The Faculty of Mathematics and Physics
Comenius University
Bratislava

Virtual Path Layouts in ATM networks

Master Thesis

Consultant:
doc. RNDr. Peter Ružička, Csc.

Diplomant:
Martin Makúch

Čestne prehlasujem, že som diplomovú prácu vypracoval samostatne s použitím materiálov, ktoré uvádzam v zozname použitej literatúry.

Bratislava, 2000 ....

Acknowledgments

I would like to thank Peter Ružička for his great support and valuable advice.

Contents

1 Introduction

1.1 Motivation

The rapid development in communication technology naturaly brings new demands from the users. To make a realization of such requests possible, a new solution was to be found. The main problem still existing is the number of distinct communication networks, such as ordinary telephone networks, satelite networks used primary for mobile communication, cable television, and computer networks.

The most promising solution to this problem seems to be the B-ISDN (Broadband Integrated Services Digital Network), the successor of quite unsuccessfull N-ISDN (Narrowband ISDN). N-ISDN was the first fully (end to end) digital telephone system. However, it came too late and was offering too few for relative high price. The B-ISDN offers solutions, where its predecessor failed.

The main advantage of B-ISDN is the high bandwidth (from 155 Mbps), allowing, for example, realtime video transmission. The goal, however, is to reach gigabit rates. For such network, new technology is needed. The standard network protocols, such as TCP/IP used in internet, wouldn't do the work, because they are too complex.

The new technology is called ATM (Asynchronous Transfer Mode). The main idea is to use small fixed size packets, called cells. This simplification, together with unreliability (no error checking and correcting in higher levels) and simple routing method (straightforward two level hierarchy), allows design of very simple switches operating on high speeds (due to simplicity). The result is very high ratio of cells passing through chip per second, implying high bandwidth.

The assuptions of no error checking/correcting are possible only when very low error rate is achieved on physical transfer. Detecting of faulty cells must be done, except for some applications, but it is left on the application. The high error rate on physical medium would result on frequent identical cell requesting from application (all faulty cells must be usually resend), delaying the network and the application itself. Hopefully, the fiber optic allows very reliable transmision together with high bandwidth.

Yet, at least one problem remained. As mentioned above, the routing mechanism is very simple. Each cell has two small routing fields, called VCI (virtual channel index) and VPI (virtual path identifier). On every switch, there are two routing tables, one for the virtual paths, the second for the virtual channels. As long as VPI is not null, routing is done according to it. Then the VCI is used to choose the following virtual path on the route. The virtual paths must be designed at the very beginning, and are never changed (they are physicaly implemented on the switching hardware). The virtual channels are then used to connect the network users, and are maintained dynamically. However, each virtual channel has to be build as the concatenation of the virtual paths.

Now the problem arises - the initial design of virtual paths to allow communication between all interesting pairs of nodes. Moreover, several restrictions come into the play. These are physical limits of the switches, and network perfomance requirements. We will deal with them in this paper.

We start with definition of the model in *Chapter 2*. The Survey of known results is then presented in *Chapter 3*. In *Chapter 4*, we study some virtual path layouts for complete binary trees. These layouts are used in further constructions of VPLs for butterfly networks (*Chapter 5*). The main result of this papaer is also presented there. It is the asymptoticaly optimal broadcast routing scheme (one-to-all VPL) for the butterfly topology. Finally, the recapitulation of achieved results appears in *Chapter 6*.

2 The Model

Now, we present the model used to describe Path Layout problems in ATM networks in more formal way. Most definitions are taken from the previous papers (eg [1]). The communication network is represented by an undirected graph $G = (V, E)$, where the set V of vertices corresponds to the ATM switches, and the set E of edges to the physical links between them. Moreover, we have a given set ζ of pairs of distinct nodes from V, between which a communication must be established. We are interesting on two special cases:

- The *one-to-all* case: the connection is required from one specified vertex to all others; so $\zeta = \{(r, u) | u \in V, u \neq r\}$, where r is the specified vertex (usually called the *root*).
- The *all-to-all* case: the connection is required between all pairs of vertices; so $\zeta = \{(u, v) | u, v \in V; u \neq v\}$.

For the following definitions, the network is G, and the ζ is either *one-to-all* or *all-to-all* case.

Definition 2.1: A *virtual path layout* (shortly VPL) Ψ is a collection of simple paths in G, termed *virtual paths* (shortly VPs).

From now, we distinguish between two types of VPLs, depending on their communication patern ζ. We will refer to an VPL with one-to-all communication pattern as *one-to-all VPL*, and to an VPL with all-to-all communication pattern as *all-to-all VPL*.

Definition 2.2: The *load* $\mathcal{L}(e)$ of an edge $e \in E$ in a VPL Ψ is the number of VPs $\psi \in \Psi$ that include e. This notion is also refered to as its *edge congestion*. The *load* $\mathcal{L}(v)$ of a vertex $v \in V$ in a VPL Ψ is the number of VPs $\psi \in \Psi$ that include v (*vertex congestion*).

Definition 2.3: The *maximal edge load* $\mathcal{L}_{max}(\Psi)$ of a VPL Ψ is $max_{e \in E} \mathcal{L}(e)$. The *maximal vertex load* of a VPL Ψ is $max_{v \in V} \mathcal{L}(v)$.

Unless otherwise specified, the loads in this paper are the edge loads.

Definition 2.4: The *average (edge) load* of a VPL Ψ is computed as

$$\mathcal{L}_{avg}(\Psi) = \frac{1}{|E|}\sum_{e\in E}\mathcal{L}(e).$$

Definition 2.5: The *hop count* $\mathcal{H}(u,v)$ between two vertices $u,v \in V$ in a VPL Ψ is the minimum number of VPs whose concatenation forms a path in G connecting u and v. This concatenation is also called *virtual channel* (shortly VC). If no such VPs exist, we define $\mathcal{H}(u,v) = \infty$.

Definition 2.6: The *maximal hop count* of a VPL Ψ is computed as

$$\mathcal{H}_{max}(\Psi) := max_{(u,v)\in\zeta}\{\mathcal{H}(u,v)\}.$$

The problem is to design an VPL with as small maximum hop count and load as possible. However, decreasing hop count naturaly increases load and vice versa, so some hop-load tradeoff is needed. The tradeoff is usually studied in one of the either way:

- For a given network (G) and maximum hop count ($\mathcal{H}_{max}$), find such a VPL on G, that has lowest possible load ($\mathcal{L}_{max}$) with respect to $\mathcal{H}_{max}$.
- For a given network (G) and maximum load ($\mathcal{L}_{max}$), find such a VPL on G, that has lowest possible hop count ($\mathcal{H}_{max}$) with respect to $\mathcal{L}_{max}$.

We will concentrate on the first case. For our topology (butterfly), it will be the more difficult one (with respect to asymptotic solutions), as we will see later. The second case was already studied, and some results are known. They are presented later.

We shall also be concerned with *stretch factor*. Informally, it is the ratio between the length of the path of a VC in the physical graph (G) and the shortest possible path between its endpoints (in G). This parameter control the efficiency of the utilization of the network. However, the exact definition is needed to prevent some confusions, as explained below. The definition for our case is as follow:

Definition 2.7: Let the $\mathcal{H}_{max}$ be the upper bound on hop count in some VPL Ψ. For each $u,v \in \zeta$, we define a set

$$VC_{u,v} := \{\tau|\tau\ is\ a\ VC\ between\ u\ and\ v\ with\ hop\ count\ less\ or\ equal\ to\ \mathcal{H}_{max}\}$$

Now let

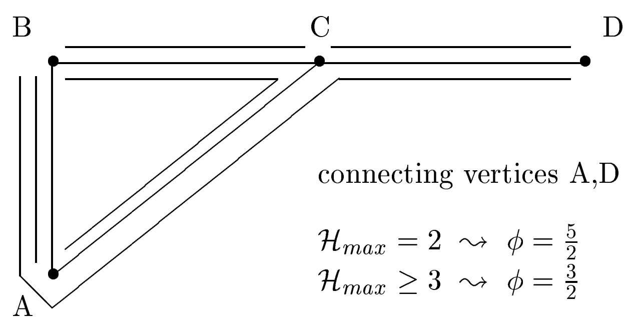
$$dst(u,v) := min_{\tau\in VC_{u,v}}\{the\ length\ of\ \tau\ in\ physical\ graph\ G\}$$

And finally we get

$$\phi := max_{u,v \in \zeta} \left\{ \frac{dst(u,v)}{d(u,v)} \right\}$$

where $d(u, v)$ is the distance between u and v in G. The ϕ is called *stretch factor* of the VPL Ψ (with respect to $\mathcal{H}_{max}$).

This definition says, that hop count has a higher priority than stretch factor, as shown on the following figure:



Suppose, we are interested only in connecting vertices A and D. If we have upper bound of $\mathcal{H}_{max} = 2$, then the only possibility is a two-hop VC consisting of VPs (A, C, B) and (B, A, C, D) of length 5. However, the shortest path between A and D (path (A, C, D)) has a length of 2, so $\phi = \frac{5}{2}$. On the other hand, if we allow greater upper bound $\mathcal{H}_{max} \geq 3$, we could connect A and D with three-hop VC using VPs (A, B), (B, C) and (C, D) with $\phi = \frac{3}{2}$.

3 Survey of Results

In this section, we summarize results related to this work, namely to the design of VPLs in ATM networks. Because of great diversity of results, my arrangement should be considered informal. The basic criterion is the topology of the network. Besides topology several other factors are considered: directional/nondirectional links, communication patterns, and the type of the results. In general, we will concentrate on the following parameters:

- $\mathcal{H}$, the maximum hop count allowed for connection (defined previously as $\mathcal{H}_{max}$)
- $\mathcal{L}$, the maximum load of edge allowed for VPL (defined previously as $\mathcal{L}_{max}$)
- ϕ, the stretch factor, or how many times are the routing routes longer than shortest paths.

3.1 Chain networks

For this subsection, assume N to be the length of the chain.

- **bidirectional links, one-to-all communiaction**
  - Simple result $\mathcal{L} \leq \mathcal{H} \times N^{\frac{1}{\mathcal{H}}}$, where $\mathcal{H} = \mathrm{O}(1)$, comes from fairly straightforward construction. It is asymptoticaly optimal layout. See eg [3] and/or [6].
  - With more complicated design, it is possible to get also optimal solution. However, the proof is somewhat more difficult. The results are:
  $$\binom{\mathcal{L}+\mathcal{H}-1}{\mathcal{L}} < N \leq \binom{\mathcal{L}+\mathcal{H}}{\mathcal{L}}$$
  for $\mathcal{L} = \mathrm{O}(1)$, and
  $$\binom{\mathcal{L}+\mathcal{H}+1}{\mathcal{H}} < N \leq \binom{\mathcal{L}+\mathcal{H}}{\mathcal{H}}$$
  for $\mathcal{H} = \mathrm{O}(1)$. These two are the only truly optimal (up to constants, not only asymptoticaly) results already known. See [10].
- **bidirectional links, all-to-all communication**
  - For $\mathcal{L} = 2$ there is a result $\sqrt{2N} - 5 < \mathcal{H} < \sqrt{2N} + 2$, and generally for $\mathcal{L} = \mathrm{O}(1)$ there is $\frac{1}{2} \times N^{\frac{1}{\mathcal{L}}} < \mathcal{H} < \mathcal{L} \times N^{\frac{1}{\mathcal{L}}}$. See [5].
  - For any $\epsilon > 0$ and $\mathcal{L} \geq \log^{1+\epsilon} N$ there is $\mathcal{H} = \Theta(\frac{\log N}{\log \mathcal{L}})$, asymptoticaly optimal layout. See [4].
  - The reverse case (given $\mathcal{H}$, find $\mathcal{L}$) is not directly mentioned.
- **unidirectional links, all-to-all communication**
  - For $\mathcal{L} = 1$ it holds $\frac{N}{2} + \log N \leq \mathcal{H} \leq \frac{N}{2} + \log N + \mathrm{O}(1)$. See [8].
  - Generally, for $\mathcal{L} = \mathrm{O}(1)$ it holds $\mathcal{H} = \Omega(N^{\frac{1}{2\mathcal{L}-1}})$ and $\mathcal{H} = \mathrm{O}(\mathcal{L} \times N^{\frac{1}{2\mathcal{L}-1}})$. See [8].
  - The reverse case (for $\mathcal{H}$, find $\mathcal{L}$) was probably not yet studied. However, since $\mathcal{L}$ is constant in the above case, the result is asymptoticaly optimal for a given $\mathcal{L}$.

