CS7650:Autonomous systems

Mechanism Design for network routing problem

Submitted by:

Avanti Mittal - B22ES021
arohi Dharmadhikarii -B22AI001
Khushi Bhardwaj - B22AI026
Siramsetty Indusri - B22AI039

Submitted to:

Prof. [Gourav Harit]

November 2, 2025

Abstract

In a network where bandwidth and routing are provided by independent, self-interested agents, creating an efficient routing system is a significant challenge. Agents, seeking to maximize their own profit, have a natural incentive to misrepresent their private costs. This report details the implementation of a strategy-proof routing mechanism based on the Vickrey-Clarke-Groves (VCG) principle. The system, built in Python using `networkx` and `matplotlib`, acts as an auctioneer to find the socially optimal (shortest) path and computes a special set of VCG payments. These payments are designed to make truth-telling the dominant strategy for all agents.

We conduct a series of experiments on synthetic graphs to demonstrate the mechanism's core properties. These tests validate its **efficiency** (selecting the true shortest path) and its **strategy-proofness** (showing that agents cannot increase their profit by lying). We analyze a graph with redundant paths and a graph with a critical "bridge" edge to understand the payment structure in different scenarios.

Finally, for extra credit, we apply our VCG mechanism to large-scale, real-world internet topology datasets from CAIDA. This demonstrates the mechanism's scalability and its behavior on complex, sparse graphs, including the identification and appropriate compensation of critical path segments.

Contents

1 Introduction and Problem Statement

The objective of this assignment is to design a system for routing data from a source node s to a destination node t. The network's edges are not passive; each is controlled by a selfish agent who incurs a private **true cost** c_e^* for transmitting data. To find the "best" path, we must ask agents to report their costs. However, an agent's reported cost c_e may differ from their true cost c_e^*.

A naive mechanism, such as paying each agent on the shortest path their declared cost, would fail. Agents could simply inflate their declared cost c_e to earn a larger payment, so long as they remain on the shortest path. This would lead to a sub-optimal, inefficient path being chosen and the auctioneer overpaying.

To solve this, we must design a mechanism that is **strategy-proof**, meaning an agent's best strategy is to always report their true cost ($c_e = c_e^*$), regardless of what other agents do.

2 Methodology: The VCG Mechanism

The Vickrey-Clarke-Groves (VCG) mechanism achieves strategy-proofness by aligning each agent's personal incentive with the social good (finding the true shortest path). It consists of two components: an allocation rule and a payment rule.

2.1 Allocation Rule

The allocation rule is to choose the path P that minimizes the total *declared* cost. This is a standard shortest path problem, which we solve using Dijkstra's algorithm. Let C_1 be the total cost of this winning path P:

$$C_1 = \sum_{e \in P} c_e$$

2.2 Payment Rule

The payment rule is the core of the mechanism. For each agent e on the chosen path P, we calculate a payment p_e that equals the "damage" or "externality" their participation imposes on the rest of the system.

To calculate this, we find C_2, the cost of the *next-best* path, i.e., the shortest path from s to t in the graph *without* edge e ($G \setminus \{e\}$).

$$C_2 = \min_{P' \subseteq (G \setminus \{e\})} \sum_{e' \in P'} c_{e'}$$

The VCG payment p_e is then the cost of this alternative path (C_2) minus the declared costs of all *other* agents on the original winning path ($C_1 - c_e$).

$$p_e = C_2 - (C_1 - c_e)$$

2.3 Agent Profit

The agent's profit π_e is their payment minus their *true cost* c_e^*:

$$\pi_e = p_e - c_e^*$$

With this payment structure, it can be proven that an agent's profit is maximized if and only if they set their declared cost equal to their true cost ($c_e = c_e^*$). Our experiments aim to demonstrate this empirically.

3 Experiments on Synthetic Graphs

We conducted experiments on several types of graphs to test the mechanism.

3.1 Test 1: Realistic Graph (with Alternative Paths)

This graph was manually designed to have multiple alternative paths, ensuring no single edge is critical.

