Pavlo Kravets, Artem Volokyta.

RESEARCH OF PROPERTIES OF SCALABLE DE BRUIJN TOPOLOGIES WITH EXESS CODING.

The article discusses properties of the topology based on the de Bruijn graph with variable length of the alphabet and the word. The issue of the relation between fault tolerance and the length of the alphabet has been considered.

Key words: fault tolerance, de Bruijn, excess code

Fig.: 6. Tables: 3. Bibl.: 4

Acute problem. The topological structure of any computational system heavily impacts its performance, fault tolerance and difficulty of organizing one. The designs of the topology are constantly evolving, based on current needs and engineering capabilities. Therefore, we propose de Bruijn topology with the method to increase robustness with use of longer alphabet.

Target setting. Due to the growing usage of distributed computational systems and especially systems with low trust and fault tolerance of the individual elements new methods of synthesis fault-tolerant topologies are required.

Actual scientific researches and issues analysis.

Fat tree

Figure 1 illustrates the general idea of fat-tree topology using switches with 4 ports.

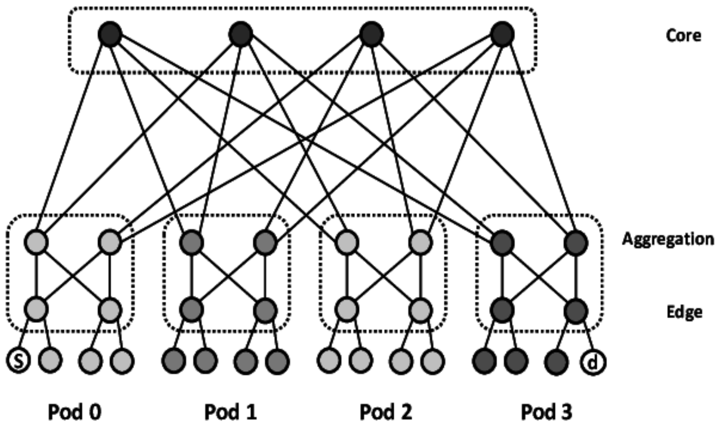

Fig. 1. Fat-Tree topology with k=4 [1]

Fat-Tree, which is the special case of the Clos network, is a highly-scalable topology, which is widely used in the construction of modern supercomputers. In their paper [2], Mohammad Al-Fares et al. propose three-tier architecture. Using *k*-port switches, they organize *k* pods, each containing 2 layers of *k/2* switches. Lower level (called *edge layer*) switches are connected to *k/2* hosts and *k/2* switches of higher level (*aggregation layer*). The aggregation layer switches are connected to *k/2 core* (in some publications it is referred to as *spine*) layer switches. There are $(k/2)^2$ switches in this layer, each connected to *k* pods.

Using k-port switches, this topology supports $k^3/4$ hosts. It provides high performance and fault tolerance and can be build using commodity switches, making it cheaper option compared to other topologies.

Dragonfly topology

Figure 2 illustrates the general idea of dragonfly topology with 9 groups, 4 routers per group, 2 hosts per router.

