SmartSharing: A CDN with Smart Contract-based Local OTT Sharing

Jiamin Fan*, Kui Wu*, Daming Liu+, Guoming Tang♯
* Department of Computer Science, University of Victoria, BC, Canada
+ Department of Computer Science, Shanghai University of Electric Power, Shanghai, China
♯ Peng Cheng Laboratory, Shenzhen, China

***Abstract*—A content delivery network (CDN) uses distributed cache servers to reduce the content delivery latency to end users. In recent years, CDN providers adopt a new content caching strategy that allows end users to share their storage/bandwidth resources. Two core questions need to answer in this strategy: (1) how to incentivize end users to contribute their resources? (2) how to facilitate transparent, secure content trading among end users?**

We propose a new CDN solution, called SmartSharing, where users contribute their over-the-top (OTT) devices as mini-cache servers. To incentivize end users to contribute resources, SmartSharing uses game theory and an Expectation-Maximization (EM) algorithm to determine the content delivery schedule and the pricing scheme. To facilitate content trading among end users, SmartSharing uses smart contracts in Ethereum to create a transparent and safe transaction platform. We thoroughly evaluate the performance of SmartSharing with real-world trace-driven simulation as well as a prototype using content metadata and the derived pricing scheme.

I. Introduction

CDNs extensively use cache servers to cache content to reduce the traffic delivery distance and latency from service providers to users.

The expansion of CDN footprint, however, may incur high equipment cost and heavy maintenance overhead for the CDN providers. The expansion costs eventually will be passed on to the users, leading to a higher price for CDN services.

The main idea is to *incentivize* and *facilitate* ordinary Internet users to contribute their OTT devices for content caching and their bandwidth for content delivery.

To *incentivize* end users, we design a centralized (virtual) reward mechanism.

To *facilitate* end users sharing their OTT storage and bandwidth resources, we need to support automatic trading without worrying about flawed transactions.

We aim at addressing the above two problems in a unified framework, called SmartSharing. To tackle the first problem, we propose a new incentive and pricing strategy. The main idea is to allow OTT owners to compete in a well-designed pricing game [1] and let them compensate each other. To tackle the second problem, we adopt smart contracts [2] in Ethereum [3] to support automatic, secure transactions among end users.

The contributions of the paper include:

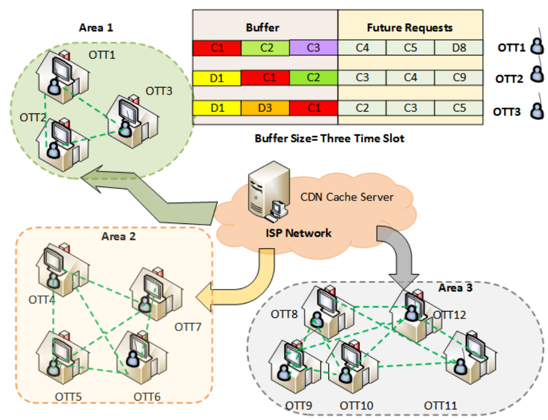


Fig. 1. Architecture of SmartSharing.

- We propose SmartSharing, a new CDN architecture where end users contribute their OTT devices as mini cache servers.
- We address two key technical challenges in SmartSharing: (a) incentivizing OTT owners to contribute and (b) facilitating content trading among OTT owners.

II. System Model and Problems

The architecture of SmartSharing is shown in Fig. 1. OTT devices in nearby neighborhood are grouped together, and OTT devices in the same group can serve each other. OTT devices have buffers to temporarily cache content, and can quickly deliver content to other OTT devices in the same group. As an example in Fig. 1, there are three OTT owners in Area 1. Assume that time is slotted and in the second time slot, the content in each OTT buffer is as follows: OTT1 (c_1, c_2), OTT2 (d_1, c_1), OTT3 (d_1, d_3), where c_i and d_i denote the indexes of content. Assume that the content requests at the next time slot are c_3 at OTT1, c_2 at OTT2, and c_1 at OTT3. We can see that for the next time slot, OTT1 can serve OTT2 and OTT3, and OTT1 and OTT2 can collectively serve OTT3 (e.g., OTT3 can fetch partial of c_1 from OTT1 and partial of c_1 from OTT2).

Annex to ISBN 978-3-903176-28-7© 2020 IFIP

III. Pricing in SmartSharing

The pricing scheme in SmartSharing consists of two steps: in the first step, we need to determine where an OTT device should download content for its owner, and in the second step, we determine the prices that a user should pay to other users who provide the services. Accordingly, we formulate and solve a bandwidth game for the first step and propose an expectation-maximization (EM) based pricing algorithm for the second step.

A. Banswidth game

Definition 1. ***Bandwidth game:***

- *Players: The set of all OTT devices, denoted by N_r.*
- *Strategies: OTT i's contribution $d_i(0 \leq d_i \leq 1)$ to SmartSharing.*
- *Payoffs: $r_i(d_i, d_{-i})$*

In order to find the Nash equilibrium, each OTT aims at maximizing its own payoff, by selecting its best strategy when the strategies of other OTTs are given. For two-player bandwidth game, a Nash equilibrium solution can be derived analytically. For multi-player bandwidth game, we propose an iterative algorithm called Best Response Update Algorithm (BRUA) 1, to find the solution.

