On Efficiency of Virtual Path Layouts in Bounded Degree ATM Networks: A Case Study for Butterfly Networks*

Martin Makúch
Comenius University, Bratislava

Peter Ružička
Comenius University, Bratislava

Abstract

The problem to design virtual path layouts in ATM networks has been intensively studied with respect to the size N, load $\mathcal{L}$ and hop count $\mathcal{H}$. The previous result due to Stacho and Vrťo [5] presents all-to-all virtual path layouts for some bounded degree networks in the form $\mathcal{H} = \Theta(\frac{\log N}{\log \mathcal{L}})$ for any $\mathcal{L}$. This general result holds for not necessarily shortest paths gossip on networks of $O(\log N)$ bounded diameter and so it can be directly applied also to butterfly topologies. However, the lower bound holds even for one-to-all virtual path layouts thus the result is tight even in this case.

In this paper, we give an optimal shortest path broadcast layout on butterfly ATM networks with load $\mathcal{L} = \Theta(N^{1/\mathcal{H}})$ for any hop count $\mathcal{H}$. That is, for specific butterfly networks it holds $\mathcal{H} = \frac{\log N}{\log \mathcal{L} - O(1)}$ even for shortest path layouts. The question remains whether such improved result holds also for other topologies of $O(1)$ bounded degree and $O(\log N)$ bounded diameter.

Keywords

ATM networks, Virtual Path Layouts, Broadcast

1 Introduction

Recent development in fiber optic media offers dramatical changes in the area of digital communication networks. The sharp distinction between computer networks, telephone networks and cable TV networks has been

*Supported in part by grand from VEGA 1/7155/20.

replaced by a unified approach. The new technology is called ATM (Asynchronous Transfer Mode). It allows very reliable transmission together with high bandwith.

In the *routing problem* for ATM networks, for certain pairs of nodes the end-to-end communication is done along predefined paths in the network, so called *virtual paths*. The problem is to design these paths optimally. The smallest number of concatenated paths between two nodes is called the *hop count*, while the *load* of a layout is the maximum number of virtual paths along any communication link. The hop count relates to the time needed to establish a connection between two nodes and the load measures the size of routing tables at nodes.

The problem to design virtual path layouts in ATM networks has been intensively studied with respect to the network size N, load $\mathcal{L}$ and hop count $\mathcal{H}$ (for results see representative overview [1]). The general result due to Stacho and Vrťo [5] presents all-to-all virtual path layouts for some bounded degree networks in the form $\mathcal{H} = \Theta(\frac{\log N}{\log \mathcal{L}})$ for any $\mathcal{L}$. This result holds for not necessarily shortest path gossip on networks of $O(\log N)$ bounded diameter and so it can be directly applied also to butterfly networks. However, the lower bound holds even for one-to-all virtual path layouts, that means this result is optimal even for broadcast.

The interesting question is what happens in the reverse case, namely, "given a hop number minimize the load". The paper suggests that from the theoretical point of view, it is more reasonable (and harder) to study this problem. A good solution to this problem may provide an efficient solution to the converse one, but not vice versa.

In this paper, we present optimal shortest path broadcast layouts on butterfly networks in the form $\mathcal{L} = \Theta(N^{1/\mathcal{H}})$ for any $\mathcal{H}$. That means, for specific butterfly networks it holds $\mathcal{H} = \frac{\log N}{\log \mathcal{L} - O(1)}$ for any $\mathcal{L}$ even for shortest path layouts. The construction can be applied also to some related networks, as wrapped butterflies or cube connected cycles, within the same bound, however by loosing the shortest path property.

There are some questions left open. The main open problem remains whether this reverse result holds also for other topologies of constant degree and $O(\log N)$ bounded diameter. Further, we have restricted just to one-to-all layouts, so the all-to-all case is another possible extension of our results. Here the unresolved problem is whether the all-to-all case have the same asymptotic solution as the one-to-all case, like in the result due to Stacho and Vrťo. Finally, a reason of difficulty to analyze the reverse problem is probably because of a strong requirement that any virtual channel to be obtained by concatenating (appropriate) virtual paths should be shortest. The question is what is the impact of the requirement "to be shortest" (or stretch factor to be equal 1) on the generalization to other (larger) class of graphs.

