Market-driven Bandwidth Allocation in Selfish Overlay Networks

Weihong Wang, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
{*wwang,bli*}*@eecg.toronto.edu*

***Abstract*— Selfish overlay networks consist of autonomous nodes that develop their own strategies by optimizing towards their local objectives and self-interests, rather than following prescribed protocols. It is thus important to regulate the behavior of selfish nodes, so that system-wide properties are optimized. In this paper, we investigate the problem of bandwidth allocation in overlay networks, and propose to use a market-driven approach to regulate the behavior of selfish nodes that either provide or consume services. In such markets, consumers of services select the best service providers, taking into account both the performance and the price of the service. On the other hand, service providers are encouraged to strategically decide their respective prices in a pricing game, in order to maximize their economic revenues and minimize losses in the long run. In order to overcome the limitations of previous models towards similar objectives, we design a decentralized algorithm that uses *reinforcement learning* to help selfish nodes to incrementally adapt to the local market, and to make optimized strategic decisions based on past experiences. We have simulated our proposed algorithm in randomly generated overlay networks, and have shown that the behavior of selfish nodes converges to their optimal strategies, and resource allocations in the entire overlay are near-optimal, and efficiently adapts to the dynamics of overlay networks.**

I. Introduction

When overlay nodes are inherently *selfish*, applications in overlay networks may not perform optimally, since selfish nodes tend to optimize towards their self-interests. For example, they may attempt to maximally exploit services from other nodes, while not willing to provide services to others. Their strategies and behavior are not easily regulated by prescribed distributed algorithms, if their self-interests are not considered.

Naturally, it is important to regulate the behavior of such selfish nodes, and even steer such behavior towards the *common good*, where system-wide properties are optimized, rather than the original local self-interests. We investigate the problem of *bandwidth allocation* in overlay networks, involving applications with long-lived and bandwidth demanding peer-to-peer data transmissions. We wish to manipulate the self-interests of overlay nodes by placing all participating nodes in a *market*, where service provisioning becomes preferable even for selfish nodes.

Let us consider the relationship between the nodes that provide services and the nodes that consume them. Each overlay node that consumes services, hereafter referred to as a *downstream node*, has the choice of using the service from one of multiple nodes that have the capability of providing it (henceforth referred to as *upstream* nodes). On the other hand, each upstream node may potentially serve multiple downstream nodes. In the bandwidth allocation problem, we can simply envision that the data source in the peer-to-peer data transfer application provides a "service" to the receiver, who benefits from receiving such data.

Two critical questions arise from this context. First, if we establish a directed overlay link (that symbolically represents the service provisioning relationship) between a successfully matched upstream and downstream node, which overlay links should we include in our service provisioning network, that connects all the upstream nodes that provide services to all the downstream nodes that consume them? Second, once these links are established, how much bandwidth should be assigned to each overlay link in order to satisfy the traffic demands of as many downstream nodes as possible? The formation of this problem is rather generic, and may find its root in various overlay application scenarios such as overlay multimedia streaming and parallel downloading of bulky data.

By placing all participating (upstream and downstream) nodes in a *market*, we can leverage the concept of *prices of providing services* to regulate the behavior of selfish nodes in contributing and consuming resources required for such services. In our problem of bandwidth allocation, such resource is the network bandwidth. A downstream node simply pays a price to an upstream node for every unit of bandwidth the data transmission service consumes.

Our market model is fundamentally different from most *single pricing* or *static pricing* models that have been previously studied in the context of overlay networks. In previous models, either a single centralized price is used in the entire system, or per-service prices are established, but remain static throughout the lifetime of the nodes. In our market mechanism, *each* upstream node has its own specific service price it prefers to charge its downstream nodes, and such a price is dynamically adjusted over time in order to maximize its economic revenue and minimize its empirical loss (due to the occupation of its bandwidth by downstream nodes) in the long run. Such a market mechanism is more flexible and realistic, as there does not exist centralized authorities to determine a single centralized price in overlay networks.

The market mechanism can be understood from two different perspectives. First, from the perspective of the downstream nodes as service consumers, they need the freedom

to select the best upstream nodes that not only deliver the best performance, but also incur the minimum economic costs. Second, from the perspective of the upstream nodes in the market, they compete in a *pricing game* in which they need to strategically decide their service prices, since their future revenues and potential losses are determined by the prices set by all players in the game. Such a pricing game, unfortunately, is rather complex in reality: it is a game with incomplete information and imperfect recall, which usually requires the nodes' supplementary knowledge on probability distributions of missing information in order to be solvable by classical game theory. In this paper, we provide practical solutions for strategic nodes to gradually solve the pricing game, by modeling them as *reinforcement learning agents* that are capable of incrementally improving their strategies through trial-and-error interactions with the external world. At equilibrium, nodes are expected to reach strategies that optimally adjust their prices.