For all the above cases the routing is done along the shortest paths (ϕ=1).

3.2 Ring networks

Now, let N be the number of vertices in a ring.

- **bidirectional links, one-to-all communication**
  - Since we still route using the shortest paths, we can treat this problem as the one-to-all case for chain with length $\frac{N}{2}$. Small technical problem arises for even N. However, it won't be a problem at all for asymptotical solutions.

- **bidirectional links, all-to-all communication**
  - For $\mathcal{H} = \mathrm{O}(1)$ we can get

$$\frac{1}{32\mathcal{H}^{\frac{1}{\mathcal{H}}}} N^{\frac{2}{\mathcal{H}}} \leq \mathcal{L} \leq \frac{\mathcal{H}(\mathcal{H}+1)}{2} N^{\frac{2}{\mathcal{H}}}$$

    what is, for constant $\mathcal{H}$, an asymptoticaly optimal layout with $\mathcal{L} = \Theta(N^{\frac{2}{\mathcal{H}}})$. See [2].

- **unidirectional links, all-to-all communication**
  - For $\mathcal{L} = 1$ we have $\mathcal{H} = 2\sqrt{2N} + \mathrm{O}(1)$. See [8]
  - In a general case $\mathcal{L} = \mathrm{O}(1)$, it holds $\mathcal{H} = \Omega(N^{\frac{1}{2\mathcal{L}}})$ and $\mathcal{H} = \mathrm{O}(\mathcal{L} \times N^{\frac{1}{2\mathcal{L}}})$, what is, for constant $\mathcal{L}$, again asymptoticaly optimal. See [8].

3.3 Mesh networks

Further, the a, b are the dimensions of the mesh. In the case of $a = b$, we set $\sqrt{N} = a = b$.

- **bidirectional links, one-to-all communication**
  - For a mesh with $\mathcal{H} = \mathrm{O}(1)$ we have a lower bound of $\mathcal{L} = \Omega((\frac{N}{\mathcal{H}})^{\frac{1}{\mathcal{H}}})$ and a layout with $\mathcal{L} \leq \mathcal{H} \times N^{\frac{1}{\mathcal{H}}}$. See [1].

- **bidirectional links, all-to-all communication**
  - Quite easy construction (for $\mathcal{H} = \mathrm{O}(1)$) leads to a layout with $\mathcal{L} = a^{\frac{2}{h_a}} = b^{\frac{2}{h_b}}$, where $h_a = \frac{\mathcal{H}}{\frac{\log a}{\log b}+1}$, $h_b = \mathcal{H} - h_a$. So for $a = b$ we get $\mathcal{L} = \mathrm{O}(\mathcal{H} \times N^{\frac{2}{\mathcal{H}}})$. See [2].
  - For $\mathcal{L} = \mathrm{O}(1)$ and $a = \mathrm{O}(1)$, we have $\mathcal{H} = \Theta(b^{\frac{1}{a\mathcal{L}}})$. Further, for the mesh $\sqrt{N} \times \sqrt{N}$ and $\mathcal{L} = \mathrm{O}(1)$, we get $\mathcal{H} = \Theta(\log \sqrt{N}) = \Theta(\log N)$. See [5].
  - Previous result is for arbitrary $\mathcal{L} \geq 2$ generalized to $\mathcal{H} = \Theta(\frac{\log N}{\log \mathcal{L}})$. See [4].

- **unidirectional links, all-to-all communication**
    - For $\mathcal{L} = \mathrm{O}(1)$ we have $\mathcal{H} = \Theta(\log N)$. See [8].
    - For torus and $\mathcal{L} = \mathrm{O}(1)$ we get $\mathcal{H} = \Omega((ab)^{\frac{1}{2a\mathcal{L}}})$ and $\mathcal{H} = \mathrm{O}(a \times b^{\frac{1}{2a\mathcal{L}}})$. See [8].
    - The reverse case ($\mathcal{L}$ for given $\mathcal{H}$) is again not considered.

3.4 Tree networks

For trees, let N be the number of their vertices.

- **bidirectional links, one-to-all communication**
    - It is shown in [10], that optimal one-to-all VPL for chain of length N can be transformed into an one-to-all VPL for a tree with N vertices. The ϕ is still 1, $\mathcal{H}$ and $\mathcal{L}$ remain the same or better. This makes some upper bounds for trees in both cases (known $\mathcal{H}$, find $\mathcal{L}$ and vice versa). However, these layouts may not be optimal for trees.
    - A recursive layout leads to $\mathcal{L} = \mathrm{O}(\mathcal{H} \times N^{\frac{1}{\mathcal{H}}})$. It is also proven, that $\mathcal{L} = \Omega(\frac{1}{\Delta \times 2^{\frac{1}{\mathcal{H}}}} \times N^{\frac{1}{\mathcal{H}}})$, where Δ is the maximum degree of any vertex in tree. For "realistic" network (Δ and $\mathcal{H}$ bounded by constant) we have $\mathcal{L} = \Theta(N^{\frac{1}{\mathcal{H}}})$. See [6].
- **bidirectional links, all-to-all communication**
    - For $\mathcal{L}$, an upper bound of $\frac{\mathcal{H}}{2(2^{\frac{2}{\mathcal{H}}}-1)} N^{\frac{2}{\mathcal{H}}}$ and a lower bound of $\mathcal{L} = \Omega(\frac{1}{\Delta \times (8\mathcal{H})^{\frac{1}{\mathcal{H}}}} N^{\frac{2}{\mathcal{H}}})$ are shown in [6]. For "realistic" networks, it leads to $\mathcal{L} = \Theta(N^{\frac{2}{\mathcal{H}}})$.
    - In the mentioned solutions all routes are using the shortest paths, so $\phi = 1$.
- **unidirectional links, all-to-all communication**
    - For $\mathcal{L} = 2$ we have a lower bound of $\mathcal{H} \geq \frac{1}{2} N^{\frac{1}{3}}$ and upper bounds of $\mathcal{H} \leq 32 N^{\frac{1}{3}}$ and $\mathcal{H} \leq D_G{}^{\frac{1}{3}} \log N$, where D_G is a diameter of the graph. See [8].
    - Generally, for $\mathcal{L} = \mathrm{O}(1)$, we have a lower bound of $\mathcal{H} = \Omega(D_G{}^{\frac{1}{2\mathcal{L}-1}})$, upper bounds of $\mathcal{H} \leq 8\mathcal{L} \times N^{\frac{1}{2\mathcal{L}-1}}$ and $\mathcal{H} = \mathrm{O}(D_G{}^{\frac{1}{2\mathcal{L}-1}} \times \log N)$. See [8].

3.5 Hypercube topology

Here, the N denotes the dimension of the hypercube. So the number of vertices is 2^N.

- **bidirectional links, all-to-all communication**
  - For $2 \le \mathcal{L} \le N$ we get $\mathcal{H} = \Theta(\frac{N}{\log N})$. See [4].
  - For the rest $\mathcal{L}$ we have $\mathcal{H} = \Theta(\frac{N}{\log \mathcal{L}})$. Again, see [4].

3.6 General networks

The N will denote the number of vertices of a graph G.

- **bidirectional links, one-to-all communication**
  - There is a construction satisfying $\mathcal{L} \le \sqrt{\mathcal{H}} N^{1+\frac{1}{\mathcal{H}}}$. See [6].
  - The decision problem was studied for the existence of the one-to-all layouts for $\mathcal{H}$ and $\mathcal{L}$ in arbitrary graphs with $\phi = 1$. It was shown, that there is a polynomial algorithm for designing such VPL in the case of $\mathcal{H} = 2$, $\mathcal{L} = 1$ and $\mathcal{H} = 1$, *any* $\mathcal{L}$. In other cases, the problem is NP-complete. See [9].
- **bidirectional links, all-to-all communication**
  - Recursive construction for a given k leads to a VPL with $\phi = 8k$ and $\mathcal{L}_V = \mathrm{O}(\mathcal{H} \times k \times \log N \times N^{\frac{1}{k}+\frac{2}{\mathcal{H}}})$, where $\mathcal{L}_V$ is the vertex load. See [1].
  - It is shown in [5], that $\mathcal{H} > \frac{\log N}{\log(\Delta\mathcal{L})} - 1$, for any $\mathcal{L} \ge 1$.
  - The VPL with $\mathcal{H} = \mathrm{O}(\frac{D_G \log \Delta}{\log \mathcal{L}})$ was constructed for $\Delta \ge 3$. It is asympoticaly optimal in the case of networks with unbounded Δ, $D_G = \mathrm{O}(\log N)$ and any $\mathcal{L} \ge \Delta$. For any $\mathcal{L} \ge 1$ and $D_G = \mathrm{O}(\log N)$ is constructed a VPL with $\mathcal{H} = \Theta(\frac{\log N}{\log \mathcal{L}})$. See [4].
  - In [6] is a recursive construction for graphs with bounded treewidth. The result is $\mathcal{L} = \mathrm{O}(\frac{k\mathcal{H}N^{\frac{2}{\mathcal{H}}}}{2((1.5)^{\frac{2}{\mathcal{H}}}-1)})$, where k is the bound on the treewidth.
  - Several other problems were studied, for instance dynamic maintenace of the links. The fault-tolerant VPLs were considered too (see [11]).

3.7 Summary of previous results

Though quite a couple of interesting results were already presented, there is still a much more left for future research. The following list shows some possibilities:

- There are still topologies not considered yet in detail (just general results could be applied on them). An example is a butterfly studied in *Chapter 4*.

- There are still problems, which are not solved even asymptoticaly optimal, so the gap could be narrowed or closed. Eg tree networks for not "realistic" case.

- There are results in one direction (eg given $\mathcal{L}$, find $\mathcal{H}$) and not in another. This is also the case for butterfly studied in this paper.

- It could be interesting to replace stretch factor by some other measure describing the difference between the shortest and used routing paths (eg dilation).

- Mostly, just two communication patterns are studied. The one-to-all and the all-to-all case, though some other are of interest too (eg permutation pattern).

- Usually, it is assumed to have constant maximal $\mathcal{L}$ for the whole network, though the different switches could have different abilities, eg future switches may allow higher congestion. Also, it is common to assume the same capacity (bandwidth) for all the links and in both directions, though different services demand different bandwidth (and sometimes also in different directions).

- There are not results for any compound communication patterns, which could for instance favour some pairs of vertices (eg some priority links for important pairs or better paying customers).

- The case of bidirectional links was more studied, though some unidirectional results already exist.

- Very few results consider faulty links.

- There is probably only one optimal (not only asymptoticaly) layout, the one-to-all VPL for chain topology.

However, on the other hand, it is necessary to consider also the simplicity of the model. Since too general model may be too complicated for the design and analysis of VPLs. So it might be good to consider, which property is quite useful (eg in real ATM networks) to model, and then, perhaps, add it to the model.

4 VPLs for complete binary trees

Now we will concentrate on the design of VPLs for complete binary trees. This case is fairly simple and was already studied for more general case (arbitrary trees, eg [3],[2]). Though these results were already presented (at least in general case), we deal with them again, since they are quite useful for the design of VPLs on butterfly network.

4.1 Complete binary trees

Here we present our labeling conventions for complete binary trees to achieve easier description of the design of VPLs on this topology.

Let $T = (V, E)$ be a complete binary tree with N vertices, $V = \{v_1, ..., v_N\}$. Then $N = 2^l - 1$, where l is the depth of T.

We number the levels of T from $'1'$ up to $'l'$ (notice that $l = lg(N+1)$), where the root is the only vertex on the first level. We also label the vertices, top-down and left-right, giving label $'1'$ to the root. So the root is referenced to as $v_1 \in V$ and a vertex $v_k \in V$, if not a leaf, has sons v_{2k} and v_{2k+1}, and if not a root, has a parent $v_{\lfloor k/2 \rfloor}$.