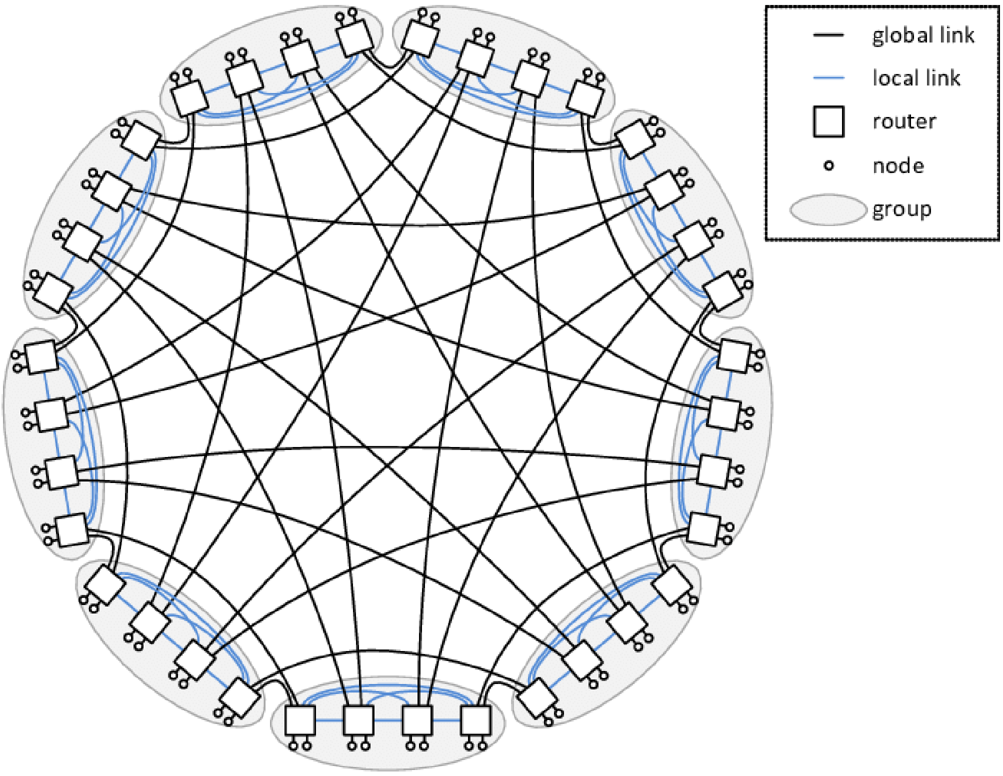

Fig. 2. Dragonfly topology with h=2, p=2, a=4 [3]

The dragonfly topology, introduced in the paper [4] by John Kim et al., is hierarchical topology with three levels: *router*, *group* and *system*. Each group consists of a routers, each connected to p terminals, all other routers within the same group (such connections are called *local channels*) and h routers in other groups (such connection are called *global channels*). All of the routers in the group act as virtual router with radix of $a(p+h)$. The topology can have $ah+1$ groups and the global diameter of 1 ($ap(ah+1)$ terminals). This helps to lessen the number of global connections that require more expensive cables, effectively decreasing the cost of organizing the network.

Uninvestigated parts of general matters defining. While de Bruijn topology itself was examined in different papers[5], its characteristics in general case are still not completely investigated.

The research objective. The purpose of this research is to study characteristics of general case de Bruijn topology and how the depend on basic properties of de Bruijn topology: power of the alphabet and length of the node identifier

The statement of basic materials. The topology synthesis is: there are V nodes with unique identifiers coded in particular way. Usually $V = B^N$, where B – length of the alphabet, N – length of the identifier. Find connected nodes (neighbours) for each node.

De Bruijn topology. Standard de Bruijn topology is synthesized using shift transformation. To get identifiers of the node's neighbours, its digits must be shifted left or right (only one way for the whole process) with freed digit being set to one of values in the alphabet. For example, for B = 3 (Alphabet {0;1;2}) and N = 4 neighbours of the node with identifier 0012 are:

Table 1

Shift results for node number 0012

New value	Shift left	Shift right
0	0120	0001
1	0121	1001
2	0122	2001

The result is the de Bruijn graph. To get de Bruijn topology, following steps must be executed:

1) Transform all directed edges into undirected
2) Eliminate loops
3) Eliminate multi-edges

Example of the synthesis. The de Bruijn topologies with B = 2 N = 2 and B=3 N=3 and will be synthesised (Shift left)

Table 2

Shifts for de Bruijn graph with B = 2 and N = 2

Node	Shift 0	Shift 1
00	00	01
01	10	11
10	00	01
11	10	11

Figure 3 shows synthesized de Bruijn graph and topology with alphabet {0;1} and 2-symbol word.