In more general scenarios where all overlay nodes may potentially assume the dual roles of being both upstream and downstream, the proposed market mechanism solves the general problems of downstream/upstream matching and bandwidth assignment, both in a fully distributed manner. In this paper, we study how well the effects of such a market mechanism approximate the optimal system-wide properties that can be achieved in overlay networks. In particular, given an overlay network, the distribution of data items, and the demands from downstreams, we evaluate the *optimality* of a specific bandwidth allocation with two metrics: (1) the percentage of transmission requests accepted by the network; and (2) the total end-to-end throughput in the resulting topology.

The remainder of the paper is organized as follows. We first discuss related work in Sec. II. The market model is formulated in Sec. III. Sec. IV defines the pricing game, and discusses our distributed solution based on reinforcement learning algorithms. Bandwidth allocation decisions to be made by upstream and downstream nodes on the market are discussed in Sec. V. Sec. VI evaluates the performance of the proposed mechanism through simulation results. Finally, Sec. VII concludes the paper.

II. Related work

Node selfishness and incentive provisioning in autonomous networks have been extensively researched, with the current literature showing several distinct while related research trends, which differ in their interpretations of self-interest, and the assumptions related to applications.

First, networked selfish nodes have been modeled as *strategic players* from a game theoretic perspective, where the self-interest of a node is studied by considering the empirical benefits of consuming or the losses for contributing resources [1], [2], [3]. A selfish node wishes to maximize its overall benefit while taking into account the negative impact from the behavior of others. For example, in the routing game discussed by Roughgarden and Tardos [4], a selfish node constantly seeks to reduce its perceived latency by routing traffic through shorter paths, while the increased traffic increases the latency of every flow going through the shared links. Unfortunately, the use of classical game theory requires strong assumptions, for instance, the exact information about the entire game — including private information of other players — is assumed to be known to each selfish player. Due to the infeasibility of making such assumptions in overlay networks, it is not possible to design practical solutions using classical game theory for each individual player to actually play the game. The objectives of previous work have been to investigate whether a specifically proposed game leads to the preferable equilibrium point, and the equilibria are usually directly computed using linear or nonlinear programming [4], [5].

Second, if we assume the existence of a *service charge* or *reward*, a selfish node may be concerned with both the empirical benefit or loss and the *economic revenue or cost*. When the service charge and reward are decided by a central authority [6], [7], [8], [9], a selfish node just needs to decide the amount of its contribution or usage of resources, and may not be aware of the behavior of others. If we assume that the central authority makes strategic decisions on prices, the interactions between one player and the other players lead to a *Stackelberg game* [10]. Although the existence of any central authorities can not be conveniently assumed in overlay networks, we still believe that it is a promising direction to further explore decentralized algorithms of settling charges and rewards, and to study the interactions between the two sides that charge and pay.

Finally, some recent work has introduced the theory of *mechanism design* [11] to the study of autonomous networks. The main focus is to exploit the strength of *strategyproof* mechanisms, which enforce selfish entities to truthfully reveal their private information by offering calculated payments, in order to derive the optimal solution to a system-wide problem. Initially, various *second-price* auctions have been extensively studied. For example, the progressive second price auction mechanism proposed by Lazar *et al.* and Semret *et al.* [12], [13] was used to differentiate QoS in bandwidth sharing problems, and the Spawn system [14] manages idle CPU times through distributed bidding. More recent research has focused on more complicated algorithms such as the VCG mechanism, and has emphasized *distributed algorithmic mechanism designs* [15]. For example, Feigenbaum *et al.* have investigated cost sharing mechanisms for multicast transmissions based on the marginal cost and the *Shapley* value [16], and have designed a distributed mechanism that computes VCG payments for intra-domain routing using the BGP model [17]. Though strategyproof mechanisms have been extensively studied, most existing approaches assume that a central entity has unlimited amount of incentives to be offered to the system in order to guarantee strategyproofness (as illustrated in the budget imbalance problem of VCG), which is not realistic in overlay applications. In this paper, we seek to design a fully distributed market-based mechanism, which still provides *incentives* for upstream nodes to provide services, without the requirements of a central authority to offer payments at its

cost.

