A 1.5-Approximation Algorithm for Broadcast Time in k-Path Graphs

Aref Afzali[1], Hovhannes A. Harutyunyan[1]

[1]Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada

Abstract

Broadcasting is a classical information-dissemination problem in which an originator vertex must transmit a message to all vertices of a network in the minimum number of synchronous time units. We study broadcasting in k-path graphs, also known as melon graphs, which consist of k internally vertex-disjoint paths between two junction vertices. Although this graph family has a restricted structure, computing an optimal broadcast scheme is NP-complete, motivating the study of simple approximation algorithms. We introduce and analyze deterministic odd–even schedules for both junction-originator and internal-originator placements. These schedules coordinate the two junction vertices through complementary parity orders on the remaining paths: one junction processes odd-indexed paths first and then even-indexed paths, while the other processes even-indexed paths first and then odd-indexed paths. We prove that the odd–even schedules achieve an unconditional 1.5-approximation for all k-path graphs and all originator placements. Once the paths are indexed by non-increasing length, the schedules are constructed in $O(k)$ time. We also show that the 1.5 analysis is asymptotically tight by giving infinite families of instances whose achieved ratios approach $3/2$.

Keywords

Interconnection networks, information dissemination, broadcasting, approximation algorithms, k-path graph, melon graph

1. Introduction

Broadcasting is a fundamental information-dissemination process in interconnected networks, where a message originating at one node, called the *originator*, must be transmitted to all other nodes as quickly as possible. This primitive appears throughout parallel and distributed computing, and many communication models have been studied [1, 2, 3]. We focus on the *classical*, or telephone, model of broadcasting.

The network is modeled as a connected undirected graph $G = (V, E)$, whose vertices represent nodes and whose edges represent communication links. The process is divided into synchronous time units; initially only the originator is informed; in each time unit, every informed vertex may call at most one uninformed neighbor; all calls occur in parallel; and the process stops when all vertices are informed. A *broadcast scheme* is an ordered list of the calls made in each time unit.

Definition 1 (Broadcast time)**.** The *broadcast time* of a vertex v in a graph G is the minimum number of time units required to broadcast in G when v is the originator, denoted $b(v, G)$. The broadcast time of G is $b(G) = \max_{v \in V} b(v, G)$.

Two standard lower bounds are used throughout the broadcasting literature. Since the number of informed vertices can at most double per round, $b(v, G) \geq \lceil \log_2 n \rceil$ for $n = |V|$; and since information advances at most one edge per round along any root-to-leaf chain, $b(v, G) \geq \mathrm{ecc}(v)$, the eccentricity of v.

Broadcasting has been studied from both network-design and scheduling perspectives. In network design, the goal is to construct sparse *minimum broadcast graphs* that still allow broadcasting in the optimal $\lceil \log_2 n \rceil$ rounds from every originator [4, 5]. In scheduling, the graph is fixed and the goal is to compute an optimal broadcast scheme; closed-form broadcast times are known for several

ICTCS 2026: Italian Conference on Theoretical Computer Science, September 7–9, 2026, Udine, Italy
aref.afzali@mail.concordia.ca (A. Afzali); haruty@cs.concordia.ca (H. A. Harutyunyan)
0009-0005-1511-4427 (A. Afzali); 0000-0001-7260-4186 (H. A. Harutyunyan)

standard topologies, including grids and tori, hypercubes and recursive circulant graphs, and Knödel graphs [6, 7, 8]. The broadcast-time decision problem is NP-complete for arbitrary graphs [9, 10], and remains NP-complete for restricted classes such as 3-regular planar graphs and bounded-degree graphs [11, 12]. This has motivated exact algorithms for structured families [13, 14, 15, 16], heuristics [17], and approximation algorithms for the general problem [18, 19, 20, 21]. Other models, including k-broadcasting and weighted-vertex variants, have also been studied [22, 23]; see [1, 2, 24, 3] for surveys.

In this paper we study broadcasting in *k-path graphs*, also known as *melon graphs*, which consist of k internally vertex-disjoint paths joining two junction vertices. Although simple to describe, this family creates nontrivial scheduling choices because both junctions may contribute to informing the remaining paths. Bhabak and Harutyunyan [25] gave specialized algorithms for this family, achieving a $(4-\varepsilon)$-approximation in $O(|V|+k\log k)$ time. Harutyunyan and Hovhannisyan [26] later gave a simpler 2-approximation based on a predetermined junction ordering, running in $O(k)$ time and achieving a $(2-\varepsilon)$-approximation when the shortest path has length at least one. More recently, the broadcast-time problem was shown to be NP-complete for k-path graphs, and polynomial-time approximation schemes were obtained for melon and cycle-star graphs [27]. A simple linear-time fixed-ratio 1.5-approximation has also been developed for the closely related k-cycle graphs [28].

