On Achieving All-Optical Failure Restoration via Monitoring Trails

János Tapolcai*, Pin-Han Ho†, Péter Babarczi*, Lajos Rónyai‡
* MTA-BME Future Internet Research Group, High-Speed Networks Laboratory (HSN*Lab*),
Dept. of Telecommunications and Media Informatics, Budapest University of Technology, {tapolcai, babarczi}@tmit.bme.hu
† Dept. of Electrical and Computer Engineering, University of Waterloo, Canada, p4ho@uwaterloo.ca
‡ Computer and Automation Research Institute Hungarian Academy of Sciences and BME, ronyai@sztaki.hu

***Abstract*—The paper investigates a novel monitoring trail (m-trail) scenario that can enable any shared protection scheme for achieving all-optical and ultra-fast failure restoration. Given a set of working (W-LPs) and protection (P-LPs) lightpaths, we firstly define the *neighborhood* of a node, which is a set of links whose failure states should be known to the node in restoration of the corresponding W-LPs. A set of m-trails is routed such that each node can localize any failure in its neighborhood according to the ON-OFF status of the traversing m-trails. Bound analysis is performed on the minimum bandwidth required for the m-trails. Extensive simulation is conducted to verify the proposed scheme.**

I. Introduction

It is considered the best strategy to locally restore an optical layer failure (e.g., fiber cut) in the optical domain within as short time as possible before the failure maliciously affects the operation of upper layer protocols such as IP or TCP. Thus, an optical layer failure should be handled without relying on any electronic signaling protocol no matter the network optical domain has central or distributed control. Currently, only dedicated protection (i.e., 1+1) and pre-configured Cycle (p-Cycle) based approaches can achieve 50ms or shorter restoration time in mesh networks due to their simplicity and pre-configured spare capacity, but at the expense of 70% or higher redundancy [1]. Note that failures are rare events, and allocating a significant amount of redundancy for failure recovery is not considered economically reasonable.

Monitoring trail (m-trail) has been proposed as an effective approach to enable all-optical and ultra-fast failure restoration in the network optical domain. An m-trail is implemented as a pair of lightpaths along a common physical route in opposite directions for sensing/monitoring the health of the links along the route. Thus, each node traversed by an m-trail will sense loss of light (LOL) via lambda monitoring when a failure hits upon any link along the m-trail. By properly allocating a set of m-trails in the network, an all-optical monitoring system is formed, so that every node can unambiguously identify the failed link by only inspecting the m-trails traversing through the node. This is also referred to as the network-wide local unambiguous failure localization (NWL-UFL) scenario [2]. In [3] NWL-UFL was taken as a building block for constructing the first all-optical failure restoration framework, which enables a general shared protection scheme to be performed in an all-optical and signaling-free fashion.

Although theoretically sound, [2], [3] assumed that each node is able to unambiguously identify all possible failures. Thus a node will monitor a remote link even if the node does not need to respond to the link failure, resulting in unnecessary monitoring resource consumption, high computation complexity, and very lengthy m-trails. In this paper, we investigate an on-demand m-trail allocation paradigm that can enable a general shared protection scheme to perform signaling-free failure restoration as in 1+1 and p-Cycle. Firstly, we define the *neighborhood* of a node as a set of links whose failures must be unambiguously localized by the node; thus each node only localizes the link failures in its neighborhood. Specifically, the neighborhood of a node should contain all the links along the W-LPs whose corresponding P-LPs traverse through the node. On the other hand, all the nodes traversed by a P-LP should be able to localize the link failure for which the P-LP is used to restore. Secondly, the spare capacity by those P-LPs can be reused to support the m-trails in order to achieve better capacity efficiency. Thirdly, a node can monitor both traversing m-trails and W-LPs for failure status acquisition, which is referred to as *out-of-band* and *in-band* monitoring, respectively.

