EIE3280 Project Final Report

Xinyu LIANG
121010060@link.cuhk.edu.cn

Tianyi JIANG
121090232@link.cuhk.edu.cn

Xiaohui CAO
122040049@link.cuhk.edu.cn

Jiahao YANG
122090646@link.cuhk.edu.cn

***Abstract*—This project presents a dynamic pricing model for Internet Service Providers that balances network congestion, revenue, and user fairness. Users are grouped by data usage, pricing is adjusted through rule-based logic and Q-learning. Simulations are performed to evaluate usage, congestion, and satisfaction. The model is set to be used in a controlled environment focusing on university campuses.**

I. Introduction

On university campuses, especially during peak hours—such as lunch time or in the evenings, students often experience slow network speeds, buffering during video calls or failed downloads. This is due to too many users consuming large amounts of bandwidth at the same time. During exam season or live-streamed lectures, network infrastructure becomes overloaded, causing delays and interruptions. Students who rely on online learning platforms for assessments or online-based classes were heavily affected. This project therefore proposes a solution based on similar logic from large Internet Service Providers (ISPs) but focusing on university campuses. In cities like Toronto or Hong Kong, ISPs often offer data caps or speed-based plans to manage demand[1], [2]. Dynamic pricing in such cases helps balance network load. This project was inspired to apply this idea in a controlled academic environment. Moreover, this project is interested in suggesting fairness in this controlled environment; more specifically, we designed a new evaluation metric—average user satisfaction—which takes into account the network usage satisfaction of all users in the network.

May 11, 2025

II. Literature Review

Real-time pricing is an effective tool for managing congestion across various networked systems, including power grids, transportation, and communication networks. This review highlights three key studies relevant to our project, although with completely different usage, some similar approaches were used.

Yang et al. [3] developed a real-time dynamic pricing framework for electric vehicle charging stations, using a constrained Markov Decision Process and an Adaptive Model-Based Safe Deep Reinforcement Learning algorithm. Their focus on simplified user decision-making—balancing travel time, wait time, and charging cost—parallels our approach, which models user behavior with rationality but avoids excessive complexity.

Zhang et al. [4] explored dynamic pricing in mobile crowdsensing, using Q-learning to adapt payments based on user privacy preferences. Although targeting privacy rather than congestion, their use of reinforcement learning aligns with our project's core Q-learning model for pricing adjustments.

Räcke [5] proposed a theoretical framework for minimizing congestion in networked systems using hierarchical decomposition, providing competitive routing solutions. Unlike our adaptive, data-driven model, Räcke's work offers provable, static guarantees, highlighting the contrast between theoretical and learning-based methods.

Our project stands out from the three referenced studies by focusing on real-time dynamic pricing for mobile data in ISPs. Unlike the other works, we designed a hybrid pricing strategy that includes static, rule-based, and Q-learning methods, optimized for balancing network congestion, revenue, and fairness. While prior studies use complex constrained optimization or offer theoretical suggestions, our approach emphasizes practicality—featuring a lightweight simulation environment and built-in fairness metrics to ensure that pricing does not disproportionately affect low-usage or price-sensitive users.

III. Methodology

A. User Design

To simplify user behavior, the model classifies users into three categories based on application usage patterns and price sensitivity.

1) User Types:

- Light users: Basic communication needs (e.g., WeChat).
- Medium users: Video streaming needs (e.g., YouTube).
- Heavy users: Gaming needs (e.g., Video games).

2) Basic Rule: The following equation models how user's data usage responds to changes in pricing. When the price p is below or equal to the user's threshold p_0, they consume at a high usage level U_h. However, when the price exceeds p_0, the usage begins to decrease exponentially. The formula incorporates an exponential decay factor, scaled by user-specific elasticity α and sensitivity coefficient k. This dynamic behavior captures realistic user reactions to price increases and forms the basis for simulating price-sensitive demand in our system. The value of α is only related to the user type (light, medium, heavy). The value of k is used to retain the extended model.

$$\text{DataUsage}(p) = \begin{cases} U_h, & p \le p_0 \\ \max\left(U_l,\ U_h \cdot \exp(-\alpha k \Delta p)\right), & p > p_0 \end{cases} \tag{1}$$

where $\Delta p = p - p_0$, p_0 = price threshold,
α = price elasticity, U_l = minimum data usage needed,
U_h = maximum data usage expected.

