

Principles of Distributed Computing

Advanced Exercise: Low-Congestion Shortcuts in Planar Networks

ETH Eidgenössische Technische Hochschule Zürich | Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2026 | Dr. G. Zuzic

Background and Definitions

Consider a planar graph $G = (V, E)$ with n nodes and diameter D . Let $S_1, S_2, \dots, S_N$ be a partition of V into disjoint subsets such that each subset induces a connected subgraph $G[S_i]$.

In distributed graph algorithms (such as Minimum Spanning Tree or Min-Cut), we often need to perform aggregate communication (e.g., convergecast or broadcast) within each part S_i simultaneously. Because the diameter of an individual part $G[S_i]$ can be much larger than the network diameter D , we augment each part with a **shortcut** subgraph $H_i \subseteq G$.

Definition 1 (Low-Congestion Shortcut). Given a graph $G = (V, E)$ and a partition of V into disjoint connected subsets $S_1, \dots, S_N$, an α -congestion shortcut with dilation β is a collection of subgraphs $H_1, \dots, H_N \subseteq G$ such that: 1. **Dilation:** For each $i \in [1, N]$, the diameter of the augmented subgraph $G[S_i] + H_i$ is at most β . 2. **Congestion:** For each edge $e \in E$, the number of subgraphs $G[S_i] + H_i$ containing e is at most α .

Let T be a Breadth-First Search (BFS) spanning tree of G rooted at an arbitrary node r , with depth at most D . Using a planar embedding of G , a left-to-right Depth-First Search traversal of T assigns each node a **left-ID** and **right-ID**. For each part S_i , we identify its **leftmost** node (smallest left-ID) and **rightmost** node (smallest right-ID). The two tree paths connecting the root r to the leftmost and rightmost nodes define the **boundary paths** of S_i , which diverge at the Lowest Common Ancestor (LCA) of S_i .

Task 1: The $O(D)$ Congestion, $O(D^2)$ Dilation Construction

To build intuition, we first analyze a baseline shortcut construction that completely excludes boundary paths.

Shortcut Rule (Version 1): For each part S_i , define H_i to be the subset of BFS tree edges e that satisfy at least one of the following two conditions: 1. **Ancestor Rule:** S_i contains a BFS-ancestor of e . 2. **Encapsulation Rule:** S_i contains a BFS-descendant of e , but e is **not** on the boundary paths of S_i . (In this case, we say S_i *encapsulates* e).

Note that edges along the boundary paths of S_i are explicitly excluded from H_i .

Question 1.1 (Congestion)

Prove that each BFS tree edge e is contained in at most $O(D)$ shortcut subgraphs H_i .

(Hint: Analyze the congestion from Condition 1 and Condition 2 separately. For Condition 2, use the Jordan Curve Theorem in the planar embedding to argue that if two parts S_i and S_j both encapsulate e , their encapsulation paths cannot be fully independent without crossing).

Question 1.2 (Dilation)

Prove that for each part S_i , the diameter of the augmented subgraph $G[S_i] + H_i$ is at most $O(D^2)$.

(Hint: Consider the boundary paths $\mathcal{P}$ of S_i , which contain at most $2D$ nodes. Group the nodes of S_i into boundary classes based on where their ancestor paths meet $\mathcal{P}$. Argue that nodes within the same class can reach their common boundary ancestor in $\leq 2D$ hops using encapsulated edges, and that traversing across all classes takes at most $O(D^2)$ hops).

Task 2: The $O(D \log D)$ Congestion and Dilation Construction

To reduce dilation from $O(D^2)$ to $O(D \log D)$, parts must be allowed to use some edges along their boundary paths. However, adding *all* boundary edges would cause congestion to blow up to N . We introduce a selective boundary inclusion rule based on geometric distances.

Junctions: For each part S_i , consider the boundary paths connecting its LCA to its leftmost and rightmost nodes. A node v on these boundary paths is called a **junction** if it touches a BFS tree edge encapsulated by S_i .

Shortcut Rule (Version 2): Add to the shortcut H_i from Version 1 any candidate boundary edge e that satisfies the following threshold rule: * Let u be the lowest junction on the boundary path strictly above e . * Let $\ell_1 = \text{dist}(e, u)$ be the tree distance from e to junction u . * Let $\ell_2 = \text{dist}(u, \text{LCA})$ be the tree distance from junction u to the LCA of S_i . * **Rule:** Include boundary edge e in H_i if and only if $\ell_2 \geq \frac{\ell_1}{2} > 0$.

Question 2.1 (Congestion)

Prove that under Version 2, each BFS tree edge e is contained in at most $O(D \log D)$ shortcut subgraphs H_i .

(Hint: Consider walking upwards from e along the tree path towards the root. Divide the path into geometric segments where the distance to e grows by a factor of $3/2$. Use the $\ell_2 \geq \ell_1/2$ rule to prove there are at most $O(\log D)$ distinct segments that can contribute parts using e . Then, use planar topology to bound the congestion within each segment by $O(D)$).

Question 2.2 (Dilation)

Prove that under Version 2, for each part S_i , the diameter of $G[S_i] + H_i$ is at most $O(D \log D)$.

(Hint: Walking down the boundary path from the LCA, junctions divide the boundary into consecutive intervals called 'pieces'. Argue that if a boundary piece is skipped (omitted from H_i), its length must be more than twice the sum of the lengths of all preceding pieces combined. Conclude that at most $O(\log D)$ boundary pieces are skipped, limiting the number of disconnected boundary classes to $O(\log D)$).