Celal Bayar University Journal of Science
Volume 18, Issue 4, 2022, p 349-353
Doi: 10.18466/cbayarfbe.970002

Celal Bayar University Journal of Science

Encoding Vertices in Rectangular Grid Graphs with Eliminating Errors

Gökçe Çaylak Kayaturan[1*]

[1] Ordu Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümü, Ordu, Türkiye
* gokcekayaturan@odu.edu.tr
*Orcid: 0000-0001-6823-7686

Received: 12 July 2021
Accepted: 18 November 2022
DOI: 10.18466/cbayarfbe.970002

Abstract

We assume that an undirected rectangular grid graph models a message transportation between computers in a computer network. We consider a routing scenario in this network and label each vertex with a binary string by using a particular approach. According to routing scenario we consider , the messages are sent with a header of a message between two distinct computers in the graph. We present a way to encode the paths that represent the routes in the network. We aim that these codes prevent errors which distruct the message transportation in the network and may cause network traffic.

Keywords: Computer Network, Routing, Rectangular grid, Undirected Graph.

1. Introduction

In this paper, we assume undirected graphs model computer networks. Chosing a mathematical graph as network model has been studied in literature [9,10]. The rectangular grid graph has been chosen for the network model (see Figure 1). A rectangular grid is defined with sets of vertices and edges, where the adjacent vertices (i,j) and (p,q) are linked by an edge if $i = p$ and $j = q \pm 1$ or $i = p \pm 1$ and $j = q$.

In this research, we inspired an encoding method invented by B.Bloom from [1] that is called as Bloom filter. A Bloom filter is a way to compress the data. This case of the Bloom filter offers the users to save space. The one can control if an element is in the set or not by using Bloom filters easily. This way of a query of an element in the set saves time. Accordingly, it is space and time efficiency method to represent a set. It has been widely used to find some solutions for data mining, reducing network traffic or some routing problems in networks [2-5].

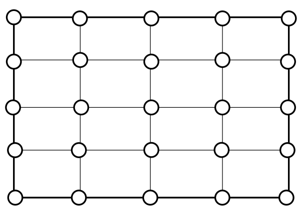

Figure 1. A rectangular grid graph as a model of a network with computers on each vertex.

For more extensive reading on network applications of Bloom filters, we recommend the study [6]. Because of the usage of the random methods to build the Bloom filter, it may produce some errors and such errors are called false positives. Therefore, the implementations of the Bloom filter widely aim to decrease the probability of false positives [11,12].

A message forwarding scenario in a rectangular grid graph representing a computer network is considered in this paper. We suppose that there is a computer in each vertex in the graph and the messages are sent through the paths. One of routing model is that a message can be sent via the shortest paths between nodes in a network [7]. This is a way to time saving during distribution. In practical applications, the demand for the shortest paths might cause the network traffic increments. In this case, the users may be forced to use any path between two distinct nodes rather than shortest paths. The model taking a rectangular grid as a network and encoding shortest paths in that graph without false positives has been studied by [8,13]. In this research, we consider to encode any path with Bloom filter which do not generate false positives. The Bloom filter in this model of routing has a role of packet header that is sent with the message between computers. A sender can send messages throght the shortest paths between distinct nodes. We assume that the path is choosen in advance, encoded with a Bloom filter, and the Bloom filter and the message are directed together to the receiver computer. We introduce a certain encoding method for the paths in this paper for this model to function.

Celal Bayar University Journal of Science
Volume 18, Issue 4, 2022, p 349-353
Doi: 10.18466/cbayarfbe.970002

We organize the paper as follows. In section 2 we introduce standard Bloom filter with probability of false positive in detail. In section 3 we give definition of some encoding methods and define our new encoding method for the paths in rectangular grids. In following sections we prove how the encoding method we built does not generate false positives, and emphasise the influence of our encoding method, then we conclude the paper with section 6.