III. Problem formulation

We consider the most generic abstraction of *one-hop flows* in overlay networks, each of which corresponds to a long-lived end-to-end data transmission session between a pair of overlay nodes. We believe that such an abstraction can be made in overlay networks without being unrealistic: most peer-to-peer applications involve one-hop unicast flows between a data source and a receiver (downloader). For other types of overlay communication sessions such as overlay multicast, each edge in the corresponding topology (single tree, multiple trees, or mesh) corresponds to a one-hop flow. We study bandwidth allocation problems with respect to one-hop flows, without assuming a specific type of overlay applications.

In our study, we assume that each overlay node is capable of measuring performance metrics regarding overlay links between itself and other overlay nodes. With respect to one-hop flows, we assume that nodes are only concerned with session throughput, and the *available bandwidth* $B_j^i(t)$ from node i to node j may be measured through bandwidth estimation algorithms at any given time. We assume that essential overlay services such as service discovery exist in the overlay network, so that each downstream node is able to identify a set of upstream candidates that are able to provide the requested data before interacting with them on the market. Finally, we assume a secure payment mechanism among peers is in place, which is complementary to this study and has been the focus of some of the existing research work [8], [18], [19].

A. Market model

Our market model is established based on the notions of downstream and upstream nodes of one-hop flows, where the downstream node may be interpreted as the buyer and consumer of the data service, and the upstream node as the seller and provider. As a potential upstream node of an one-hop flow to be established, each overlay node i maintains a *transmission price* $p_i(t)$ for time slot t, which is to be charged to any of its one-hop downstream nodes, for *each unit* of bandwidth they consume in that time slot. $p_i(t)$ may be adjusted by node i over time, for the purpose of maximizing its utility based on its accumulated experience.

Each downstream node aims to achieve the highest benefit from the one-hop flows it receives, and minimize the payments made to the respective upstream nodes. Therefore, it *selects* upstream nodes based on their prices, as well as the maximum possible session throughput from each of them. Each downstream node determines the actual session throughput — or the amount of bandwidth to be purchased per unit time — by maximizing its own utility function. Since traffic loads at both sides and within the underlying network may change over time, downstream nodes have the freedom to switch to better upstream nodes, since it wishes to always enjoy the best performance at the minimum cost.

When establishing one-hop flows, we assume that upstream nodes accept any downstream nodes as long as the resulting one-hop flows improve their utilities. In other words, for requests that come in sequentially, an upstream node simply processes them on a *first come first serve* basis, without skipping or waiting for "better" requests to come.

B. Utility function

In our market mechanism, the consistent objective of any selfish node is to maximize its self-interest for every time slot that it participates in overlay data transmissions. Mathematically, we may characterize a node's self-interest using a *utility function*, which includes the *empirical benefits* and *losses* for consuming and contributing bandwidth resources, and the *economic revenues* and *costs* incurred in trading the resource.

Since an overlay node usually assumes the dual roles of both downstream and upstream in the overlay, its utility function includes the utility in both roles. For the time slot t, suppose that node i is currently receiving flows from a set $U_i(t)$ of upstream nodes, each at a rate of $b_i^j(t)$ and a unit charge of $p_j(t)$, $j \in U_i(t)$; it also delivers flows to a set $D_i(t)$ of downstream nodes, each at rate $b_k^i(t)$. If the local bandwidth capacity at node i is C_i, the utility of node i participating in overlay data transmissions can be expressed as:

$$\begin{aligned} u_i(t) = \quad & \epsilon_1 \log\left(1 + \frac{\sum_{j\in U_i(t)} b_i^j(t)}{C_i}\right) \\ + \quad & \epsilon_2 \log\left(1 - \frac{\sum_{k\in D_i(t)} b_k^i(t)}{C_i}\right) \\ - \quad & \sum_{j\in U_i(t)} p_j(t) b_i^j(t) + p_i(t) \sum_{k\in D_i(t)} b_k^i(t) \end{aligned} \tag{1}$$