The existence of an EPTAS for melon graphs shows that approximation ratios arbitrarily close to 1 are achievable in polynomial time [27]. However, the stated running-time upper bound for using that scheme to obtain a fixed 1.5-approximation has a substantially larger dependence on the input parameters than the construction bound for the schedules considered here. Setting the approximation parameter to $\epsilon = 1/2$ gives $O\big(2^{64}k(k+n)\log n\big)$, where $n=|V|$. In the worst case, when $k=\Theta(n)$, this bound becomes $O\big(2^{64}n^2\log n\big)$. In contrast, the schedules analyzed in this paper achieve the same fixed approximation factor through simple deterministic rules and are constructed in $O(k)$ time once the paths are indexed by non-increasing length.

Our contribution We study a deterministic odd–even schedule for k-path graphs, in which the two junctions follow complementary parity orders: one processes odd positions first and then even positions, while the other processes even positions first and then odd positions. We prove that this schedule gives an unconditional 1.5-approximation for both junction and internal originators. We also show that the analysis is asymptotically tight by giving infinite families of instances whose achieved ratios approach $3/2$.

2. k-Path Graphs

A k-path graph $G=(P_1,P_2,\ldots,P_k)$ is obtained from two vertices u and v by adding $k\geq 2$ internally vertex-disjoint paths $P_1,P_2,\ldots,P_k$ between them. The vertices u and v are the *junctions* of G (Fig. 1). We assume the paths are indexed in non-increasing order of their lengths; formally $l_1\geq l_2\geq\cdots\geq l_k\geq 0$, where l_i is the number of internal vertices on P_i (excluding u and v). Since the graph is simple, at most one path can have $l_i=0$, namely P_k. As k-path graphs are among the simplest families containing intersecting cycles, they form a natural setting for studying the broadcast-time problem in structured graphs.

The broadcasting process begins at an originator $s\in V$. We distinguish two cases: s is a junction vertex (u or v), or s is an internal vertex on one of the paths. For an algorithm $\mathcal{A}$, we write $b_{\mathcal{A}}(s,G)$ for the broadcast time achieved by its scheme, and compare it to the optimal broadcast time $b(s,G)$. For each path P_i, let t_i denote the first time unit by which all vertices of P_i are informed under the schedule being analyzed.

Definition 2 (Approximation ratio)**.** An algorithm $\mathcal{A}$ is an α-approximation, where $\alpha\geq 1$, if $b_{\mathcal{A}}(s,G)\leq \alpha\cdot b(s,G)$ for all k-path graphs G and all originator vertices s.

The following two structural lemmas guide the algorithm design.

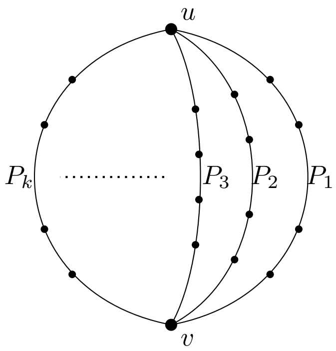

Figure 1: An example of a k-path graph with junction vertices u and v.

Lemma 1 ([25]). *There exists a minimum broadcast scheme from a junction originator u in which u starts propagation on the shortest path P_k in the first time unit.*

Lemma 2 ([25]). *There exists a minimum broadcast scheme from an internal originator $w \in V(P_m)$ in which w first sends the message along a shortest subpath of P_m toward a junction vertex.*

In any broadcast scheme where a junction vertex is the originator, an internal (non-junction) vertex has at most one uninformed neighbor once it becomes informed. Hence, to describe a scheme it suffices to specify the order of calls placed by the junction vertices.

Throughout the fixed-order schedules, each path retains its prescribed position in the corresponding order. If a scheduled call is unnecessary because the path is already fully informed, that call is omitted, but the prescribed times of paths occurring later in the order remain unchanged.

3. Fixed-Order k-Path Schedules

We study a deterministic odd–even scheduling rule. The junction that is informed first considers the odd positions in the relevant path order first and then the even positions. The other junction, once informed, follows the complementary order: it considers the even positions first and then the odd positions. We denote the resulting schedules by C_J and C_I for junction and internal originators, respectively.

The parity refers to positions in the non-increasing path-length order, not to the parity of the path lengths. The purpose of the complementary orders is to balance the prescribed starting times of each path: a path considered early by one junction is considered later by the other junction, and conversely. For a path that is started from both junctions, its informing time is controlled by its length and by the sum of these two starting times. The odd–even order keeps this sum sufficiently small to obtain the 1.5 approximation bound. For an internal originator, the same rule is applied to the corresponding active list of paths.