2. Standard Bloom Filter

A Bloom filter is method to store the elements in a set with a compressed form [1]. Consider a subset S with n elements of a universe U. $\forall x \in U$, each x is described by a binary string. The Bloom filter introduced in [1] represents the subset S with a binary string of length m. Each binary string of the elements in U is kept. Hence, one can easily check if any bits in binary strings of the set and the elements match in any bit positions. By this comparision we can conclude whether $x \in S$ or $x \notin S$. We may denote the model binary strings of an element by $\beta(x)$ and the Bloom filter of the set S by $\beta(S)$. If $\beta(x) \nleq \beta(S)$, then it can be concluded that x is definitely not in the set S. However, if $\beta(x) \leq \beta(S)$, then we cannot be certain about the existence of the element x in the set S. Since, the bits 1s take place in the array of each element in the random bit positions and $\beta(S)$ is obtained by adding these binary arrays together. Specifically, the $\beta(S)$ can be produced by applying the bitwise OR operation to $\beta(x)$ for all elements in S. The binary OR operation generates a bit 1 if at least one of the inputs is 1, in other case generates a bit 0.

Some elements may seem like an element of the set S, but they may not be in the set S. These elements are denoted by false positives. Simple calculation yields the probability of false positives as

$$(1-(1-\tfrac{1}{m})^{kn})^k \approx \left(1-e^{\frac{-kn}{m}}\right)^k \qquad \textbf{(2.1)}$$

where m is the number of the bits in the Bloom filter, $n = |S|$ and k is the number of bit 1 in the Bloom filter of an element [1].

3. Edge Encoding Methods in Rectangular Grid Graphs

3.1. Bit-per-edge Encoding

Suppose $G = (V, E)$ is a rectangular grid graph with sets of vertices V and edges E and U is a universal set of Bloom filters. The edge e is described by one bit in $\beta(U)$.

The edges have particular bit positions in $\beta(U)$. This labelling way is called bit-per-edge labelling. Obviously $U = E$. The number of edges in a ractangular grid of size $M \times N$, that is, M is the number of horizontal edges in each row and N is the number of vertical edges in each column, is $2MN + M + N$. Hence, if the edges in the graph are encoded by bit-per-edge encoding method, then $2MN + M + N$ is obtained as the length of a Bloom filter.

Theorem 3.1. If the edges in the graph are labelled by bit-per-edge encoding method, then no false positives is produced.

Proof. Suppose the edges in the graph are labelled with bit-per-edge encoding method. Consider a path P between the vertices v_0 and v_k consisting of sequence of k edges that are $e_0, e_1, \ldots, e_{(k-1)}$. The edges in the path P are presented by the bit 1 in $\beta(S)$. Obviously, k number of bits in $\beta(S)$ are set to 1 and $n - k$ number of bits are 0 where $n = |E|$. These k bits are situated in distinct bit places in $\beta(P)$. Suppose the representative bit 1 of an edge f from the graph is placed an ith bit position in $\beta(f)$ among n bits. If $\beta(f) \leq \beta(P)$, then we can conlude that $f \in \{e_0, e_1, \ldots, e_{(k-1)}\}$. Otherwise, there must be $k + 1$ number of bit 1 in $\beta(P)$. This is a contradiction with the number of edges on the path P. Therefore, the edge f is not a false positive to the Bloom filter of the path.

3.2. A New Approach for Encoding in Rectangular Grid Graphs

In order to encode the paths in the rectangular grid, we assume that there are imaginary diagonal rows intersecting vertices. The compass direction of the rows is north-west (or eqaually south-east) and they are numbered starting from the bottom corner to top corner of the grid (see Figure 2). The first row intersects one vertex and the second row intersects two vertices and similarly the following rows intersect the vertices regarding their numbers. The total number of rows is $M + N + 1$ where the size of the rectangular grid is $M \times N$. We may denote the rows as $r_1, r_2, \ldots, r_{(M+N+1)}$.

Each vertex is represented by binary strings which we call block and the lenght of these representative blocks depends on the number of vertices lying on the same row. Also each block contains one bit 1. For instance: if a row intersects two vertices then the representative blocks of these vertices are 01 and 10, similarly if a row intersects three vertices, then the blocks 100, 010 and 001 represent any vertex on the row. The number of the row determines the places for blocks of the vertices in $\beta(v)$ where $v \in V$.

