Blockchain-Based Decentralized Peer-to-Peer Negawatt Trading in Demand-Side Flexibility Driven Transactive Energy System

By Mahdis Rahmani

Overview

Motivation

The Pizza Problem in a Hungry Neighborhood

- There’s only **one pizza shop** in your neighborhood.
- It makes only **100 slices per day**.
- Everyone is hungry around **7 PM**, and demand is high.

❗ Supply is limited, but demand keeps rising.
What do we do?

Motivation

Pizza Saving = Negawatt

🍕 The pizza shop says:

> “If you **give up some of your pizza**, I’ll give you tokens.”

You’re now being rewarded **not for eating**, but for **not eating**!
This saved pizza = **negawatt** in energy terms.

Flat payment? Unfair!

Each friend says:

- “I can give up 3 slices.”
- “I want 10 tokens.”

Motivation

Reverse Auction Begins

- 🧠 The pizza shop runs a **reverse auction**:
- “I need 50 slices saved. Who can give up pizza?
  And how much do you want in return?” Friends submit offers:

Wants Tokens	Can Save	Person
15	10	A
40	20	B
45	15	C

The shop selects the **cheapest mix** of offers just like choosing ingredients for the best price.

Motivation

Smart Selection with Knapsack

The shop uses a **fractional knapsack algorithm**:

- Sorts offers by **cost per slice** (tokens/slice)
- Picks offers **from cheapest to most expensive**
- Takes only what it needs to meet the target (50 slices)

🎯 Result = Goal achieved **at lowest total cost** = **Social cost minimized**

Motivation

What If Someone Fails

Friend A promised 20 slices…
But only saved 15!
What now?

✔️ They open a **second auction**:

"I need 5 slices — I'll buy from friends!"

This is the **secondary market**
→ You trade saved pizza **with each other**, not just with the shop.

Motivation

What's New? (Contribution of the Paper)

🎁 In the past:

- You could **only trade** with the pizza shop.
- Prices were **fixed**, no matter how hard it was to give up pizza.
- People with higher effort got **underpaid**.

🚀 This paper's idea:

- **You can trade with friends too!**
- **You decide** your price.
- A smart auction **picks the best deals**
- You're paid **fairly** using VCG rules.

Motivation

The Pizza Dilemma in the Hungry Neighborhood

Real Energy System ⚡	Pizza World 🍕
DSRA (Demand-Side Response Aggregator)	Pizza shop
Kilowatt-hours (kWh)	Pizza slices
Saving electricity	Giving up slices
Payments / energy tokens	Tokens
Demand response auction	Reverse auction
Peer-to-peer negawatt trading	Secondary market
Optimization algorithm (Fractional Knapsack)	Knapsack selection
Incentive-compatible fair compensation	VCG payments

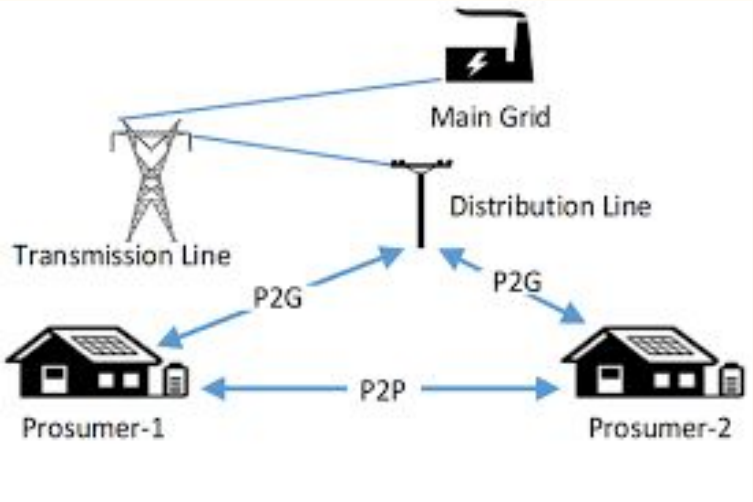
Motivation

The Pizza Dilemma in the Hungry Neighborhood

The Pizzeria (DSRA): Only one in town. It has a **limited daily supply of pizzas (energy)** — let's say 100 slices/day.

You and Your Friends (Consumers): Everyone orders when they're hungry.