3.1. Junction-originator schedules

Assume w.l.o.g. that the junction vertex u is the originator. In the first time unit, the schedule starts propagation on the shortest path P_k, so the opposite junction v is informed in time unit l_k+1. Meanwhile, u considers the odd-indexed paths first and then the even-indexed paths. Once v is informed, it follows the complementary order, considering the even-indexed paths first and then the odd-indexed paths. The full procedure is given in Algorithm 1.

Let $a = \lceil (k-1)/2 \rceil$ and $c = \lfloor (k-1)/2 \rfloor$. Thus a is the number of odd indices and c is the number of even indices in $\{1, \ldots, k-1\}$.

The path P_k is excluded from the two parity orders above, since it is used in the first time unit to inform v. If a scheduled call is no longer needed because the path has already been fully informed, the call is skipped. If both junctions would inform the same remaining vertex in the same time unit, only

Algorithm 1 Broadcasting from a junction vertex, odd–even order

Input: a k-path graph $G = (P_1, \ldots, P_k)$, junction vertices u, v, originator u.
Output: a broadcast scheme with broadcast time $b_{C_J}(u, G)$.

procedure OddEvenJunction
In the first time unit, u passes the message along P_k.
for $1 \le r \le a$ **do**
  if P_{2r-1} contains an uninformed vertex **then**
    u passes the message along P_{2r-1} in time unit $r + 1$.
  end if
end for
for $1 \le r \le c$ **do**
  if P_{2r} contains an uninformed vertex **then**
    u passes the message along P_{2r} in time unit $a + r + 1$.
  end if
end for
v receives the message in time unit $l_k + 1$.
for $1 \le r \le c$ **do**
  if P_{2r} contains an uninformed vertex **then**
    v passes the message along P_{2r} in time unit $l_k + r + 1$.
  end if
end for
for $1 \le r \le a$ **do**
  if P_{2r-1} contains an uninformed vertex **then**
    v passes the message along P_{2r-1} in time unit $l_k + c + r + 1$.
  end if
end for
end procedure

one of the two calls is made. This only removes a redundant call and does not increase the broadcast time.

Example. Consider a 5-path graph with

$$(l_1, l_2, l_3, l_4, l_5) = (6, 5, 3, 2, 0),$$

and let u be the originator. Since P_5 is the edge uv, the junction v is informed in time unit 1. Here $a = c = 2$. The order followed by u on the remaining paths is P_1, P_3, P_2, P_4, while the order followed by v is P_2, P_4, P_1, P_3. Thus the prescribed starting times are

	P_1	P_2	P_3	P_4
u	2	4	3	5
v	4	2	5	3

and the corresponding times by which the paths are fully informed are $t_1 = 5, t_2 = 5, t_3 = 5, t_4 = 4$. In particular, P_4 is already fully informed when u reaches its prescribed call in time unit 5, so that call is omitted. Since P_5 is fully informed in time unit 1, the broadcast time produced by the schedule is $b_{C_J}(u, G) = 5$. Note that the optimal broadcast time is $b(u, G) \ge 4$, because there is a cycle with length 8 formed by the paths P_1 and P_4, so the approximation ratio is $b_{C_J}(u, G)/b(u, G) \le 1.5$.

Lemma 3. *For the odd–even junction schedule, all vertices of P_k are informed in time unit $l_k + 1$, and for every $1 \le i \le k - 1$,*

$$t_i \le \left\lceil \frac{l_i + l_k + i + \lceil k/2 \rceil}{2} \right\rceil . \tag{1}$$

Consequently,

$$b_{C_J}(u,G) \le \max\left\{ l_k+1, \max_{1\le i\le k-1} \left\lceil \frac{l_i+l_k+i+\lceil k/2\rceil}{2} \right\rceil \right\}. \tag{2}$$

Proof. All vertices of P_k are informed in time unit l_k+1. Now fix $1 \le i \le k-1$.

First suppose that i is odd, say $i = 2r-1$. Then P_i is the r-th odd-indexed path. The vertex u starts propagation on P_i in time unit $r+1$, while v considers P_i in time unit $l_k+c+r+1$. Before v reaches P_i, the message has already moved along P_i from the u-side for

$$(l_k+c+r+1)-(r+1) = l_k+c$$

time units.

If $l_i \le l_k+c$, then all vertices of P_i are informed before v needs to start it. In this case, $t_i \le r+l_i$. Since $l_i \le l_k+c$, we have $2(r+l_i) \le l_i+l_k+c+2r$, and therefore

$$t_i \le \left\lceil \frac{l_i+l_k+c+2r}{2} \right\rceil.$$

Now suppose that $l_i > l_k+c$. When v reaches P_i, the number of uninformed internal vertices remaining on P_i is $l_i-(l_k+c)$. From that time on, the remaining part of P_i is informed from both sides. Hence

$$\begin{aligned} t_i &\le l_k+c+r+\left\lceil \frac{l_i-(l_k+c)}{2} \right\rceil \\ &= \left\lceil \frac{l_i+l_k+c+2r}{2} \right\rceil. \end{aligned}$$