The paper has the following structure. In Section 2 we present the graph-theoretic model of ATM networks and define basic efficiency measures. Section 3 includes two basic virtual path layouts for complete binary trees, which are used as building blocks for butterfly layouts. Section 4 contains optimal one-to-all virtual shortest path layouts on butterfly ATM networks.

2 The Model

We exploit the graph-theoretic model of ATM networks as defined in [1]. The communication network is presented by an undirected graph $G = (V, E)$, where the set of vertices V corresponds to ATM switches, and the set of edges E to physical links between them. Moreover, we have a given set ζ of pairs of distinct vertices from V, between which a communication must be set up. We are interested in two special communication patterns:

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

In the following text, the network is denoted by G, and the ζ is either *one-to-all* or *all-to-all* pattern.

Definition 1 *A* virtual path layout *(shortly VPL)* Ψ *on* G *is a collection of simple paths in* G*, called* virtual paths *(shortly VPs).*

From now on, we distinguish between two types of VPLs, depending on their communication pattern ζ, namely *one-to-all* or *all-to-all VPL.*

Definition 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*.*

$\mathcal{L}(e)$ is also referred as *edge congestion* of e.

Definition 3 *The* maximal edge load $\mathcal{L}_{max}(\Psi)$ *of a VPL* Ψ *is* $max_{e \in E}\mathcal{L}(e)$*.*

Definition 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 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 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 a VPL such that hop count and load are minimized simultaneously. The hop–load tradeoff has been studied for various topologies.

The last parameter we are interested in is *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. Layouts with *strech factor* equal to one always use the shortest paths for routing.

3 VPLs for complete binary trees

Now we present some basic layouts for complete binary trees, known from literature. The presented layouts are useful for the design of VPLs on butterfly networks.

Layout 1

Let $T = (V, E)$ be a complete binary tree with a root v_1. Our VPL Ψ will consist of the following simple paths: for each vertex $v_k \in V$, $v_k \neq v_1$, add the shortest path from v_1 to v_k, i.e. $\Psi = \{\psi \mid \psi = (v_k, v_{\lfloor k/2 \rfloor}, \ldots, v_1),\ v_k \in V,\ k \neq 1\}$.

Analysis of VPL. It is easy to see that $\mathcal{H}_{max} = 1$, $\mathcal{L}_{max} = \lfloor N/2 \rfloor$, $\phi = 1$, where N is the number of vertices in T. □

For the next layout we need the following definition:

Definition 7 *Let* $T = (V, E)$ *be a rooted complete binary tree and* $u \in V$ *be a vertex of* T*. Let* T_u *denote a subtree of* T *rooted at* u*. For any* $k \in \mathcal{N}$*,* $k \geq 1$*, we define* $T(u,k) = (V', E')$ *as 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*, and edges* $E' = \{(a,b) | (a,b) \in E \wedge a, b \in V'\}$*.*

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

Layout 2

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

First assume, for simplicity, that $l-1 = k\mathcal{H}$, so $\mathcal{H}$ divides $l-1$ (notice, that there are $l-1$ levels of edges in T). Now 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 denote arbitrary tree among them. There are $\mathcal{H}$ levels of such trees, as shown in Fig. 1.

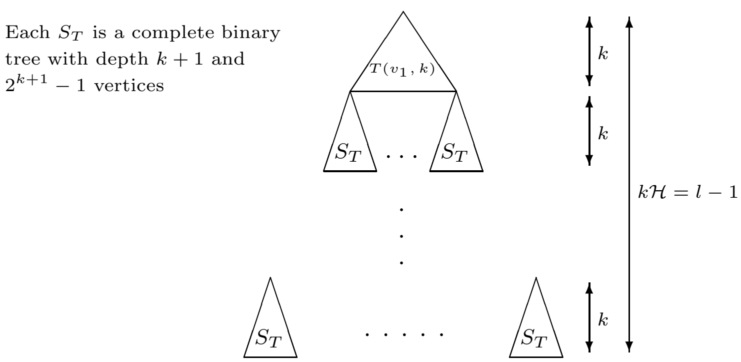


Figure 1: Tree T partitioned into trees S_T

The roots of the trees $T(v_i, k)$ are usually called $p$ivots. Note 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 for each $T(v_i, k)$ construct a one-to-all VPL $\Psi_{T(v_i,k)}$ using *Layout 1*. The VPL Ψ for the tree T is the union of all layouts for trees $T(v_i, k)$.