Let $e \in E$ be an edge of T. We define its level to be equal to the level of the vertex incident with e with smaller label (so the two edges from root are on the first level).

Further, for any vertex $u \in V$, T_u denotes the subtree of T rooted at u.

4.2 Lower bounds

Theorem 4.1: Let $T = (V, E)$ be a tree network with N nodes rooted at r, let Δ be the maximum degree of a node, and take any $h > 1$. For every one-to-all VPL from r with h hops, there exists an edge $e \in E$ with load $\mathcal{L}(e) = \Omega(\frac{1}{\Delta^{\frac{1}{h}}} N^{\frac{1}{h}})$.

Proof: See [6] and/or [3].

Theorem 4.2: Let T be a complete binary tree with the root r. For every one-to-all VPL from r it holds $\mathcal{L} = \Omega(N^{\frac{1}{\mathcal{H}}})$.

Proof: A special case of the previous theorem. Let T be a complete binary tree, $h = \mathcal{H}$, so $\Delta = 3$. Now, from the existence of such edge e and definition of $\mathcal{L}_{max}$, we have $\mathcal{L} = \Omega(\frac{1}{3^{\frac{1}{\mathcal{H}}}} N^{\frac{1}{\mathcal{H}}})$. Since $3^{\frac{1}{\mathcal{H}}}$ is bounded by a constant between 1 and 3 for $\mathcal{H} \geq 1$, we have $\mathcal{L} = \Omega(N^{\frac{1}{\mathcal{H}}})$□

Theorem 4.3: Let $T = (V, E)$ be a tree network with N nodes, let Δ be the maximum degree of a node, and $h > 1$. For every all-to-all VPL with h hops, there exists an edge $e \in E$ with load $\mathcal{L}(e) = \Omega(\frac{1}{\Delta^{\frac{1}{h}}} N^{\frac{2}{h}})$.

Proof: Again, see [6] and/or [3].

Theorem 4.4: Let T be a complete binary tree. For every all-to-all VPL it holds $\mathcal{L} = \Omega(N^{\frac{2}{\mathcal{H}}})$.

Proof: A special case of the previous theorem. Let T be a complete binary tree, $h = \mathcal{H}$, so $\Delta = 3$. Now, from the existence of such edge e and definition of $\mathcal{L}_{max}$, we have $\mathcal{L} = \Omega(\frac{1}{3^{\frac{1}{\mathcal{H}}}} N^{\frac{2}{\mathcal{H}}})$. Since $3^{\frac{1}{\mathcal{H}}}$ is bounded by a constant between 1 and 3 for $\mathcal{H} \geq 1$, we have $\mathcal{L} = \Omega(N^{\frac{2}{\mathcal{H}}})\Box$

4.3 1-hop VPLs for complete binary trees

Now, we will look closer at VPLs for complete binary trees with $\mathcal{H}_{max} = 1$. Although not very interesting as a special case, they are usefull as a basic block in future constructions.

Layout 4.5: Firstly, we describe the most basic one-to-all layout connecting root with all other vertices. Let $T = (V, E)$ be a complete binary tree (with the root v_1). Our VPL Ψ will consist of the following simple paths: For each vertex $v_k \in V$, $v_k \neq v_1$ (v_k is not a root), add the shortest path from v_1 to v_k. This path is unique, since T is a tree. We can write it formally as

$$\Psi = \{\psi | \psi = (v_k, v_{\lfloor k/2 \rfloor}, \dots, v_1);\ v_k \in V;\ k \neq 1\}$$

Analysis of VPL: For the above one-to-all VPL Ψ, one can easily see, that $\mathcal{H}_{max} = 1$ and $\mathcal{L}_{max} = \lfloor N/2 \rfloor$, since the largest load is on the two topmost edges ((v_1, v_2) and (v_1, v_3)), each sharing VPs to the whole subtree of $\lfloor N/2 \rfloor$ vertices. Since we are interested mostly on asymptotical results, it will usually suffice, that $\mathcal{L}_{max} = \mathrm{O}(N)$. It is easy to see, that routing is done along the shortest paths, so $\phi = 1.\Box$

Layout 4.6: Although previous layout is obviously optimal (for $\mathcal{H}_{max} = 1$, not only asymtotically), we present another assymptoticaly optimal one-to-all layout (again from the root v_1) which will be exploited in further constructions. Let $T = (V, E)$ be a complete binary tree. Now we connect every vertex $v_k \in V$ with all the vertices in its subtree (this was done only with the root v_1 in the previous layout). Formally, our VPL Ψ is

$$\Psi = \{\psi | \psi = (v_k, v_{\lfloor k/2 \rfloor}, \dots, v_m);\ v_k, v_m \in V;\ \exists j \in \mathcal{N}:\ m = \lfloor \frac{k}{2^j} \rfloor\}$$

Simply written, v_k must be in the subtree rooted at v_m, but $k \neq m$.

Analysis of VPL: In this layout, we only added some new paths to the previous one, so the routing is still using the shortest paths and $\phi = 1$. Also $\mathcal{H}_{max} = 1$. To determine the $\mathcal{L}_{max}$ is somewhat more difficult. Let $e \in E$, $e = (v_a, v_b)$, $a \leq b$ w.l.o.g., be an edge in E on ith level, $1 \leq i < l$. Let W be a set of vertices on the path from the root v_1 to the vertex v_a ($W = \{v_a, v_{\lfloor a/2 \rfloor}, \dots, v_1\}$). In the VPL Ψ, every path, which includes e, starts at some vertex from W, since only vertices from W have e in their subtrees. Each vertex $u \in W$ is directly connected (by path in Ψ) with all vertices 'below' e (with all vertices in subtree rooted at v_b). No other paths use e, since each path, which uses e, must end in the subtree rooted at v_b, and these paths were mentioned above (see *Fig.4.1*).

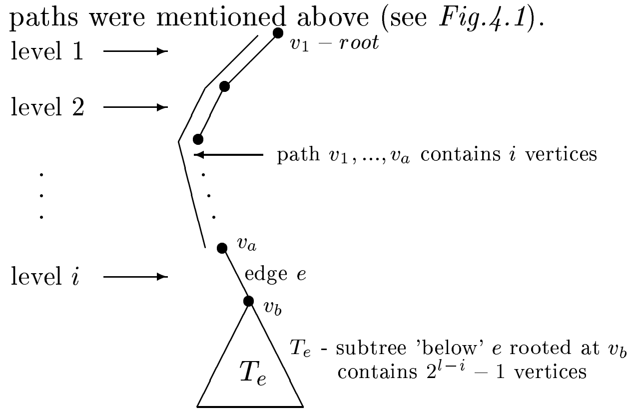


Fig.4.1: schematic picture of the tree T

Since the edge $e = (v_a, v_b)$ is on the ith level, the path from v_1 to v_a has i vertices (v_a is on the ith level). These vertices form W, so $|W| = i$. The subtree T_e rooted at v_b has $2^{l-i} - 1$ vertices. Since each vertex from W has direct connection to every vertex in T_e, the total number of paths in Ψ, which include e, is equal to $|W|.|T_e| = i(2^{l-i} - 1)$. So we have

$$\mathcal{L}(e) = i(2^{l-i} - 1) \leq 2^{i-1}(2^{l-i} - 1) = 2^{l-1} - 2^{i-1} \leq 2^{l-1} - 1 = \frac{N-1}{2} \leq \lfloor \frac{N}{2} \rfloor$$

Thus, it holds $\mathcal{L}_{max} = \lfloor \frac{N}{2} \rfloor$, since it is the load of the two uppermost edges ((v_1, v_2) and (v_1, v_3)). Asymptoticaly, it still holds $\mathcal{L}_{max} = O(N)$□

4.4 Asymptoticaly optimal one-to-all VPL on complete binary trees

Now we will use *Layout 4.5* as a basic building block for constructing asymptoticaly optimal one-to-all VPL for complete binary tree rooted at v_1 for any given $\mathcal{H}_{max}$. Again, the ideas are taken from constructions already presented (eg [3]) in more general case.

We need following definition for next layout:

Definition 4.7: Let $T = (V, E)$ be a complete binary tree with the root v_1. Let $u \in V$ be an vertex of T. For any $k \in \mathcal{N}$, $k \geq 1$, we define

$$T(u, k) = (V', E')$$

to be a subgraph of T, with vertices

$$V' = \{w | w \in T_u \wedge d_T(u, w) \leq k\}$$

where d_T is the distance in the tree T. The set of edges

$$E' = \{(a,b)|(a,b) \in E \wedge a,b \in V'\}$$

comes directly from T. It is easy to see, that $T(u,k)$ is a complete binary tree with the depth of $k+1$ or smaller (if k is greater than the number of levels below the vertex u).

Layout 4.8: Let $T = (V,E)$ be a complete binary tree with a root v_1. Let $\mathcal{H}$ be the upper bound for hop count in VPL on T. We will construct one-to-all VPL on T from v_1 as follows:

Firstly assume, for simplicity, that $l-1 = k\mathcal{H}$, so the $\mathcal{H}$ divides $l-1$ (notice, that there are $l-1$ levels of edges in T). Now we partition the tree T into smaller complete binary trees, each with depth $k+1$, as follows: For any vertex v_i on level m, $m = ck+1$, $c \in \mathcal{N}_0$, if v_i is not a leaf, take a tree $T(v_i,k)$ from the previous definition. By S_T we denote arbitrary tree among them. There would be $\mathcal{H}$ levels of such trees, as shown in *Fig.4.2*:

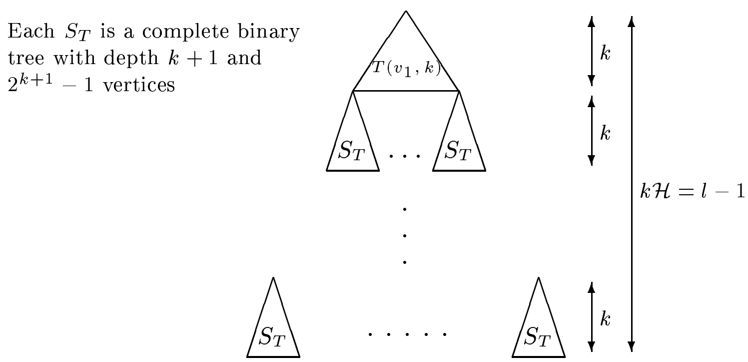


Fig.4.2: Tree T partitioned into the trees S_T

The roots of the trees $T(v_i,k)$ are usually called pivots. Notice, that each leaf of $T(v_i,k)$ is identical to a root of some tree $T(v_j,k)$ from lower level (except for downmost trees). Now we construct one-to-all VPL $\Psi_{T(v_i,k)}$ for each of the $T(v_i,k)$ using *Layout 4.5* . The VPL Ψ for the tree T is then union of all layouts for trees $T(v_i,k)$.

Analysis of VPL: Let $u \in V$ be a vertex of the tree T. From the previous construction, there exists at least one (at most two) tree $T(v_i,k)$, which includes u. Now construct a path P from the root v_1 to the vertex v_i in T (the path is unique, since T is a tree). The path includes at most $\mathcal{H}$ pivots (the first is v_1, last v_i). Every pivot v_j on P, except for v_i, is directly connected to the succeeding pivot in P by single path from Ψ, since in the tree $T(v_j,k)$ the pivot v_j is directly connected to every leaf (see *Layout 4.5*), and the succeeding pivot is one of the leaves. So we can get from the root v_1 to the vertex v_i in at most $\mathcal{H}-1$ hops. From v_i we can get to the vertex u on single hop due to layout $\Psi_{T(v_i,k)}$, which is included in Ψ. So $\mathcal{H}_{max} = \mathcal{H}$.

Since the path constructed above was also the shortest path between the root v_1 and the vertex u, we get $\phi = 1$.

In the case of load, no two trees $T(v_i, k)$, $T(v_j, k)$ have a common edge, so the load is the same as in every tree $T(v_i, k)$. The load is, recalling from *Layout 4.5* ,

$$\mathcal{L} = \left\lfloor \frac{2^{k+1} - 1}{2} \right\rfloor = \left\lfloor \frac{2(\frac{N}{2})^{\frac{1}{\mathcal{H}}} - 1}{2} \right\rfloor = \mathrm{O}(N^{\frac{1}{\mathcal{H}}})$$

since each tree $T(v_i, k)$ has $2^{k+1}-1$ vertices and $2^{k\mathcal{H}} = 2^{l-1} = \frac{N}{2}$. Comparing with the lower bound, our layout is asymtoticaly optimal.