Thus, for odd $i = 2r-1$,

$$t_i \le \left\lceil \frac{l_i+l_k+c+2r}{2} \right\rceil.$$

Since $2r = i+1$ and

$$c+1 = 1+\left\lfloor \frac{k-1}{2} \right\rfloor = \left\lceil \frac{k}{2} \right\rceil,$$

we obtain

$$t_i \le \left\lceil \frac{l_i+l_k+i+\lceil k/2\rceil}{2} \right\rceil.$$

Now suppose that i is even, say $i = 2r$. Then P_i is the r-th even-indexed path. The vertex v considers P_i in time unit l_k+r+1, while u considers P_i in time unit $a+r+1$.

First assume that $l_k \ge a$. Then u reaches P_i no later than v. Before v reaches P_i, the message has already moved along P_i from the u-side for $(l_k+r+1)-(a+r+1) = l_k-a$ time units.

If $l_i \le l_k-a$, then all vertices of P_i are informed before v needs to start it, and $t_i \le a+r+l_i$. Since $l_i \le l_k-a$, we have $2(a+r+l_i) \le l_i+l_k+a+2r$, and therefore

$$t_i \le \left\lceil \frac{l_i+l_k+a+2r}{2} \right\rceil.$$

If $l_i > l_k-a$, then when v reaches P_i, the number of uninformed internal vertices remaining is $l_i-(l_k-a)$. Hence

$$\begin{aligned} t_i &\le l_k+r+\left\lceil \frac{l_i-(l_k-a)}{2} \right\rceil \\ &= \left\lceil \frac{l_i+l_k+a+2r}{2} \right\rceil. \end{aligned}$$

It remains to consider the case $l_k < a$. Then v reaches P_i before u. Before u reaches P_i, the message has already moved along P_i from the v-side for $(a+r+1)-(l_k+r+1) = a-l_k$ time units.

If $l_i \le a - l_k$, then all vertices of P_i are informed before u needs to start it, and $t_i \le l_k + r + l_i$. Since $l_i \le a - l_k$, we have $2(l_k + r + l_i) \le l_i + l_k + a + 2r$, and therefore

$$t_i \le \left\lceil \frac{l_i + l_k + a + 2r}{2} \right\rceil .$$

If $l_i > a - l_k$, then when u reaches P_i, the number of uninformed internal vertices remaining is $l_i - (a - l_k)$. Hence

$$\begin{aligned} t_i &\le a + r + \left\lceil \frac{l_i - (a - l_k)}{2} \right\rceil \\ &= \left\lceil \frac{l_i + l_k + a + 2r}{2} \right\rceil . \end{aligned}$$

Therefore, for even $i = 2r$,

$$t_i \le \left\lceil \frac{l_i + l_k + a + 2r}{2} \right\rceil .$$

Since $i = 2r$ and

$$a = \left\lceil \frac{k-1}{2} \right\rceil \le \left\lceil \frac{k}{2} \right\rceil ,$$

we obtain

$$t_i \le \left\lceil \frac{l_i + l_k + i + \lceil k/2 \rceil}{2} \right\rceil .$$

Combining the odd and even cases proves the lemma. □

3.2. Internal-originator schedules

Now let the originator w be an internal vertex on P_m. Let d be the length of the subpath from w to the nearer junction u, and let $\tau(m) = l_m + 2 - 2d$. Since w starts propagation toward v in time unit 2, the junction v is informed through the originator path P_m in time unit $d + \tau(m) = l_m + 2 - d$.

Alternatively, after u is informed in time unit d, it can inform v through the shortest path P_k in time unit $d + l_k + 1$. Thus the comparison between $\tau(m)$ and $l_k + 1$ determines which of these two routes informs v earlier. The schedule C_I first sends the message from w toward u and then toward v along P_m. If $\tau(m) \le l_k + 1$, then v is informed through P_m, and the odd–even rule is applied to the list obtained from $P_1, \dots, P_k$ after deleting P_m and also deleting P_k if $l_k = 0$. If $\tau(m) > l_k + 1$, then u informs v through P_k, and the odd–even rule is applied to $(P_1, \dots, P_{k-1})$, with P_m skipped by u but kept in the order followed by v. The full procedure is given in Algorithm 2.

If a scheduled call is no longer needed because the corresponding path has already been fully informed, the call is skipped. If both junctions would inform the same remaining vertex in the same time unit, only one of the two calls is made. This only removes a redundant call and does not increase the broadcast time.