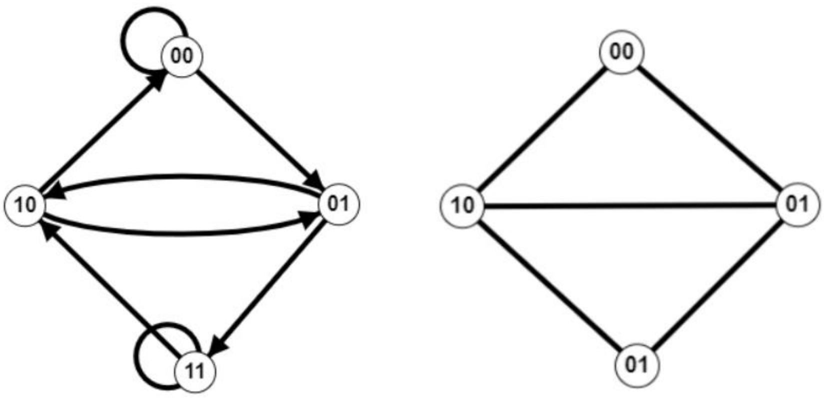

Fig. 3. De Bruijn graph and topology with B=2 and N=2

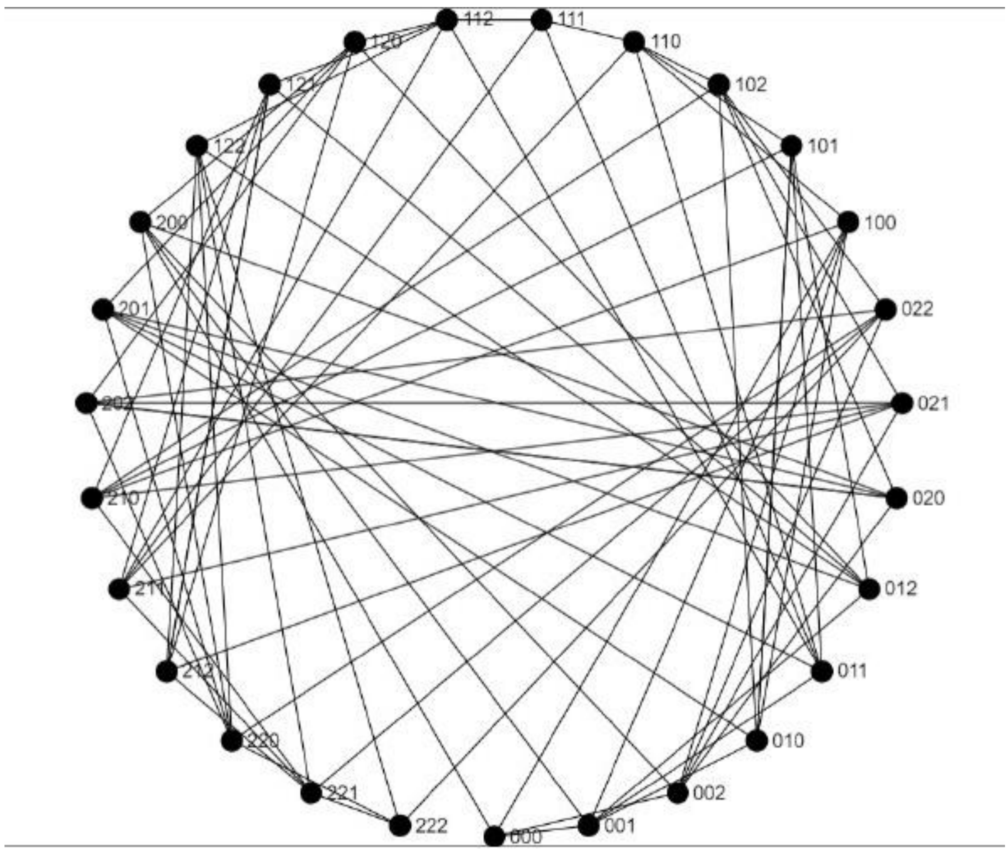

Fig. 4. De Bruijn topology with B=3 and N=3

Figure 4 shows synthesized de Bruijn topology with alphabet {0;1;2} and 3-symbol word [6].