The Grid's Problem: When everyone's hungry at the same time (peak demand), **pizza runs out**



Title Breakdown:

Blockchain-Based Decentralized Peer-to-Peer Negawatt Trading in Demand-Side Flexibility Driven Transactive Energy System

Blockchain-Based

Decentralized

Peer-to-Peer (P2P)

Negawatt

Trading

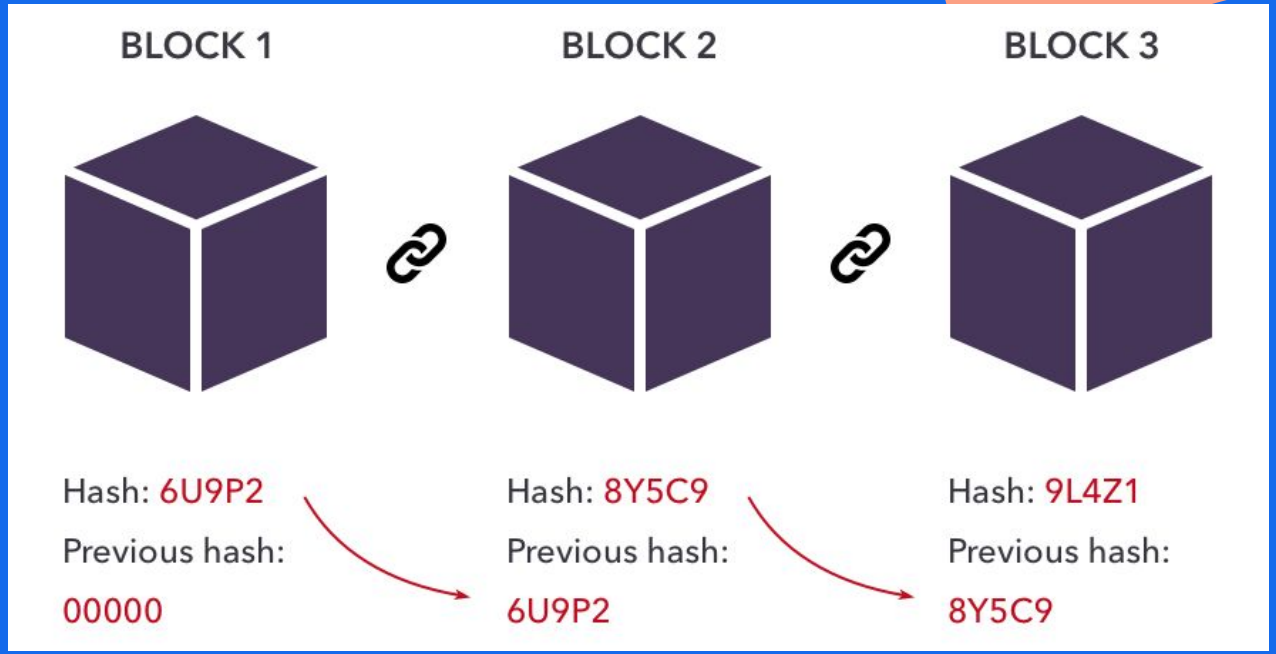
Demand-Side Flexibility

Transactive Energy System

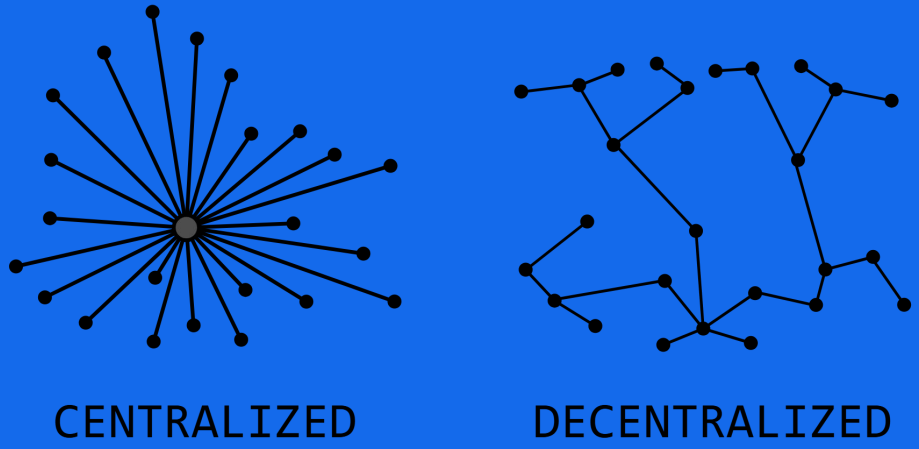
Blockchain-based:

A digital database or ledger that is distributed among the nodes of a P2P network.