The first two terms represent node i's empirical benefit of receiving flows, and the empirical loss for delivering flows. The third and fourth term represent the economic cost and revenue in the market. As is evident from the $\log(\cdot)$ function, the empirical benefit $\epsilon_1 \log\left(1 + \frac{\sum_{j\in U_i(t)} b_i^j(t)}{C_i}\right)$ increases quickly from zero as the total receiving throughput increases from zero, then increases more slowly. This reflects the intuition that the initial increase in receiving throughput is more important to a node. On the contrary, the empirical loss $\epsilon_2 \left|\log\left(1 - \frac{\sum_{k\in D_i(t)} b_k^i(t)}{C_i}\right)\right|$ increases relatively slowly from zero at the beginning but rapidly later, which reflects the natural judgement of a selfish node that becomes increasingly reluctant to sell bandwidth when its available capacity is decreasing. The $\log(\cdot)$ function is also analytically convenient, since it is increasing, strictly concave and continuously differentiable. The coefficients ϵ_1 and ϵ_2 in Eq. (1) are positive parameters that indicate the relative importance of empirical benefit and loss in comparison with economic factors. They also keep the four terms on the same order of magnitude. For ease of illustration in our subsequent studies, we assume that

all nodes use the same form of utility functions, but they may have different parameters that are only privately known.

C. Decision problems

Nodes have different decisions to make as they appear on the market as downstream and upstream nodes. As a downstream node, since the transmission prices of its upstream candidates are given, the decision problem of node i is to select the best upstream node and the optimal receiving throughput, so that it receives the highest positive utility from the transmission, given the constraints of available bandwidth between itself and the selected upstream node. As an upstream node, node i faces two kinds of decisions. First, for any downstream node that requests for service, node i decides the range of its acceptable outgoing bandwidth, beyond which its utility is going to decrease. Second, node i strategically decides the transmission price it charges in order to maximize its utility in each upcoming time slot.

Why should node i dynamically decide and adjust its price as an upstream node? Due to the nature of the market model, whether or not node i will be selected by a downstream node depends not only on its transmission price and the performance of the overlay link between the two, but also on the transmission prices set by other upstream candidates and performance of their respective overlay links. Therefore, if node i's transmission price is too high, few downstream nodes may wish to receive flows from it; if it is too low, too much bandwidth may be consumed during the time slot, so that some downstream nodes may decide to switch to other upstream nodes or to reduce their receiving throughput, due to the degraded performance.

IV. The Pricing Game

We divide our discussions on the behavior of selfish nodes in the market setting into two parts. In this section, we discuss the decision problem at upstream nodes with respect to their transmission prices. In the subsequent section, we proceed to the problem of making upstream selections at downstream nodes, when transmission prices are available.

A. Game formulation and its properties

Since all selfish nodes involved in overlay data transmissions need to decide their transmission prices strategically, they form the player set I, and the pricing game can be formed as follows. Each node i has a set of actions $A_i = \{a_i\}$ to be chosen under various situations, and a strategy set $S_i = \{f_i\}$ containing all the possible mappings from *distinguishable information sets* $\{H_i\}$ perceived by node i to node i's actions, *i.e.*, $f_i : H_i \rightarrow a_i$. Asynchronously in time, nodes sequentially take their optimal actions, following their optimal strategies that maximize their utilities as expressed in Eq. (1).

To reduce the complexity of the game, we assume transmission prices to be *integer-valued*, and interpret node i's actions as the possible incremental changes to be made to the prices: $A_i = \{-1, 0, 1\}$. For such a finite game — one with finite number of players and finite action sets for each player, the classical game theory has proved that it has at least one mixed strategy Nash equilibrium [20].

However, further reflections show that classical game theory does not provide any practical solutions to such a pricing game. First, it is very hard for an arbitrary node to identify the player set I, *e.g.*, how many players there are in the game, or which nodes they are, due to the lack of global information. Second, even if the set I is identified, a node still has *incomplete information* about the game itself and about other players. Knowing only its own utility, a node has no exact knowledge of how a bandwidth allocation outcome is reached, or how other players' strategies impact its utility. Third, the game cannot be treated as a Bayesian game either, since a node does not have the knowledge of the concrete form of other nodes' utility functions, as well as any probability distributions of them. Finally, it is infeasible for a node to observe the previous actions and states of all other players, and thus to perform backward induction.

Summarizing these difficulties, we may recognize the pricing game as a *dynamic sequential move game* with *incomplete information* and *imperfect recall*, which provides insufficient knowledge for nodes to really derive their equilibrium strategies. However, since nodes are still capable of *observing* their own actions and utilities, as well as relevant system outcomes in the history, we may design an appropriate solution for nodes to gradually *learn* the optimal strategies through past experience. Being iterative and incremental, the well-studied *reinforcement learning* algorithms have become our choice that our proposed solution is based upon.