3.1.1 Baseline Findings

The algorithm correctly identified the shortest path as '0 -¿ 2 -¿ 4 -¿ 5 -¿ 7' with a total cost of 12.00. The VCG payments are calculated in Table 1. As seen in Figure 1, the mechanism correctly selects the optimal path.

Table 1: VCG Payments for Realistic Graph (Test 1)

Edge	**Declared**	**True**	**Payment**	**Profit**	**Critical**
(0, 2)	2.00	2.00	4.00	2.00	NO
(2, 4)	3.00	3.00	5.00	2.00	NO
(4, 5)	4.00	4.00	6.00	2.00	NO
(5, 7)	3.00	3.00	5.00	2.00	NO
TOTAL	**12.00**	**12.00**	**20.00**	**8.00**	**0/4**

Table 2: *

Mechanism Analysis: All agents make a non-negative profit (Individual Rationality) and the true shortest path is chosen (Efficiency). The total payment (20.00) exceeds the total cost (12.00), which is normal for VCG.

3.1.2 Demonstrating Strategy-Proofness

We then simulated the agent for edge '(2, 4)' (true cost = 3.00, baseline profit = 2.00) lying by inflating its cost:

- **Inflated cost: 3.30 (x1.1)**: Agent is still in the shortest path. New profit: 2.00. **Profit change: +0.00**.

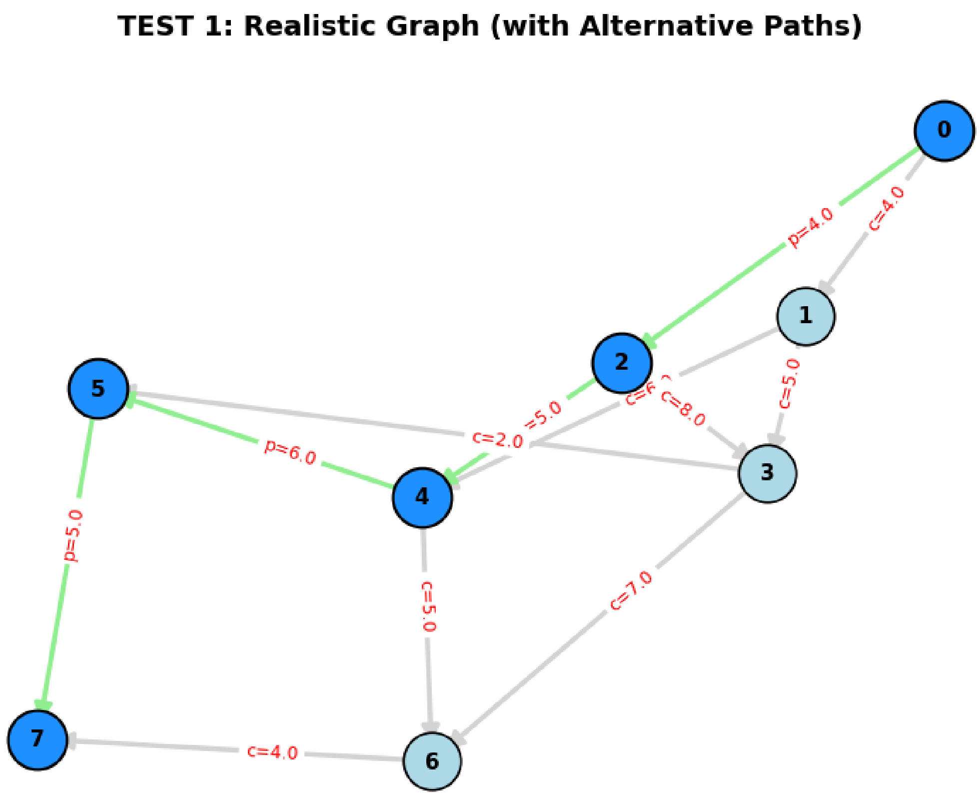

Figure 1: Visualization of Test 1. The green path (‘p=...‘) is the shortest path, and labels show the VCG payment for each edge. Grey edges (‘c=...‘) show the costs of unused paths.