A technology that securely records transactions across many computers, making it difficult to alter or hack.

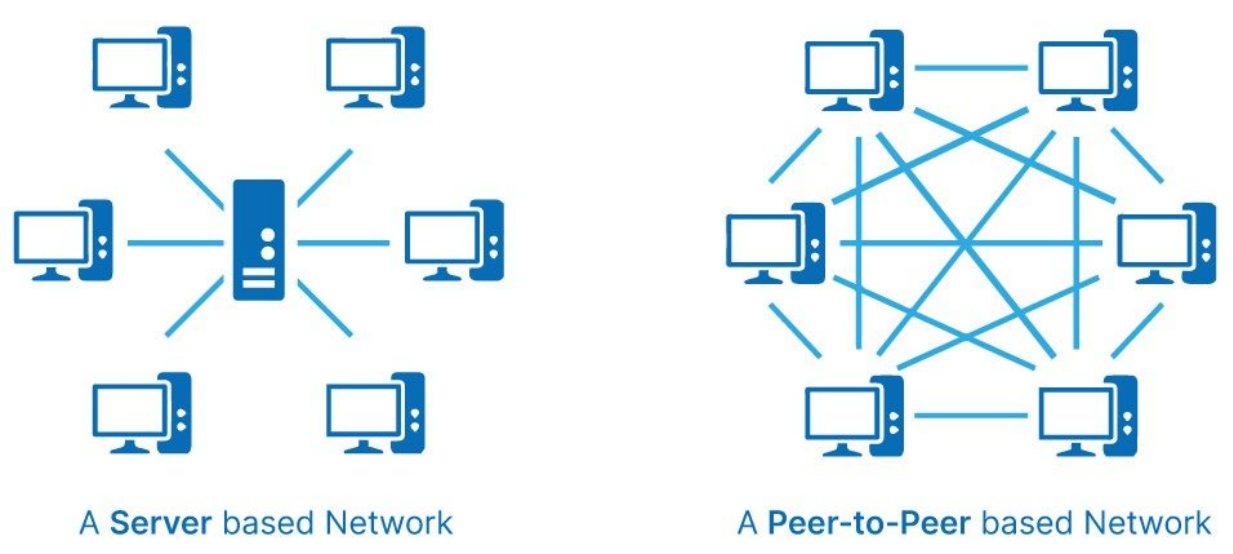


Decentralized:



A system that does not have a central authority or control; instead, power and decision-making are distributed among participants.

Peer-to-Peer (P2P):