The rest of the paper is organized as follows. Section II provides the system model, including the targeted m-trail scenario and problem formulation. Section III is on a bound analysis of the proposed problem. A heuristic solution to the proposed problem is given in Section IV, while Section V presents the simulation results.

II. System Model

A. Proposed M-Trail Scenario

The inputs are the W-LPs and P-LPs, the *neighborhood* of each node is given, and a node is said to meet the *neighborhood failure localization* (NFL) requirement if it can localize a link failure in its neighborhood. The proposed m-trail scenario enables each node to localize the failed link in its neighborhood based on the ON-OFF status of a subset of the m-trails and/or W-LPs that pass through the node.

Fig. 1 shows an example of the proposed m-trail scenario in a topology with 4 nodes and two W-LPs denoted as W_1

J.T. was supported by the project TÁMOP - 4.2.2.B-10/1–2010-0009 . R.L. was supported by the Hungarian Research Fund (OTKA grants NK 105645, K77476, and K77778). P. -H. Ho was supported by National Science and Engineering Research Council (NSERC), Canada.

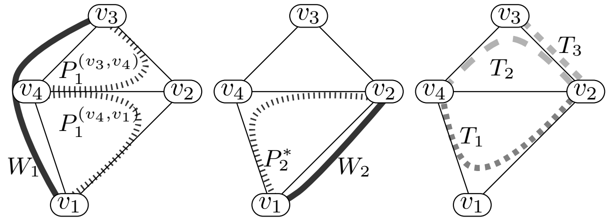
and W_2, each being provisioned with two physical lightpaths on the same route in opposite directions. W_1 is protected by two P-LPs, namely $P_1^{(v_3,v_4)}$ for link failure (v_3,v_4), and $P_1^{(v_4,v_1)}$ for (v_4,v_1) as in Fig. 1(a), while W_2 is protected by a single P-LP denoted as P_2^* as in Fig. 1(b). To ensure signaling-free restoration for W_1 and W_2, v_1 should be able to unambiguously identify the failure of (v_4,v_1) and (v_2,v_1), such that W_1 (or W_2) can be switched over to $P_1^{(v_4,v_1)}$ (or P_2^*) when (v_4,v_1) (or (v_2,v_1)) fails. Thus the neighborhood of v_1 must contain the two links (v_4,v_1) and (v_2,v_1). On the other hand, since v_2 is traversed by all the three P-LPs (i.e., $P_1^{(v_3,v_4)}$, $P_1^{(v_4,v_1)}$, and P_2^*), it needs to react to any failure upon W_1 or W_2. Therefore, the neighborhood of v_2 should contain (v_3,v_4), (v_4,v_1) and (v_2,v_1). Similarly, we can define the neighborhood of v_3 as (v_3,v_4), etc.

To achieve the NFL requirement according to the above nodal neighborhoods, three m-trails T_1, T_2 and T_3 are needed as shown in Fig. 1(c), by which the alarm code table (ACT) for each node is formed as shown in Fig. 1(d)-(g). Each row of an ACT on top of the separator corresponds to a failure state within its the neighborhood, and the rows below the separator are for the alarm codes seen at the node due to a link failure outside the neighborhood. All these codes in an ACT should be unique. For example, v_1 keeps the ACT as in Fig. 1(f) by observing the ON-OFF status of T_1 and W_2, so as to uniquely identify the failure on (v_4,v_1) or (v_2,v_1) in its neighborhood. If v_1 finds that T_1 becomes unexpectedly off while W_2 is still on, an alarm code $[1,0]$ is obtained; so the node will consider link (v_4,v_1) as failed by matching the first row of its ACT and be ready to switch W_1 over to $P_1^{(v_4,v_1)}$. Meanwhile and in parallel, v_2 and v_4 will be able to identify the failure of (v_4,v_1) by matching the second row in their ACTs as in 1(g) and (d), respectively, and instantly configure their OXCs to support $P_1^{(v_4,v_1)}$. Thus W_1 can be restored an all-optical and deterministic fashion upon the failure of (v_4,v_1).