For general case $l-1 = k\mathcal{H} + p,\ 0 < p < \mathcal{H}$, the design is only slightly modified s.t. subtrees in p levels have the height $k+1$ instead of k. This does not influence the asymptoticity of the result.

Analysis of VPL. $\mathcal{H}_{max} = \mathcal{H}$, $\mathcal{L} = O(N^{\frac{1}{\mathcal{H}}})$, $\phi = 1$. □

4 VPLs for butterfly network

The design of VPLs for butterfly networks has not yet been studied. However, some results can be obtained as a consequence of known general theorems. We will present some new lower and upper bounds together with related VPL designs.

4.1 Butterfly topology

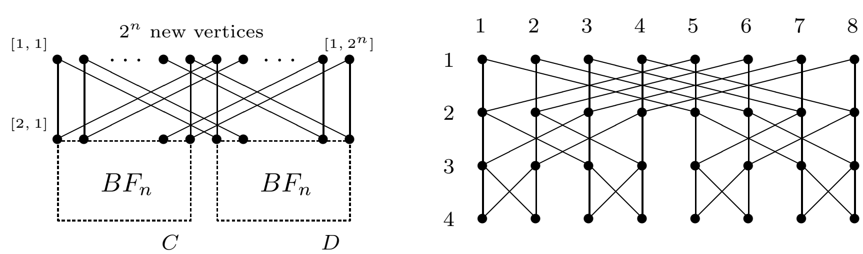
First, we describe the butterfly topology and introduce our labeling conventions to simplify the rest of the section.

The butterfly network BF_n consists of $n2^{n-1}$ vertices, usually represented by n rows and 2^{n-1} columns. Let v be a vertex in 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. Let C, D be two BF_n networks. 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 2^n new vertices $v_{1,1}, ..., v_{1,2^n}$. Finally, add the 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 schematically shown on the *Figure 2*.



Construction of BF_{n+1} *BF_4 network*

Figure 2: Butterfly topology

We will sometimes refer to the rows as levels. The number of rows (levels) is denoted by n, and the total number of vertices as N, so $N = n2^{n-1}$.

Definition 8 *Let BF_n be a butterfly. For any $1 \leq i \leq 2^{n-1}$, define a tree T_i corresponding to the vertex $v_{1,i}$ as follows: Set the vertex $v_{1,i}$ to be the root of the T_i; for every vertex $v_{a,b}$ already in T_i, recursively add its two neighbours in BF_n from the level $b+1$, unless $b = n$, where $v_{a,b}$ is a leaf in T_i.*

Similary, we can define the tree $T_i^{'}$ corresponding to the vertex $v_{n,i}$ oriented bottom-up. All these trees are complete binary trees with depth n and $2^n - 1 = 2\frac{N}{n} - 1$ vertices. These trees make a backbone used for our VPLs.

4.2 Known results

Some known results for common interconnection networks follow from the following theorem (see [5])

Theorem 1 *Let G be a graph of order N with the maximal vertex degree* O(1) *and the diameter* O($logN$)*. Then for not necessarily shortest path gossip on G it holds $\mathcal{H} = \Theta(\frac{logN}{log\mathcal{L}})$ for any given $\mathcal{L}$.*

The lower bound part of the above result holds even for one-to-all layouts, so it is tight also for broadcast virtual path layouts. However, this result can be directly applied to butterfly networks. In this paper, however, we will study the reverse problem for butterfly topology, namely to determine load $\mathcal{L}$ for a given hop count $\mathcal{H}$. Moreover, it will turn out, that this reverse relation for one-to-all layouts (i.e. to compute $\mathcal{L}$ for given $\mathcal{H}$) is more difficult to prove, since it will imply the above result and not vice versa.

4.3 Lower bound

In this subsection we present a lower bound on the load for butterfly topologies.

Theorem 2 *Let BF_n be a butterfly with N vertices. For every one-to-all VPL with given hop count $\mathcal{H}$ it holds $\mathcal{L} = \Omega(N^{\frac{1}{\mathcal{H}}})$.*

Proof. For $\mathcal{H} = 1$ the lower bound $\mathcal{L} = \frac{1}{4}(N-1)$ is trivial. So assume $\mathcal{H} \geq 2$.

Let r be a source in BF_n. Assume Ψ to be an one-to-all layout from r (on BF_n) with load bounded by $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} f(N))$, where $f(N) = o(lg^{\frac{1}{\mathcal{H}}} N)$.