For general case $l - 1 = k\mathcal{H} + m,\ 0 \leq m < \mathcal{H}$, the design is only slightly modified. Recall, that we have $\mathcal{H}$ levels of trees $T(v_i, k)$ in the previous simplified case. Now, we need two different types of trees, namely $T(v_i, k)$ and $T(v_j, k+1)$. So we replace first m levels of trees $T(v_i, k)$ with corresponding trees $T(v_i', k+1)$ (with the depth equal to $k+2$) keeping the property, that the trees on neighbouring levels have only one common vertex (the pivot). The $\mathcal{H} - m$ downmost trees remain of depth $k+1$. The resulting $\mathcal{H}_{max}$ and ϕ don't change (there are still $\mathcal{H}$ levels of trees $T(v_j, k[+1])$). The load is twice as large due to increasing the depth of some trees to $k+2$. This, however, don't affect the assymptotical result.□

4.5 Modified asymptoticaly optimal one-to-all VPL on complete binary trees

Now we will add some VPs to the previous layout, similary as we did with the 1-hop case. This modification does not affect the asymptotical optimality, though it is quite useful for the butterfly network.

Layout 4.9: Let $T = (V, E)$ be a complete binary tree with the root v_1. Let $\mathcal{H}$ be the upper bound on the hop count for VPL on T. We construct one-to-all VPL Ψ from v_1 exactly as in the previous *Layout 4.8* , only replacing *Layout 4.5* for trees $T(v_i, k)$ ($\Psi_{T(v_i,k)}$) by a *Layout 4.6* .

Analysis of VPL: Since we only added new VPs to the previous layout, it is still possible to connect the root with any vertex by an VC of at most $\mathcal{H}$ hops. The routing is still possible among the shortest paths ($\phi = 1$). Only the load of $T(v_i, k)$ is taken from different layout. However, it is still the same, since the $\mathcal{L}_{max}$ of the layouts *Layout 4.5* and *Layout 4.6* do not differ.

Notice, that in this layout, there exist also connections (VCs) from arbitrary vertex u to all vertices in its whole subtree with at most $\mathcal{H}$ hops.

Morevoer, this is also an all-to-all layout, connecting any pair of vertices in T with at most $2\mathcal{H}$ hops using the shortest path, what makes it asymtoticaly optimal all-to-all layout.□

5 VPLs for butterfly network

This section is the heart of this paper. The design of VPLs for butterfly networks was not studied yet, though some results are already known as a consequence of more general theorems. We will present some new lower and upper bounds together with releated VPL designs.

5.1 Butterfly topology

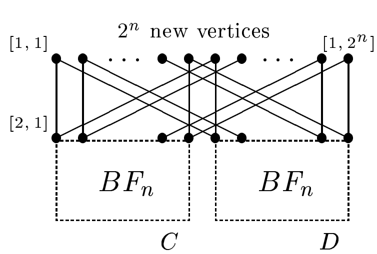
Here we describe the butterfly topology and introduce our labeling conventions to simplify the rest of the section.

The butterfly network BF_n consists of $n2^{n-1}$ vertices, usually represented by n rows and 2^{n-1} columns. We will label the vertices by their row/column position, exactly, let v be a vertex on the rth row and in the cth column. Then we label v as $v_{r,c}$. We now define the interconnection of vertices recursively.

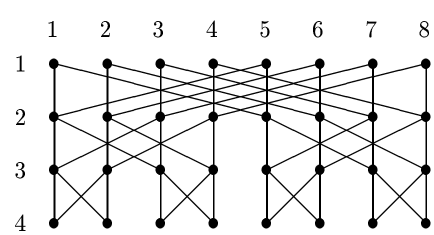
BF_1 network is a single vertex. Now let C,D be two BF_n networks. We construct BF_{n+1} as follows. Add 1 to each row of C and D, so they now have rows $2, ..., n+1$. Also add 2^{n-1} to each column of D. Now add new 2^n vertices $v_{1,1}, ..., v_{1,2^n}$. Finally, add following edges to the resulting graph:

- $\forall i,\ 1 \leq i \leq 2^n$, add an edge $(v_{1,i}, v_{2,i})$
- $\forall i,\ 1 \leq i \leq 2^{n-1}$, add an edge $(v_{1,i}, v_{2,i+2^{n-1}})$
- $\forall i,\ 2^{n-1}+1 \leq i \leq 2^n$, add an edge $(v_{1,i}, v_{2,i-2^{n-1}})$

The process is schematicaly shown on the following figure:



Construction of BF_{n+1}



BF_4 network

We will sometimes refer to the rows as levels, as we did on the trees. The number of rows/levels is denoted by n, or l (as in trees), and the total number of vertices as N, so $N = n2^{n-1} = l2^{l-1}$.

Now we will look at some properties of the butterfly topology, which will be helpful later. Firstly notice, that the butterfly is the same in the topdown and bottomup direction, in the sense, that there exists an isomorphism relabelling the vertices and reversing their topdown orientation. Also, all the vertices in the same row are symmetric in similar way (the order within a row is not significant). Now we define some special subtrees included in the butterfly network.

Definition 5.1: Let BF_n be a butterfly. For any $1 \leq i \leq 2^{n-1}$, we define a tree T_i corresponding to the vertex $v_{1,i}$ as follows: We set the vertex $v_{1,i}$ to be a root of the T_i. Now, for every vertex $v_{a,b}$ already in T_i, we recursively add its two neighbours in BF_n from the level $b+1$, unless $b = n$, when the $v_{a,b}$ is a leaf in T_i.

Similary, we can define the tree $T_i^{'}$ corresponding to the vertex $v_{n,i}$ going bottom-up. All these trees are complete binary trees with the depth n and $2^n - 1 = 2\frac{N}{n} - 1$ vertices. These trees make a backbone used for some of our VPLs. On the butterfly, we will be interested in two cases:

- one-to-all VPLs from any vertex $v_{r,c}$.
- all-to-all VPLs

5.2 Known results

As mentioned above, no special designs for butterfly were constructed. However, several general results offer some solutions. We will look closely only on one such result, which affects butterfly topology in significant way.

Theorem 5.2: Let G be a graph of order N with $\Delta = \mathrm{O}(1)$ and $diam(G) \leq \mathrm{O}(logN)$. Then $\mathcal{H} = \Theta(\frac{logN}{log\mathcal{L}})$ for any $\mathcal{L}$.

Proof: See [4].

It is easy to check, that butterfly network satisfies the conditions of this theorem, so the result can be apllied for it. However, the construction from [4] does not use only the shortest paths. In this paper, we will study the reverse problem for butterfly topology, namely to bound $\mathcal{L}$ for given $\mathcal{H}$.

5.3 Lower bounds

Firstly, we use some lower bounds from trees and apply them on the butterfly topology.

Lemma 5.3: Let Ψ be an one-to-all VPL on butterfly BF_n from the vertex $v_{1,1}$ with the maximal load of $\mathcal{L}$, the maximal hop count of $\mathcal{H}$ and stretch factor of one ($\phi = 1$). Then there exists a one-to-all VPL Ψ_t on complete binary tree T with n levels and $2^n - 1$ vertices from the root r with the same maximal load $\mathcal{L}$, the same maximal hop count $\mathcal{H}$ and the stretch factor of one.

Proof: Let $T = T_1$ (T_1 from *Definition 5.1*), since both are complete binary trees with n levels. Now we can construct Ψ_t as

$$\Psi_t = \{\psi | \psi \in \Psi \wedge \forall e \in \psi : \; e \in T_1\}$$

so we will restrict only to the VPs inside the T_1. The Ψ_t has clearly a load at most $\mathcal{L}$, since it is derived from Ψ, omitting some VPs.

Now look at the VCs in Ψ_t, namely, if the connection to all the vertices in T_1 from $v_{1,1}$ is still possible. Let u be an arbitrary vertex of tree T_1. Then there exists a VC θ connecting $v_{1,1}$ and u (using VPs from Ψ). Let $\psi \in \theta$ (ψ is a VP in Ψ). We will show that $\psi \in \Psi_t$.

Notice, that there is unique shortest path between $v_{1,1}$ and $u \in T_1$ in BF_n, because the path must go from $v_{1,1}$ only downwards (otherwise, it wouldn't be the shortest path). But all paths from $v_{1,1}$, coming only downwards, are in T_1, since there are allways two possibilities of going down, and both are in T_1 by its definition. So, the shortest path between $v_{1,1}$ and u in BF_n is also in T_1. And since T_1 is complete binary tree, this path is unique.

Now recall, that θ must go over the shortest path between $v_{1,1}$ and u. This path is in T_1, so is the θ too. However, this imply, that $\psi \in T_1$, due to $\psi \in \theta$. Thus, finally we have $\psi \in \Psi_t$ from the definition of Ψ_t.

As a result, all VPs necessary for connecting $v_{1,1}$ and u are also in the derived VPL Ψ_t. The routing is done using the same VC θ as in Ψ. So the hop count is at most $\mathcal{H}$, and the stretch factor is equal to one.□

Theorem 5.4: Let BF_n be a butterfly with N vertices. For every one-to-all VPL from $v_{1,1}$ with $\phi = 1$ it holds $\mathcal{L} = \Omega([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}})$.

Proof: Let Ψ be an one-to-all VPL for BF_n with $\phi = 1$, maximal hop count $\mathcal{H}$ and a load $\mathcal{L}$. From the previous *Lemma 5.3* , there exists an one-to-all VPL for complete binary tree with $2^n - 1$ vertices with the same $\mathcal{H}$, $\mathcal{L}$, and ϕ. From the lower bound for trees (*Theorem 4.2*), we get $\mathcal{L} = \Omega([2^n - 1]^{\frac{1}{\mathcal{H}}})$, what leads to:

$$\mathcal{L} = \Omega([2^n - 1]^{\frac{1}{\mathcal{H}}}) = \Omega([2^{n-1}]^{\frac{1}{\mathcal{H}}}) = \Omega(\left[\frac{N}{n}\right]^{\frac{1}{\mathcal{H}}})$$

since $N = n2^{n-1}$. We have also

$$lg(N) = lg(n2^{n-1}) = lg(n) + n - 1 = \Theta(n)$$

and next

$$lg(N) = \Theta(n) \Longrightarrow n = \Theta(lgN)$$

so finaly mixing these results

$$\mathcal{L} = \Omega(\left[\frac{N}{n}\right]^{\frac{1}{\mathcal{H}}}) = \Omega(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}})$$

proving the theorem.□

Theorem 5.5: Let BF_n be a butterfly with N vertices. For every one-to-all VPL with upper bound on hop count $\mathcal{H} \geq 2$ it holds $\mathcal{L} = \Omega([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N) = \Omega(N^{\frac{1}{\mathcal{H}}})$.

Proof: Let $r \in BF_n$ be any vertex of BF_n. Let Ψ be an one-to-all VPL from r for BF_n with load bounded by function $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} f(N))$. Suppose that $f(N) = o(lg^{\frac{1}{\mathcal{H}}} N)$.

Since the degree of any vertex in BF is bounded by constant ($\Delta_{max} = 4$), on one hop from r we can get to at most $O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} f(N))$ vertices (due to the upper bound for $\mathcal{L}$). On the second hop, we can get from these vertices to at most $O([\frac{N}{lgN}]^{\frac{2}{\mathcal{H}}} f^2(N))$ vertices and so on. After $\mathcal{H} - 1$ hops we can reach at most $O([\frac{N}{lgN}]^{\frac{\mathcal{H}-1}{\mathcal{H}}} f^{\mathcal{H}-1}(N))$ vertices.

Let S_1 be a set of vertices reached from r on at most $\mathcal{H} - 1$ hops in the VPL Ψ. Let $S_2 = BF_n \setminus S_1$. There are at least $\Omega(N)$ vertices in S_2, since $|S_1| = O([\frac{N}{lgN}]^{\frac{\mathcal{H}-1}{\mathcal{H}}} f^{\mathcal{H}-1}(N))$ and $f(N) = o(lg^{\frac{1}{\mathcal{H}}} N)$. These sets are connected with at most $O([\frac{N}{lgN}]^{\frac{\mathcal{H}-1}{\mathcal{H}}} f^{\mathcal{H}-1}(N))$ edges, since every vertex in S_1 has at most four outgoing edges. See Fig.5.2.