B. Problem Definition

It is critical to route a set of m-trails such that each node meets the NFL requirement. Given an undirected graph $G = (V,E)$ with node set V and link set E, where the number of nodes is denoted by $n = |V|$ and the number of links by $m = |E|$. Given a set of W-LPs, denoted by $\mathcal{W}$, each of which can be in-band monitored by the traversed nodes by it for failure status acquisition. If a working path $W_i \in \mathcal{W}$ that traverses e is interrupted due to the failure of e, the corresponding protection path $P_{i,e}$ should be activated at the switching node for restoration. Let the *neighborhood* of node v be denoted by E_v, which is a set of links whose failure states should be unambiguously identified by v, formally

(R1): a node v has to failure-localize link e if and only if node v is involved in the restoration process of the link failure e according to the current traffic distribution; i.e., v is along the P-LP protecting the link failure e[1].



(a) Connection 1 (b) Connection 2 (c) M-trails T_1,T_2,T_3

Link	T_1	T_2	W_1	switch to
(v_3,v_4)	0	1	1	$P_1^{v_3,v_4}$
(v_4,v_1)	1	0	1	$P_1^{v_4,v_1}$
(v_2,v_1)	1	0	0	P_2^*
(v_3,v_2)	0	1	0	
(v_2,v_4)	0	0	0	

(d) ACT at v_4

Link	T_2	T_3	W_1	switch to
(v_3,v_4)	1	0	1	$P_1^{v_3,v_4}$
(v_2,v_1)	0	0	0	
(v_3,v_2)	1	1	0	
(v_4,v_1)	0	0	0	
(v_2,v_4)	0	0	0	

(e) ACT at v_3

Link	T_1	W_2	switch to
(v_4,v_1)	1	0	$P_1^{v_4,v_1}$
(v_2,v_1)	1	1	P_2^*
(v_3,v_2)	0	0	
(v_3,v_4)	0	0	
(v_2,v_4)	0	0	

(f) ACT at v_1

Link	T_1	T_2	T_3	W_2	switch to
(v_3,v_4)	0	1	0	0	$P_1^{v_3,v_4}$
(v_4,v_1)	1	0	0	0	$P_1^{v_4,v_1}$
(v_2,v_1)	1	0	0	1	P_2^*
(v_3,v_2)	0	1	1	0	
(v_2,v_4)	0	0	0	0	

(g) ACT at v_2

Fig. 1. An illustrative example for the proposed m-trail scenario.

Conversely, let *visibility region* of e be denoted by V_e, as a set of nodes each being able to unambiguously identify the failure of e.

Our target is to establish a set of m-trails to meet the following two requirements (R2) and (R3) as follows.

(R2): each m-trail is a loopless path of G at most one hop longer than the shortest path between its end nodes.

The set of m-trails is denoted by $\mathcal{T} = \{T_1,\ldots,T_b\}$ where b is the number of m-trails. The objective is to minimize:

$$\|\mathcal{T}\| = \sum_{i=1}^{b} |T_i|, \tag{1}$$

where $|T_i|$ is the number of links in m-trail T_i. We expect that each node $v \in V$ can achieve NFL according to the ON-OFF status of m-trails and W-LPs in $\mathcal{T}^v$ - the subset of $\mathcal{T} \cup \mathcal{W}$ containing the m-trails and W-LPs passing through v. Let $\underline{A}^v$ denote the *alarm code table* (ACT) at node v, where the i^{th} bit of alarm code for link e at v will be denoted by $a_{e,i}^v$ for $1 \le i \le |\mathcal{T}^v|$, where $|\mathcal{T}^v|$ is the number of m-trails in $\mathcal{T}^v$. We have $a_{e,i}^v = 1$, if the i^{th} m-trail passing through node v has link e, and 0 otherwise.

To achieve NFL at node v, (R3) should hold:

(R3): every link e in neighborhood E_v has a unique non-zero alarm code seen at v denoted by A_e^v, and meanwhile different from all the possible link codes that v can see outside the neighborhood.