B. Reinforcement learning

Reinforcement learning (RL) is a branch of machine learning that enables a decision maker, or an *agent* having a set A of alternative actions, to involve an optimal *decision policy* through systematic *trial-and-error* interactions with the external *environment*, which is characterized by a set S of states.

A decision policy is defined as a mapping from each environment state to a probability distribution over the agent's actions, *i.e.*, $\pi : (s, a) \rightarrow \pi(s, a)$, where $\pi(s, a)$ is the probability of taking action a at state s. The agent incrementally improves its decision policy towards an optimal one, based on feedback provided by the environment, known as *reinforcement* r. An optimal decision policy is to incur the highest accumulated reinforcement values. The most familiar example of RL is the training of a chess player: a chess player gradually learns the best moves at different positions, by repeatedly taking his moves, and receiving rewards (*e.g.*, $r >= 0$) or penalties (*e.g.*, $r < 0$) for its moves from the trainer.

In the discrete-time domain, RL models the interaction between an agent and the environment as a *Markov decision process* [21] (Fig. 1). Suppose the environment is at state $s(t)$ in time slot t, after the agent performs action $a(t)$, it shifts to state $s(t+1)$ in the next slot with probability $P^a_{ss'} = P\{s(t+1) = s' | s(t) = s, a(t) = a\}$. The agent then

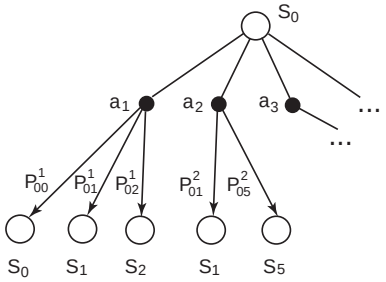

Fig. 1. An example of the Markov decision process. Empty circles represent system states, and solid dots represent the corresponding actions, respectively. For each state of the system, there exist multiple choices of feasible actions, taking each one of them may lead the system to various states probabilistically.

receives a reinforcement $r(t+1) \in \mathcal{R}$, the expected value of which may be expressed as $R^a_{ss'} = E_{\mathcal{R}}\{r(t+1)|s(t+1) = s', s(t) = s, a(t) = a\}$. The interaction between an agent and its environment may be illustrated by Fig. 2.

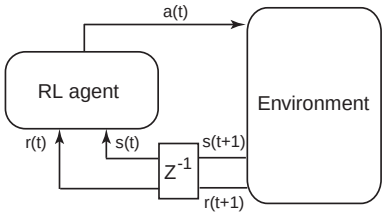

Fig. 2. The interaction interface between an agent and the environment. The Z^{-1} unit represents a delay of one time slot.

A decision policy is incrementally improved as the agent *tries* by choosing an optimal action following the current policy, and then *makes corrections* by adjusting the policy based on the most recent observation $\langle s(t-1), a(t-1), s(t), r(t)\rangle$.

For improved efficiency, we adopt the *Q-learning* method that improves decision policies with the aid of *Q-value functions*, $Q : (s, a) \to Q(s, a), s \in S, a \in A$, where $Q(s, a)$ is the Q-value associated with the state-action pair (s, a), and represents the *expected return* when taking action a in state s and then following the current policy to the end. The expected return is expressed as follows:

$$\sum_{k=1}^{\infty} E\{\gamma^k r(t+k)\}$$

where $\gamma \in [0, 1)$ is a *discounting factor* that discriminates the impact of reinforcements that are farther away. The standard updating rule for Q-learning is given as:

$$\begin{cases} Q(s(t-1), a(t-1)) \leftarrow Q(s(t-1), a(t-1)) + \Delta \\ \Delta \leftarrow \beta\left[r(t) + \gamma \max_a Q(s(t), a) - Q(s(t-1), a(t-1))\right] \end{cases} \quad (2)$$

where β indicates the *learning rate*. The determination of a decision policy starts with the iteration of a Q-value function; the latter is considered to have converged to the optimal when its value for each state-action pair is no smaller than that of any other value functions.