Since BF_n is of constant vertex degree ($\Delta_{max} = 4$), on the first hop from r one can get to at most $O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} f(N))$ vertices. On the second hop, from these vertices one can get to at most $O([\frac{N}{lgN}]^{\frac{2}{\mathcal{H}}} f^2(N))$ vertices and so on. Hence, after $\mathcal{H} - 1$ hops one 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 by 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. 3*.

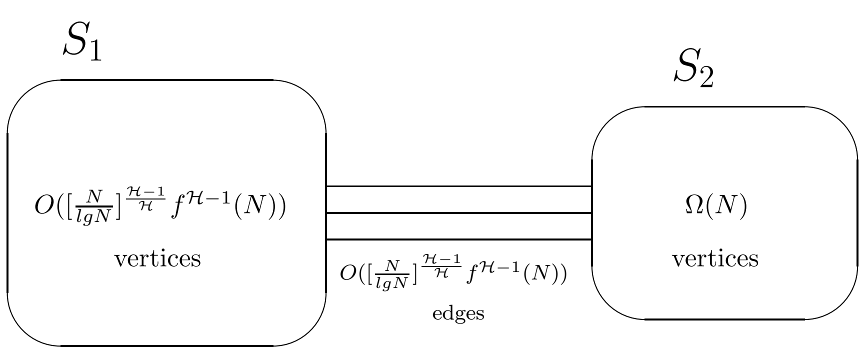


Figure 3: Edge Cut for sets S_1 and S_2

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

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

As $f(N) = o(lg^{\frac{1}{\mathcal{H}}} N)$, 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 the layout Ψ. Thus for arbitrary one-to-all VPL for BF_n with load bounded by $\mathcal{L} = O([\frac{N}{lgN}]^{\frac{1}{\mathcal{H}}} f(N))$ it holds $f(N) = \Omega(lg^{\frac{1}{\mathcal{H}}} N)$, proving the theorem. □

Note that we are not aware of better lower bound for all-to-all case.

4.4 One-to-all VPLs for butterfly networks

In this section we design one-to-all VPLs for butterfly networks. Firstly, we look at one-to-all layouts from the "corner" vertex $v_{1,1}$. Later on, we generalize these results for any source $v_{i,j}$.

We start with an asymptoticaly optimal one-to-all layout from the vertex $v_{1,1}$ on the butterfly.

Layout 3

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

3. For each column c of BF_n, by $v_{i_c,c}$ denote a vertex from T_1 on the column c with the minimum possible row, i.e. $i_c = min\{i | v_{i,c} \in T_1\}$. If there exist connections from $v_{1,1}$ to $v_{i_c,c}$ with at most $\mathcal{H}-1$ hops (in the layout for T_1 from step 2), 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 Ψ (these are one-hop layouts for chains $(v_{i_c,c}, \ldots, v_{1,c})$). This is the case, where $v_{i_c,c}$ is not in the lowest level of trees in Layout 2 ($i_c < n - k$).

4. For the rest of vertices $v_{i_c,c}$ (not included in the previous step 3) find the pivot $v_{i,j}$ of their subtree in the Layout 2. Now, for each path $(v_{i,j}, \ldots, v_{i_c,c})$ already included in Ψ (in fact, there is only one such path), add paths $(v_{i,j}, \ldots, v_{i_c,c}, v_{i_c-1,c}), \ldots, (v_{i,j}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{1,c})$ to Ψ, so there is one-hop connection from the pivot $v_{i,j}$ to any vertex $v_{m,c}$ for $m \leq i_c$.

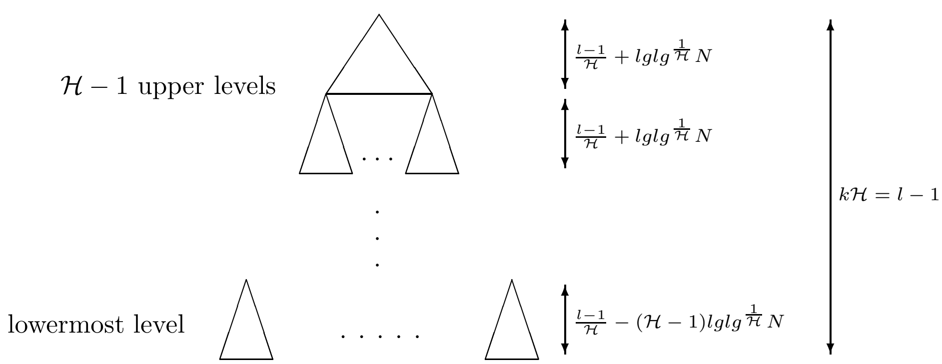


Figure 4: Tree T_1 partitioned into the levels of trees

Analysis of VPL. We need the following lemma:

Lemma 1 *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 defined 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 the 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 $\mathcal{H}$ hops are sufficient to get from $v_{1,1}$ to any vertex $v_{i,c}$. Consider three cases:

- $v_{i,c} \in T_1$: The fact follows directly from the correctness of Layout 2 and from 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 connects $v_{1,1}$ and $v_{i_c,c}$ prolonging it with the path $(v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{i,c})$ from step 3 of construction. Following Lemma 1 it holds $i < i_c$.

- $v_{i,c} \in BF_n \backslash T_1$ 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 connects $v_{1,1}$ 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 2 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 previous path $(v_{m,j}, \ldots, v_{i_c,c})$ in VC between $v_{1,1}$ and $v_{i_c,c}$. The resulting VC connects $v_{1,1}$ and $v_{i,c}$ with at most $\mathcal{H}$ hops.

It is easy to check that all above connections are shortest paths in BF_n. Now we will concentrate on the parameter load.

Paths from step 3 are edge disjoint with any path from other steps. Their worst-case load is $O(lgN)$.

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

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

since the load of each subtree is equal to the 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 subtrees on the lowest level, each path in 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 (the maximum of 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 2. □

Claim 1 *Let BF_n be a butterfly network. Let $\mathcal{H} \geq 2$ be an upper bound on the hop count. Due to the previous layout and the Theorem 2, 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 can be used to broadcast from any vertex of butterfly lying on the first (topmost) level and from any vertex on the last (bottommost) level.

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

Definition 9 *Let BF_n be a butterfly and $v_{r,c}$ be any vertex from BF_n. Define $T_{r,c}$ to be a complete binary tree rooted at $v_{r,c}$ spreading downwards in BF_n (to rows $r+1, r+2, \ldots, n$) as in Definition 8. Similary, define $T'_{r,c}$ to be a complete binary tree rooted at $v_{r,c}$ going upwards.*

Recalling Definition 8 we have $T_{1,c} \equiv T_c$ and $T'_{n,c} \equiv T'_c$.

Definition 10 *Let BF_n be a butterfly and $v_{r,c}$ be any vertex from BF_n. Define $T_{r,c}[p]$ to be a complete binary tree rooted at $v_{r,c}$ spreading down to p levels in BF_n. Similary define $T'_{r,c}[p]$ which spreads upwards.*

The following layout is rather complicated. The reader is suggested to assume that $K < \mathcal{H}$ (in step 2) for the first time and ignore all parts (in the 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 4

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 a one-to-all VPL for BF_n from the vertex $v_{n,1}$ with upper bound for hop count $\mathcal{H}$ using Layout 3. Add all VPs from this VPL to Ψ. This layout divides the butterfly into $\mathcal{H}$ levels (see *Fig. 5*). We define j as $j = max\{v_{j,1}$ *is pivot in Layout* $3 | \ j \leq k\}$. Let K be the number of levels (from Layout 3) below $v_{j,1}$ in BF_n.

   {In our example (*Fig. 5*), $K = 4$, 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 vertices from $\{v_{r,c}|\ k \leq r \leq n\ \wedge\ 1 \leq c \leq 2^{n-k}\}$.

   {These paths are not needed in our construction. However, this step is optional, since leaving these paths in our VPL Ψ will not affect asymptotical optimality.}

4. Construct a VPL for a complete binary tree $T_{k,1}$ (see Definition 9) with hop count K (from step 2) using slightly modified Layout 2. 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 with height $\frac{n-1}{\mathcal{H}} - (\mathcal{H}-1) lglg^{\frac{1}{\mathcal{H}}} N - (k-j)$ if $K = \mathcal{H}$. The k is taken from initial assumptions, j is from step 2. Add these VPs to Ψ.