Lemma 4. *For the odd–even internal schedule, the following bounds hold.*

If $\tau(m) \le l_k + 1$ and $\mathcal{L}_1 = (Q_1, \dots, Q_N)$ is the active list used by the algorithm, then for every $P_i = Q_r \in \mathcal{L}_1$,

$$t_i \le \left\lceil \frac{l_i + 2d + \tau(m) + r + \lceil N/2 \rceil - 1}{2} \right\rceil . \tag{3}$$

If $\tau(m) > l_k + 1$, then for every $1 \le i \le k-1$ with $i \ne m$,

$$t_i \le \left\lceil \frac{l_i + 2d + l_k + i + \lceil k/2 \rceil}{2} \right\rceil . \tag{4}$$

Algorithm 2 Broadcasting from an internal vertex, odd–even order

Input: a k-path graph $G = (P_1, \ldots, P_k)$, junction vertices u, v, originator $w \in V(P_m) \setminus \{u, v\}$.
Output: a broadcast scheme with broadcast time $b_{C_I}(w, G)$.

procedure OddEvenInternal
Let d be the length of the subpath wu on P_m and let $\tau(m) = l_m + 2 - 2d$.
In the first time unit, w passes the message along P_m toward u.
In the second time unit, w passes the message along P_m toward v.
u receives the message in time unit d.
if $\tau(m) \leq l_k + 1$ **then**
    v receives the message in time unit $t_v = d + \tau(m)$.
    Let $\mathcal{L}_1$ be the ordered list obtained from $P_1, \ldots, P_k$ after deleting P_m and also deleting P_k if $l_k = 0$.
    u follows the order of the paths in odd positions of $\mathcal{L}_1$ first and then the paths in even positions of $\mathcal{L}_1$.
    v follows the order of the paths in even positions of $\mathcal{L}_1$ first and then the paths in odd positions of $\mathcal{L}_1$.
else
    u passes the message along P_k in time unit $d + 1$.
    v receives the message in time unit $t_v = d + l_k + 1$.
    Let $\mathcal{L}_2 = (P_1, P_2, \ldots, P_{k-1})$.
    u follows the order of the paths in odd positions of $\mathcal{L}_2$ first and then the paths in even positions of $\mathcal{L}_2$, skipping P_m whenever it is reached.
    v follows the order of the paths in even positions of $\mathcal{L}_2$ first and then the paths in odd positions of $\mathcal{L}_2$.
end if
end procedure

Moreover, in this case let r_m be the position of P_m in the odd–even order followed by v on $\{P_1, \ldots, P_{k-1}\}$. Then

$$t_m \leq \left\lceil \frac{l_m + l_k + r_m + 1}{2} \right\rceil . \tag{5}$$

In particular, since $r_m + 1 \leq m + \left\lceil \frac{k}{2} \right\rceil$, we have

$$t_m \leq \left\lceil \frac{l_m + l_k + m + \lceil k/2 \rceil}{2} \right\rceil . \tag{6}$$

Proof. The proof was omitted because of the space constraints. □

Complexity analysis Once the paths are indexed by non-increasing length, the odd–even orders are determined directly from the path positions. No adaptive path selection or dynamic scheduling is required. For each path, the relevant start times can be computed from its index, or from its position in the active list in the internal-originator case. Therefore a single scan over the k paths is enough to compute the broadcast time, and the compact description of the schedule is constructed in $O(k)$ time.

If the full broadcast scheme is written explicitly as a list of calls in every time unit, then the output size may be $\Theta(|V|)$, because internal vertices forward the message along the paths. Thus the expanded scheme can be generated in $O(|V|)$ time from the compact representation. If the paths are not already indexed by non-increasing length, an additional $O(k \log k)$ preprocessing step is needed for sorting.

4. Broadcasting from a Junction Vertex

Assume w.l.o.g. that u is the originator.

Lower bound. The following lower bound appears in prior work and will be used in the approximation analysis.

Lemma 5 ([25]). *Let $G = (P_1, \ldots, P_k)$ be a k-path graph with $l_1 \geq \cdots \geq l_k \geq 0$, and let the originator be a junction vertex u. Then, for every $1 \leq i \leq k-1$,*

$$b(u, G) \geq \left\lceil \frac{l_i + l_k + i + 1}{2} \right\rceil \geq \frac{l_i + l_k + i + 1}{2}.$$

Theorem 1. *The schedule C_J is a 1.5-approximation for k-path graphs when the originator is a junction vertex.*

Proof. Let $b = b(u, G)$ be the optimal broadcast time. First, every broadcast scheme from u must inform the opposite junction v. Since P_k is a shortest path from u to v, we have $b \geq l_k + 1$. Thus all vertices of P_k are informed by Algorithm 1 within b time units, and therefore P_k does not violate the desired approximation bound.