However, Eq. (2) only updates the Q-value for one state-action pair after each iteration with the environment, which might be very slow, especially for systems in which performing real interactions with the environment is expensive. To greatly improve the convergence speed, this paper adopts the *Dyna-Q architecture* [22], which, after each real interaction, performs one iteration on the Q-value of previous state-action pair, then iterates on k hypothetical interactions simulated by the learned system model, *i.e.*, $\{R^a_{ss'}\}$ and $\{P^a_{ss'}\}$. The updating rule for *Dyna-Q* based on hypothetical interactions is as follows:

$$\begin{cases} Q(s, a) \leftarrow Q(s, a) + \Delta \\ \Delta \leftarrow \beta\left\{\sum_{s'}\left[\hat{R}^a_{ss'} + \gamma \max_{a'} Q(s', a')\right]\hat{P}^a_{ss'} - Q(s, a)\right\} \end{cases} \quad (3)$$

where $\hat{R}^a_{ss'}$ and $\hat{P}^a_{ss'}$ are the estimates of expected reinforcement value $R^a_{ss'}$ and state transition probability $P^a_{ss'}$, based on real interactions.

The details of the algorithm are given in Table I. Note that the algorithm loops infinitely, because we are especially interested in cases where the dynamics of the external environment possesses unpredictable time-varying characteristics that need to be learned on line. The convergence is faster when k is higher.

TABLE I

THE Q-LEARNING ALGORITHM: DYNA

For all $s \in S$ and $a \in A(s)$, $Q(s, a) \leftarrow 0$
while (true)
  Increment the time of transitions $s(t-1) \overset{a(t-1)}{\to} s(t)$
  Update $\hat{P}^a_{ss'}$
  Record the latest reinforcement value $r(t-1)$
  Update $\hat{R}^a_{ss'}$
  Update $Q(s(t-1), a(t-1))$ by Eq. (2)

  Randomly choose k state-action pairs $\{\bar{s}, \bar{a}\}$
  for all $(\bar{s}, \bar{a})$
    Update $Q(\bar{s}, \bar{a})$ by Eq. (3)

Given all the Q-values of state-action pairs associated with the current state s, the probability of taking action a follows the *Boltzmann distribution* given by:

$$P(a|s) = \frac{e^{\alpha Q(s,a)}}{\sum_{a'} e^{\alpha Q(s,a')}} \quad (4)$$

where α is a positive constant that controls the "sharpness" of differentiating actions corresponding to different Q-values.

C. Playing by RL

We wish to leverage reinforcement learning methods to solve the decision making problem for the pricing game, based on the following justifications.

– First, each node is presented with an external environment composed of all other competitive upstream nodes as well as downstream nodes. Each node i may locally observe its own residual bandwidth $b_i(t)$, which is the original capacity C_i subtracted by the total bandwidth being consumed by all ongoing flows, as the state of the

environment. By adjusting the transmission price $p_i(t)$, node i receives the corresponding utility (characterized by the second $\log(\cdot)$ term and the economic revenue term in Eq. (1)) as feedback. All upstream nodes seek to maximize the sum of the feedback in the long run.

– Second, it is not feasible to obtain information about the game itself, including (1) utility functions of other players; (2) prices and performance metrics observed by downstream nodes; (3) preferences of downstream nodes, and (4) distribution of service requests. It is more practical to model the interaction between the node and the remainder of the game as a Markov process.
– Third, as the Q-function converges, the agent will form a decision policy that maps an environment state s to a probability distribution $P(a|s)$ over all possible actions (Eq. (4)), which exactly corresponds to the spirit of a mixed strategy equilibrium in our game.
– Finally, the *Dyna-Q* learning algorithm is very amenable to incremental implementations.

Formally, we define the RL-based solution for the pricing game players as follows. A node i, when acting as an upstream node, is represented by a RL agent that locally observes the environment state $b_i(t)$ at the end of time slot t, and maintains the *integer-valued* transmission price $p_i(t+1) \leftarrow p_i(t) + \Delta p$, by choosing actions from the action space $\Delta p = \{-1, 0, 1\}$. When its new price is exposed to the environment in time slot $(t+1)$, the agent receives a reinforcement of

$$\begin{aligned} r(t+1) = \quad & p_i(t+1) \sum_{k \in D_i(t+1)} b_k^i(t+1) \\ + \quad & \epsilon_2 \log \left(1 - \frac{\sum_{k \in D_i(t+1)} b_k^i(t+1)}{C_i} \right) \end{aligned} \tag{5}$$

by the end of that slot. The objective of the agent is to obtain an optimal decision policy $\{P(\Delta p|b_i(t))\}$, so that $\sum_{k=1}^{\infty} E\{\gamma^k r(t+k)\}$ is maximized at any time t.

Before deploying the *Dyna-Q* algorithm, there are a few outstanding problems that we need to address.