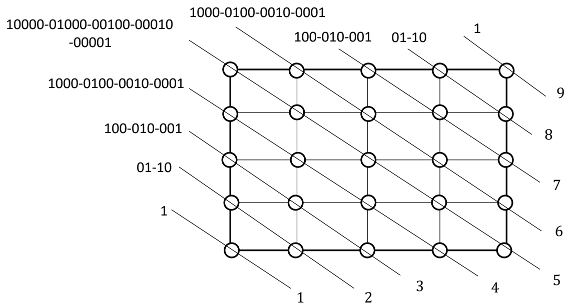

Figure 2. The numbers of the rows intersecting the vertices and the representative blocks of the vertices on the rows in a 4×4 sized rectangular grid.

There are $(M + 1)(N + 1) = MN + M + N + 1$ vertices in total in a rectangular grid of size of $M \times N$. Obviously, the lenth of the Bloom filter in the grid is $MN + M + N + 1$.

In the routing scenario, the message and the header of the message, that is $\beta(P)$, are sent through the shortest paths in the graph together. A path can be defined as a sequence of the consecutive distinct edges and an edge can be denoted by $e = \{v_i, v_j\}$ where v_i and v_j are the end vertices of the edge e. The path $P = \{v_0, v_1, \dots, v_n\}$ where v_i is a vertex and $i \in \{0, 1, 2, \dots, n\}$ is represented by $\beta(P)$ that is obtained by using bitwise OR operation to all bits of $\beta(v\_i)$, where $\forall v\_i \in P$.

Note that the vertices, edges and paths in the grid have same length Bloom filters. Yet, the number of the bit 1 in $\beta(v)$, an edge and a path are one, two and n, which is the number of veritices in the path, respectively.

4. No-False Positives

According to routing scenario, the header describes the route for the message and it is sent with the message to the last node in the particular shortest path. When a computer v from the path gets the message, it examines each vertex connected to v and compares $\beta(P)$ and $\beta(v\_i)$ incidental to v in all bits. Therefore, the adjacent vertices stand for the false positives of these networks.

Lemma 4.1. The shortest paths in rectangular grids do not have cycles.

Proof. Suppose a path $\boldsymbol{P}$ is one of the shortest path between the vertices $\boldsymbol{p}$ and $\boldsymbol{q}$ which are not adjacent. The shortest distance between these vertices is a straigth line. Through this imaginary line the edges on the path $\boldsymbol{P}$ have two different directions, and hence the shortest path includes only such edges. Suppose the path $\boldsymbol{P}$ contains a cycle on a vertex $\boldsymbol{u}$ (see the Figure 3.). A cycle contains the edges with four different directions in a rectangular grid. Therefore this means that if a path

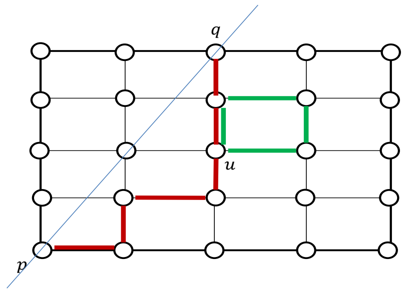

Figure 3. The red path is the shortest path between the nodes $\boldsymbol{p}$ and $\boldsymbol{q}$ and contains a cycle on the vertex $\boldsymbol{u}$.

contains a cycle then the path includes at least two more edges whose directions do not belong to the shortest distance between the vertices $\boldsymbol{p}$ and $\boldsymbol{q}$.

Theorem 4.1. $\boldsymbol{\beta(P)}$, where the path $\boldsymbol{P}$ is the shortest path, is obtaied by using the new approach (see Section $\mathbf{3.2}$) in a rectangular grid graph does not generate any false positives.

Proof. Consider a rectangular grid graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ and a shortest path $\boldsymbol{P}$ with the sequence of vertices $\boldsymbol{v_0}, \boldsymbol{v_1}, \dots, \boldsymbol{v_n}$. Suppose that there is a false positive $\boldsymbol{v_i} \in \mathbf{V}$ and this vertex $\boldsymbol{v_i}$ is intersected by the encoding row $\boldsymbol{r_i}$. In this case, $\boldsymbol{r_i}$ may intersect more than one vertices from the path or only the vertex $\boldsymbol{v_i}$.