III. Bound Analysis

This section presents our bound analysis for the coverlength in the proposed m-trail allocation problem for $\mathcal{W} = \emptyset$. We will first consider the lower bound on a generalized version of Combinatorial Group Testing (CGT) and then apply them to

[1] it either the switching, intermediate, or merging node of a P-LP which protects link e along an active W-LP.

the NFL requirement at each node. The key idea is to define a special cost function for the m-trails at each node such that the lower bound can be summed up to get a lower bound on the total coverlength.

A. General Lower Bound for CGT

Let us consider a non-adaptive CGT problem where the goal is to find one faulty item among a set of items with group tests, where each group test is on a set of items and has two outcomes: the test contains a faulty item or not. Note that the problem at each node v is a special version of CGT, where the tests are the m-trails passing through v, and the items are the links and we have three additional constraints (R1), (R2), and (R3). It is clear that a valid solution at node v is a valid CGT solution over the links in its neighborhood.

Next, let us formalize the CGT problem with a cost function on each test. The cost of test T_i depends on its size according to a given cost function ω. The input of the CGT problem is a set of items denoted by $E = \{e_1, \dots, e_m\}$ and a cost function ω, where $m = |E|$ is the number of items. The goal is to establish a set of b group tests, denoted by $T_1, \dots, T_b$, such that a single faulty item can be unambiguously identified according to the outcomes of the group tests. Each test has a cost defined as follows[2]

Definition 1: The cost of test T_i with $t_i = |T_i|$ is $\omega(t_i)$, where function ω has the following properties:

(i) $\omega(1) = 1$, means testing one element has a unit cost.

(ii) $\omega(x+1) \geq \omega(x)$ for every positive integer $1 \leq x \leq m-1$. Testing a larger group cannot decrease the cost.

(iii) $\frac{\omega(x)}{x} \geq \frac{\omega(x+1)}{x+1}$ whenever $1 \leq x < m$.

The goal is to identify the faulty item with minimum cost:

$$\text{Minimize } \Omega = \sum_{i=1}^{b} \omega(t_i) \tag{2}$$

Theorem 1: Suppose there are $m > 1$ items and assume (i)-(iii) holds for the cost function ω. Then for the cost of finding precisely one faulty item with group tests is at least

$$\Omega \geq \min_{1 \leq x \leq \frac{m}{2}} \omega(x) \left(\log_2 x + \frac{m}{x} - 1 \right). \tag{3}$$

The proof is relegated to the Appendix.

B. Lower Bound of the Problem

Let $r(T_i)$ denote the number of nodes the m-trail T_i passes through. These nodes are aware of the ON-OFF status of T_i. A trivial upper bound on it is $r(T_i) \leq |T_i| + 1$ because T_i is a connected linkset.

We divide the cost of each m-trail equally among the nodes it traverses, and represent the cost in a matrix Ω which has n rows and b columns, where

$$\omega_{v,i} = \begin{cases} \frac{|T_i|}{r(T_i)} & \text{the } i^{\text{th}} \text{ m-trail traverses node } v, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The size of T_i can be expressed as

$$\sum_{v=1}^{n} \omega_{v,i} = \sum_{v \in T_i} \frac{|T_i|}{r(T_i)} = |T_i|. \tag{5}$$

Thus we have

$$\|\mathcal{T}\| = \sum_{i=1}^{b} |T_i| = \sum_{i=1}^{b} \sum_{v=1}^{n} \omega_{v,i} = \sum_{v=1}^{n} \left(\sum_{i=1}^{b} \omega_{v,i} \right) = \sum_{v=1}^{n} \Omega_v \ , \tag{6}$$

where

$$\Omega_v = \sum_{i=1}^{b} \omega_{v,i} = \sum_{i | v \in T_i} \omega_{v,i} \ , \tag{7}$$