1) Dividing the state space: The residual bandwidth of a node is essentially a continuous variable, hence the state space potentially contains an infinite number of states. Solutions to this problem involve either *dividing* continuous states into sections, or *generalizing* Q-functions to the continuous domain. Since generalization methods usually require a neural network to approximate the continuous function, while discrete solutions only require table-based mapping, we choose to divide the state space to reduce the complexity. In our mechanism, since the residual bandwidth at node i is bounded by $[0, C_i]$, it may be simply divided into m_b equal *sections*, though equal division is not required for the RL algorithms to work successfully.

2) Filtering unnecessary state transitions: On observing $b_i(t)$ from the environment, agent i needs to decide which state the environment currently resides in. To avoid mistakenly believing environment states are frequently in transition when observed values vary around section borders, we introduce a *transition threshold* ϵ that is 0.15 times of the average size of a section to filter unnecessary transitions. Thus, a state s may still belong to a section $[a, b)$, even if $s < a$ but $s \geq (a - \epsilon)$, or if $s > b$ but $s \leq (b + \epsilon)$.

3) Convergence of learning: Conventional reinforcement learning models require that the state transition of the environment, as well as the distribution of reinforcement values, be *stationary*, meaning that $\{P_{ss'}^a\}$ and $\{R_{ss'}^a\}$ do not change over time. Overlay networks do not satisfy this requirement, since both quantities are affected by unpredictable network dynamics and variations of transmission requests. However, as long as $\{P_{ss'}^a\}$ and $\{R_{ss'}^a\}$ are only varying mildly and gradually, RL algorithms are still effective solutions to optimal decision making problems, where we have difficulty to mathematically characterize the dynamics of the environment [23].

V. Trading on the market

As described by the market model in Sec. III, once transmission prices are determined, the downstream node needs to make decisions to select the best upstream node and the session throughput of the one-hop flow. Such decisions are made based on the respective utilities brought by the upstream candidates.

A. Evaluating utilities

For each upstream candidate, a downstream node i may envision its benefit or loss when receiving from node j at rate $b_i^j(t)$ based on the utility function Eq. (1). Two design alternatives are possible: node i may either evaluate its *total utility* that considers all current incoming flows (including the new flow) using Eq. (6), or consider the *additional utility* brought by the new flow using Eq. (7).

$$\begin{aligned} u_{i,D}^j(t) = \quad & \epsilon_1 \log \left(1 + \frac{b_i^j(t) + \sum_{j' \in U_i(t)} b_i^{j'}(t)}{C_i} \right) \\ - \quad & p_j(t) b_i^j(t) - \textstyle\sum_{j' \in U_i(t)} p_{j'}(t) b_i^{j'}(t) \end{aligned} \tag{6}$$

$$\begin{aligned} u_{i,D}^j(t) = \quad & \epsilon_1 \log \left(1 + \frac{b_i^j(t) + \sum_{j' \in U_i(t)} b_i^{j'}(t)}{C_i} \right) \\ - \quad & \epsilon_1 \log \left(1 + \frac{\sum_{j' \in U_i(t)} b_i^{j'}(t)}{C_i} \right) - p_j(t) b_i^j(t) \end{aligned} \tag{7}$$

The subscript D and superscript j in $u_{i,D}^j(t)$ indicate that the value is computed by node i as a downstream node of node j. In subsequent discussions, we assume all nodes use Eq. (7), as it requires less computation. Our market model, however, does not favor one over the other, and leave it as an option to applications or different selfish nodes.

B. Determining optimal transmission throughput

In addition to evaluating each upstream candidate in terms of its induced utility, a downstream node i also computes the most preferable receiving throughput from the candidate, by maximizing Eq. (7):

$$\begin{aligned} r_{i,D}^{j}(t) = \quad & \arg\max_{b_i^j(t)}[\epsilon_1 \log(1+\frac{b_i^j(t)+\sum_{j'\in U_i(t)} b_i^{j'}(t)}{C_i}) \\ - \quad & p_j(t) b_i^j(t)] \\ \text{s.t.} \quad & b_{\min} \le b_i^j(t) \le b_{\max} \end{aligned} \tag{8}$$

where $b_{\min}$ and $b_{\max}$ are feasibility constraints on the possible end-to-end throughput from node j to node i. Eq. (8) also suggests the order of magnitude ϵ_1 has to take in order to keep the value of Eq. (8) positive. For instance, $\epsilon_1 \sim 2C_iP_i$ should be a valid choice, where P_i is the maximum acceptable transmission price to node i.