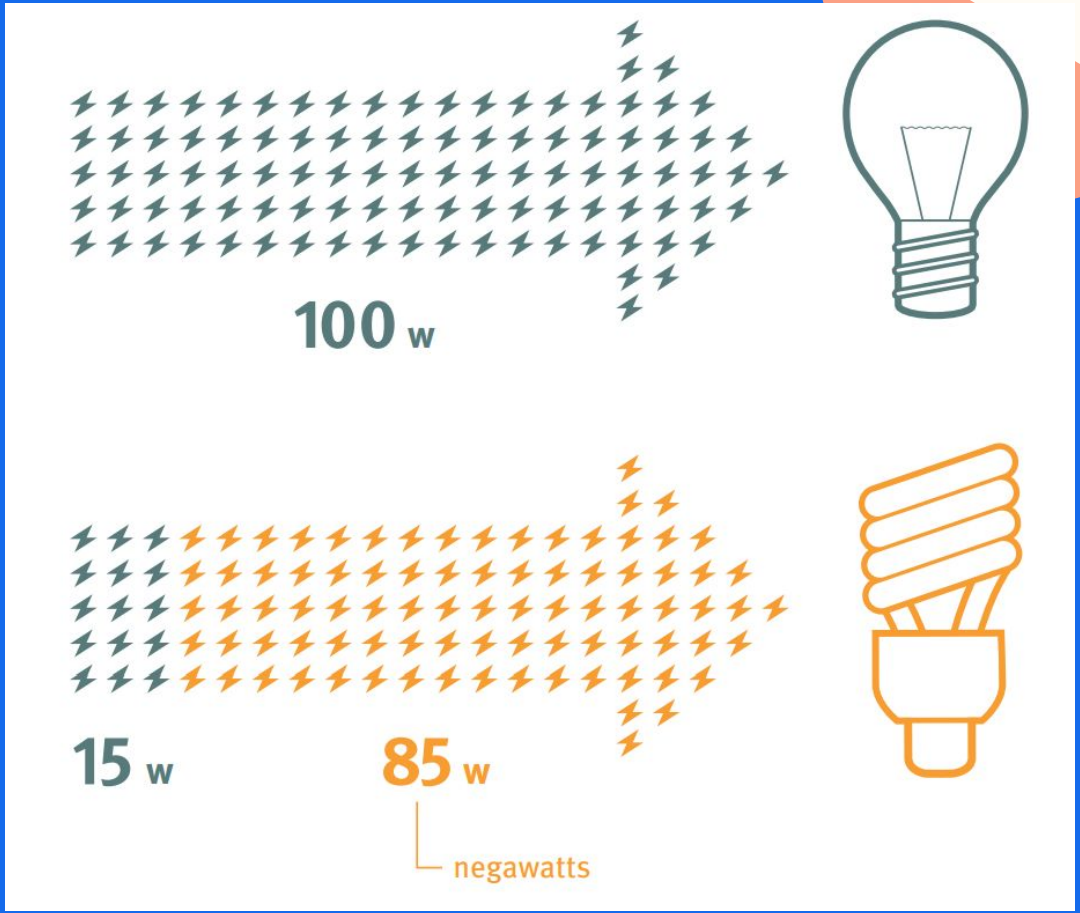


A network model where participants interact directly with each other rather than through an intermediary.

Negawatt:

This is about selling *saved electricity* (negawatts), not electricity you generate.

A concept referring to energy savings or reduction in energy consumption, essentially representing "negative" electricity usage.



Trading:

The act of buying and selling goods or services; in this context, it refers to exchanging energy savings or negawatts.

Demand-Side Flexibility:

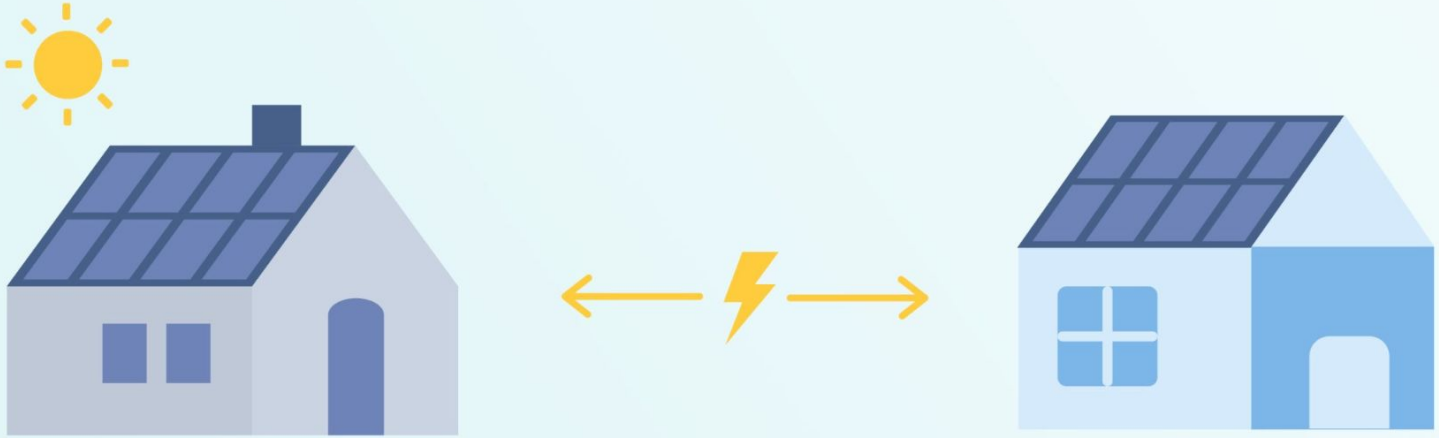
The ability of consumers to adjust their energy usage in response to supply conditions, such as reducing use during peak demand times.

Transactive Energy System:

A smart system where people trade energy (or savings) dynamically, based on prices and needs.

