FINDING HOT-IPs IN NETWORK USING GROUP TESTING METHOD – A REVIEW

Huynh Nguyen Chinh¹, Nguyen Dinh Thuc² and Tan Hanh³
¹ Ho Chi Minh University of Technical Education
² University of Science, HCMC-VNU
³ Post and Telecommunications Institute of Technology

ABSTRACT
Group Testing is an applied mathematical theory proposed by Robert Dorfman and is applied in many different fields. This paper presents a survey on using group testing in detecting Hot-IPs in some cases, such as DoS attacks, worms, and viruses in network. Specifically, we focus on the solution of the optimization of the space, minimize the number of tests and time to run algorithm in order to make the efficiently in fast finding Hot-IPs in network.

KEYWORDS: Hot-IP, DoS, group testing, worm, virus, scan network….

1. INTRODUCTION
In network environment, servers are important role for serving clients. It is why, they are often the victims of attacks such as DoS attacks. Besides that, worms and viruses are also the big problems in network environment. There are two problems we focus on:

Problem 1:
In the case of DoS attacks [8] or scan network, attackers send a lot of traffic to a destination in a short time. Router receives and must process a lot of packets in network. Every packet has a destination IP address. If there are many packets pass through router that has the same IP destination, it may be a DoS attack.

Problem 2:
In the case of worms or virus [9][12], if there are many packets which have the same source IP addresses, this host may be infected by worms or virus, and they are scanning network. How to find items that appear with high rate in data stream in network? In this case, items are hosts in network.

Both problems above can be solved based on group testing method proposed by Robert Dorfman. [7]

In this paper, we consider using the group testing method to find all Hot-IPs in network; in particular, we focus on the solution of optimization of space, minimize number tests. This paper is organized as follows: Section II describes group testing. Section III describes some methods and optimize algorithms in order to fast finding Hot-IPs. Conclusion is giving in section IV.
2. GROUP TESTING

The basic idea of group testing is to identify the set of “positives” items from a large population of “items” using as few tests as possible. A test is subset of items, which returns positive if there is a positive in the subset.

There are two types of group testing [1]: adaptive group testing and non-adaptive group testing [1]. In adaptive testing, later stages are designed depending on the test results of the earlier stages. In non-adaptive testing, all tests must be specified without knowing the outcomes of other tests.

From the idea of group testing, we describe our problem as follows: assume that, there are N IP addresses (N hosts) that come to router, we need to design as few tests as possible such that shows all Hot-IPs in network. Hot-IP is IP which appear with high rate in a short time.

Assume that router receives N IP addresses, in which has at most d Hot-IPs. The problem is finding all Hot-IPs in network. This is called a combinatorial non-adaptive group testing [1][16].

Assume that we use t tests and construct an \((t \times N)\)-matrix \(M = (m_{ij})\), where:

\[
m_{ij} = \begin{cases} 1 & \text{if } IP_j \text{ belongs to group of } i^{th} \text{ test} \\ 0 & \text{otherwise} \end{cases}
\]

It is well known that if matrix M is d-disjunct, we can show all at most d Hot-IPs in network.

**Definition 1.** (d-disjunct matrix) [1] A matrix \(M\) with \(t\) rows and \(N\) columns is called d-disjunct matrix if and only if the union of \(d\) columns does not contain any other column.

We can construct a matrix \(t \times N\) d-disjunct with the minimum of tests [5][13]:

\[ t = O(d^2 \log N) \]

and time for decoding [5]

\[ poly(d).t \log^2 t + O(t^2) \]

Detail of the solutions for this problem will be presented in the later section.

3. FINDING HOT-IPs

This section presents some solutions for find Hot-IPs using group testing method.

3.1 Cormode-Muthukrishnan’s method [2]

Given a sequence of \(m\) items from \([n]\), an item is considered “hot” if it occurs more than \(m = d + 1\) times. Note that there can be at most \(d\) hot items.

Given the matrix \(M_{t \times N} = (m_{ij})\) d-disjunct, \(m_{ij} = 1\) if \(IP_j\) belong to group test \(i^{th}\).

Using counters \(c_1, c_2, \ldots, c_t, (c_i \in [t])\), when an item \(j \in [n]\) arrives, increment all of the counters \(c_i\) such that \(m_{ij} = 1\). From these counters, is defined a result vector \(r \in \{+, -\}^t\) as follows: Let \(r_i = +\) if \(c_i > m = (d + 1)\) and \(r_i = -\), otherwise. A test’s outcome is positive if and only if it contains a hot item.