3) Price Elasticity:

- Price elasticity affects how much a user's data usage changes in the face of rising data pricing. Typically, we would expect heavy users to have lower price elasticity because their demand for data is often immediate.

According to the above formula, user data consumption only vary with current network pricing. While additional influencing factors could be incorporated for refined modeling, our code implementation focuses on the core pricing mechanism.

B. Pricing Model

We introduce three pricing strategies.

1) Static Pricing: This is the simplest model, where the data price is fixed and does not respond to any changes in network conditions. It is calculated in advance based on the distribution of user types and their corresponding demand characteristics. Once the number of each user type is determined, a base price is adjusted using a weighted formula that reflects overall demand. However, this model cannot adapt to real-time fluctuations, making it less effective in dynamic environments. Its main advantage lies in its simplicity and predictability. The static pricing formula is defined as follows:

$$P = P_{\text{base}} + \underbrace{\left(\frac{\sum_k N_k U_h}{C}\right)}_{\text{demand ratio}} \cdot \underbrace{\left(\sum_k \frac{N_k}{N_{\text{total}}} p_k\right)}_{\text{weighted threshold}} \tag{2}$$

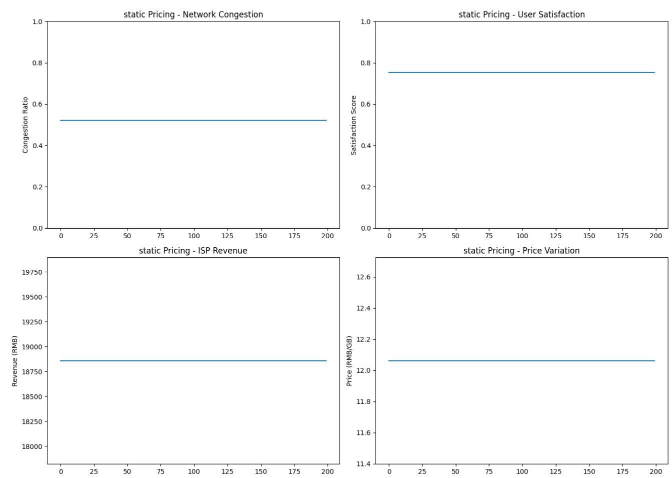

Fig. 1. Static Pricing

where:

- P_{base} is the base price,
- k is the type of users k,
- N_k is the number of users of type k,
- U_h is the maximum data usage expected k,
- C is the network capacity,
- p_k is the maximum price user k willing to pay (price threshold),
- $N_{\text{total}} = \sum_k N_k$ is the total number of users.

Fig. 1 presents the performance of the Static Pricing model across four key dimensions: network congestion, user satisfaction, ISP revenue, and price variation. As shown, all metrics remain constant over the simulation period due to the fixed nature of static pricing. The congestion ratio remains stable, user satisfaction remains moderate, and ISP revenue is steady, reflecting the lack of dynamic adjustment. This demonstrates the predictability and simplicity of static pricing but also highlights its limitation in adapting to changing network conditions and user behavior.

2) Rule-Based Pricing: This method is a dynamic pricing algorithm. The model will adjust the pricing in real time based on the current congestion rate level in the network.

Our pricing formula, achieves the following expectations: when the price tends to be stable, the difference between the current network congestion rate and the target congestion rate will approach 0, achieving our goal of regulating the network congestion rate. The Rule-Based Pricing formula is as follows:

$$p_{\text{next}} = p + \eta(c_c - c_t) \tag{3}$$

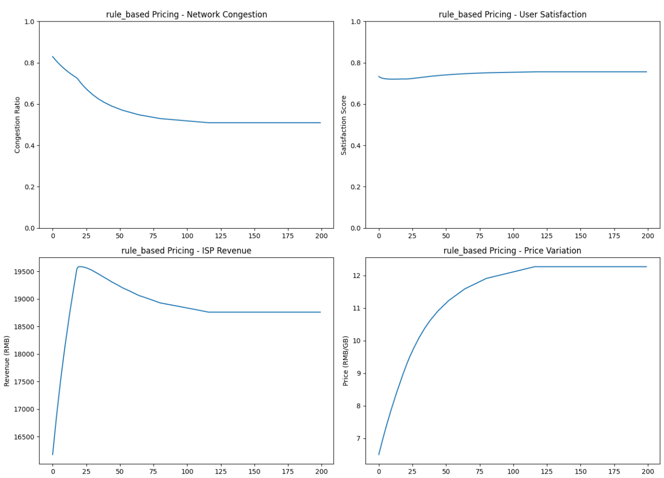

Fig. 2. Rule-Based Pricing

where:

- p: Current price
- p_{next}: New price
- η: Adjustment factor
- c_c: Real-time congestion level
- c_t: Target congestion level

Fig. 2 presents the performance of the Rule-Based Pricing model, in which it dynamically adjusts data prices based on real-time congestion feedback. The top-left plot shows a clear decline in the network congestion ratio as prices gradually increase, demonstrating the model's ability to reduce overload over time. Correspondingly, user satisfaction (top-right) improves slightly, stabilizing at a higher level than at the beginning. ISP revenue (bottom-left) peaks early as prices rise, then stabilizes as congestion is managed and user demand adjusts. The bottom-right chart displays price variation, where the price increases smoothly and eventually plateaus—highlighting the rule-based system's reactive but non-oscillatory nature. Overall, this model achieves moderate adaptability with predictable performance improvements.

3) Q-Learning Pricing: This is the best model in our simulations, using reinforcement learning to dynamically adjust prices in response to observed network conditions. The model trains the ISP agent to learn and to make the best decisions, such as increasing, maintaining, or decreasing the price based on the current state of the network, which includes factors like congestion and price. The model rewards or punishes the agent based on whether it can successfully adjust the network congestion rate towards the target congestion rate level in each iteration. Over time, the Q-learning algorithm learns the most effective pricing strategy, offering a data-driven and highly adaptive solution to congestion management. The framework of Q-Learning Pricing is as follows:

- Agent: The ISP acts as the Q-learning agent, responsible for dynamically adjusting data prices to optimize network performance. The agent learns from interaction with the environment over multiple iterations and can store learned policies for future reuse, enabling rapid deployment without the need for retraining.
- State Space: Each state is represented as a combination of the current network congestion level and pricing tier. By discretizing these values, we construct a finite state table that allows the agent to track system conditions and associate them with appropriate pricing responses.
- Action Space: The action space is defined as discrete pricing adjustments within a restricted range. This design ensures that changes in price are smooth and gradual during training, which helps stabilize learning and prevents erratic behavior.
- Reward Function: The reward is computed based on how closely the network congestion aligns with a predefined target. If the gap between actual and target congestion narrows compared to the previous time step, the agent receives a positive reward. Conversely, if the gap increases, it incurs a penalty. This incentivizes the agent to learn pricing strategies that gradually guide the system toward optimal load conditions.

Fig. 3 presents the performance of the Q-Learning Pricing model, which dynamically adjusts prices based on learned strategies that balance congestion, user satisfaction, and revenue. The top-left plot shows that network congestion remains consistently close to the target level, indicating that the learning agent effectively stabilizes network usage. User satisfaction (top-right) also remains stable and relatively high, reflecting the model's fairness and responsiveness. Unlike previous models, ISP revenue (bottom-left) and price variation (bottom-right) exhibit more fluctuations, a result of the model's ongoing exploration and adaptation. These controlled oscillations suggest that the Q-learning agent is actively balancing short-term adjustments with long-term performance optimization.