because $\omega_{v,i} = 0$ if the i^{th} m-trail does not traverse node v. Next we give a lower bound on $\omega_{v,i}$ as a function of size T_i, denoted by $\omega(|T_i|)$, if an m-trail traverses v as follows

$$\omega_{v,i} \geq \frac{\omega(|T_i|)}{2} \ , \text{ if } v \in T_i \ , \tag{8}$$

where

$$\omega(|T_i|) = \begin{cases} \dfrac{2|T_i|}{1+|T_i|} & \text{if } |T_i| \leq n-1 \ , \quad (9a) \\ \dfrac{2|T_i|}{n} & \text{otherwise.} \quad (9b) \end{cases}$$

We have (8) because for $v \in T_i$ $2\omega_{v,i} = \frac{2|T_i|}{r(T_i)} \geq \frac{2|T_i|}{|T_i|+1}$, and also $r(T_i) \leq n$.

To give a lower bound on Ω_v, we may consider this subproblem as a general version of a CGT problem where the cost of a group test T_i is $\omega(|T_i|)$. It is a function of its size and meets the requirements in Definition 1. In this case, the cost function (9) is defined separately on two intervals, (9a) is a reciprocal function and (9b) is linear.

Theorem 2: The total cover length for a solution is at least

$$\|\mathcal{T}\| \geq \sum_{v \in V} \left(1 - \frac{2}{m_v + 2} \right) \log_2(m_v) \tag{10}$$

if $n - 1 \geq \frac{m_v}{2}$ for all $v \in V$. Here m_v denotes the number of edges in node v's neighborhood.

Proof: By assumption, $n - 1 \geq \frac{m_v}{2}$, thus we need to consider $x = |T_i| \leq \frac{m_v}{2} \leq n - 1$ only. Putting together the lower bound on the cost in (6), (8) and applying Theorem 1 on each node we get a lower bound on Ω_v

$$\Omega_v \geq \min_{1 \leq x \leq \frac{m_v}{2}} \frac{2x}{1+x} \left(\log_2 x + \frac{m_v}{x} - 1 \right), \tag{11}$$

where inside the $\min$ there is a decreasing function of x on $\left[1, \frac{m_v}{2}\right]$ as proved in Lemma 2 in [3]. Thus, it leads to

$$\begin{aligned} \Omega_v \geq \frac{2\frac{m_v}{2}}{\frac{m_v}{2}+1} \left(\log_2 \left(\frac{m_v}{2} \right) + \frac{m_v}{\frac{m_v}{2}} - 1 \right) &= \\ = \frac{2m_v}{m_v+2} \left(\log_2(m_v) - 1 + 2 - 1 \right) &= \\ = \left(2 - \frac{4}{m_v+2} \right) \log_2(m_v). & \quad (12) \end{aligned}$$

Putting it together with (6), we get (10). ■

[2] In the traditional CGT problem we have $\omega(t) = 1$ for every t.

IV. Proposed Heuristic

A simple yet effective heuristic to solve the proposed problem defined in Section II-B is implemented. The basic idea is to successively and incrementally construct the ACT at each node such that every link code is unique among any other link codes visible by the node.

A detailed description of the proposed heuristic is given in Algorithm 1 and is explained step by step as follows.

In Step (2) an initial solution is taken using the W-LPs $\mathcal{W}$ and single-hop m-trails for every link. In Step (3) each node $v \in V$ is considered one after the other to meet the NFL requirement such that each link code in the neighborhood E_v is unique. Specifically, E_v is loaded with W-LPs in Step (4), and the current ACT $\underline{\underline{A}}^v$ is constructed based on the m-trails traversing through node v in Step (5).