Now fix $1 \leq i \leq k-1$. By Lemma 5,

$$b \geq \left\lceil \frac{l_i + l_k + i + 1}{2} \right\rceil.$$

Therefore $l_i + l_k + i + 1 \leq 2b$, and hence

$$l_i + l_k + i \leq 2b - 1. \tag{7}$$

We also need a lower bound on b in terms of k. By Lemma 1, there exists a minimum broadcast scheme in which u starts propagation on P_k in the first time unit. After this first time unit, u has at most $b-1$ remaining opportunities to start propagation on the other paths by time b. Moreover, v cannot start propagation on any other path before it becomes informed. Since v is informed no earlier than time $l_k + 1$, it can start propagation on the remaining paths only in time units $l_k+2, l_k+3, \ldots, b$. Hence v has at most $b - l_k - 1$ such opportunities.

Because G is simple, each of the paths $P_1, \ldots, P_{k-1}$ contains at least one internal vertex. In any broadcast scheme completing by time b, each of these $k-1$ paths must therefore be started from at least one of the two junctions. Thus $k - 1 \leq (b-1) + (b - l_k - 1)$. Hence $k - 1 \leq 2b - l_k - 2 \leq 2b - 2$, and so $k \leq 2b - 1$. It follows that

$$\left\lceil \frac{k}{2} \right\rceil \leq b. \tag{8}$$

Combining (7) and (8), we get $l_i + l_k + i + \lceil \frac{k}{2} \rceil \leq (2b-1) + b = 3b - 1$. Using (1), we obtain

$$t_i \leq \left\lceil \frac{l_i + l_k + i + \lceil k/2 \rceil}{2} \right\rceil \leq \left\lceil \frac{3b-1}{2} \right\rceil.$$

Since b is an integer,

$$\left\lceil \frac{3b-1}{2} \right\rceil \leq \frac{3b}{2}.$$

Therefore all vertices of every path P_i, $1 \leq i \leq k-1$, are informed by time $3b/2$. Together with the bound for P_k, this gives $b_{C_J}(u, G) \leq \frac{3}{2} b(u, G)$. Hence Algorithm 1 is a 1.5-approximation. □

Theorem 2. *The 1.5 approximation guarantee of the odd–even junction schedule C_J is asymptotically tight.*

Proof. Let $q \geq 2$, let $k = 2q$, and let $G_q = (P_1, \ldots, P_k)$ have path lengths $l_1 = 2q, l_2 = l_3 = \cdots = l_{k-1} = 1, l_k = 0$. Thus P_k is the edge uv, and the originator is the junction u.

We first show that $b(u, G_q) = q + 1$. The path P_1 has $2q$ internal vertices. In any broadcast scheme, let x and y be the first time units in which propagation starts on P_1 from the u-side and from the v-side, respectively. If u informs v through P_k in time unit 1, then neither junction can start propagation on P_1 before time unit 2, so $x, y \geq 2$. Otherwise, v cannot be informed before time unit 2, and hence $y \geq 3$ while $x \geq 1$. Thus, in all cases, $x + y \geq 4$, and so

$$b(u, G_q) \geq \left\lceil \frac{2q + 4 - 2}{2} \right\rceil = q + 1.$$

This lower bound is attainable: in time unit 1, u informs v through P_k; in time unit 2, both junctions start propagation on P_1; and in time units $3, \dots, q + 1$, the two junctions start all remaining length-one paths. Hence $b(u, G_q) = q + 1$.

Under the schedule C_J, the junction u starts P_1 in time unit 2. The junction v first considers the even-indexed paths $P_2, P_4, \dots, P_{2q-2}$ and then considers P_1 in time unit $q + 1$. None of these even-indexed paths is skipped before v reaches P_1, since each has one internal vertex and has not yet been fully informed when v considers it. Therefore all vertices of P_1 are informed by C_J in time unit

$$\left\lceil \frac{2q + 2 + (q + 1) - 2}{2} \right\rceil = \left\lceil \frac{3q + 1}{2} \right\rceil.$$

All other paths have length at most one and are fully informed no later than P_1. Thus

$$b_{C_J}(u, G_q) = \left\lceil \frac{3q + 1}{2} \right\rceil,$$

and hence

$$\frac{b_{C_J}(u, G_q)}{b(u, G_q)} = \frac{\lceil (3q + 1)/2 \rceil}{q + 1}$$

approaches $3/2$ as q increases. □

5. Broadcasting from an Internal Vertex

Let the originator w be an internal vertex on P_m, with d and $\tau(m)$ as defined in Section 3.2.

Lower bounds. We first state the lower bounds used in the internal-originator analysis.

