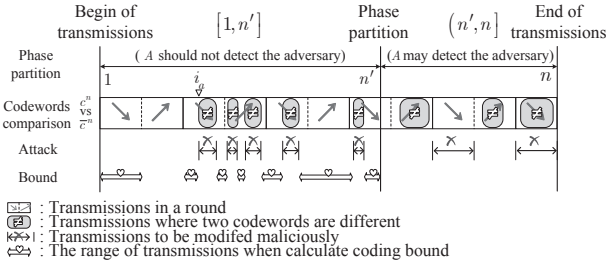


Fig. 4. Divide all transmissions into two phases: $[1, n']$ and $(n', n]$.

B. Attack Error-Correcting Codes

The analysis for error-correction codes is similar to that for error-detection codes. The equivalent characterization of the z-correctable codes is:

Proposition 2 (z-Correctable Code, Theorem 1 of [16]): A code is an z-correctable code only if for any two distinct messages $m, \bar{m} \in \mathcal{M}$ ($m \neq \bar{m}$), for any malicious modification

of message m transmitting such that $d_H\left(y^n, x^n\right) \leq z$ and any malicious modification of message $\bar{m}$ transmitting such that $d_H\left(\bar{y}^n, \bar{x}^n\right) \leq z$, we have $y_+^n \neq \bar{y}_+^n$.

The proposition tells us, for an interactive code, if there exist two distinct messages $m, \bar{m}$ such that the adversary can make B receive the same sequence in transmissions $[1,n]_+$ when either of messages is being transmitted, node B has no way to tell which message of these two are the original message. Thus, B is unable to recover the message in this case, and this interactive code can not correct the adversarial errors.

Theorem 2 (Not z-Correctable Code): For an interactive code, c^n and $\bar{c}^n$ are the codewords of two different messages. If there exist integers $n' \in [0,n]$ and $n'' \in [0,n']$ such that $|I_a| \leq z$, $\left|\bar{I}_a\right| \leq z$, and

$$|I_a| + \left|\bar{I}_a\right| + \left|(n',n]_+\right| \leq 2z$$

where

$$\begin{aligned} I_a &= \{i \in [1,n''] : c_i \neq \bar{c}_i\} \bigcup (n'',n']_-, \\ \bar{I}_a &= \{i \in (n'',n'] : c_i \neq \bar{c}_i\} \bigcup (n'',n']_-, \end{aligned}$$

then the code is not z-correctable.

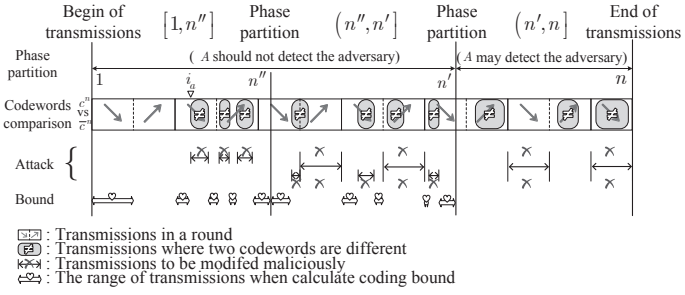


Fig. 5. Divide all transmissions into three phases: $[1,n'']$, $(n'',n']$, and $(n',n]$.

In this theorem, all transmissions are divided into three phases (See Fig. 5). The comparisons of entries matter in the transmissions $[1,n'']$; both the comparisons of entries and the directions matter in the transmissions $(n'',n']$; and the directions matter in the transmissions $(n',n]$. This partition is according to whether A and B detect the errors. Specifically, A is not allowed to detect the adversarial errors in both the first and the second phase, and B is not allowed to detect in merely the first phase. Such partition assumes that B detects the errors earlier than A. In fact, this assumption is without loss of generality: If A detects the errors before B, subsequent transmissions from A to B are not defined by codewords and need to be modified. At the time, it does not matter whether B detects the errors or not.

IV. Coding Bounds

Recall that the size of code is defined as the cardinality of message set of an interactive code. In this section, we consider the upper bounds on the size of interactive codes that can detect/correct arbitrary z adversarial errors.