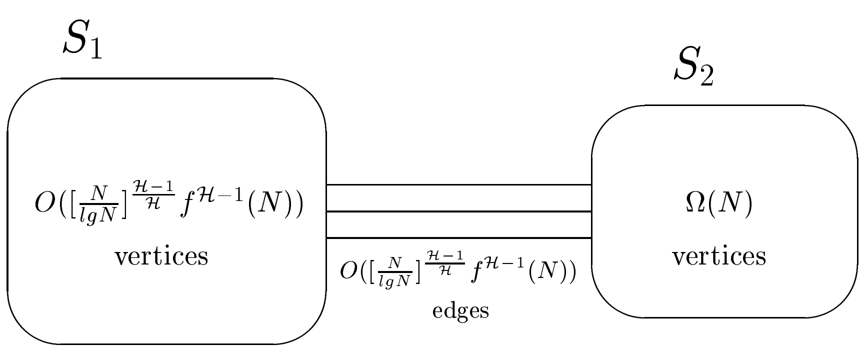


Fig.5.2: Edge Cut for sets S_1 and S_2

The vertices in S_2 must be connected on last single hop to vertices in S_1. Since there are $\Omega(N)$ vertices in S_2, there must be at least $\Omega(N)$ virtual paths going through $O([\frac{N}{lgN}]^{\frac{\mathcal{H}-1}{\mathcal{H}}} f^{\mathcal{H}-1}(N))$ connecting edges. Thus we get lower bound for the load of these connection edges equal to

$$\mathcal{L} = \frac{\Omega(N)}{O([\frac{N}{lgN}]^{\frac{\mathcal{H}-1}{\mathcal{H}}} f^{\mathcal{H}-1}(N))} = \Omega\left(\frac{N}{[\frac{N}{lgN}]^{\frac{\mathcal{H}-1}{\mathcal{H}}} f^{\mathcal{H}-1}(N)}\right)$$

For $f(N) = lg^{\frac{1}{\mathcal{H}}} N$ we get $\mathcal{L} = \Omega([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$. At the beginning of the proof we assumed, that $f(N) = o(lg^{\frac{1}{\mathcal{H}}} N)$. So the edges on the cut have load at least $\mathcal{L} = \omega([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$. This is a contradiction, since we supposed, that $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} f(N))$ with $f(N) = o(lg^{\frac{1}{\mathcal{H}}} N)$ for all edges in VPL Ψ. Thus for all one-to-all VPLs for BF_n with load bounded by $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} f(N))$ we must have $f(N) = \Omega(lg^{\frac{1}{\mathcal{H}}} N)$ proving the theorem.□

Corollary 5.6: Let BF_n be a butterfly with N vertices. For every all-to-all VPL with upper bound on hop count $\mathcal{H} \geq 2$ it holds $\mathcal{L} = \Omega([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N) = \Omega(N^{\frac{1}{\mathcal{H}}})$.

Proof: Since every all-to-all VPL is also an one-to-all VPL from any vertex, the bound comes directly from the previous theorem.□

5.4 One-to-all VPLs for butterfly networks

In this section, we concentrate on one-to-all VPLs for butterfly networks. Firstly, we will look at some layouts connecting the vertex $v_{1,1}$ with the rest of the network. Later, we generalize these results for any root $v_{i,j}$. We will study only layouts with $\mathcal{H}_{max} \geq 2$, since construction of 1-hop layouts is straightforward (connect v_1 with every other vertex using some shortest path, this is always asymptoticaly optimal layout with $\mathcal{L} = \mathrm{O}(N)$).

We start with an interesting layout, which is both simple and quite good.

Layout 5.7: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count. We construct an one-to-all VPL from $v_{1,1}$ as follows:

1. Start with empty VPL Ψ
2. Construct an VPL for complete binary tree T_1 from vertex $v_{1,1}$ with upper bound for hop-count equal to $\mathcal{H} - 1$ using *Layout 4.8* . Add all VPs from the VPL for T_1 to the VPL Ψ.
3. For each column c of BF_n, let $v_{i_c,c} \in T_1$ be a vertex from T_1 on column c with the minimum possible row, exactly

   $$i_c = min\{i | v_{i,c} \in T_1\}$$

   Add paths $(v_{i_c,c}, v_{i_c-1,c})$,...,$(v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{1,c})$ to the VPL Ψ (these are one-hop layouts for chains $(v_{i_c,c}, \ldots, v_{1,c})$).

The resulting VPL is schematicaly shown on the *Fig.5.3*.

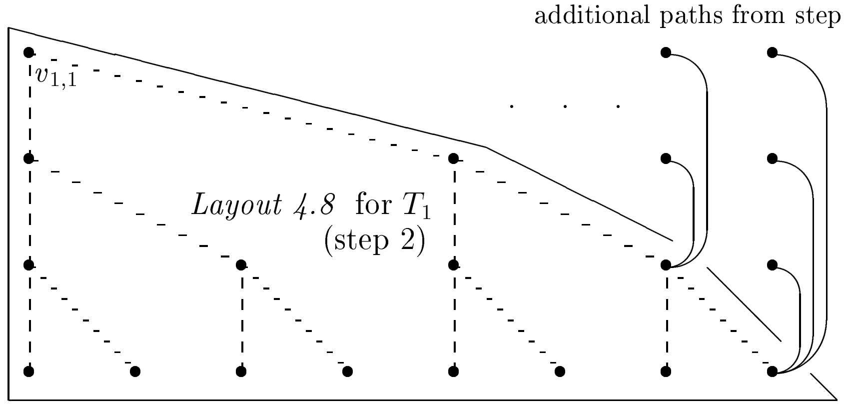


Fig.5.3: Basic one-to-all layout for butterfly

Analysis of VPL: Firstly, we prove the correctness of this layout. We use the following lemma:

Lemma 5.8: Let BF_n be a butterfly. Now take any vertex $v_{i,c} \in BF_n \setminus T_1$. Then $i < i_c$, where i_c is the same as in the previous layout.

Proof: Suppose, that $i \geq i_c$. From the definition of i_c we get $v_{i_c,c} \in T_1$. Now, from definition of T_1, all neighbours of $v_{i_c,c}$ in level $i_c + 1$ are in T_1. So $v_{i_c+1,c} \in T_1$. By induction, any vertex $v_{k,c},\ k \geq i_c$ is in T_1. This is a contradiction, since $v_{i,c} \in BF_n \setminus T_1$.□

Now, we show that we can get from $v_{1,1}$ to any vertex $v_{i,c}$ in at most $\mathcal{H}$ hops. Let $v_{i,c} \in BF_n \setminus \{v_{1,1}\}$ be any vertex in BF_n, except the root $v_{1,1}$. There are two possible situations.

Firstly assume, that $v_{i,c} \in T_1$. Then, we can get to the $v_{i,c}$ by at most $\mathcal{H} - 1$ hops using the shortest path. This follows directly from *Layout 4.8* , which is included in VPL Ψ.

Now, let $v_{i,c} \in BF_n \setminus T_1$. We connect $v_{1,1}$ with this vertex through vertex $v_{i_c,c}$. There is a connection from $v_{1,1}$ to $v_{i_c,c}$ with at most $\mathcal{H} - 1$ hops using the shortest path, since both vertices are in T_1 (see previous paragraph). Now we prolong this route by a path $(v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{i,c})$ (remember, that from *Lemma 5.8* it holds $i < i_c$). This path is in VPL Ψ due to the step 3 of layout construction. The resulting connection from $v_{1,1}$ to $v_{i,c}$ has at most $\mathcal{H} - 1 + 1 = \mathcal{H}$ hops. It is also easy to check, that it uses the shortest path between these vertices (the shortest paths in butterfly graphs are studied for example in [7]).

The last thing remaining is the load of this layout. The load of the *Layout 4.8* is $\mathrm{O}(N^{\frac{1}{\mathcal{H}}})$ for a tree with N vertices and hop-count at most $\mathcal{H}$. In our case, the tree T_1 has $\mathrm{O}(\frac{N}{n}) = \mathrm{O}(\frac{N}{lgN})$ vertices and the upper bound for hop-count in this tree is $\mathcal{H}-1$. So the resulting load for this tree from the step 2 of construction of VPL Ψ is $O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}-1}})$. All paths from the second step of construction use only edges of T_1. On the other hand, the paths added in step 3 have no edge in T_1, so their load is independent from the load of T_1. In each column, we have at most n edges forming the chain. The greatest load in each column c is on the bottom-most edge $(v_{i_c,c}, v_{i_c-1,c})$ and is at most $n-1$ (the number of paths from $v_{i_c,c}$). So the final load is the maximum of loads from steps 2 and 3, since no paths from different steps have a common edge.

$$\mathcal{L} = max(O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}-1}}), O(n))$$

and since $n = \Theta(lgN)$ (see *Theorem 5.4*) and $lgN = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}-1}})$, we have at last

$$\mathcal{L} = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}-1}})$$

with the hop-count bounded by $\mathcal{H}$ and $\phi = 1$ as shown above.

Claim 5.9: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count. According to the previous layout, we can construct an one-to-all VPL from $v_{1,1}$ for BF_n with $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}-1}})$.

Now, we will enhance previous layout a bit, giving better asymptotic result. The main idea is to connect the lower level of trees in the layout for T_1 directly with the chains from step 3, saving one hop. The hop is used to increase the upper bound of hop-count for the tree T_1.

Layout 5.10: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count.We construct an one-to-all VPL from $v_{1,1}$ as follows:

1. Start with empty VPL Ψ
2. Construct an VPL for complete binary tree T_1 from vertex $v_{1,1}$ with upper bound for hop-count equal to $\mathcal{H}$ using *Layout 4.8* . Add all VPs from the VPL for T_1 to the VPL Ψ.
3. For each column c of BF_n, let $v_{i_c,c} \in T_1$ be a vertex from T_1 on column c with the minimum possible row, exactly

$$i_c = min\{i | v_{i,c} \in T_1\}$$

If there exist connection form $v_{1,1}$ to $v_{i_c,c}$ with at most $\mathcal{H}-1$ hops (in the layout for T_1), add paths $(v_{i_c,c}, v_{i_c-1,c})$,...,$(v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{1,c})$

to the VPL Ψ (these are one-hop layouts for chains $(v_{i_c,c}, \ldots, v_{1,c})$). This is the case, when $v_{i_c,c}$ is not in the lowest level of trees in *Layout 4.8* $(i_c < n - k)$.

4. For the rest vertices $v_{i_c,c}$ (not included in the previous step) find the pivot $v_{i,j}$ of their subtree in the *Layout 4.8* . Now, for each path $(v_{i,j}, \ldots, v_{i_c,c})$ already in VPL Ψ (in fact, there is only one such path), add paths

$$(v_{i,j}, \ldots, v_{i_c,c}, v_{i_c-1,c}), \ldots, (v_{i,j}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{1,c})$$

so there is one-hop connection from the pivot $v_{i,j}$ to any vertex $v_{m,c}$ for $m \leq i_c$.

Analysis of VPL: Firstly, we show that we can get from $v_{1,1}$ to any vertex $v_{i,c}$ in at most $\mathcal{H}$ hops. The proof is similar to the one in the previous layout. There are three kinds of vertices:

- $v_{i,c} \in T_1$: The fact follows directly from the correctness of *Layout 4.8* and the fact, that it is used in VPL Ψ for the tree T_1.

- $v_{i,c} \in BF_n \setminus T_1$ and the corresponding $v_{i_c,c}$ is reachable from $v_{1,1}$ in at most $\mathcal{H} - 1$ hops (using layout for T_1): We use the VC from T_1 which connect $v_{1,1}$ and $v_{i_c,c}$ prolonging it with path $(v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{i,c})$ from step 3 of construction. Remeber, that from *Lemma 5.8* it holds $i < i_c$.