Main properties of the topology. Main properties of the topology are number of nodes, number of edges, maximum and minimum degree (radix) of the node, diameter and minimal number of faulty nodes that will make routing impossible

Number of nodes V by definition

$$V = B^N \quad (1)$$

Example: $B = 2$ and $N = 2$. Then $V = 2^2 = 4$

Number of edges E**.** By definition of de Bruijn topology synthesis:

E = Number of edges of the graph - (Number of loops + Number of duplicate edges).

By definition of de Bruijn topology synthesis number of edges of the graph = *V*B* (Number of nodes by number of possible shifts)

Loops in de Bruijn graph are possible only if the shift does not change the identifier, which is possible only if all symbols in identifier are the same and freed digit is set to the same symbol. So, the number of loops is equal to B.

Multi-edges in de Bruijn graph are possible only if the result of two different shifts is the original identifier (if shifts are identical, it is the loop).

Therefore:

1) There can be only 2 edges parallel to each other at most

2) It is only possible, if the identifier is the interchanging sequence of two symbols, for example 1010 and 0101. If N < 3, then all identifiers fulfill this condition.

To calculate the number of duplicate edges, we must calculate the number of possible unordered pairs of non-identical symbols. Therefore number of duplicate edges is $(B^2-B)/2$

Hence , $E = V * B - B - \frac{B^2-B}{2} =$

$$\boldsymbol{E = B^{N+1} - \frac{B^2+B}{2}} \tag{2}$$

Example: $B = 2$ and $N = 2$. Then $E = 2^{2+1} - (2^2 + 2)/2 = 5$

Maximum Δ and minimum δ degree of the node. By definition of de Bruijn topology synthesis Δ = number of possible shifts from node’s identifier + number of possible shifts that lead to node’s identifier. Number of possible shifts that lead to node’s identifier can be expressed as the number of possible opposite (for left shift – rights and vice versa)

Therefore

$$\boldsymbol{\Delta = 2B} \tag{3}$$

Because of elimination of loops and multi-edges δ < Δ. Since either one loop or one pair of parallel edges can be incident to the node, then

$$\boldsymbol{\delta = 2B - 2} \tag{4}$$

If N < 3, then every node has either loop or multi-edge. Therefore, if N < 3, then

$$\boldsymbol{\Delta = \delta = 2B - 2}$$

Diameter d. By definition of de Bruijn topology synthesis minimal path between nodes, that do not have same symbols in identifiers (for example, 000 and 111), is equal to N. By definition of de Bruijn topology, there can be no pair of nodes?

Minimal path between which is greater than N. Therefore

$$d = N \tag{5}$$

Routing trees and minimal number of faulty nodes that will make routing impossible. Let's denote minimal number of faulty nodes that will make routing impossible as $\mathbf{F_{min}}$. In de Brujin topology, this property is equal to the number of trees (let's call them routing trees) that can be constructed by shifts from root node that has identifier that consists from the same symbol.

Proof. Consider de Bruijn topology, constructed with left shift length of the alphabet B and length of the identifier N. Then B trees with height N can be constructed, because there are B nodes that has identifier that consists from the same symbol and diameter of the topology is equal to N.

Figure 5 shows one of the routing trees for de Bruijn topology with B=3 N=3:

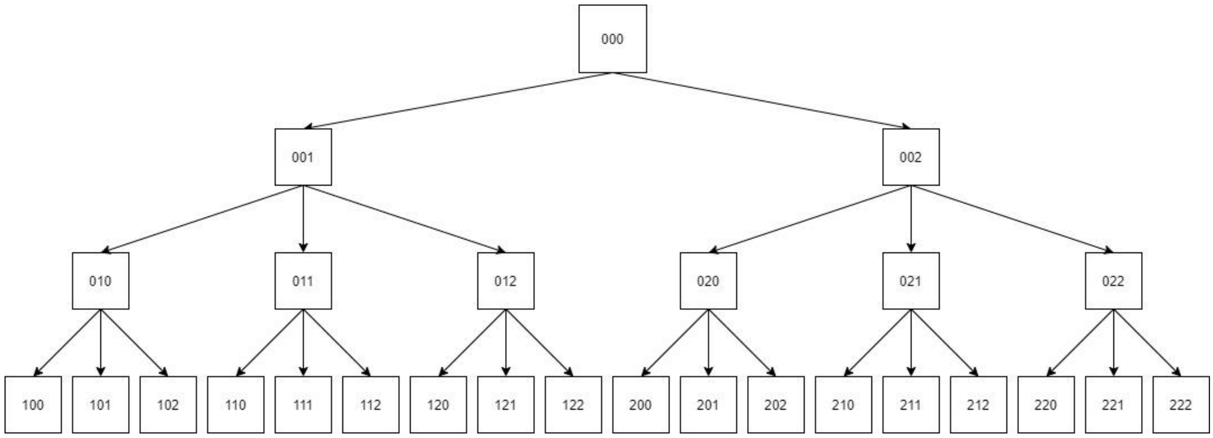

Fig. 5. Routing tree example

The important property of such trees is that identifiers of all non-leaf nodes start with the same symbol as root? And every such tree includes all nodes of the topology.

Suppose F nodes do not transfer messages. Then the set of the symbols with which the identifiers of these nodes start has no more elements than *F*. If $F < B$, then exists such routing tree, where all faulty nodes are leafs and do not impact routing. Analogical proof for right shift

Therefore

$$F_{min} = B \tag{6}$$

Summary of characteristics

Table 3

De Bruijn topology characteristics

Property	Value
Number of nodes V	B^N
Number of edges E	B^{N+1} - $(B^2+B)/2$
Maximal degree Δ	2B
Minimal degree δ	2B-2
Diameter d	N
Minimal number of faults to break routing F_{min}	B