A. Bounds on the Size of Error-Detecting Codes

Theorem 3 (z-Detectable Bound): The maximum size of z-detectable codes is upper bounded by

$$A_{s^n \circ q^n}^{(z,0)} \leq \min_{n' \in [0,n]:\left|(n',n]_+\right| \leq z} Q_{s^{n'} \circ q^{n'}}^{\left(z-\left|(n',n]_+\right|\right)}$$

where $Q_{s_{i_1}^{i_2} \circ q_{i_1}^{i_2}}^{(z)}$ is defined in (2) (See the bottom of the page).

Proof Outline: First, we use contradiction to show that, for any $n' \in [0,n]$ and $I_a \subseteq [1,n']$ such that

(1) $|I_a| + \left|(n',n]_+\right| \leq z$, and
(2) $s_{i_a} = +1$, where $i_a = \min_{i \in I_a} i$,

the size of any z-detectable codes is not greater than

$$A_{s^n \circ q^n}^{(z,0)} \leq \prod_{i \in I_{\text{EQ}}} q_i,$$

where

$$I_{\text{EQ}} = [1,i_a)_+ \bigcup \left([i_a,n'] \setminus I_a\right).$$

Consider an interactive code. If its size is greater than $\prod_{i \in I_{\text{EQ}}} q_i$, due to the Pigeon Principle, there are two codewords c^n and $\bar{c}^n$ for distinct messages such that $c_{I_{\text{EQ}}} = \bar{c}_{I_{\text{EQ}}}$. Since $c_+^{i_a-1} = \bar{c}_+^{i_a-1}$, B can not distinct the difference between the two messages before the (i_a-1)th transmission. Therefore, $c^{i_a-1} = \bar{c}^{i_a-1}$, and

$$d_H\left(c^{n'}, \bar{c}^{n'}\right) = |I_a|.$$

Consequently, we have

$$d_H\left(c^{n'}, \bar{c}^{n'}\right) + \left|(n',n]_+\right| = |I_a| + \left|(n',n]_+\right| \leq z.$$

According to Theorem 1, the interactive code is not z-detectable. Thus, the size of all z-detectable codes is upper bounded by $\prod_{i \in I_{\text{EQ}}} q_i$. Then Theorem 3 is obtained by minimizing $\prod_{i \in I_{\text{EQ}}} q_i$ over all possible I_{EQ}. ∎

In this theorem, all transmissions are also partitioned into two phases. Only the first phase contributes the coding bound. Especially, there is also a transmission i_a that further divides the first phase into two parts. In the first part, all transmissions from A to B contribute to the coding bound; in the second phase, a limited number of transmissions contribute.

$$Q_{s_{i_1}^{i_2} \circ q_{i_1}^{i_2}}^{(z)} = \begin{cases} \prod_{i \in [i_1,i_2]_+} q_i, & z = 0 \\ \min_{i_a \in [i_1,i_2]_+} \prod_{i \in [i_1,i_a)_+} q_i \cdot \min_{I^* \subseteq (i_a,i_2]:|I^*|<z} \prod_{i \in (i_a,i_2] \setminus I^*} q_i, & z > 0 \end{cases} \tag{2}$$

B. Bounds on the Size of Error-Correcting Codes

Similar to the case for error-detecting codes, the size of error-correcting codes are upper bounded by the following theorem:

Theorem 4 (z-Correctable Bound): The maximum size of z-correctable codes is upper bounded by

$$A^{(z,z)}_{s^n \circ q^n} \leq \min Q^{(z'')}_{s^{n''} \circ q^{n''}} P^{(z')}_{s^{n'}_{n''+1} \circ q^{n'}_{n''+1}} \tag{3}$$

where the minimum is over all $n' \in [0, n], n'' \in [0, n']$ such that

$$2\left|(n'', n']_-\right| + \left|(n', n]_+\right| \leq 2z,$$

and all $z', z'' \in \left[0, z - \left|(n'', n']_-\right|\right]$ such that

$$z' + z'' \leq 2z - 2\left|(n'', n']_-\right| - \left|(n', n]_+\right|.$$

In the inequality (3), $Q^{(z)}_{s^{i_2}_{i_1} \circ q^{i_2}_{i_1}}$ is defined in (2) and

$$P^{(z)}_{s^{i_2}_{i_1} \circ q^{i_2}_{i_1}} = \min_{I^* \subseteq [i_1, i_2]_+ : |I^*| \leq z} \prod_{i \in [i_1, i_2]_+ \setminus I^*} q_i.$$

In this theorem, all transmissions are partitioned into three phases and the first two phases contribute to the coding bound. The first phase is further divided in a way that is identical to that in Theorem 3, resulting in the same $Q^{(z)}_{s^{i_2}_{i_1} \circ q^{i_2}_{i_1}}$ function. In the second phase, a limited number of transmissions from A to B contribute to the value of the bound.