C. User Satisfaction

As shown in the formula, a user satisfaction formula was adopted, which considers the usage of data, the current price of the data and the level of congestion in

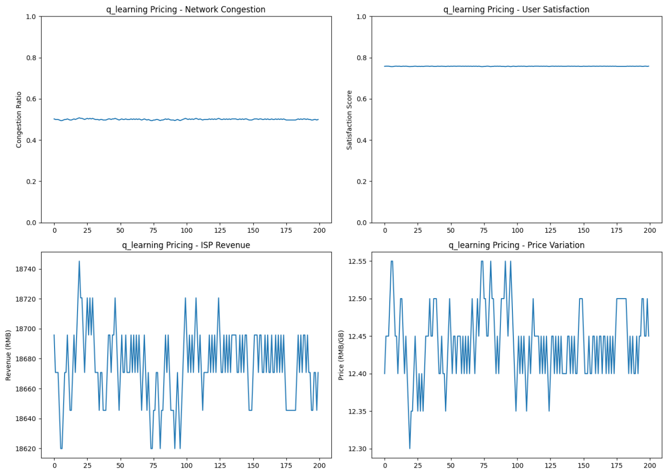

Fig. 3. Q-Learning Pricing

the network. By measuring the average user satisfaction within this network, we can evaluate the effectiveness of the pricing model.

$$s = 0.3 \cdot U_a/U_h + 0.1 \cdot P + 0.6 \cdot C$$

- **Price Tolerance** P:

$$P = \begin{cases} 1.0 & \text{if } p \leq p_0 \\ \frac{1}{1+\alpha \cdot \Delta p} + \lambda & \text{otherwise} \end{cases}$$

  where λ is the price penalty:

$$\Delta p = p - p_0$$
$$\lambda = \begin{cases} \frac{p_0 - p}{p_0} & \text{if } U_a = U_l \\ 0 & \text{otherwise} \end{cases}$$

- **Congestion Tolerance** C:

$$C = \begin{cases} 1.0, & \text{if } c_c \leq c_t \\ 1 + c_t - c_c, & \text{otherwise} \end{cases}$$

Symbol Definitions:

- U_h: maximum data usage expected
- U_l: minimum data usage needed
- U_a: actual user's data usage
- p_0: price threshold
- α: price elasticity coefficient
- p: current price
- c_c: current congestion level $\in [0, 1]$
- c_t: target congestion level $\in [0, 1]$

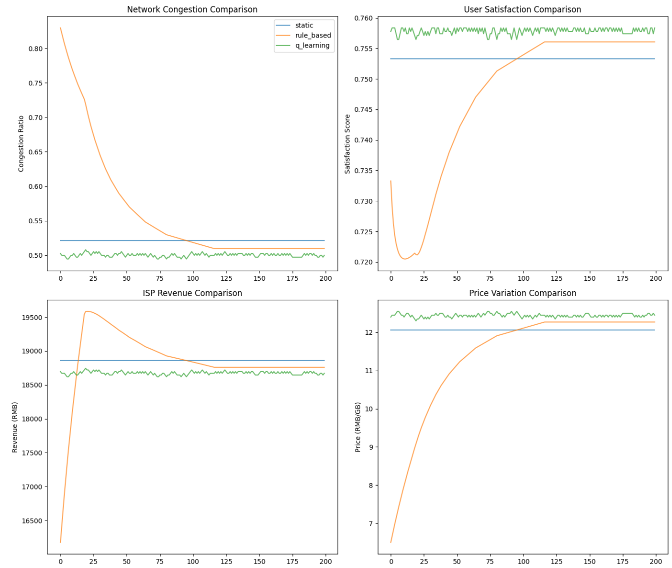

Fig. 4. Strategy Comparison

IV. Network Simulation

A. Strategy Comparison

Using the user satisfaction model, the effectiveness of the three pricing strategies can be visualized, as shown in Fig. 4

- Network Congestion Comparison: Q-Learning Pricing consistently maintains the lowest congestion ratio (~ 0.49) throughout the simulation, showing its ability to dynamically balance network load. Rule-Based Pricing starts with high congestion (~ 0.85) but steadily decreases as the system adjusts prices based on real-time conditions, eventually stabilizing near 0.52. In contrast, Static Pricing remains flat at around 0.54, reflecting its inability to adapt to fluctuating demand.
- User Satisfaction Comparison: Q-Learning delivers the highest and most stable user satisfaction score ~ 0.757), demonstrating its effectiveness in tailoring prices to user experience. Static Pricing provides moderately high satisfaction (~ 0.745), due to consistent and predictable pricing. Rule-Based Pricing starts lower (~ 0.72) because of early pricing or congestion issues, but improves gradually and converges close to the level of Static Pricing over time.
- ISP Revenue Comparison: Static Pricing yields the highest and most stable revenue ($\sim 19,050 RMB$) due to its fixed pricing structure. Rule-Based Pricing initially spikes above static levels due to early