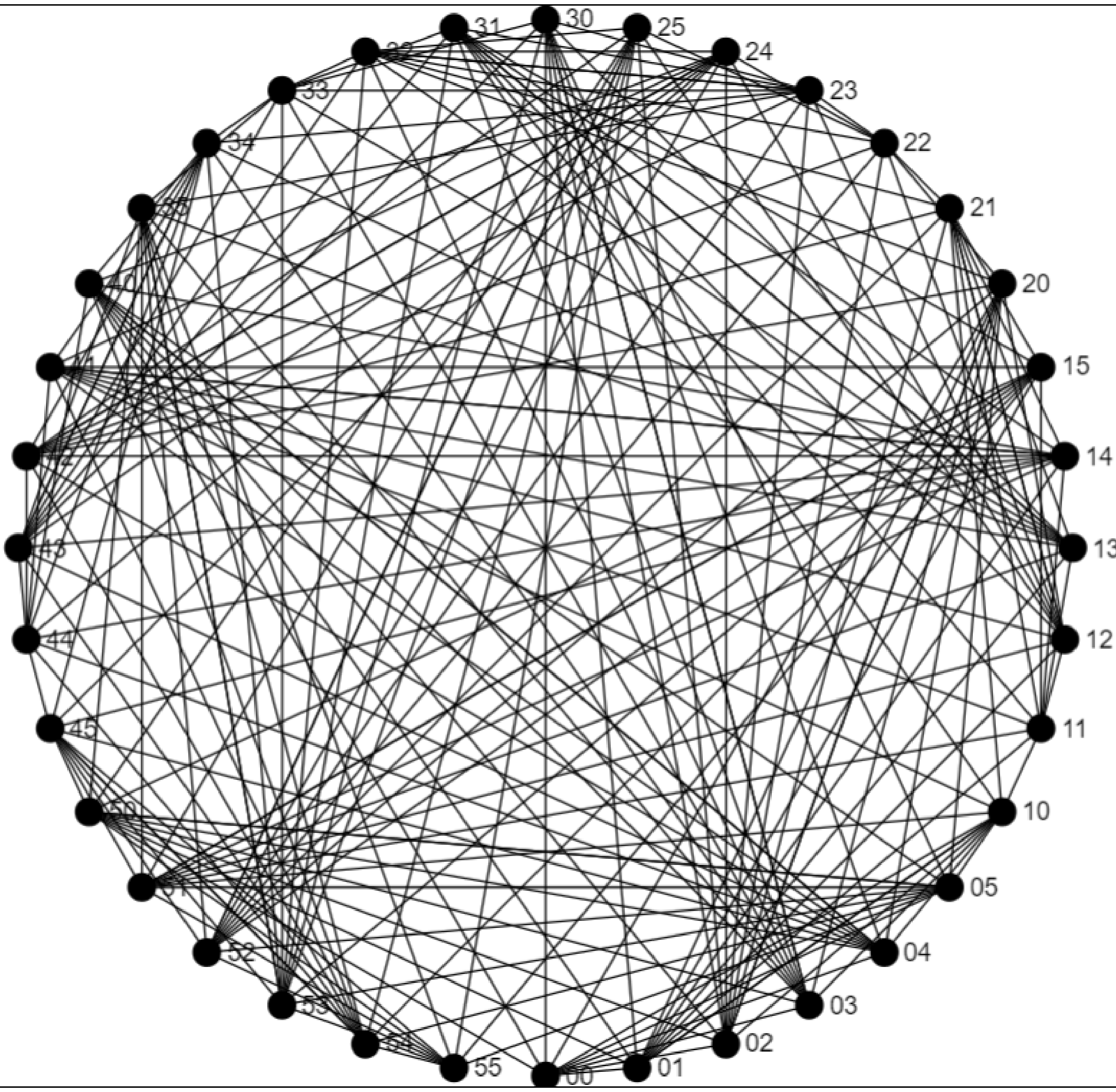

Fig. 6. De Bruijn graph with B=6 N=2

Figure 6 shows synthesized de Bruijn topology with alphabet {0;1;2;3;4;5} and

2-symbol word. It has 36 nodes, 195 edges, its nodes have a radix of 10, ith has diameter of 2 and at least 6 nodes must stop transporting messages for routing to become impossible

Conclusions. Fault tolerance is becoming increasingly valuable in modern distributed computation system. In this paper, we propose new highly robust topology based on the use of de Bruijn graph.

The advantages of the proposed topology are high fault tolerance, scalability, variable radix and diameter. But de Bruijn topology has disadvantages, too: for large networks, it requires either high diameter or high radix and number of connections.

This topology can be further improved by using clusters in place of nodes or creating additional connections to decrease the diameter.

References

1. The adopted fat tree topology for SDN switches of order k=4. URL:https://www.researchgate.net/figure/A-The-adopted-fat-tree-topology-for-SDN-switches-of-order-k4-48-B-Network-sizes-used_fig5_305228967 (request date 11.05.2020)

2. Mohhamed Al-Fares, Alexander Loukissas, Amin Vahdat.: A Scalable, Commodity Data Center Network Architecture. Proceedings of the ACM SIGCOMM 2008 conference on Data communication. ACM. pp. 63–74

3. Sample Dragonfly topology with h=2 (p=2, a=4), 36 routers and 72 compute nodes. URL: https://www.researchgate.net/figure/Sample-Dragonfly-topology-with-h2-p2-a4-36-routers-and-72-compute-nodes_fig2_261313973 (request date 11.05.2020)

4. J.Kim,W.J.Dally,S.Scott,andD.Abts.2008.Technology-Driven,Highly-Scalable DragonflyTopology.In2008 International Symposium on Computer Architecture. 77–88. https://doi.org/10.1109/ISCA.2008.19

5. H. Loutskii, A. Volokyta, P. Rehida, O. Honcharenko, B. Ivanishchev and A. Kaplunov, "Increasing the fault tolerance of distributed systems for the Hyper de Bruijn topology with excess code," 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2019, pp. 1-6, doi: 10.1109/ATIT49449.2019.9030487.

6. Olexandr G., Rehida P., Volokyta A., Loutskii H., Thinh V.D. (2020) Routing Method Based on the Excess Code for Fault Tolerant Clusters with InfiniBand. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham

AUTHORS

Pavlo Kravets – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

Artem Volokyta (supervisor) – associate professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.