Let:

- \(M\) be d-disjunct \(t \times N\) matrix
- \(C := (c_1, \ldots, c_t) \in N^t\)
- \(R := (r_1, \ldots, r_t) \in \{+, -\}^t\)
- \(IP \in [n]^*:\) sequence of IPs

We have:

- For \(i = 1\) to \(t\) do \(c_i := 0\)
- For each \(j \in IP\),
  - for \(i = 1\) to \(t\) do
    - if \(m_{ij} = 1\) then \(c_i := +\)
  - For \(i = 1\) to \(t\) do
    - If \(c_i > d\) then \(r_i := -\)
    - Else \(r_i := +\)

Determining Hot-IPs (naive algorithm):

Given:

- \(M\) be d-disjunct \(t \times N\) matrix
- \(R \in \{+, -\}^t\)
• Find Hot-IPs

\[
\begin{bmatrix}
m_{11} & m_{12} & \cdots & m_{1n} \\
m_{21} & m_{22} & \cdots & m_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
m_{n1} & m_{n2} & \cdots & m_{nn}
\end{bmatrix}
\begin{bmatrix}
r_1 \\
r_2 \\
\vdots \\
r_n
\end{bmatrix}
\]

With each \( r_i = - \) do

for i=1 to n do

if \((m_{ij}) = 1 \) Then

\( \text{IP:} = \text{IP} \setminus \{j\} \)

Return IP

The naive algorithm can be implemented in time \( O(tN) \). It’s very cost, therefore we need an efficiently decoding algorithm and optimize the space for matrix \( M \).

To optimize the space of matrix \( M \), using code concatenation to construct matrix \( M \) \( d \)-disjunct, in which \( outC \) is a Reed Solomon (RS) [15] code and \( inC \) is a \( qI \) (Identity code).

3.2 Code concatenation

Reed Solomon [6][15]

For a message \( m = (m_0, \ldots, m_{k-1}) \in F_q^k \), let \( P \) be a polynomial

\[ P_m(X) = m_0 + m_1X + \cdots + m_{k-1}X^{k-1} \]

in which the degree of \( P_m(X) \) is at most \( k-1 \). RS code \([n,k]_q\) with \( k \leq n \leq q \) is a mapping \( RS: F_q^k \rightarrow F_q^n \) is defined as follows. Let \( \{\alpha_1, \ldots, \alpha_n\} \) be any \( n \) distinct members of \( F_q \)

\[ RS(m) = (P_m(\alpha_1), \ldots, P_m(\alpha_n)) \]

It is well known that any polynomial of degree at most \( k-1 \) over \( F_q \) has at most \( k-1 \) roots. For any \( m \neq m' \) the Hamming distance between \( RS(m) \) and \( RS(m') \) is at least \( d = n-k+1 \). Therefore, RS code is a \([n,k,n-k+1]_q\) code.

**Code concatenation**

Let \( C_{out} \) be a \( (n_1,k_1) \) code with \( q = 2^{k_1} \) is an outer code, and \( C_{in} \) be a \( (n_2,k_2) \) binary code. Given \( n_1 \) arbitrary \( (n_2,k_2) \) code, denoted by \( C^i_{in} \). It means that \( \forall i \in [n_1], C^i_{in} \) is a mapping from \( F_{2}^{k_2} \) to \( F_{2}^{k_2} \).

A concatenation code \( C = C_{out} \circ (C^1_{in}, \ldots, C^n_{in}) \) is a \( (n_1n_2,k_1k_2) \) code defined as follows: given a message \( m \in F_{2}^{k_2} \) and let \((x_1, \ldots, x_{n_1}) = C_{out}(m) \), with \( x_i \in F_{2}^{k_2} \) then

\[ C_{out} \circ (C^1_{in}, \ldots, C^n_{in})(m) = (C^1_{in}(x_1), \ldots, C^n_{in}(x_{n_1})) \]

in which \( C \) is constructed by replacing each symbol of \( C_{out} \) by a codeword in \( C_{in} \).

For example:

\[
C_{out} : \begin{bmatrix}
0 & 1 & 2 & 0 & 1 & 2
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 2 & 1 & 2 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 2 & 2 & 0 & 1
\end{bmatrix}
\]

\[
C_{in} : \begin{bmatrix}
1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1
\end{bmatrix}
\]

\[
C_{out} \circ C_{in} : \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}
\]

**Constructing d-disjunct matrix [3][17]**

Let inner code be the identity code \( I_q : F_{2}^{k_2} \rightarrow F_{2}^{2k_2} \) and \( C_{out} \) is a \([n_1,k_1,n_1-k_1+1]_q\) code. Matrix \( M \) has size \( n_12^{k_2} \times 2^nk_2 \). Each column of \( M \) has
weight is \( n_i \). Distance of two columns is at least \( n_i - k_i + 1 \). Therefore, \( M \) is a \( d \)-disjunct with
\[
d = \left\lfloor \frac{n_i - 1}{k_i - 1} \right\rfloor > \left\lfloor \frac{n_i}{k_i} \right\rfloor.
\]
If we begin with \( N \) and \( d \), \( d \)-disjunct matrix \( t \times N \) is constructed with \( N = 2^{k_1 + k_2} \), \( n_i = d k_i \). Therefore we have \( t = n_i 2^{k_2} = d k_i 2^{k_2} \).