🤔 What is Peer-to-Peer (P2P) Energy Trading?

Peer-to-peer energy trading enables grid-connected users to buy and sell excess energy directly with others through a secure platform, giving consumers the freedom to choose their energy buyers and sellers.



Two Worlds, One Idea:

Section	🍕 Pizza Analogy (Simple)	⚡ Real Energy System
Introduction	There's only one pizza shop with limited slices. People get hungry at the same time. The shop needs a better way to reduce demand.	The power grid has peak demand periods and limited supply. Utilities need to manage demand to avoid blackouts or high costs.
Novelty & Contribution	Before, only the pizza shop could offer tokens for saved slices. Now friends can trade saved slices with each other too. And prices are fair!	Adds peer-to-peer (P2P) negawatt trading and fair pricing through fractional knapsack + VCG — not just grid-to-consumer like before.

Two Worlds, One Idea:

Problem Statement	The pizza shop used to pay the same amount to everyone, even if some gave up slices easily while others sacrificed a lot. That was unfair and inefficient.	Existing demand response models ignore individual costs and aren't incentive-compatible → leads to inefficiency and dropout.
Model	The shop announces "Save 50 slices by 7 PM!" Friends respond with how many slices they'll give up and for how many tokens. Smart devices enforce it.	DSRA announces demand reduction events. Consumers respond via Home Energy Management systems. The system collects offers via auction.

Two Worlds, One Idea:

Blockchain Implementation	A public whiteboard (blockchain) records all trades. Rules are run by a robot judge (smart contract) that picks winners and pays automatically.	Ethereum blockchain hosts the auction as a smart contract: receives bids, selects winners, calculates VCG payments, ensures transparency.
Case Study	In a real example, friends submitted slice-saving offers. The shop picked the best mix to reach 50 slices. Someone failed and had to buy slices in a second auction.	A simulated auction with 5 consumers. DSRA reduced 100 kW at minimum cost. A peer later failed and traded in the secondary market.

Two Worlds, One Idea:

Simulation & Discussion	The shop saved tokens compared to before. People who saved pizza got fair rewards. Truthful offers led to best results.	Simulation shows lower social cost, truthful bidding, and higher fairness. Confirms theory with empirical results.
Conclusion	A smart system lets pizza saving happen fairly, efficiently, and automatically. The shop saves money. Everyone eats better later.	A decentralized, fair, automated P2P energy trading system based on fractional knapsack and VCG improves efficiency and participation.

Game Theoretic Approach:

📚 Formal Definition:

In **game theory**, an **agent** is:

> "An independent decision-making entity that has preferences, takes actions (strategies), and seeks to maximize its own utility in an environment where other agents are also acting."

🔍 More Simply:

An **agent** is:

- A **player** in the game
- Has **private information** (like how hungry they are, or how much saving pizza costs them)
- Chooses a **strategy** (like how much to bid or save)
- Wants to **maximize their payoff** (e.g., get paid well or spend little)
- And crucially: agents **interact** — your choice affects others, and vice versa

👫 In Our Pizza Example:

Agent Type	Goal
DSRA	Buy saved slices at lowest cost
Consumer A	Sell saved pizza, earn the most
Consumer B	Same — maximize their own tokens

So in a **reverse auction**, each consumer is an **agent** who:

- Knows their true cost (private info)
- Submits a bid (strategy)
- Gets paid based on the mechanism (VCG)

Knapsack Problem vs. Fractional Knapsack

Feature	0-1 Knapsack	Fractional Knapsack
Decision Type	Binary (include item or not)	Continuous (can include partial items)
Problem Type	NP-Hard	Greedy-solvable in O(n log n) time
Solution Method	Dynamic Programming / Branch & Bound	Greedy algorithm based on value/weight
Use Case in Paper	Not used	✅ Used to minimize **social cost**
Relevance	Used when partial allocation is not realistic	Used when partial allocation is allowed (like negawatts)

Fractional Knapsack

Maximize:

$$\sum_{i=1}^{n} x_i \cdot v_i$$

Subject to:

$$\sum_{i=1}^{n} x_i \cdot w_i \leq W, \quad 0 \leq x_i \leq 1$$

Where:

- x_i: fraction of item i selected
- v_i: value of item i
- w_i: weight of item i
- W: total weight capacity

This exact model is used in the paper to choose negawatt offers.