5. For each column $1 \leq c \leq 2^{n-k}$ of BF_n, let $v_{i_c,c}$ be a vertex from $T_{k,1}$ on column c with the minimum possible row, i.e. $i_c = min\{i | v_{i,c} \in T_{k,1}\}$. Add paths $(v_{i_c,c}, v_{i_c-1,c}), \ldots, (v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{k,c})$ to Ψ (these are one-hop layouts for chains $(v_{i_c,c}, \ldots, v_{k,c})$).

6. For each column $1 \leq c \leq 2^{n-k}$ of BF_n such that there exist connection from $v_{k,1}$ 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 2. Add path $(v_{i,y}, \ldots, v_{i_c,c}, v_{i_c-1,c}, \ldots, v_{k,c})$ to Ψ. Construct one-hop one-to-all VPL for tree $T^{'}_{k,c}[k-j]$ using Layout 1. Add paths from this VPL to 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.e. $i_d = max\{a \mid v_{a,d} \in T^{'}_{k,c}[k-j]\}$ 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 Ψ.

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

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

   to Ψ. Moreover, if $K = \mathcal{H}$,

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

   to Ψ, 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.e.

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

and add paths

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

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

to Ψ.

The layout is schematicaly shown in Fig. 5.

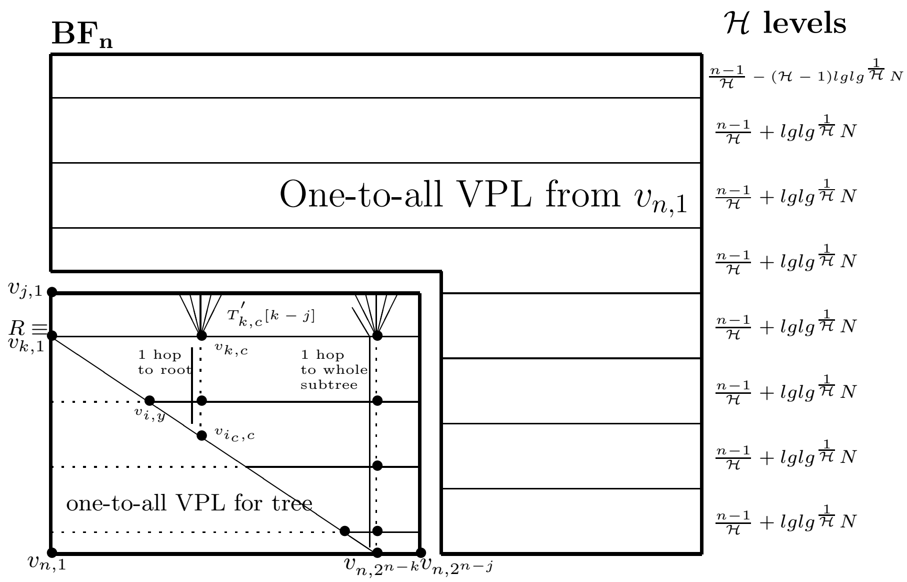


Figure 5: One-to-all VPL for BF_n from any vertex ($\mathcal{H} = 8$)

Analysis of VPL.
PART A - correctness 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. The number of hops used is at most $K+1\le\mathcal{H}$. 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"
  $$\{v_{a,b}|\ k\le a\le n\ \wedge\ 1\le b\le 2^{n-k}\}$$
  and it is easy to verify, 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 $v_{j,c}$, $c\le 2^{n-j}$ in at most K hops. It is still easy to see, that we use shortest paths (for complete characterization of the shortest paths in butterfly networks see e.g. [8]).

- *Vertices in* $\{v_{a,b}|\ j\le a\le n\ \wedge\ 2^{n-k}<b\le 2^{n-j}\}$, *not included in the 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 $K+1\le\mathcal{H}$ 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 of 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. 2*. 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 $x=min\{a\mid 1\le a\le 2^{n-j}\ \wedge\ v_{j,a}\in q\}$. 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 $\{v_{j,a}\mid 1\le a\le 2^{n-j}\}$ is a 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 3rd 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 2 *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 [8]. Let p be the 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 the 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 p starts going up (decrease row). It can be transformed into the 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 the whole procedure ($v_{k,1}$ to $v_{1,c}$ to $v_{r,c}$) can be done in a single hop.
  - The path starts downward (increasing row). This is identical with downward path in the following case.
- The path p changes (top-down) direction twice.
  - The path p starts going up (decrease row). From the properties of shortest paths, the path must finish at row k (or below, when $r > k$, but this is not the case of Lemma). Such path can be transformed into path p_2, which starts going downwards (the necessary column changes on rows $\geq k$ are taken first). Use p_2 in the following case.