- $v_{i,c} \in BF_n \setminus T_1$ and the corresponding $v_{i_c,c}$ is reachable from $v_{1,1}$ in $\mathcal{H}$ hops (using layout for T_1): We take the VC from T_1 which connect $v_{1,1}$ and $v_{i_c,c}$. The last path in this VC is the path $(v_{m,j}, \ldots, v_{i_c,c})$, where $v_{m,j}$ is the pivot for the subtree (in *Layout 4.8* for T_1) which contains the vertex $v_{i_c,c}$. From the step 4 of construction, there is a path $(v_{m,j}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{i,c})$ in VPL Ψ. We use this path to replace the last path $(v_{m,j}, \ldots, v_{i_c,c})$ in VC between $v_{1,1}$ and $v_{i_c,c}$. The resulting VC connect $v_{1,1}$ and $v_{i,c}$ with at most $\mathcal{H}$ hops.

It is easy to check, that all above connections use the shortest paths in BF_n, like in the previous layout. We will take look at the load now.

The paths from step 3 have no common edge with any path from other steps. The worst load for them, as in the previous layout, is $\mathrm{O}(lgN)$.

Now we will look at the edges in the tree T_1. For any subtree from *Layout 4.8* , except trees on the lowest level, there are only paths from step 2, with the total load of $\mathrm{O}([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}})$. For the subtrees on the lowest level, each path is in the step 4 replaced by at most $\mathrm{O}(lgN)$ paths, so the load is at most $\mathrm{O}([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lgN)$. The final load is the maximum of the previous three loads, so we get

$$\mathcal{L} = \mathrm{O}(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lgN)$$

slightly better than the previous result. Comparing with the lower bound, we have only logarithmic factor between them.

Claim 5.11: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count. According to the previous layout, we can construct an one-to-all VPL from $v_{1,1}$ for BF_n with $\mathcal{L} = \mathrm{O}([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lgN)$.

Now, we are ready to present an asymptoticaly optimal layout for the butterfly topology from the vertex $v_{1,1}$. The main idea is to shrink a bit the lowermost level of trees in the VPL design for T_1, since there are the most loaded edges. This will, however, increase somewhat height of the trees from upper (not lowermost) levels. The layout is identical to the previous one (*Layout 5.10*) except for the step 2 of construction.

Layout 5.12: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count. We construct an one-to-all VPL from $v_{1,1}$ as follows:

1. same as in *Layout 5.10*
2. Construct an VPL for complete binary tree T_1 from vertex $v_{1,1}$ with upper bound for hop-count equal to $\mathcal{H}$ using slightly modified *Layout 4.8* . In the *Layout 4.8* we divided the tree T_1 into $\mathcal{H}$ levels of trees with depth equal to $\lfloor\frac{l-1}{\mathcal{H}}\rfloor + 1$ or $\lfloor\frac{l-1}{\mathcal{H}}\rfloor + 2$ (l is the depth of T_1). In our modification, we increase the depth of trees from *Layout 4.8* by a factor of $lglg^{\frac{1}{\mathcal{H}}}N$ (N is the number of vertices in the whole butterfly) except for the lowermost level. The lowermost level is decreased by a factor of $(\mathcal{H}-1)lglg^{\frac{1}{\mathcal{H}}}N$, so the number of levels does not change. Exactly, the topmost $\mathcal{H}-1$ levels of trees have a depth of $\lfloor\frac{l-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}}N\rfloor + 1[+1]$ and the lowermost level of trees have a depth of $\lfloor\frac{l-1}{\mathcal{H}} - (\mathcal{H}-1)lglg^{\frac{1}{\mathcal{H}}}N\rfloor + 1[+1]$. See Fig.5.4. Finally, add all VPs from the VPL for T_1 to the VPL Ψ.
3. same as in *Layout 5.10*
4. same as in *Layout 5.10*

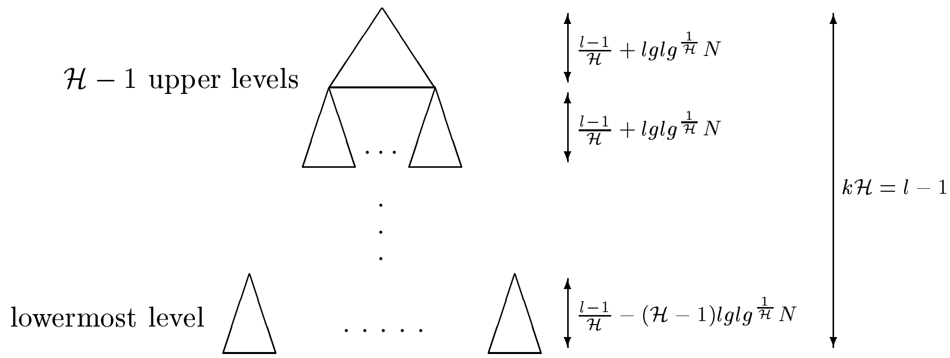


Fig.5.4:Tree T_1 partitioned into the levels of trees

Analysis of VPL: We can get from $v_{1,1}$ to any vertex $v_{i,c}$ in at most $\mathcal{H}$ hops. The proof is exactly the same as in the *Layout 5.10* . We will concentrate on the load now.

The paths from step 3 have are edge disjoint with any path from other steps. The worst load for them, as in the previous layout, is O(lgN).

Now we will look at the edges in the tree T_1. For any subtree from the *modified Layout 4.8* for T_1, except trees on the lowest level, there are only paths from step 2. Their load can be computed as:

upper levels : $\mathcal{L} = O(2^{(\frac{l-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}}N)}) = O(2^{\frac{l-1}{\mathcal{H}}}.2^{lglg^{\frac{1}{\mathcal{H}}}N}) = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}}N)$

since the load of each subtree is equal to number of its vertices (on upper levels), that is $2^{(\frac{l-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}}N)}$, and it holds $2^l = \Theta(\frac{N}{lgN})$.

For the subtrees on the lowest level, each path in the step 4 is replaced by at most O(lgN) paths, so the load is at most:

$$\mathcal{L} = O(lgN.2^{(\frac{l-1}{\mathcal{H}} - lglg^{\frac{\mathcal{H}-1}{\mathcal{H}}}N)}) = O(lgN.\frac{2^{\frac{l}{\mathcal{H}}}}{2^{lglg^{\frac{\mathcal{H}-1}{\mathcal{H}}}N}}) = O(lgN.\frac{\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}}}{lg^{\frac{\mathcal{H}-1}{\mathcal{H}}}N}) =$$

$$= O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}}N)$$

Since both these loads are equal, the final load is (maximum of the previous three loads)

$$\mathcal{L} = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}}N) = O(N^{\frac{1}{\mathcal{H}}})$$

which is asymptoticaly optimal layout recalling the lower bound from *Theorem 5.5* .

Claim 5.13: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count. According to the previous layout and recalling the *Theorem 5.5* , we can construct an one-to-all VPL from $v_{1,1}$ for BF_n with $\mathcal{L} = \Theta(N^{\frac{1}{\mathcal{H}}})$.

Note, that the same scheme could be used for any vertex from butterfly on the first (topmost) level, using automorphism from Appendix A - Column symmetry. Then, applying automorphism from Appendix A - Bottom Up symmetry, we get a VPL from any vertex in last (bottommost) level, too.

Finally, we are ready for an asymptoticaly optimal one-to-all VPL from any vertex of the butterfly network. We begin with some definitions.

Definition 5.14: Let BF_n be a butterfly and $v_{r,c} \in BF_n$ be any vertex from it. We define $T_{r,c}$ to be a complete binary tree rooted at $v_{r,c}$ spreading downwards in BF_n (to rows $r+1$,$r+2$,...,n) as in *Definition 5.1* . Similary, we define $T'_{r,c}$ to be a complete binary tree rooted at $v_{r,c}$ going upwards.

Recalling *Definition 5.1* we have $T_{1,c} \equiv T_c$ and $T'_{n,c} \equiv T'_c$.

Definition 5.15: Let BF_n be a butterfly and $v_{r,c} \in BF_n$ be any vertex from it. We define $T_{r,c}[p]$ to be a complete binary tree rooted at $v_{r,c}$ spreading downwards in BF_n with p levels. Similary we define $T'_{r,c}[p]$ which spreads upwards.

The following layout is quite complicated. The reader is suggested to assume that $K < \mathcal{H}$ (in step 2) for the first time and ignore all parts (in design and analysis) which concern the $K = \mathcal{H}$ possibility. Once the layout is understood in this way, the $K = \mathcal{H}$ possibility should be taken into account.

Layout 5.16: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count. Let $R \in BF_n$ be any vertex of the leftmost column in BF_n ($R \equiv v_{k,1}$ for some $1 \leq k \leq n$). We construct an one-to-all VPL from R as follows:

1. Start with empty VPL Ψ
2. Construct an VPL for BF_n from vertex $v_{n,1}$ with upper bound for hop-count equal to $\mathcal{H}$ using *Layout 5.12* . Add all VPs from this VPL to the VPL Ψ. This layout divides the butterfly into $\mathcal{H}$ levels (see

Fig.5.5). We define j as

$$j = max\{v_{j,1} \text{ } is \text{ } pivot \text{ } in \text{ } Layout \text{ } 5.12 \mid j \leq k\}$$

Let K be the number of levels (from *Layout 5.12*) below $v_{j,1}$ in BF_n. In our example, $K = 4$, since there are three levels of pivots below $v_{j,1}$. It is possible to have $K = \mathcal{H}$, in this case $j = 1$.

3. Remove VPs added in step 2 which include any of the following vertices:

$$\{v_{r,c}| \ k \leq r \leq n \ \wedge \ 1 \leq c \leq 2^{n-k}\}$$

We will not need these paths in our construction. However, this step is optional, since leaving these paths in our VPL Ψ will not affect asymptotical optimality.

4. Construct an VPL for complete binary tree $T_{k,1}$ (see *Definition 5.14*) with upper bound for hop-count equal to K (from step 2) using slightly modified *Layout 4.8* . In our modification, we divide the tree $T_{k,1}$ into $K-1$ levels of height $\frac{n-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}} N$ (equal to the height of levels in layout for BF_n from step 2) and the last Kth level with height $\frac{n-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}} N - (k-j)$ if $K < \mathcal{H}$ or $\frac{n-1}{\mathcal{H}} - (\mathcal{H}-1)lglg^{\frac{1}{\mathcal{H}}} N - (k-j)$ if $K = \mathcal{H}$. The k is taken from initial assumptions, j is from step 2. Add these VPs to the VPL Ψ.

5. For each column $1 \leq c \leq 2^{n-k}$ of BF_n, let $v_{i_c,c} \in T_{k,1}$ be a vertex from $T_{k,1}$ on column c with the minimum possible row, exactly

$$i_c = min\{i | v_{i,c} \in T_{k,1}\}$$

Add paths $(v_{i_c,c}, v_{i_c-1,c})$,...,$(v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{k,c})$ to the VPL Ψ (these are one-hop layouts for chains $(v_{i_c,c}, \ldots, v_{k,c})$).

6. For each such column $1 \leq c \leq 2^{n-k}$ of BF_n, that there exist connection form $v_{k,1}$ to $v_{i_c,c}$ with at most $K-1$ hops (in the layout for $T_{k,1}$), find the pivot $v_{i,y}$ of $v_{i_c,c}$'s subtree in *Layout 4.8* . Add path

$$(v_{i,y}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{k,c})$$

to the VPL Ψ. Construct one-hop one-to-all VPL for tree $T'_{k,c}[k-j]$ using *Layout 4.5* . Add paths from this VPL to the VPL Ψ. Moreover, if $K = \mathcal{H}$, for each column d (except of column c) of the tree $T'_{k,c}[k-j]$ find a vertex $v_{i_d,d} \in T'_{k,c}[k-j]$ with maximal possible row i_d

$$i_d = max\{a \mid v_{a,d} \in T'_{k,c}[k-j]\}$$

and add paths

$$(v_{k,c}, \ldots, v_{i_d,d}, v_{i_d+1,d}), \ldots, (v_{k,c}, \ldots, v_{i_d,d}, v_{i_d+1,d}, \ldots, v_{n,d})$$

to the VPL Ψ.