Fractional Knapsack

🔧 Proposed Solution

- **A decentralized P2P auction model** where:
  - Buyers initiate auctions.
  - Sellers submit bids: amount A, price B.
  - Winner selection = **fractional knapsack problem**.
  - Payment = **VCG (Clarke pivot rule)** → ensures:
    - **Truthfulness**
    - **Individual Rationality**
    - **Efficiency**

Fractional Knapsack

Optimization Problem (Fractional Knapsack):

Minimize:

$$\sum_{i=1}^{n} x_i \cdot b_i$$

Subject to:

$$\sum_{i=1}^{n} x_i \cdot a_i \geq T, \quad 0 \leq x_i \leq 1$$

Where:

- x_i: fraction of bid accepted
- a_i: negawatts
- b_i: bid price

Greedy strategy: Sort by cost per negawatt $\frac{b_i}{a_i}$, pick lowest first.

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

> **VCG = a mechanism that pays people based on how much their presence helps the group.**

💡 Intuition:

Let's say you're trying to build a football team. You want to pay players fairly.

In a VCG world:

- You don't pay a player based on what they *asked for*
- You pay them based on:

  > "How much worse would the team do **if they didn't play**?"

🎯 That's called their **externality** — the value they add to the team.

In short:

> 🟢 **Help the group more → get paid more**
> 🔴 **Lie about your value → you might lose out**

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

In a **reverse auction** (as in your paper), the VCG formula for how much a selected participant i is paid is:

$$\boxed{\text{VCG Payment}_i = \text{Total Cost Without } i - \text{Cost of Others With } i}$$

Where:

- $\text{Total Cost Without } i$ = the **minimum cost** the buyer would pay **if participant i didn't exist**
- $\text{Cost of Others With } i$ = the total cost of **everyone else** in the winning set (excluding i)

💡 Intuition:

> You are paid based on how much **extra cost you saved** the system by being present.

This is sometimes called the **Clarke pivot rule**.

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

The **pizza manager (DSRA)** needs to save **5 slices** today.

Three friends offer to save some slices and submit their bids.

Person	Can Save	Asks For	Tokens/slice
Alice	3	9	3.00
Bob	2	6	3.00
Charlie	2	10	5.00

🎯 Goal: Choose a combination that saves **at least 5 slices**, with **minimum total cost**, and compute **VCG payments** for winners.

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

🔢 Step 1: Choose Winners (Greedy/Fractional Knapsack)

Target = 5 slices.

- Take Alice: 3 slices, cost = 9
- Take Bob: 2 slices, cost = 6
- ✅ Total = 5 slices, total cost = **15 tokens**

Charlie is excluded.

💸 Step 2: Compute VCG Payments

💡 Remember:

> Each winner is paid based on the difference they made to total cost.

Let's calculate **how much each winner helped**.

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

🧑🍳 Payment for Alice

Step 1: Remove Alice from the system

Now we have:

- Bob: 2 slices for 6
- Charlie: 2 slices for 10

We need 5 slices → Uh oh! Only 4 slices available.

To reach 5 slices, DSRA must buy 1 slice from a fallback (e.g., generator) at **reservation price = 6 tokens**.

So new total cost =
= Bob (6) + Charlie (10) + Fallback (6) = **22 tokens**

Step 2: Compare with original

- With Alice: Cost = 15
- Without Alice: Cost = 22
  ➡️ **Alice saved the system 7 tokens**

🎯 So Alice gets **paid 7 tokens** (even though she only asked for 9)

✅ She's happy, and incentivized to **be honest** in the future.

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

🧑🍳 Payment for Bob

Step 1: Remove Bob

Now we have:

- Alice: 3 slices
- Charlie: 2 slices

That's exactly 5 slices.

Cost = Alice (9) + Charlie (10) = **19 tokens**

Step 2: Compare with original

- With Bob: 15 tokens
- Without Bob: 19 tokens

➡️ Bob saved the system **4 tokens**

🎯 Bob gets paid **4 tokens**

✅ Even though he asked for 6

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

Summary Table

Person	Asked For	VCG Payment	Got More/Less	Reason
Alice	9	7	❌ Less	Still worth it
Bob	6	4	❌ Less	Still worth it
Charlie	10	0	❌ Not chosen	Too expensive

📌 **Everyone still earns non-negative utility** (Individual Rationality)

📌 **Best strategy = tell the truth** (Incentive Compatibility)

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

🍕 Scenario: Overpaid Participant via VCG

Setup:

The pizza shop needs **5 slices saved**.