– The path p starts 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).

$\square$

This property is exploited in Layout 5 to get a 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. $\mathcal{L} = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$ since it is the load of Layout 3.
- Step 3. $\mathcal{L} = 0$. We only remove paths.
- Step 4. $\mathcal{L} = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$ 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 3).
- Step 5. $\mathcal{L} = O(lgN)$, it is the length of chains.
- Step 6. $\mathcal{L} = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$. 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 3).
- Step 7. $\mathcal{L} = O(\left[\frac{N}{lgN}\right]^{\frac{1}{\mathcal{H}}} lg^{\frac{1}{\mathcal{H}}} N)$. Let $a = k - j$ and let b be the number of rows on the lowest level in tree $T_{k,1}$ from the Layout 2 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 3). 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 3).

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 2 *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, $1 \leq k \leq n-1$. Due to the previous layout and Theorem 2, 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 can be used for any vertex $v_{r,c}$ from BF_n using automorphism which maps $v_{r,c}$ into the vertex $v_{r,1}$.

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

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 4. Add all VPs from it to 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 4. 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. Now $v_{n-k,1}$ match the vertex $v_{k,1}$ from the previous step. Add all VPs (after change of orientation) to Ψ.

Analysis of VPL. The load $\mathcal{L}$ is at most twice the load from Layout 4, so it still holds

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

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

Theorem 3 *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 $1 \leq k \leq n-1$. Due to the previous layout and Theorem 2, 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 can be used for any vertex $v_{r,c}$ from BF_n by mapping it firstly into the vertex $v_{r,1}$.

5 Conclusions

We have presented an optimal shortest path broadcast layout on butterfly ATM networks of size N with load $\mathcal{L} = \Theta(N^{1/\mathcal{H}})$ for any hop count $\mathcal{H}$. The main question is whether such a result holds also for other $O(1)$ bounded degree and $O(\log N)$ bounded diameter topologies and also for all-to-all case.

References

[1] Shmuel Zaks: *Path Layout in ATM Networks – A Survey.* In the 24th Annual Conference on Current Trends in Theory and Practice of Informatics (SOFSEM), Lecture Notes in Computer Science 1338, Springer-Verlag, Berlin, 1997, pp. 144–160

[2] Shmuel Zaks: *Design Issues in ATM and Optical Networks.* Computing and Informatics, Vol 20, No 2, 2001

[3] Ornan Gerstel and Shmuel Zaks. *Path Layout in ATM Networks.* In the 2nd International Colloquium on Structural Information and Communication Complexity (SIROCCO), Carleton University Press, Ottawa, 1995

[4] Israel Cidon, Ornan Gerstel and Shmuel Zaks. *A scalable approach to routing in ATM networks.* In the 8th International Workshop on Distributed Algorithms (WDAG), Lecture Notes in Computer Science 857, Springer-Verlag, Berlin, 1994, pp. 209–222

[5] Ladislav Stacho and Imrich Vrťo. *Virtual Path Layouts for some Bounded Degree Networks.* In the 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO), Carleton Scientific, Ottawa, 1996, pp. 269–278

[6] Evangelos Kranakis, Danny Krizanc and Andrzej Pelc. *Hop-congestion tradeoffs for ATM networks.* In the 7th IEEE Symposium on Parallel and Distributed Processing, 1995, pp. 662–668

[7] Israel Cidon, Ornan Gerstel and Shmuel Zaks. *The layout of virtual paths in ATM networks.* ACM/IEEE Transactions on Networking, Vol 6, No 4, 1996, pp. 873–884

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

[9] Jean-Claude Bermond, Nausica Marlin, David Peleg and Stephane Pérennes. *Directed Virtual Path Layouts in ATM Networks.* In the 12th International Symposium on Distributed Computing (DISC), Lecture Notes in Computer Science 1499, Springer-Verlag, Berlin, 1998, pp. 75–88

[10] 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.* In the 24th International Colloquium on Automata, Languages, and Programming (ICALP), Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1997

[11] Ornan Gerstel, Avishai Wool and Shmuel Zaks. *Optimal Layouts on a Chain ATM Network.* In the 3rd Annual European Symposium on Algorithms (ESA), Lecture Notes in Computer Science 979, Springer-Verlag, Berlin, 1995, pp. 508–522

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