7. For each such $v_{i_c,c}$, $1 \le c \le 2^{n-k}$, that we can get from $v_{k,1}$ to the $v_{i_c,c}$ at exactly K hops ($v_{i_c,c}$ is at the lowest level in *Layout 4.8*) find the pivot $v_{i,y}$ of $v_{i_c,c}$'s subtree in *Layout 4.8* . Now

$$\forall v_{a,b} \in T'_{k,c}[k-j] \text{ add path } (v_{i,y}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{k,c}, \ldots, v_{a,b})$$

Moreover, if $K = \mathcal{H}$,

$$\forall v_{a,c} \in BF_n,\ k < a < i_c \text{ add path } (v_{i,y}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{a,c})$$

so in this case the chain (see step 5) is reachable directly from $v_{i,y}$. Finally, still only if $K = \mathcal{H}$, for each column d (except of column c) of the tree $T'_{k,c}[k-j]$ find a vertex $v_{i_d,d} \in T'_{k,c}[k-j]$ with maximal possible row i_d

$$i_d = max\{a \mid v_{a,d} \in T'_{k,c}[k-j]\}$$

and add paths

$$(v_{i,y}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{k,c}, \ldots, v_{i_d,d}, v_{i_d+1,d}), \ldots$$

$$\ldots, (v_{i,y}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{k,c}, \ldots, v_{i_d,d}, v_{i_d+1,d}, \ldots, v_{n,d})$$

to the VPL Ψ.

The layout is schematicaly shown in *Fig.5.5.*

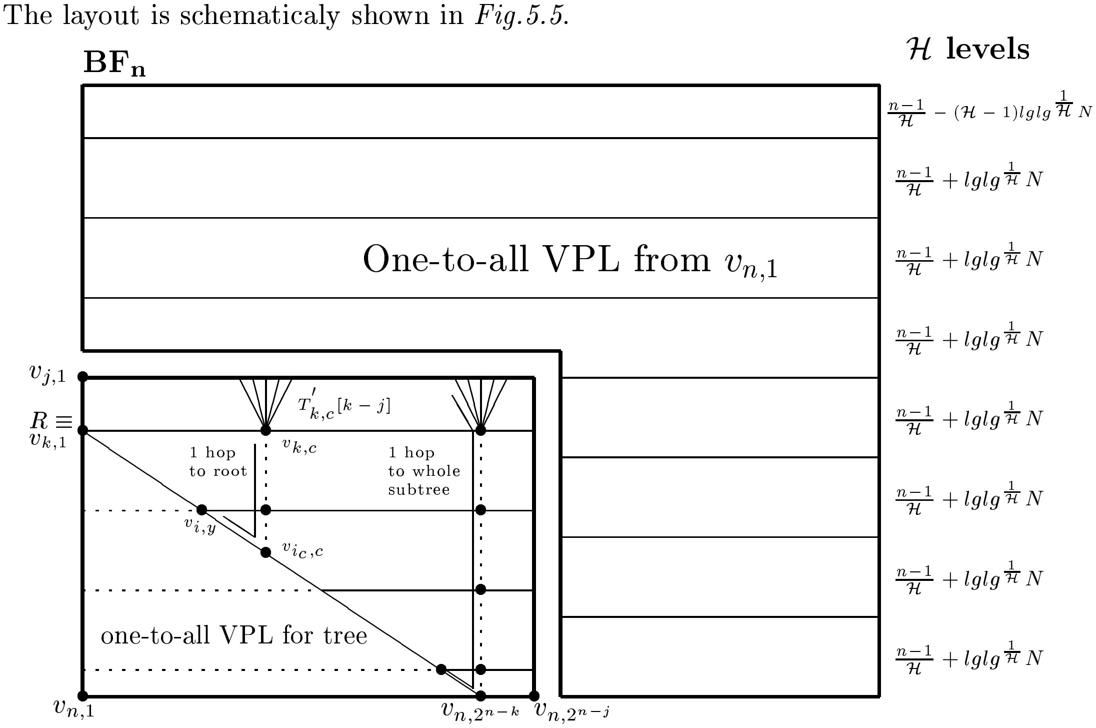


Fig.5.5: One-to-all VPL for BF_n from any vertex ($\mathcal{H} = 8$)

Analysis of VPL:

PART A - correctnes of VPL

Let $v_{r,c}$ be any vertex of BF_n. It is included in at least one of the following sets:

- Complete binary tree $T_{k,1}$. In this case, we can get from the $v_{k,1}$ to the $v_{r,c}$ in at most K hops using layout from step 4. The VPL for complete binary trees uses the shortest paths.

- Chain $(v_{i_c,c}, \ldots, v_{k,c})$. We can get from $v_{k,1}$ to $v_{i_c,c}$ in at most K hops (step 4). If $K < \mathcal{H}$ one more hop is needed from $v_{i_c,c}$ to the $v_{r,c}$ from step 5. Number of hops used is at most $K + 1 \leq \mathcal{H}$. If $K = \mathcal{H}$ find $v_{i_c,c}$'s pivot $v_{i,y}$ in $T_{k,1}$. We can get from $v_{k,1}$ to $v_{i,y}$ in at most $K - 1$ hops. Since $K = \mathcal{H}$, there is a single hop path from $v_{i,y}$ to $v_{r,c}$ (from step 7). So $\mathcal{H}$ hops are needed in this case.
  All these vertices are in "subbutterfly"

  $$\{v_{a,b} | \; k \leq a \leq n \;\wedge\; 1 \leq b \leq 2^{n-k}\}$$

  and it is easy to check, that they use shortest path from $v_{k,1}$ to $v_{r,c}$.

- One of the subtrees $T^{'}_{k,b}[k-j]$. We firstly find pivot $v_{i,y}$ for vertex $v_{i_b,b}$ from step 6 or step 7 (this pivot could be also the vertex $v_{k,1}$ itself). We begin with VPs from $v_{k,1}$ to $v_{i,y}$. If it takes $K-1$ hops (K hops to $v_{i_b,b}$), then we can get from $v_{i,y}$ to $v_{r,c}$ through vertices $v_{i_b,b}$ and $v_{k,b}$ in one hop (step 7). If, on the other hand, VP from $v_{k,1}$ to $v_{i,y}$ takes at most $K - 2$ hops, we can add two VPs, $v_{i,y}$ *to* $v_{k,b}$ and $v_{k,b}$ *to* $v_{r,c}$ from step 6. In both cases we use at most K hops. It is important that we can get to any vertex $v_{j,c}$, $c \leq 2^{n-j}$ in at most K hops. It is still easy to see, that we use shortest paths (for detail on the shortest paths in butterfly networks see [7]).

- Vertices in set $\{v_{a,b} | \; j \leq a \leq n \;\wedge\; 2^{n-k} < b \leq 2^{n-j}\}$ not included in previous step. We set

  $$i_r = max\{a \mid v_{a,c} \in T^{'}_{k,c \; mod \; 2^{n-k}}[k-j]\}$$

  Such index exists, since $v_{j,c} \in T^{'}_{k,c \; mod \; 2^{n-k}}[k-j]$. Now, if $K < \mathcal{H}$, we construct an VP from $v_{k,1}$ to $v_{i_r,c}$ with at most K hops (see previous step) and add one hop from $v_{i_r,c}$ to $v_{r,c}$ from step 2 of construction for total of $K + 1 \leq \mathcal{H}$ hops. If $K = \mathcal{H}$, let $b = c \; mod \; 2^{n-k}$, so $v_{i_r,c} \in T^{'}_{k,b}[k-j]$. Let

  $$i_b = min\{a \mid v_{a,b} \in T_{k,1}\}$$

  and let $v_{i,y}$ be $v_{i_b,b}$'s pivot in tree $T_{k,1}$. Then there exist a path from $v_{k,1}$ to $v_{i,y}$ in at most $K - 1$ hops (step 4) and a single-hop path from $v_{i,y}$ to $v_{r,c}$ through vertices $v_{i_b,b}, v_{k,b}$, and $v_{i_r,c}$ from step 7.

- The rest vertices. If $K = \mathcal{H}$ this set is empty, so we can assume, that $K < \mathcal{H}$. These vertices are from the larger part of *Fig.5.5*. Let q denote the shortest path between $v_{n,1}$ and $v_{r,s}$ used in layout from step 2 to connect these vertices. We set

  $$x = min\{a \mid 1 \leq a \leq 2^{n-j} \;\wedge\; v_{j,a} \in q\}$$

  The minimum operator is only for syntax, since there is exactly one vertex in the specified set. The set is not empty, because the set $\{v_{j,a} \mid 1 \leq a \leq 2^{n-j}\}$ is an vertex cut in BF_n, so any path from $v_{n,1}$ to $v_{r,s}$ go through it. Firstly we connect $v_{k,1}$ with $v_{j,x}$ with at most K hops (see third item on this list) and from $v_{j,x}$ to $v_{r,c}$ we use VPs from step 2 (the rest of the path q). We can do it, since $v_{j,x}$ is pivot in layout from step 2. The connection of $v_{n,1}$ and $v_{r,c}$ used at most $\mathcal{H}$ hops. The connection from $v_{n,1}$ to $v_{j,x}$ uses K hops, so the rest of the path - from $v_{j,x}$ to $v_{r,s}$ is in at most $\mathcal{H} - K$ hops. Combining with path from $v_{k,1}$ to $v_{j,x}$, we can get from $v_{k,1}$ to $v_{r,s}$ in at most $K + \mathcal{H} - K = \mathcal{H}$ hops.

PART B - shortest path analysis

Lemma 5.17: Let $v_{r,c} \in \{v_{a,b} \mid 1 \leq a \leq k \wedge\; 1 \leq b \leq 2^{n-1}\}$. Then we can get from $v_{k,1}$ to $v_{r,c}$ in at most $\mathcal{H}$ hops using shortest path.

Proof: For detailed description of shortest paths in butterfly topology see *Appendix B* and [7]. Let p be shortest path between $v_{k,1}$ and $v_{r,c}$, $v_{r,c} \in \{v_{a,b} \mid 1 \leq a \leq k \wedge\; 1 \leq b \leq 2^{n-1}\}$. There are three possibilities

- The path p does not change direction (each row between k and r is visited exactly once). Combine VPL from step 6 and 2 to get $\mathcal{H} - K + 1$ hop layout for tree $T'_{k,1}$. Since $v_{r,c} \in T'_{k,1}$ (because path does not change direction), we can use this combined VPL to get from $v_{k,1}$ to $v_{r,c}$. Since it is common VPL for tree, the used path is the shortest one.

- The path p changes (top-down) direction once.

  - The path start going up (decrease row). It can be transformed into path p_2, which changes column only before changing direction. This can be done due to $r \leq k$. The combined VPL from step 6 and step 2 is again useful. If $K < \mathcal{H}$, we get firstly from $v_{k,1}$ to $v_{1,c}$ using VPL for $T'_{k,1}$. The rest of the path p_2 is straight chain on column c. One hop path from the layout in step 2 can be used to get from $v_{1,c}$ to $v_{r,c}$. If $K = \mathcal{H}$ we use only layout from step 6 and whole procedure ($v_{k,1}$ to $v_{1,c}$ to $v_{r,c}$) can be made on single hop.

- The path start downward (increasing row). This is identical with downward path in the following case.

- The path p changes (top-down) direction twice.
  - The path start going up (decrease row). From the properties of shortest paths (see *Appendix B*), the path must finish at row k (or below, when $r > k$, but this is not case of the Lemma). Such path can be transformed into path p_2, which start going downwards (the necessary column changes on rows $\geq k$ are taken first). Use p_2 in the following case.
  - The path start going down (increase row). If $v_{r,c} \in \{v_{a,b} \mid j \leq a \leq k \wedge\ 1 \leq b \leq 2^{n-j}\}$, we can use connection from part A, the last but one case (the path from that construction has the same length as p). Otherwise, we use connection from part A, the last case. Again, the segmets between $v_{k,1}$ to $v_{j,x}$ and $v_{j,x}$ to $v_{r,c}$ in the path p might be replaced by equaly long segments from this connection (The rows are not changed, only columns are shifting differently).

This property is exploited in *Layout 5.19* to get an VPL which uses only the shortest paths for routing.

PART C - load analysis

We will look at the load contributed from each step of construction.