If the row $\boldsymbol{r_i}$ intersects vertices $\boldsymbol{v_j}, \boldsymbol{v_k}, \dots$ from the path and the vertex $\boldsymbol{v_i}$ and a fragment $\boldsymbol{F}$ of the path $\boldsymbol{P}$ contains these vertices, then $\boldsymbol{\beta(v_i)} \leq \boldsymbol{\beta(F} = \{\boldsymbol{v_i}, \boldsymbol{v_j}, \boldsymbol{v_k}, \dots\})$. If $\boldsymbol{v_i} \in \{\boldsymbol{v_0}, \boldsymbol{v_1}, \dots, \boldsymbol{v_n}\}$, then $\boldsymbol{v_i}$ is not a false positive.

Suppose that $\boldsymbol{v_i} \notin \{\boldsymbol{v_0}, \boldsymbol{v_1}, \dots, \boldsymbol{v_n}\}$. By encoding method we build, the number of the bit 1 in the $\boldsymbol{\beta(P)}$ in the rectangular grid graph is the number of vertices, since each Bloom filter of a vertex contain one bit $\mathbf{1}$ and a path consists of a sequence of vertices. By Lemma $\mathbf{4.1}$ a shortest path does not have cycles, therefore the rows intersecting the vertices in this particular shortest path come consecutively. In this case, each row intersects one vertex from the shortest path. By assumption, the model blocks of the vertices crossed by the same row contain the bit $\mathbf{1}$ in different bit positions from each other. Therefore, the number of $\mathbf{1}$ in $\boldsymbol{\beta(P)}$ is obtained as $\boldsymbol{n} + \mathbf{2}$. Yet this contradicts the number of vertices in the path $\boldsymbol{P}$ which is $\boldsymbol{n} + \mathbf{1}$. Therefore, the vertex $\boldsymbol{v_i}$ is not a false positive of $\boldsymbol{\beta(P)}$.

If the row $\boldsymbol{r_i}$ intersects only the vertex $\boldsymbol{v_i}$ and the other vertices in the path $\boldsymbol{P}$ are intersected by other rows in the graph, then the model block of the vertex $\boldsymbol{v_i}$ is placed in the block position numbered by the row $\boldsymbol{r_i}$ in $\boldsymbol{\beta(P)}$. The row $\boldsymbol{r_i}$ does not intersect other vertices from the path $\boldsymbol{P}$, hence no other represenatative blocks of the vertices in the path $\boldsymbol{P}$ coincides with the representative block of the vertex $\boldsymbol{v_i}$ in the same block position of $\boldsymbol{\beta(P)}$. Accordingly, the number of the bits $\mathbf{1}$ in the $\boldsymbol{\beta(P)}$ is obtained as $\boldsymbol{n+2}$, yet this contaradicts the number of $\mathbf{1}$ in $\boldsymbol{\beta(P)}$ that is $\boldsymbol{n+1}$. This conludes that the vertex $\boldsymbol{v_i}$ does not represent a false positive.

5. Results for Vertex Encoding Method

The Bloom filter provides the users to access the data very fast and to save space when storing the data. Yet it may generate false positives and this may cause some trouble for some implementations. In this research we build Bloom filter for routing scenarios in a regular network model. The vertex-coding method which puts the bits 1 in the Bloom filters in certain bit-positions. Therefore, one does not find any false positives for shortest paths in a rectangular grid.

Besides, bit-per-edge labelling method (introduced in Section **3.1**) does not generate false positives. Yet, the number of edges in a rectangular grid is $2MN + M + N$ in a $M \times N$ sized rectangular grid and hence the length of the Bloom filter in bit-per-edge labelling method is $2MN + M + N$. However, the length of the Bloom filter we built is $MN + M + N + 1$. Obviously, $MN + M + N + 1 < 2MN + M + N$. Accordingly, the method introduced in this paper results Bloom filters with less number of bits than the Bloom filter produced by applying bit-per-edge encoding method. This can be advantage for some applications which has limited spaces during message routing.