Algorithm 1 Best Response Update Algorithm (BRUA)

Input: Initial contribution values $d_i^0(i \in N_r)$, constant k_i associated with OTT i, small threshold ϵ, the supply-demand relationship matrix $[\mathbf{Y}_{ij}]$

Output: all d_i^* values

1: t=0 and Flag=0.
2: **while** Flag==0 **do**
3: **Calculate** $d_i^{t+1} = T(d_{-i}^t)$ for all $i \in N_r$.
4: **if** $|d_i^{t+1} - d_i^t| \leq \epsilon$ for all $i \in N_r$ **then**
5: Flag=1.
6: t++.
7: Return $d_i^* = d_i^n$ for all $i \in N_r$.

B. EM algorithm

1) E-step: The solution of E-step denotes the proportion of OTT devices that would accept each price.

2) M-step: The solution of M-step denotes the probability of each price been accepted. The solution is denoted as $\boldsymbol{\theta}_{(t+1)} = (\theta_{z(1),(t+1)}, \theta_{z(2),(t+1)}, ..., \theta_{z(K),(t+1)})$, where

$$\theta_{z(l),(t+1)} = \frac{\sum_{i=1}^{n} (Q_{i,z(l),(t)} \sum_{j=1}^{n} x_{ij})}{n \sum_{i=1}^{n} Q_{i,z(l),(t)}}. \quad (1)$$

The $\boldsymbol{\theta}_{(t+1)}$ values will be used in the next-round E step. We iterate the E step and the M step until convergence. The pricing strategy for each OTT is obtained from the convergence result of the last-round E step.

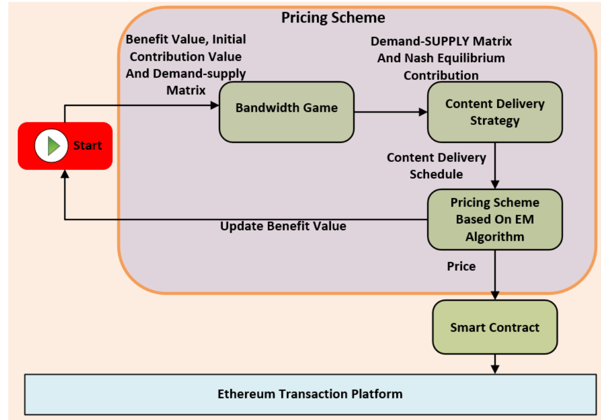


Fig. 2. Setting prices in smart contracts

IV. Evaluation of SmartSharing

We evaluate SmartSharing with prototyping and trace-driven simulation. Together, trace-driven simulation and smart contract-based implementation show a comprehensive picture on the performance of SmartSharing. Fig. 2 shows how prices are set in smart contracts.

We build a web interface to help OTT owners manage their Ethereum wallet and view content trading metadata. We also built a GUI. More implementation detail about the backend and frontend of SmartSharing could be found at [4].

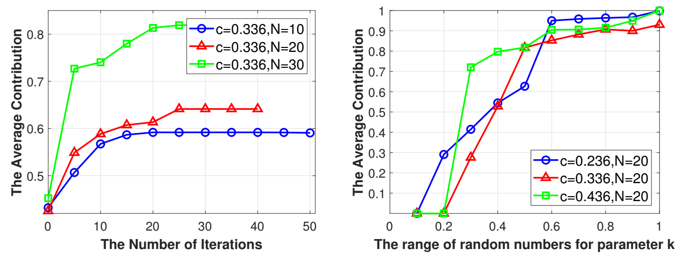
We generate synthetic trace data based on the data set from MMSyS2015[1], which includes thousands of live streaming sessions of two major service providers: Twitch.tv and YouTube Live. From the data, we identify the most popular 50 videos based on number of views. Since each video includes log data regarding the number of viewers over time but the information regarding individual viewers is mostly missing (except for those who wrote comments), we simulate the behavior of each viewer in the following steps: (1) we divide time into time slots of duration 5 seconds; (2) at each time slot, each viewer would view content with probability proportional to the popularity of the content; and (3) based on the above content viewing procedure, an OTT can share content to other OTT devices if the content is already in its buffer. We set the buffer size at each OTT to hold content for 4 time slots. We simulated different number of viewers (i.e., 10 and 20).

Using the synthetic traffic trace, we illustrate how SmartSharing converges to equilibrium in the bandwidth game (Section) and how it reaches the desired pricing scheme (Section IV-A).

A. Pricing Scheme

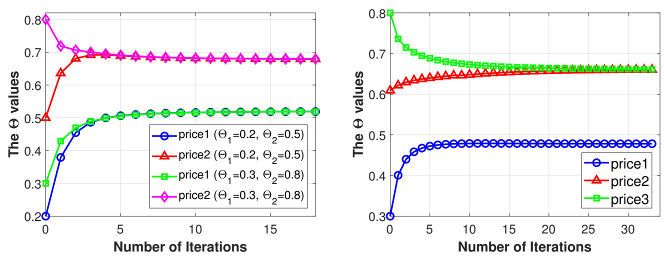
This section discloses the detailed intermediate steps on how OTT devices reach the pricing scheme with our

[1]Available from http://dash.ipv6.enstb.fr/dataset/live-sessions/.