- Step 1. $\mathcal{L} = 0$.
- Step 2. $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$ since it is the load of *Layout 5.12* .
- Step 3. $\mathcal{L} = 0$. We only remove paths.
- Step 4. $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$ since the largest level of the tree $T_{k,1}$ has $\frac{n-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}} N$ rows (see *Layout 5.12*).
- Step 5. $\mathcal{L} = O(lgN)$, it is the length of chains.
- Step 6. $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$. If $K < \mathcal{H}$, the trees $T^{'}_{k,c}[k-j]$ have at most $\frac{n-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}} N$ rows, so $\mathcal{L}$ is as stated. If $K = \mathcal{H}$, the trees $T^{'}_{k,c}[k-j]$ have at most $\frac{n-1}{\mathcal{H}} - lglg^{\frac{\mathcal{H}-1}{\mathcal{H}}} N$ rows. To each path at most lgN new paths are added, hence $\mathcal{L} = O(lgN \cdot 2^{(\frac{n-1}{\mathcal{H}} - lglg^{\frac{\mathcal{H}-1}{\mathcal{H}}} N)})$ as stated (see *Layout 5.12*).
- Step 7. $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$. Let $a = k - j$ and let b be the number of rows on the lowest level in tree $T_{k,1}$ from the *Layout 5.8* in step

4 of construction. So $a+b = \frac{n-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}}N$ if $K < \mathcal{H}$ and $a+b = \frac{n-1}{\mathcal{H}} - lglg^{\frac{\mathcal{H}-1}{\mathcal{H}}}N$ if $K = \mathcal{H}$. If $K < \mathcal{H}$, each path from $T_{i,y}[b]$ is prolonged by at most 2^a paths (vertices of $T'_{k,c}[a]$), leading finally to $\mathcal{L} = O(2^a 2^b) = O(2^{a+b}) = O(2^{(\frac{n-1}{\mathcal{H}} + lglg^{\frac{1}{\mathcal{H}}}N)})$ as stated (see *Layout 5.12*). If $K = \mathcal{H}$, each path from $T_{i,y}[b]$ is prolonged by at most 2^a paths to $T'_{k,c}[a]$, which are further prolonged by another $O(lgN)$ paths to chain $(v_{i_d,d}, \ldots, v_{n,d})$. Independently, each path from $T_{i,y}[b]$ is prolonged by $O(lgN)$ paths to chain $(v_{i_c,c}, \ldots, v_{k,c})$. So we have finally $\mathcal{L} = O(2^a(2^b lgN + lgN)) = O(2^{a+b} lgN) = O(2^{(\frac{n-1}{\mathcal{H}} - lglg^{\frac{\mathcal{H}-1}{\mathcal{H}}}N)} lgN)$ as stated (see *Layout 5.12*).

Each step has a load of at most $O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}}N)$, so we have

$$\mathcal{L} = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}}N) = O(N^{\frac{1}{\mathcal{H}}})$$

Claim 5.18: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count and k be arbitrary number, $1 \leq k \leq n-1$. According to the previous layout and recalling the *Theorem 5.5* , we can construct an one-to-all VPL from $v_{k,1}$ for BF_n with $\mathcal{L} = \Theta(N^{\frac{1}{\mathcal{H}}})$.

Note, that the same scheme could be used for any vertex $v_{r,c}$ from BF_n using automorphism which maps $v_{r,c}$ into the vertex $v_{r,1}$ (see *Appendix A - Column symmetry*).

In the following VPL we exploit previous layout to get asymptoticaly optimal one-to-all VPL from any vertex which uses shortest paths for routing.

Layout 5.19: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count. Let $R \in BF_n$ be any vertex of the leftmost column in BF_n ($R \equiv v_{k,1}$ for some $1 \leq k \leq n$). We construct an one-to-all VPL from R as follows:

1. Start with empty VPL Ψ
2. Construct one-to-all VPL for BF_n from vertex $v_{k,1}$ with upper bound for hop-count equal to $\mathcal{H}$ using *Layout 5.16* . Add all VPs from it to VPL Ψ.
3. Construct one-to-all VPL for BF_n from vertex $v_{n-k,1}$ with upper bound for hop-count equal to $\mathcal{H}$ using *Layout 5.16* . Change top-down orientation of BF_n (use a bijection $v_{r,c} \rightarrow v_{n-r,1+rev(c-1)}$ where rev is reverse function for binary numbers, see *Appendix A - Bottom-up symmetry*). Now $v_{n-k,1}$ match the vertex $v_{k,1}$ from the previous step. Add all VPs (after change of orientation) to the VPL Ψ.

Analysis of VPL: The load $\mathcal{L}$ is at most twice of the load from *Layout 5.16* , so it still holds

$$\mathcal{L} = O(\left\lceil \frac{N}{lgN} \right\rceil^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N) = O(N^{\frac{1}{\mathcal{H}}})$$

Similary, we can still get from $v_{k,1}$ to any vertex in at most $\mathcal{H}$ hops, we only have more alternatives.

According to *Lemma 5.17* and layout from step 2, we can get to any vertex $v_{r,c}$ with $r \leq k$ using shortest path. Similary, according to *Lemma 5.17* and layout from step 3, we can get to any vertex $v_{r,c}$ with $r \geq k$ using shortest path. So we can get to any vertex of BF_n using the shortest path.

Theorem 5.20: Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop-count and k be arbitrary number $1 \leq k \leq n-1$. According to the previous layout and recalling the *Theorem 5.5* , we can construct an one-to-all VPL from $v_{k,1}$ for BF_n with $\mathcal{L} = \Theta(N^{\frac{1}{\mathcal{H}}})$ in which the shortest paths are used for routing.

Note, that the same scheme could be used for any vertex $v_{r,c}$ from BF_n by mapping it firstly into the vertex $v_{r,1}$ (see *Appendix A - Column symmetry*).

6 Conclusion

We presented some virtual path layouts for complete binary trees. Next we proved a lower bound $\mathcal{L} = \Omega(N^{\frac{1}{\mathcal{H}}})$ for one-to-all VPLs on butterfly topology. Then we presented different one-to-all VPLs for butterfly networks, leading finally to an asymptoticaly optimal one-to-all VPL for butterfly topology with $\mathcal{L} = \Theta(N^{\frac{1}{\mathcal{H}}})$.

Appendix A - Butterfly symmetries

In the following text, we assume that the columns of butterfly BF_n are numered from 0 to $2^{n-1} - 1$, not from 1 to 2^{n-1} as in previous sections. This is because we adapt binary representation of the column index. So in BF_n, there is an edge between two vertices iff they are in consecutive i-th and $(i+1)$-st levels, respectively, and their labels are either equal or differ in the i-th most significant bit. In our $v_{row,column}$ represantion, there are edges $(v_{i,c}, v_{i+1,c})$ and $(v_{i,c}, v_{i+1,c\ xor\ 2^{n-1-i}})$ where xor stands for *bitwise exclusive or*.

Bottom-Up symmetry

Let BF_n be a butterfly. We define following bijection on its vertices:

$$f(r, c) = (n - r + 1, rev(c))$$

where rev stand for binary $reverse$ function. We show, that it is an automorphism on BF_n. Let e be an edge of butterfly BF_n. There are two possibilities:

- Edge $e = (v_{i,c}, v_{i+1,c})$. After apllying f to its vertices, we get $f(e) = (v_{n-i+1,rev(c)}, v_{n-i,rev(c)})$. This is an edge in BF_n, since the rows differ only by one and the columns are the same.

- Edge $e = (v_{i,c}, v_{i+1,c\ xor\ 2^{n-1-i}})$. After apllying f to its vertices, we get $f(e) = (v_{n-i+1,rev(c)}, v_{n-i,rev(c\ xor\ 2^{n-1-i})}) = (v_{n-i+1,rev(c)}, v_{n-i,rev(c)\ xor\ 2^{i-1}})$. This is an edge in BF_n, since the rows differ only by one and the columns differ on the $n-i$-th most significant bit.

The bijection f changes top-down orientation of BF_n (the first row becomes last and vice versa). What is important for our layouts, all vertices on first column (in our case column 0) remain there ($rev(0) = 0$).

Column symmetry

Let BF_n be a butterfly. Let $0 \leq x \leq 2^{n-1} - 1$ be arbitrary column of BF_n. We define following bijection on its vertices:

$$f(r, c) = (r, c\ xor\ x)$$

This is an automorphism, since the rows remain unchanged and if two columns c_1 and c_2 differ, they will differ in exactly the same bits after the apllication of the function f. Notice, that the vertices in x-th column are mapped into the vertices of the first (number 0) column of $BF - n$ by the bijection f.

Appendix B - Shortest paths in butterfly

Let v_{r_1,c_1}, v_{r_2,c_2} be two vertices in BF_n. Let r_{min} be the lowest bit in which c_1 and c_2 differ, r_{max} be the highest such bit. If $r_1 < r_2$, the shortest path from v_{r_1,c_1} to v_{r_2,c_2} start upwards to the level r_{min}, then turns downward toward level r_{max} and finaly turning back upwards on the r_{max}-th level to reach v_{r_2,c_2}. if $r_1 > r_2$ the paths starts going downwards to level r_{max} then upwards to r_{min} and finally downwards toward v_{r_2,c_2}. In each row we change the correspoding bit of the column c_1 if necessary to become finaly the column c_2. However, we might pass some levels more times, so the path is not always unique. For detailed description see [7]. In the case of $r_1 = r_2$ we could start either up or down, the choice is ours, and then follow the previous description. There are at most two bottom-up turns on the shortest path and each level of BF_n is crossed at most twice, once upwards and once downwards.

References

[1] Shmuel Zaks. *Path layout in ATM Networks - A Survey.* The DIMACS Workshop on Networks in Distributed Computing, DIMACS Center, Rutgers University, Oct. 1997.

[2] Ornan Gerstel and Shmuel Zaks. *Path Layout in ATM Networks.* International Colloquium on Structural Information and Communication Complexity (SIROCCO), 1994.

[3] Israel Cidon, Ornan Gerstel and Shmuel Zaks. *A scalable approach to routing in ATM networks.* 8th International Workshop on Distributed Algorithms (WDAG), Terschelling, Netherlands, October 1994, (LNCS 857) pp. 209-222.

[4] Ladislav Stacho and Imrich Vrťo. *Virtual Path Layouts for some bounded degree networks.* In Proc. 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO), Siena, Italy, June, 1996.

[5] Evangelos Kranakis, Danny Krizanc and Andrzej Pelc. *Hop-congestion tradeoffs for high-speed networks.* 7th IEEE Symp. on Parallel and Distributed Processing, 1995, pp. 662-668.

[6] Ornan Gerstel, Israel Cidon and Shmuel Zaks. *The layout of virtual paths in ATM networks.* ACM/IEEE Transactions on networking, 4(6):873-884, December 1996. A preliminary version of this paper appeared in [3].

[7] Rastislav Královič, Peter Ružička and Daniel Štefankovič. *The Complexity of Shortest Path and Dilation Bounded Interval Routing.* Theoretical Computer Science 234(2000), 85-107.

[8] Jean-Claude Bermond, Nausica Marlin, David Peleg and Stéphane Perennes. *Directed Virtual Path Layouts in ATM Networks.* 12th International Symposium on Distributed Computing (DISC), Andros, Greece, September 1998, pp.75-88.

[9] Tamar Eilam, Michele Flammini and Shmuel Zaks. *A Complete Characterization of the Path Layout Construction Problem for ATM Networks with Given Hop Count and Load.* 24th International Colloq. on Automata, Languages, and Programming, 1997.

[10] Ornan Gerstel, Avishai Wool and Shmuel Zaks. *Optimal Layouts on a Chain ATM Network.* 3rd Annual Europan Symposium on Algorithms (ESA), Corfu, Greece, September 1995, (LNCS 979) pp. 508-522. To appear in Discrete Appliead Mathematics (DAM).

[11] Leszek Gasieniec, Evangelos Kranakis, Danny Krizanc and Andrzej Pelc. *Minimizing Congestion of Layouts for ATM Networks with Faulty Links.* Proc. 21st International Symposium on Mathematical Foundations of Computer Science, Lecture notes in Computer Science, Springer Verlag, Berlin, 1996.