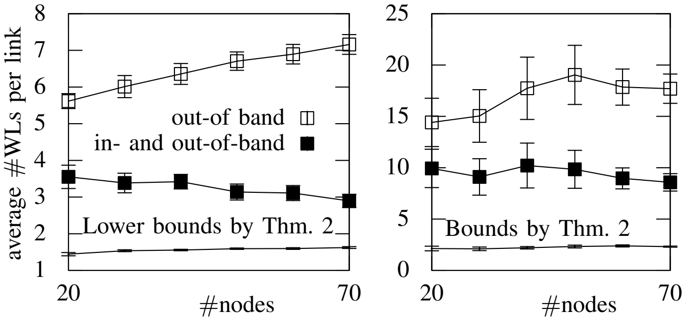
Then, the heuristic enters the loop in Step (6)-(7) for each node v, by checking whether links e_1 and e_2, where $e_1 \in E_v$ and $e_2 \in E$, have the same alarm code seen at v or not. If yes in Step (8), we place an m-trail starting from v and traversing either e_1 or e_2, but not both. To make this information local at node v, we use Dijkstra's shortest path finding algorithm in Step (9) between v and the two adjacent nodes of the corresponding link, and select the one with the shortest distance in Step (10). Finally, we add the shortest possible path to $\mathcal{T}$ in Step (12) or Step (14), and refresh the ACT of v in Step (15).

Algorithm 1:

```
   Input: $G = (V, E)$
   Result: $\mathcal{T}$ set of m-trails
1  begin
2    Use a single-hop m-trail for each link and W-LPs $\mathcal{W}$ as
     an initial guess;
3    for $v \in V$ do
4      Load the set of neighborhood links $E_v \subseteq E$;
5      Construct current ACT $\underline{\underline{A}}^v$ at node $v$;
6      for $e_1 = (u_1, w_1) \in E_v$ do
7        for $e_2 = (u_2, w_2) \in E$ do
8          if $e_1 \neq e_2 \wedge A^v_{e_1} = A^v_{e_2}$ then
9            Use Dijkstra's algorithm to get the shortest paths
             from $v$ to $\{u_1, w_1, u_2, w_2\}$;
10           Set $\mathcal{P}_1$ and $\mathcal{P}_2$ to the shortest path to $\{u_1, w_1\}$
             and $\{u_2, w_2\}$, respectively;
11           if $|\mathcal{P}_1| \leq |\mathcal{P}_2|$ then
12             Add m-trail $\forall e \in \mathcal{P}_1 \cup e_1$ to $\mathcal{T}$;
13           else
14             Add m-trail $\forall e \in \mathcal{P}_2 \cup e_2$ to $\mathcal{T}$;
15           Refresh $\underline{\underline{A}}^v$;
```

V. Experimental results

A set of experiments was conducted to verify the proposed m-trail scenario. Two classes of random planar graphs were generated: one for dense and the other for sparse networks, with typical vertex number 4 and 7 of the inner faces, and an average nodal degree 4.0 and 2.8, respectively. 30% of all node pairs are randomly selected for being loaded, where a pair of W-LPs are shortest-path routed for each loaded node pair on the same route in both directions, which is protected by a set of P-LPs shortest and diversely routed from the failed link of the W-LP.



(a) Dense networks

(b) Sparse networks

Fig. 2. Average number of WLs per link.

We first observe that the size of neighborhoods grow very mildly as the network size increases and never exceeds 15 links, compared with NWL-UFL where all the links are contained in the neighborhood of each node.

Fig. 2 shows the average WLs per link with and without in-band monitoring. Firstly we have seen that the number of WLs per link scales very well when the network sizes increase. We can see the price we pay for the complete independence between the monitoring and data planes. The results derived in Theorem 2 are also sketched. It is seen that some gaps exist between the derived lower bound, mostly due to the fact that the analysis was purely conducted based on CGT theory and can only modestly capture the additional complexity of the proposed problem. However, we claim that the analytical results not only contribute to the general CGT topics, but serve as a design guideline for the proposed solutions.