With the code concatenation described above, we have lower space for matrix \( M \).

There are two problems when using this method: [5]
(i) How to decode?
(ii) \( t = O(d^2 \log^2 N) \), which stills large compared to the random matrix construction \( t = O(d^2 \log N) \).

3.3 An efficiently decoding: [5]
Let result vector \( r = \{r_1, \ldots, r_n\} \), \( r_i \in \{0, 1\}^q \).
Every \( r_i \) is a characteristic vector of \( S_i \subset [q] \).
There are at most \( d \) positive columns. We have \( | S_i | \leq d, \forall i \).

Consider a positive column \( c = (c_1, \ldots, c_n) \), we have \( c_i \in S_i \). Note that \( c_i \) is an identity vector.
Guruswami-Sudan [4] shows that exist an algorithm that shows all positive columns in time \( \text{poly}(q, d) = \text{poly}(d, \log N) \).

Main idea of Indyk-Ngo-Rudra [5]: constructing \( (d,d) \)-list disjunct matrices \( M_i \) such that inner code with size \( n_i \times q \). If outer code is a \( d \)-disjunct matrix \( M \), we can decode \( M \) in time \( n_i O(n_i 2^{k_2}) + O(n_i 2^{k_2} / k_i) \). Parameters are selected so that time to run the algorithm is \( \text{poly}(d, t) \), and \( t = O(d^2 \log N) \).

\( (d,d) \)-list disjunct matrix is defined as follows:

**Definition 2:** [5][13] \( (d,d) \)-list disjunct matrix

We call a \( t \times N \) boolean matrix \( M \) \( (d,l) \)-list disjunct if for any two disjoint subsets \( S \) and \( T \) of \( [n] \) where \( |S| \leq d \), \( |T| \geq l \), there exists a row of \( M \) in which some column in \( S \) has a 1 but all columns in \( T \) have 0s.

**Construct \( (d,d) \)-list disjunct matrix**
There is important theorem as follow:[5]

**Theorem:** Let \( d, k_1, k_2 \) be any given positive integers such that \( 10d k_1 \leq 2^{k_2} \).
Define \( n_1 = 10d k_1, n_2 = 480d k_2, t = n_1 n_2 \), and \( N = 2^{k_1 + k_2} \). Note that \( n_1 \leq 2^{k_2} \) and \( t = O(d^2 \log N) \).

Let \( C_{out} \) be any RS code
\[
(n_1, k_1)_{d} = \frac{n_1}{10d}, n_1(1 - \frac{1}{10d})_{d}.
\]
Then, there exist inner codes \( C_{in}^{(1)}, \ldots, C_{in}^{(n)} \), each of which is an \( (n_2, k_2) \) code such that the following hold:

(a) Let \( C^* = C_{out} \circ (C_{in}^{(1)}, \ldots, C_{in}^{(n)}) \), then \( M_C \) is \( t \times N \) matrix that is \( d \)-disjunct

(b) For every \( i \in [n_1] \), \( M_{C_i} \) is a \( (d,d) \)-list disjunct matrix

**Corollary:** [5] Let \( N > d \geq 1 \) be any given integers. There exists a \( t \times N \) \( d \)-disjunct matrix with \( t = O(d^2 \log N) \) that can be decoded in time \( \text{poly}(d) t \log^2 t + O(t^2) \).

From the theorem and corollary, we can optimize the space and time to run decoding algorithm. Therefore, applying the group testing to find Hot-IPs in network is a efficiently solution.

4 CONCLUSION
Applying group testing in finding Hot-IPs in network is an efficiently solution. It helps to detect anomaly in network such that DoS attacks, scan network, virus, worm…This helps administrators recognize anomalies in network, and they can control firewall to block traffics from them or
redirect to honeypot [10][11].

5 REFERENCES


Contact
Huynh Nguyen Chinh –
Tel: +84 983929445

378
Email: chinhhm@fit.hcmute.edu.vn
Nguyen Dinh Thuc – Tel: +84 903339944
Email: thucnd@fit.hcmus.edu.vn
Tan Hanh – Tel: +84 907106251
Email: tanhanh@ptithcm.edu.vn