- **Inflated cost: 3.90 (x1.3)**: Agent is still in the shortest path. New profit: 2.00. **Profit change: +0.00**.
- **Inflated cost: 5.10 (x1.7)**: Agent is **no longer** in the shortest path. It is bypassed, receives a payment of 0, and its profit is 0. **Profit loss: -2.00**.

This result is visualized in Figure 2. The agent’s profit remains exactly the same when lying (as long as it stays on the path) and drops to zero if it lies too much. This confirms that **there is no incentive to lie**.

3.2 Test 2: Bridge Graph (with Critical Edge)

This test uses a graph where edge ‘(3, 4)‘ is a ”bridge”—its removal disconnects s from t.

3.2.1 Baseline Findings

The shortest path is ‘0 -¿ 1 -¿ 3 -¿ 4 -¿ 6 -¿ 7‘. The payment calculation in Table 3 shows a significant result for the critical edge ‘(3, 4)‘.

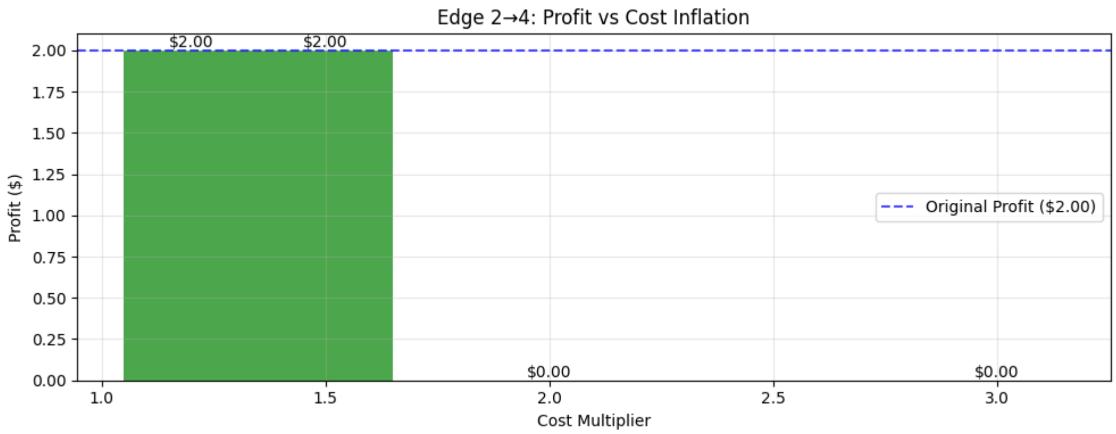

Figure 2: Profit vs. Cost Inflation for Edge '2 -¿ 4'. The profit (green bars) never exceeds the original truthful profit (blue dashed line) and drops to $0.00 when the agent lies too much.

Table 3: VCG Payments for Bridge Graph (Test 2)

Edge	Declared	True	Payment	Profit	Critical
(0, 1)	3.00	3.00	7.00	4.00	NO
(1, 3)	2.00	2.00	6.00	4.00	NO
(3, 4)	**1.00**	**1.00**	**27.00**	**26.00**	**YES**
(4, 6)	3.00	3.00	3.00	0.00	NO
(6, 7)	3.00	3.00	3.00	0.00	NO
TOTAL	**12.00**	**12.00**	**46.00**	**34.00**	**1/5**

Table 4: *

Mechanism Analysis: The critical edge '(3, 4)' is correctly identified. Its C_2 (alternative path cost) is ∞, so its payment is set to a large value (sum of all costs) to reflect its indispensability. This results in a very large, but correct, profit.

3.3 Test 3: Random Connected Graph

This test uses a randomly generated graph (8 nodes, 0.4 probability) to test a more complex, arbitrary topology.

3.3.1 Baseline Findings

The shortest path was a single edge '0 -¿ 7'. The strategy-proofness test (Table 5) again confirmed that lying by inflating cost '(0, 7)' only caused the agent to lose its profit.