VI. conclusions

The paper explored the possibility of signaling-free fault management in all-optical mesh networks that can achieve on-demand monitoring resource allocation, near shortest m-trails, and both out-of-band and in-band monitoring at each node. Bound analysis was conducted via novel general group testing results which was applied to the proposed problem. Subsequently we suggested a simple heuristic that can yield fast yet effective solution. Simulation results showed that the proposed m-trail scenario can achieve superb capacity efficiency with nearly constant monitoring resource consumption when the network size grows.

References

[1] W. Grover, J. Doucette, M. Clouqueur, D. Leung, and D. Stamatelakis, "New options and insights for survivable transport networks," *IEEE Commun. Mag.*, vol. 40, no. 1, pp. 34–41, Jan. 2002.

[2] J. Tapolcai, P.-H. Ho, L. Rónyai, and B. Wu, "Network-wide local unambiguous failure localization (NWL-UFL) via monitoring trails," *IEEE/ACM Transactions on Networking*, 2012.

[3] J. Tapolcai, P.-H. Ho, P. Babarczi, and L. Rónyai, "On signaling-free failure dependent restoration in all-optical mesh networks," *Technical Report*, 2012.

Appendix

Suppose there are $m > 1$ items and assume (i)-(iii) holds for the cost function ω. Then for the cost of finding precisely one faulty item with group tests is at least

$$\Omega \geq \min_{1 \leq x \leq \frac{m}{2}} \omega(x) \left(\log_2 x + \frac{m}{x} - 1 \right) \ . \tag{13}$$

Proof: Let us sort the tests in descending size, so that T_1 has the largest number of items while T_b has the least: we assume that

$$t_1 \geq t_2 \geq \cdots \geq t_b$$

where $t_i = |T_i|$ denotes the number of items in test T_i.

Also, we may assume that $t_i \leq \frac{m}{2}$ for every i. Indeed, a test set T_i with $|T_i| \geq \frac{m}{2}$ can be replaced by its complementary set $E \backslash T_i$. The resulting test collection still remains separating if the original one distinguished all single faults.

We build up the $b \times m$ matrix of alarm codes, by adding the rows one-by-one, and in each step we count the number of different columns in the matrix. Let f_i denote the number of different columns when the matrix has i rows, i.e. tests $T_1, \ldots, T_i$ are present, the others are not. For convenience we set $f_0 = 1$. Adding a row the number of different columns cannot decrease, thus $f_{i-1} \leq f_i$ for $i = 1, \ldots, b$. As we have a separating system, all the m columns will be different when the last row is added, giving that $f_b = m$.

When we add T_i, the number of different columns is at most doubled, hence $f_i \leq 2 f_{i-1}$, or

$$\log_2(f_i) - \log_2(f_{i-1}) \leq 1 \tag{14}$$

for $i = 1, \ldots, b$.

Similarly, by adding test T_i to the collection $T_1, \ldots, T_{i-1}$ can increase the number of different columns in the matrix by at most t_i, giving $f_i \leq f_{i-1} + t_i$, or

$$\frac{f_i - f_{i-1}}{t_i} \leq 1 \tag{15}$$

for $i = 1, \ldots, b$.

Now fix an integer k with $1 \leq k < b$. We have

$$\Omega = \sum_{i=1}^{b} \omega(t_i) \geq \sum_{i=1}^{k} \omega(t_i) \left(\log_2(f_i) - \log_2(f_{i-1}) \right) + \\ + \sum_{i=k+1}^{b} \omega(t_i) \left(\frac{f_i - f_{i-1}}{t_i} \right) . \tag{16}$$

In the first sum we used (14), and (15) in the second.