C. Tightness of Bounds

Finding the exact value of the maximum size of code is an intractable problem. At least, this problem is more complex than finding the exact value of $A_2(n, d)$, the maximum size of binary block codes with length n and minimum distance d. However, the bounds in this paper are indeed tight in some cases.

For example, for 1-correctable codes in Fig. 2(b), letting $n' = n'' = 2$ and $z' = z'' = 0$, we have

$$\begin{aligned} Q^{(0)}_{s^2 \circ q^2} &= \prod_{i \in [1,2]_+} q_i = 4 \\ P^{(0)}_{s^2_3 \circ q^2_3} &= 1. \end{aligned}$$

Therefore, our error-correcting bound reduces to

$$A^{(1,1)}_{2,2,-4,2,-4,3} \leq 4.$$

At the same time, we can construct an interactive code of size 4 [1]. Therefore, the upper bound is tight in this case.

V. Conclusion

This paper considered the coding in finite interactive transmissions in the presence of omniscient adversary. When the codebook and the number of possible error transmissions satisfy some conditions, we can assert that the code is unable to detect or correct the errors. Furthermore, upper bounds on the size of code are derived. An example is provided to illustrate the tightness of the bound.

Acknowledgment

We thank Prof. Shenghao Yang, Jiang Zhu, and Britt Fugitt for their valuable suggestions on improving this paper.

This work was supported by the National Basic Research Program of China under Grant 2013CB329002, the National High Technology Research and Development Program of China under Grant 2014AA01A703, the National Science and Technology Major Project under Grant 2013ZX03004007, the Program for NCET in University under Grant NCET-13-0321, the International Science and Technology Cooperation Program under Grant 2012DFG12010, and the Tsinghua Research Funding under Grant 2010THZ03-2.

References

[1] Z. Xiao, Y. Li, M. Zhao, and J. Wang, "Interactive code to correct and detect omniscient Byzantine adversaries (extended version)," [Online] Available: https://www.dropbox.com/s/vwcuwr14dfb60ds/interactive.pdf.

[2] M. Braverman and A. Rao, "Towards coding for maximum errors in interactive communication," in *ACM Symp. Theory of Computing*, 2011, pp. 159–166.

[3] Z. Brakerski and Y. Kalai, "Efficient interactive coding against adversarial noise," in *IEEE Symp. Foundations of Computer Science*, 2012, pp. 160–166.

[4] M. Ghaffari and B. Haeupler, "Optimal error rates for interactive coding II: Efficiency and list decoding," 2013, [Online] Available: http://arxiv.org/abs/1312.1763.

[5] R. Yeung and N. Cai, "Network error correction, I: Basic concepts and upper bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 19–35, 2006.

[6] N. Cai and R. Yeung, "Network error correction, II: Lower bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 37–54, 2006.

[7] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, "Byzantine modification detection in multicast networks with random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2798–2803, Jun. 2008.

[8] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient network coding in the presence of Byzantine adversaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.

[9] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 209–218, Jan. 2008.

[10] S. Yang, R. Yeung, and C. Ngai, "Refined coding bounds and code constructions for coherent network error correction," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1409–1424, Mar. 2011.

[11] X. Guang, F. Fu, and Z. Zhang, "Construction of network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 1030–1047, Feb. 2013.

[12] S. Kim, T. Ho, M. Effros, and A. Avestimehr, "Network error correction with unequal link capacities," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1144–1164, Feb. 2011.

[13] T. Ho, S. Kim, Y. Yang, M. Effros, and S. Avestimehr, "On network error correction with limited feedback capacity," in *Inf. Theory Appl. Workshop*, 2011, pp. 1–3.

[14] Y. Yang, T. Ho, and W. Huang, "Network error correction with limited feedback capacity," 2013, [Online] Available: http://arxiv.org/abs/1312.3823.

[15] A. Brouwer, H. Hamalainen, P. Ostergard, and N. Sloane, "Bounds on mixed binary/ternary codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 140–161, Jan. 1998.

[16] S. Yang, R. Yeung, and Z. Zhang, "Weight properties of network codes," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 371–383, May 2008.