Lemma 6 (Active-list lower bound)**.** *Let w be an internal originator on P_m, let d be the distance from w to the nearer junction u, and let v be the other junction. Suppose that, in a minimum broadcast scheme from w satisfying Lemma 2, the junction v cannot be informed before time unit $d + \sigma$, where $\sigma \geq 1$.*

Let $\mathcal{L} = (Q_1, Q_2, \dots, Q_N)$ be an ordered list of non-originator paths, each having at least one internal vertex, listed in non-increasing order of length. If $Q_r = P_i$, then

$$b(w, G) \geq d + \left\lceil \frac{l_i + \sigma + r - 1}{2} \right\rceil.$$

Proof. The proof was omitted because of the space constraints. □

Lemma 7 (Delayed active-list lower bound)**.** *Under the assumptions of Lemma 6, suppose additionally that the junction u cannot start propagation on any path of $\mathcal{L}$ before time unit $d + 2$. If $Q_r = P_i$, then*

$$b(w, G) \geq d + \left\lceil \frac{l_i + \sigma + r}{2} \right\rceil.$$

Proof. The proof was omitted because of the space constraints. □

For every $j \neq m$, define

$$\rho_m(j) = \begin{cases} j, & j < m, \\ j-1, & j > m. \end{cases}$$

Thus $\rho_m(j)$ is the position of P_j in the sequence obtained from $P_1, \ldots, P_k$ by deleting the originator path P_m.

Lemma 8. *Let $G = (P_1, \ldots, P_k)$ be a k-path graph with $l_1 \geq \cdots \geq l_k \geq 1$, and let the originator w be an internal vertex on P_m. Let d be the length of the subpath from w to the nearer junction u along P_m, and let $\tau(m) = l_m + 2 - 2d$. Then:*

1. *if $\tau(m) < l_k + 1$, then for every $1 \leq j \leq k$ with $j \neq m$,*

$$b(w, G) \geq d + \left\lceil \frac{l_j + \tau(m) + \rho_m(j) - 1}{2} \right\rceil;$$

2. *if $\tau(m) \geq l_k + 1$, then for every $1 \leq j \leq k-1$ with $j \neq m$,*

$$b(w, G) \geq d + \left\lceil \frac{l_j + l_k + \rho_m(j)}{2} \right\rceil;$$

3. *if $\tau(m) > l_k + 1$, then the stronger bound*

$$b(w, G) \geq d + \left\lceil \frac{l_j + l_k + j}{2} \right\rceil$$

holds for every $1 \leq j \leq k-1$ with $j \neq m$.

Proof. The proof was omitted because of the space constraints. □

Theorem 3. *The schedule C_I is a 1.5-approximation for k-path graphs when the originator is an internal vertex.*

Proof. The proof was omitted because of the space constraints. □

Theorem 4. *The 1.5 approximation guarantee of the odd–even internal schedule C_I is asymptotically tight.*

Proof. The proof was omitted because of the space constraints. □

6. Conclusion

We studied broadcasting in k-path graphs under the classical model and analyzed a simple deterministic odd–even schedule based on complementary parity orders at the two junction vertices. We proved that this schedule is an unconditional 1.5-approximation for both junction and internal originators. Once the paths are indexed by non-increasing length, the schedule is constructed and evaluated in $O(k)$ time, with an additional $O(k \log k)$ preprocessing step only if sorting is required. We also showed that the analysis is asymptotically tight for the odd–even schedules by giving infinite families of instances, for both choices of originator, whose achieved ratios approach $3/2$. Thus, improving beyond the 1.5 guarantee would require a different scheduling rule or additional structural ideas. An interesting direction for future work is to determine whether comparably simple fixed-order schedules can beat this bound, or whether stronger guarantees for k-path graphs necessarily require more adaptive methods.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT by OpenAI for grammar and spelling checks, and paraphrasing and rewording. The authors reviewed and verified all content and take full responsibility for the final manuscript.

References

[1] S. M. Hedetniemi, S. T. Hedetniemi, A. L. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (1988) 319–349. doi:10.1002/NET.3230180406.

[2] P. Fraigniaud, E. Lazard, Methods and problems of communication in usual networks, Discrete Applied Mathematics 53 (1994) 79–133. doi:https://doi.org/10.1016/0166-218X(94)90180-5.

[3] H. Harutyunyan, A. Liestman, J. Peters, D. Richards, Broadcasting and Gossiping, 2013, pp. 1477–1494. doi:10.1201/b16132-87.

[4] A. Farley, S. Hedetniemi, S. Mitchell, A. Proskurowski, Minimum broadcast graphs, Discrete Mathematics 25 (1979) 189–193. doi:https://doi.org/10.1016/0012-365X(79)90022-0.

[5] L. Gargano, U. Vaccaro, On the construction of minimal broadcast networks, Networks 19 (1989) 673–689. doi:https://doi.org/10.1002/net.3230190606.

[6] A. M. Farley, S. T. Hedetniemi, Broadcasting in grid graphs, in: Proceedings of the 9th SE Conference on Combinatorics, Graph Theory and Computing, Utilitas Mathematica, Winnipeg, 1978, pp. 275–288.