The sequence $\omega(t_i)$ is nonincreasing for $i = 1, \ldots, b$ by (i) and our numbering of the tests, hence

$$\sum_{i=1}^{k} \omega(t_i)(\log_2(f_i) - \log_2(f_{i-1})) \geq \\ \geq \sum_{i=1}^{k} \omega(t_k)(\log_2(f_i) - \log_2(f_{i-1})) = \omega(t_k) \log_2(f_k). \tag{17}$$

Similarly, the sequence $\frac{\omega(t_i)}{t_i}$ is nondecreasing because of (iii) and our numbering of the tests, giving that

$$\sum_{i=k+1}^{b} \omega(t_i) \left(\frac{f_i - f_{i-1}}{t_i} \right) \geq \\ \geq \sum_{i=k+1}^{b} \omega(t_{k+1}) \left(\frac{f_i - f_{i-1}}{t_{k+1}} \right) = \frac{\omega(t_{k+1})}{t_{k+1}} (m - f_k). \tag{18}$$

By substituting (17) and (18) into (16), we have

$$\Omega \geq \omega(t_k) \log_2(f_k) + \frac{\omega(t_{k+1})}{t_{k+1}} (m - f_k) \geq \\ \geq \omega(t_k) \left(\log_2(f_k) + \frac{m - f_k}{t_k} \right) . \tag{19}$$

This inequality is valid for any k with $1 \leq k < b$. Let we set now k to be the first index j for which $t_j \leq f_j$. Such index clearly exists and $k < b$ because $f_{b-1} \geq \frac{m}{2}$, while $t_i \leq \frac{m}{2}$ for every i. We need to consider two cases:

(1) If $f_{k-1} \leq t_k$, then we start from

$$\Omega \geq \omega(t_k) \left(\log_2(f_k) + \frac{m - f_k}{t_k} \right) .$$

Note that $t_k \leq f_k \leq 2 f_{k-1} \leq 2 t_k$, hence for δ defined by $f_k = t_k + \delta$ we have $0 \leq \delta \leq t_k$. Moreover,

$$\Omega \geq \omega(t_k) \left(\log_2(t_k + \delta) + \frac{m - t_k - \delta}{t_k} \right) = \\ = \omega(t_k) \left(\log_2(t_k + \delta) - \frac{\delta}{t_k} + \frac{m}{t_k} - 1 \right) . \tag{20}$$

On the interval $0 \leq x \leq 1$ we have the inequality $x \leq \log_2(1 + x)$. We apply this for $x = \frac{\delta}{t_k}$. Note that $0 \leq \delta \leq t_k$ implies that $0 \leq x \leq 1$. We obtain the following inequality

$$\log_2(t_k + \delta) - \frac{\delta}{t_k} \geq \log_2(t_k + \delta) - \log_2 \left(1 + \frac{\delta}{t_k} \right) = \\ = \log_2 \left(\frac{t_k + \delta}{1 + \frac{\delta}{t_k}} \right) = \log_2 \left(\frac{t_k + \delta}{\frac{t_k + \delta}{t_k}} \right) = \log_2(t_k). \tag{21}$$

Substituting (21) into (20) we get

$$\Omega \geq \omega(t_k) \left(\log_2(t_k) + \frac{m}{t_k} - 1 \right) .$$

(2) If $t_k < f_{k-1}$, then $k > 1$ because $f_0 = 1$ by definition. Thus $f_{k-1} < t_{k-1}$ and based on the first inequality of (19) we have

$$\Omega \geq \omega(t_{k-1}) \log_2(f_{k-1}) + \frac{\omega(t_k)}{t_k} (m - f_{k-1}).$$

Since $\frac{\omega(t)}{t}$ is a nonincreasing function of t, we have

$$\Omega \geq \omega(f_{k-1}) \log_2(f_{k-1}) + \frac{\omega(f_{k-1})}{f_{k-1}} (m - f_{k-1}) \geq \\ \geq \omega(f_{k-1}) \left(\log_2(f_{k-1}) + \frac{m}{f_{k-1}} - 1 \right) . \tag{22}$$

In both cases there is an integer x in the interval $[1, \frac{m}{2}]$ such that

$$\Omega \geq \omega(x) \left(\log_2(x) + \frac{m}{x} - 1 \right) .$$

This is because $f_{k-1} < t_{k-1} \leq \frac{m}{2}$ and $t_k \leq \frac{m}{2}$ hold. This proves the theorem. ■