price increases but declines slightly as congestion stabilizes, ending just below static. Q-Learning Pricing produces moderately lower but stable revenue ($\sim 18,800 RMB$), as it optimizes for long-term fairness and satisfaction rather than maximizing immediate profit. However, it should be noted that this ISP revenue does not take into account the costs of exceeding the target congestion rate. This is an aspect that future optimization would consider.
- Price Variation Comparison: Static Pricing remains completely flat at $\sim 11.6 RMB/GB$, as expected from a non-adaptive model. Rule-Based Pricing begins at a low price ($\sim 6RMB$) and steadily rises to above 12 RMB, reflecting gradual congestion control through predefined rules. Q-Learning Pricing fluctuates within a narrow range ($\sim$ 12.3–12.6RMB), indicating small, controlled adjustments as the agent fine-tunes prices to maintain performance across multiple objectives. In any case, the adjustment of prices is actually the essence of changes in congestion rates, which also motivates ISPs to set a reasonable price for the current network.

Although Static and Rule-Based pricing strategies offer ease of implementation, both strategy exhibit limitations in dynamic network conditions. The Static Pricing model relies on predefined parameters, such as user usage tiers and capacity ratios, to calculate a fixed price. While this may be adequate under normal operating conditions, it fails to respond to real-time fluctuations in demand. During peak periods or in scenarios with highly unbalanced user distributions, static pricing can lead to persistent congestion and user dissatisfaction due to its inability to adjust to network stress. The Rule-Based Pricing model addresses this issue partially by introducing reactive adjustments based on predefined congestion thresholds. However, these rules are hard-coded and do not evolve with the environment. For instance, in cases where the proportion of high-elasticity users increases significantly, the fixed thresholds may fail to prevent overload. Another problem with rule-based pricing strategies is that their adjustment speed is highly dependent on the factor η, setting a reasonable η value needs additional work. In contrast, the Q-learning approach addresses many of these shortcomings by continuously updating its policy through interaction with the environment. However, its complexity and computational cost must be considered when selecting an appropriate pricing strategy for specific deployment scenarios.

B. *Interface Implementation*

To facilitate interactive exploration of pricing strategies and their impact on network dynamics, we developed a web-based simulation dashboard. This tool allows users to simulate data usage behavior, pricing adjustments, and network congestion over a specified time horizon under configurable conditions.

In the project, we prioritized building a visual front-end for the rule-based pricing algorithm. The sidebar provides sliders for adjusting simulation inputs, these controls allow users to test the model's behavior in a range of real-world scenarios:

- Number of Users (10–500): Simulates traffic volume in the network.
- Network Capacity (100–1000 GB): Defines the available bandwidth.
- Initial Price per GB (1.0–20.0 RMB): Sets the starting data price.
- Target Congestion Level (0.1–1.0): Desired congestion ratio to maintain.
- Price Adjustment Factor (0.01–0.5): Determines how sensitively the price reacts to congestion.
- Simulation Times (6–48): Specifies the simulation duration in discrete "hourly" steps.

For each simulated time step, the following operations are performed:

1) User Data Demand: Each user's data usage is sampled from a normal distribution based on a base usage value reduced by a sensitivity factor times the current price. Usage is bounded below by zero.
2) Congestion Estimation: The total usage is summed and divided by network capacity to compute the congestion ratio.
3) Price Adjustment: The price is updated using the rule (same as Rule-based):
$$p_{\text{new}} = \max(0,\ p_{\text{current}} + \eta(c_c - c_t)),$$
where η is the adjustment factor, c_c is current congestion, and c_t is the target.
4) Revenue Calculation: Revenue is computed as the product of price and total usage.
5) User Satisfaction (Simpified version):
   - Basic satisfaction decreases with higher price and congestion using:
   $$\text{satisfaction} = \max(0,\ 1.5 - \text{congestion} - 0.05 \cdot \text{price})$$
   - Tweaked satisfaction penalizes excessive congestion (> 0.6) and high prices (> 8 RMB).