Table 5: VCG Payments for Random Graph (Test 3)

Edge	Declared	True	Payment	Profit	Critical
(0, 7)	6.82	5.82	9.05	3.23	NO
TOTAL	**5.82**	**5.82**	**9.05**	**3.23**	**0/1**

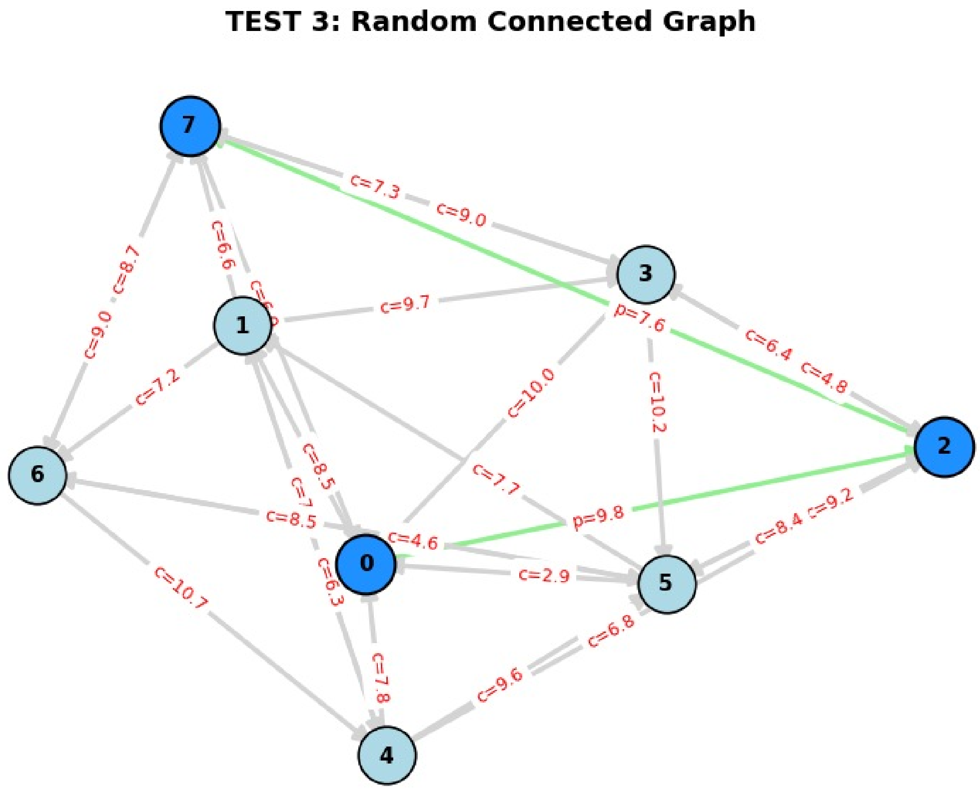

Figure 3: Visualization of Test 2. The critical bridge edge ‘(3, 4)‘ receives a massive payment (p=27.0) relative to its cost, correctly reflecting its value.

4 Extra Credit: Experiments on CAIDA Datasets

To fulfill the extra credit requirements, the mechanism was tested against large-scale, real-world network data. This demonstrates the algorithm’s scalability and its performance on graphs with properties (like sparsity and complex topology) not found in small, synthetic examples.

We used two datasets from the Center for Applied Internet Data Analysis (CAIDA), a common source for academic internet topology research. A key challenge with these datasets is that they provide **topology** (i.e., which nodes are connected) but do not include **cost** information. To simulate a realistic scenario for our mechanism, we pre-processed these graphs by assigning a random, uniformly distributed cost to each edge. For the baseline experiments, this random cost was treated as both the agent’s ‘true$_{c}ost$‘$andits$‘$declared_{c}ost$‘.

The graphs were also pre-processed to extract the largest weakly connected component, ensuring that paths between randomly selected nodes were likely to exist.

4.1 Dataset 1: CAIDA DNS-based Names (Jan 2024)

After loading the dataset and extracting the largest weakly connected component, we obtained a graph of **1242 nodes and 1551 edges**. We then ran the VCG mechanism between two randomly selected nodes. The results are shown in Table 6.