If the bits 1 in $\beta(v)$ are placed randomly , then the model may generate false positives. We compute the probability of false positives with the parameters that we found for the bit-per-vertex encoding method. These parameters are the number of vertices on the path n, the length of the Bloom filter m and the number of the bit 1 denoted by k in $\beta(v)$. In our method of encoding, the number of the bit 1 is one in $\beta(v)$. As seen in the Table 1 the probability of false positives are computed for $m = 256$ and variety values of n. Note that $n < m$, since a shortest path contains less number of vertices than the graph. Obviously, if the more vertices in a path occurs, the more probability of false positives is obtained when k and the length of the Bloom filter are fixed.

Table 1. Probability of false positives where $k = 1, m = 256$ and n with different values

m	n	p
256	256	0,632120
256	240	0,608394
256	200	0,542166
256	100	0,323366
256	50	0,177422

As seen from the Table 1, if we encode the vertices by using random labelling methods, then we have false positives with high probability. The probability of false positives p is computed by using the formula $(1 - e^{\frac{-kn}{m}})^k$ (see Section 2).

6. Conlusion

The paths in a graph can be represented by using random methods, but this approach can produce errors denoted by false positives. Our coding method is to find particular Bloom filters for the paths. In our appraoch of encoding method, the algebraic properties of the graph such as the length of the edges do not affect either encoding method or routing scenario. Also, this method can be used by the network users who need spaces for data storage and error-free zone for message transport in a particular network.

Author's Contributions

Gökçe Çaylak Kayaturan: Drafted and wrote the manuscript, performed the experiment and result analysis.

Ethics

There are no ethical issues after the publication of this manuscript.

References

[1] B. H. Bloom, "*Space/time trade-offs in hash coding with allowable errors*," Communications of the ACM, vol. 13, no. 7, 1970, pp. 422–426.

[2] Y. Lu, B. Prabhakar, and F. Bonomi, "*Perfect hashing for network applications*," in Information Theory, 2006 IEEE International Symposium on. IEEE, 2006, pp. 2774–2778.

[3] C. E. Rothenberg, C. A. B. Macapuna, M. F. Magalhaes, F. L. Verdi, ˜ and A. Wiesmaier, "*In-packet bloom filters: Design and networking applications*," Computer Networks, vol. 55, no. 6, 2011, pp. 1364–1378.

[4] L. Carrea, A. Vernitski, and M. Reed, "*Yes-no bloom filter: A way of representing sets with fewer false positives for in-packet path encoding,*" Arxiv eprint arXiv:1603.01060v1, 2016.

[5] X. Yang, A. Vernitski, and L. Carrea, "*An approximate dynamic programming approach for improving accuracy of lossy data compression by bloom filters*," Europen Jornal of Operational Research, Elsevier, vol. 252, no.3, 2016 ,pp. 985-994

[6] A. Broder and M. Mitzenmacher, "*Network applications of bloom filters: A survey*," Internet mathematics, vol. 1, no. 4, 2004, pp. 485–509.

[7] A. Vernitski, M. Reed and L. Carrea, "*Optimized hash for network path encoding with minimized false positives*". Computer networks, 58:180–191, 2014.

[8] G. C. Kayaturan and A. Vernitski. "*A way of eliminating errors when using bloom filters for routing in computer networks*". In Networks, ICN 2016. The Fifteenth International Conference on, pages 52–57. IARIA, 2016.

[9] X. Li, L. Peng, and C. Zhang, "*Application of bloom filter in grid information service*". in Multimedia Information Networking and Security (MINES), 2010 International Conference on. IEEE, 2010, pp. 866–870.

[10] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "*Grid information services for distributed resource sharing*," in High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on. IEEE, 2001, pp. 181–194.

[11] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. "*Summary cache: a scalable wide-area web cache sharing protocol*". IEEE/ACM Transactions on Networking (TON), 8(3):281–293, 2000.

[12] M. Mitzenmacher. "*Compressed bloom filters*". IEEE/ACM Transactions on Networking (TON), 10(5):604–612, 2002.

[13] G. Ç. Kayaturan. "*Representing Shortest Paths in Graphs Using Bloom Filters without False Positives and Applications to Routing in Computer Networks*". PhD dissertation, University of Essex, UK, 2018.