Person	Can Save	Asks For	Cost/slice
Alice	3	6	2.0
Bob	2	8	4.0
Charlie	2	15	7.5

Step 1: Choose Winners

- Alice (3 slices for 6)
- Bob (2 slices for 8)

  ✅ Total = 5 slices, total cost = **6 + 8 = 14 tokens**

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

Step 2: Compute Alice's VCG Payment

a. Remove Alice

- Left: Bob (2) and Charlie (2) = 4 slices only
- Need 1 more slice → fallback cost = 6 tokens

➡️ Total cost without Alice = Bob (8) + Charlie (15) + Fallback (6) = **29 tokens**

b. Cost of others with Alice = Bob (8)

➡️ VCG Payment to Alice = 29 – 8 = **21 tokens**

🎉 She asked for 6 tokens, but **gets paid 21 tokens!**

Why? Because **without her, the system would pay 29 instead of 14** — she **saved 15 tokens**.

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

Step 3: Compute Bob's VCG Payment

a. Remove Bob:

- Alice (3) + Charlie (2) = 5 slices
- Cost = Alice (6) + Charlie (15) = **21 tokens**

b. Cost of others with Bob = Alice (6)

➡️ VCG Payment to Bob = 21 – 6 = **15 tokens**

He asked for 8, but gets **15 tokens**

✅ Also overpaid!

Vickrey-Clarke-Groves Payment (Clarke Pivot Rule):

✅ Summary

Person	Asked For	VCG Payment	Overpaid?
Alice	6	21	✅
Bob	8	15	✅
Charlie	15	0	❌

🧠 This happens because both Alice and Bob were **very important** to keeping the cost low — **their absence would've forced expensive fallback options** or inclusion of Charlie.

Case 1: Primary Market

- **DSRA target:** 100 kW
- **Fallback cost (reservation):** 500 tokens
- **Bids:**

Consumer	Available (kW)	Price (tokens)
C1	30	120
C2	25	100
C3	45	150
C4	10	20
C5	20	60

- **Outcome:**
  - Chosen bidders: C3, C4, C5, partial C1
  - **Total cost:** 443 tokens (saves 57 tokens)
  - **All winning consumers gain positive utility** based on VCG

➤ Case 2: Secondary Market

- **Consumer 3** fails to meet 40% of their obligation → initiates P2P auction:
  - Needs: 18 kW
  - Willing to pay: 90 tokens
- **Other consumers offer their saved units**
- **Outcome:**
  - C1 and C2 cover the shortfall
  - Cost: 87 tokens (cheaper than penalty)
  - **All participants gain utility**

Fractional Knapsack

The DSRA wants to save **50 pizza slices**.

It posts an auction asking people to submit:

- **How many slices they'll save**
- **How many tokens they want**

📤 Offers (Bids):

Person	Can Save	Wants (tokens)	Cost per Slice
A	20	40	2.0
B	15	45	3.0
C	10	20	2.0
D	10	50	5.0
E	10	15	1.5

Fractional Knapsack

□ Step 1: Sort by cost per slice (lowest first):

Person	Can Save	Wants	Cost/slice
E	10	15	1.5
A	20	40	2.0
C	10	20	2.0
B	15	45	3.0
D	10	50	5.0

Fractional Knapsack

🎒 Step 2: Greedy Fractional Knapsack

Keep picking until we save 50 slices.

- Take all of E → 10 slices, +15 tokens → **Running total: 10 slices, 15 tokens**
- Take all of A → 20 slices, +40 tokens → **Total: 30 slices, 55 tokens**
- Take all of C → 10 slices, +20 tokens → **Total: 40 slices, 75 tokens**
- Take **10/15 of B** → 10 slices, $\frac{10}{15} \times 45 = 30$ tokens

✔ Total saved = 50 slices

✔ Total tokens = **15 + 40 + 20 + 30 = 105 tokens**

Conclusion

Proposed: Decentralized P2P negawatt trading system

Based on:

- Greedy fractional knapsack (optimal selection)
- VCG mechanism (fair payments)
- Blockchain smart contracts (secure, transparent)

Challenges & Future Work

Greedy + VCG = computational cost

Ethereum scalability limits (gas, PoW)

Thank you!