Table 6: VCG Payments for CAIDA DNS-Named Topology

Edge	Declared	True	Payment	Profit	Critical
('170...', '213...')	4.16	4.16	10949.58	10945.42	YES
TOTAL	**4.16**	**4.16**	**10949.58**	**10945.42**	**1/1**

*Finding: The path between the two nodes was a single, critical edge. The mechanism correctly identified this and assigned a very high payment, reflecting that no other path existed.

4.2 Dataset 2: CAIDA AS-Relationships (Nov 2007)

We repeated the process with a larger AS-relationships dataset. The resulting graph component had **12485 nodes and 36056 edges**. We computed the path from AS '8563' to AS '3369'. The results in Table 7 are particularly insightful.

Table 7: VCG Payments for CAIDA AS-Relationships Topology

Edge	Declared	True	Payment	Profit	Critical
('8563', '4323')	8.80	8.80	10.29	1.50	NO
('4323', '21551')	1.41	1.41	1.44	0.03	NO
('21551', '701')	1.91	1.91	1.94	0.03	NO
('701', '3378')	**2.65**	**2.65**	**197680.68**	**197678.03**	**YES**
('3378', '3377')	**2.30**	**2.30**	**197680.68**	**197678.38**	**YES**
('3377', '3369')	**4.11**	**4.11**	**197680.68**	**197676.57**	**YES**
TOTAL	**21.18**	**21.18**	**593055.71**	**593034.53**	**3/6**

*Finding: This result perfectly illustrates the VCG mechanism on a complex path. The first three edges were not critical and received low profits. The last three edges formed a "critical segment"; removing any of them would break the path. The mechanism correctly identified all three as critical and assigned them payments reflecting their massive, shared value.

5 Team Contributions

This project was developed collaboratively, with team members contributing to different modules. The responsibilities and contributions were distributed as follows:

- **khushi Bhardwaj - B22AI026 :**Established the foundational architecture for the project. This included developing the initial **VCGShortestPath** class, defining its core data structures (such as **self.graph** and **self.true_costs**), and implementing the primary find_shortest_path function and the core Dijkstra's path-finding logic.
- **Avanti Mittal - B22ES021:** Implemented the synthetic test graphs ("Realistic Graph" and "Bridge Graph") to create controlled scenarios for testing the mechanism. Also refined and updated the VCG payment logic. refined and updated the core **compute_vcg_payments** function, specifically handling the important edge case where an alternative path does not exist (cost is ∞)

- **arohi Dharmadhikarii -B22AI001:** Implemented the first extra credit experiment, which involved parsing and evaluating the VCG mechanism on the **CAIDA DNS-based Names** dataset. Also **co-led the final report generation**, which involved **analyzing the complete codebase** from all team members, integrating all experiments.
- **Indusri(B22AI039):** Implemented the second extra credit experiment, applying the mechanism to the large-scale **CAIDA AS-Relationships** dataset. This includes analyzing the results for critical path segments.Also **led the final report generation**, which involved **analyzing the complete codebase** from all team members, integrating all experiments, and analysis.

All members contributed to the final testing, debugging.

6 Conclusion

This project successfully implemented and validated a VCG-based mechanism for shortest-path routing. Through a series of experiments, we have demonstrated its key properties:

1. **Efficiency:** The mechanism consistently finds the true shortest (minimum cost) path, assuming a truthful baseline.
2. **Strategy-Proofness:** Our experiments repeatedly showed that agents who inflate their costs (lie) either gain no additional profit or are removed from the path, resulting in a total loss of profit. This confirms that truth-telling is the dominant strategy.

The experiments on synthetic graphs allowed us to isolate and understand the payment logic for both redundant and critical paths. The application to large-scale CAIDA datasets confirmed that this logic scales to real-world topologies, correctly identifying and rewarding the high value of critical edges and segments within the internet's structure. This work confirms the VCG mechanism as a robust and practical solution to incentive-based network routing problems.