(a) (b)

Fig. 3. Bandwidth game: (a) Average contribution against number of iterations to reach Nash equilibrium for 10, 20 and 30 devices. The average initial value of contribution per OTT is 0.432; parameter k_i is randomly chosen in $(0, 0.4)$. (b) Average contribution at Nash equilibrium against parameter k_i under different c for 20 OTT devices.

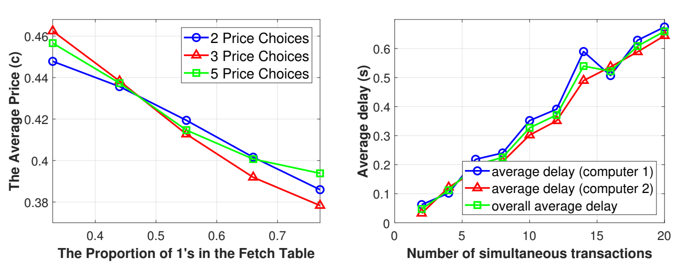


(a) (b)

Fig. 4. The convergence of $\boldsymbol{\theta}$ values. (a) The initial probabilities that an OTT would fetch content from other OTT devices with price 1 and price 2 are set to 0.2 and 0.5 (the two blue curves), respectively, or to 0.3 and 0.8 (the two red curves), respectively. (b) The initial probabilities that an OTT would fetch content from other OTT devices with price 1, price 2 and price 3 are set to 0.3, 0.6 and 0.8, respectively.

EM algorithm. In this experiment, we show the pricing schemes of 10 OTT devices. Fig. 4 (a) shows how $\boldsymbol{\theta}$ converges to stable values in our EM algorithm.

We are interested in how the number of price options and the amount of content that an OTT shares with others impact the final price schemes. For this, we tested three cases: (1) two price options $\{0.3c, 0.8c\}$, (2) three price options $\{0.3c, 0.6c, 0.8c\}$, and (3) five price options $\{0.3c, 0.5c, 0.6c, 0.7c, 0.8c\}$.



(a) (b)

Fig. 5. (a) Average price vs. the proportion of 1's in the fetch table (2 price choices: $\{0.8c, 0.3c\}$, 3 price choices: $\{0.8c, 0.6c, 0.3c\}$, 5 price choices: $\{0.8c, 0.7c, 0.6c, 0.5c, 0.3c\}$). (b) Average transaction delay (seconds) vs the number of simultaneous transactions.

Fig. 5(a) shows the average price versus the proportion of 1's in an OTT's fetch table, with different numbers of price options. We can see that the number of price options offered to an OTT does not have a clear impact on the average price charged by the OTT. Nevertheless, the average price drops with the increase of content amount that an OTT can share with other OTT devices, since all the three curves show a clear down trend along the x-axis. This is reasonable because when an OTT shares more content with others, it can reduce the average price for the same level of profit.

B. *Overhead of Smart Contracts*

We emulate the real-world transactions by deploying virtual machines (VMs) over two desktop computers connected by a local 10 Mbps Ethernet router. The two machines have same system configuration: Intel Core i7-7700 quad core 3.6 GHz CPU, 8 GB 2400 MHz DDR4 memory, and Window 10 Enterprise OS. Each VM represents an OTT and runs smart contracts in SmartSharing. We test different total numbers of OTT devices, and for each test we deploy the same number of OTT devices in the two computers. We test the *worst-case scenarios* where transactions occur in burst, i.e., we start a test by configuring several OTT devices to submit content transaction simultaneously and end the test when all transactions have been executed. For each transaction, we record its delay calculated by its finish time minus its start time.

Fig. 5(b) shows the average transaction delay vs the number of simultaneous transactions, including the average delay for OTT devices in each computer and the average delay for all OTT devices. The average gas fee per transaction is 0.00005 ether, which amounts to 0.03 USD according to the market price.

The above results demonstrate that the overhead in terms of delay and monetary cost for executing smart contracts in SmartSharing is very small.

V. Conclusion

This paper presents a new CDN solution, SmartSharing, which integrates a pricing scheme to incentivize end users and a smart contract-based content trading mechanism to facilitate content trading among end users.

Overall, SmartSharing brings a win-win-win solution for CDN providers, end users, and content providers (CPs).

References

[1] Q. Ma, L. Gao, Y.-F. Liu, and J. Huang, "Economic analysis of crowdsourced wireless community networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1856–1869, 2017.

[2] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *IEEE Symposium on on Security and Privacy (SP)*, 2016, pp. 839–858.

[3] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[4] J. Fan, K. Wu, D. Liu, and G. Tang, "Smartsharing: A cdn with smart contract-based local ott sharing [tech-report]," 2019. [Online]. Available: https://www.dropbox.com/s/260x7djaa7kedey/technical%20report.pdf?dl=0