The dashboard provides two outputs:

- Formatted Data Table: A structured view of the time-series metrics including price, usage, congestion, revenue, and both satisfaction scores.
- Line Chart: Six time-series plots visualize the evolution of metrics across simulation hours.
- Additionally, users can export the results as a downloadable `.csv` file, enabling offline analysis.

V. Conclusion

This project presents a practical and adaptive model for managing network congestion in campus environments through dynamic pricing strategies. By categorizing users based on data sensitivity and employing three distinct pricing models—Static, Rule-Based, and Q-Learning—we demonstrate how price adjustments influence user behavior, optimize resource allocation, and improve overall network performance. Among the strategies evaluated, Q-Learning Pricing proved to be the most effective in achieving stable congestion levels and high user satisfaction, while also maintaining reasonable revenue through adaptive data-driven decision-making. The integration of a simulation dashboard further enhances the model's accessibility and transparency, offering a tool for testing ISP pricing policies in real time. Our approach highlights the value of machine learning and behavioral modeling in modern network management and sets the stage for future deployment in real-world academic and ISP-managed systems.

Acknowledgment

Xinyu LIANG was responsible for designing and implementing the simulation environment.

Tianyi JIANG was responsible for defining user types and modeling the data usage behaviors.

Xiaohui CAO was responsible for evaluation, visualization, and documentation.

Jiahao YANG was responsible for designing and implementing the pricing strategy.

All team members have made outstanding contributions, thanks to everyone in the team!

FAQ

A. Q&A 1

Q: The logic of Q-learning seems very similar to rule-based. Why is this approach still used? (Professor's Question during in-class Pre)

A: In fact, we can define more actions for the Q-learning action table (set different price adjustment ranges). Q-learning can quickly adjust prices to approach the target congestion rate in the early stage of training, and adjust prices with small fluctuations in the later stage. In contrast, the adjustment speed of rule-based pricing is very dependent on the factor η. We believe that Q-learning can perform better in extreme and initial situations. In addition, Q-learning can store learning results for real-time applications.

B. Q&A 2

Q: What are the application scenarios of your model? Can you give an example?

A: Our model is designed for a small network usage environment. In scenarios where network congestion occurs, the model can be dynamically invoked to optimize congestion management. For example, in a certain college, there is a group of similar network users every night, network congestion typically peaks around 10:00 PM, our model can be deployed at this time to adjust the price, thus reducing network congestion.

C. Q&A 3

Q: What is the difference between your pricing algorithm and the pricing algorithm discussed in Chapter 12 of the textbook?

A: The pricing algorithm described in the textbook has time factors that affect users' network usage (such as daytime and nighttime), and the pricing strategy alleviate congestion by attracting users to transfer data usage to other time periods. Our pricing algorithm focuses more on changes within a small time range, and does not consider the possible changes in user data usage caused by time changes.

References

[1] PlanHub, https://www.planhub.ca/, accessed Apr. 30, 2025.

[2] HK City Guide, https://www.hk-cityguide.com/, accessed Apr. 30, 2025.

[3] H. Yang, Y. Xu, and Q. Guo, "Dynamic Incentive Pricing on Charging Stations for Real-Time Congestion Management in Distribution Network: An Adaptive Model-Based Safe Deep Reinforcement Learning Method," IEEE Transactions on Sustainable Energy, vol. 15, no. 2, pp. 1100–1112, Apr. 2024. DOI: 10.1109/TSTE.2023.3327986

[4] X. Zhang, Y. Guo, J. Cao, and X. Liu, "A Q-Learning Based Pricing Strategy in EV Charging Station Network," IEEE Transactions on Industrial Informatics, vol. 15, no. 1, pp. 638–648, Jan. 2019. DOI: 10.1109/TII.2018.2832644

[5] J. Ahn and R. Doverspike, "Efficient Algorithms for Layered Network Optimization With Multiple Constraints," IEEE/ACM Transactions on Networking, vol. 11, no. 2, pp. 205–217, Apr. 2003. DOI: 10.1109/TNET.2003.810319

GitHub Resource

https://github.com/LayOneA/Pricing-Algorithm-for-ISP