[7] J.-H. Park, K.-Y. Chwa, Recursive circulants and their embeddings among hypercubes, Theoretical Computer Science 244 (2000) 35–62. doi:https://doi.org/10.1016/S0304-3975(00)00176-6.

[8] W. Knödel, New gossips and telephones, Discrete Mathematics 13 (1975) 95. doi:https://doi.org/10.1016/0012-365X(75)90090-4.

[9] P. J. Slater, E. J. Cockayne, S. T. Hedetniemi, Information dissemination in trees, SIAM Journal on Computing 10 (1981) 692–701. doi:10.1137/0210052.

[10] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.

[11] M. Middendorf, Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2, Information Processing Letters 46 (1993) 281–287. doi:https://doi.org/10.1016/0020-0190(93)90066-I.

[12] M. Dinneen, The complexity of broadcasting in bounded-degree networks (2000). doi:10.48550/arXiv.math/9411222.

[13] Proskurowski, Minimum broadcast trees, IEEE Transactions on Computers C-30 (1981) 363–366. doi:10.1109/TC.1981.1675796.

[14] H. A. Harutyunyan, E. Maraachlian, On broadcasting in unicyclic graphs, Journal of Combinatorial Optimization 16 (2008) 307–322. doi:10.1007/s10878-008-9160-2.

[15] M. Čevnik, J. Žerovnik, Broadcasting on cactus graphs, Journal of Combinatorial Optimization 33 (2017) 292–316. doi:10.1007/s10878-015-9957-8.

[16] A. Ambashankar, H. A. Harutyunyan, Broadcasting in stars of cliques, in: Combinatorial Algorithms, Springer Nature Switzerland, Cham, 2024, pp. 485–496.

[17] H. A. Harutyunyan, N. Hovhannisyan, R. Magithiya, Deep heuristic for broadcasting in arbitrary networks, in: 2022 21st International Symposium on Parallel and Distributed Computing (ISPDC), 2022, pp. 1–8. doi:10.1109/ISPDC55340.2022.00010.

[18] R. Ravi, Rapid rumor ramification: approximating the minimum broadcast time, in: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, SFCS '94, IEEE Computer Society, 1994, p. 202–213. doi:10.1109/SFCS.1994.365693.

[19] G. Kortsarz, D. Peleg, Approximation algorithms for minimum-time broadcast, SIAM Journal on Discrete Mathematics 8 (1995) 401–427. doi:10.1137/S0895480193245923.

[20] P. Fraigniaud, S. Vial, Approximation algorithms for broadcasting and gossiping, Journal of Parallel and Distributed Computing 43 (1997) 47–55. doi:https://doi.org/10.1006/jpdc.1997.1318.

[21] M. Elkin, G. Kortsarz, A combinatorial logarithmic approximation algorithm for the directed telephone broadcast problem, SIAM Journal on Computing 35 (2005) 672–689. doi:10.1137/S0097539704440740.

[22] H. A. Harutyunyan, A. L. Liestman, Improved upper and lower bounds for k-broadcasting, Networks 37 (2001) 94–101. doi:https://doi.org/10.1002/1097-0037(200103)37:2<94::AID-NET4>3.0.CO;2-6.

[23] H. A. Harutyunyan, S. Kamali, Efficient broadcast trees for weighted vertices, Discrete Applied Mathematics 216 (2017) 598–608. doi:https://doi.org/10.1016/j.dam.2016.09.029.

[24] J. Hromkovič, R. Klasing, B. Monien, R. Peine, Dissemination of Information in Interconnection Networks (Broadcasting & Gossiping), Springer US, 1996, pp. 125–212. doi:10.1007/978-1-4757-2491-2_5.

[25] P. Bhabak, H. A. Harutyunyan, Approximation algorithm for the broadcast time in k-path graph, Journal of Interconnection Networks 19 (2019) 1950006:1–22. doi:10.1142/S0219265919500063.

[26] H. Harutyunyan, N. Hovhannisyan, Improved approximation for broadcasting in k-path graphs, in: Proceedings of the 17th Annual Conference on Combinatorial Optimization and Applications (COCOA), volume 14462 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 111–122. doi:10.1007/978-3-031-49614-1_7.

[27] J. Bringolf, H. A. Harutyunyan, S. Kamali, S.-M. Seyed-Javadi, On hardness and approximation of broadcasting in structured graphs, 2026. doi:10.48550/arXiv.2510.20026. arXiv:2510.20026, arXiv:2510.20026v2.

[28] J. Bringolf, A.-L. Ehresmann, H. A. Harutyunyan, A linear-time 1.5-approximation for broadcasting in k-cycle graphs, in: X. Li, Z. Zhang (Eds.), Algorithmic Aspects in Information and Management, Lecture Notes in Computer Science, Springer Singapore, 2026. Proceedings of the 20th International Conference on Algorithmic Aspects in Information and Management, AAIM 2026, accepted for publication.