C. Rate negotiation

Since delivering flows incurs empirical loss to the upstream node, it might be possible, especially when the transmission price is low, that node j's utility at time slot t becomes negative, if it delivers a flow to node i at rate $r_{i,D}^j$. To address this problem, we introduce a two-step rate negotiation mechanism to determine the bandwidth allocated to an one-hop flow. Since downstream nodes are the primary decision makers regarding one-hop flows, we require node j to first compute the range of its acceptable transmission rates $(r_{\min}^{j,U}, r_{\max}^{j,U})$ that keeps its utility positive, and then advertise it to node i.

The values of $r_{\min}^{j,U}$ and $r_{\max}^{j,U}$ may be computed as the roots to the following equation:

$$\begin{aligned} &\epsilon_2 \log\left(1-\frac{x+\sum_{k\in D_j(t)} b_k^j(t)}{C_j}\right) \\ &-\epsilon_2 \log\left(1-\frac{\sum_{k\in D_j(t)} b_k^j(t)}{C_j}\right) + p_j(t)x = 0 \end{aligned} \tag{9}$$

which may be numerically approximated, since Eq. (9) is transcendental.

Knowing node j's acceptable range $(r_{\min}^{j,U}, r_{\max}^{j,U})$, and having measured the available bandwidth B_i^j from node j to itself, node i may decide the feasibility constraints as:

$$\begin{cases} b_{\min} = r_{\min}^{j,U} \\ b_{\max} = \min\{B_i^j, r_{\max}^{j,U}\} \end{cases} \tag{10}$$

and computes $r_{i,D}^j$ by Eq. (8). The achievable utility can be further evaluated by Eq. (7). If the resulting utility is positive and the highest among all the relevant upstream candidates, node i then proceeds to establish the one-hop flow with node j and start to receive data at rate $r_{i,D}^j$.

In summary, we have designed a market-based mechanism, which encourages selfish nodes to contribute their spare bandwidth and prevents them from excessively consuming bandwidth at other nodes by means of transmission prices. Two properties of the mechanism help to provide high-performance bandwidth allocation: (1) upstream nodes have the capability to wisely control their revenue and residual bandwidth using their prices; and (2) downstream nodes aim to maximize their private utility by receiving data from nodes that have both high residual bandwidth and low price.

We should point out that our proposed mechanism is not confined to any particular data dissemination application. Nodes may adjust their behavior in different applications at the same time, based on the same mechanism, as long as their selfishness about the bandwidth resource can be integrally characterized by the same utility function. We discuss outstanding issues with respect to the implementation of the proposed market mechanism as follows.

D. Price and bandwidth probing

As described above, a downstream node needs to probe each upstream candidate for its transmission price and acceptable transmission rates, and measures the available bandwidth between the two. Practically, these two probing tasks can be combined in one step.

Initially, the downstream node i sends a *price probe* (PP) message to each upstream candidate j, which contains the source node ID i and the message ID (PP). Upon receiving the PP message, node j immediately returns *four* identical *price reply* (PR) messages back-to-back. A PR message contains the source ID j, its transmission price, the current local time at node j, and its minimum and maximum acceptable transmission rates $r_{\min}^{j,U}$ and $r_{\max}^{j,U}$.

Hence, node i may estimate the available bandwidth between them based on the arrival times of the messages, as in the simple *Receiver Only Packet Pair* method [24]. Since PR messages are short and only four messages are sent on each request, they do not form intrusive traffic to the network, while still giving overlay nodes a reasonable estimate of available bandwidth between them [25].

TABLE II
MESSAGES IN MARKET-DRIVEN BANDWIDTH ALLOCATION MECHANISM

Price probe (PP)		
Source	Type	Message body
i	PP	NULL
Price reply (PR)		
Source	Type	Message body
j	PR	$r_{\min}^{j,U}, r_{\max}^{j,U}, \langle p_j, t_j, j\rangle_{priv_j}$
Start request (SR)		
Source	Type	Message body
i	SR	$r_{i,D}^j, \langle\langle p_j, t_j, j\rangle, \langle p_j, t_j, j\rangle_{priv_j}\rangle_{priv_i}$

E. Transmission request and avoidance of price dispute

Knowing $r_{\min}^{j,U}$ and $r_{\max}^{j,U}$, node i computes $r_{i,D}^j$ and the corresponding utility. If satisfied with the utility, it then includes $r_{i,D}^j$ in a *start request* (SR) message and sends it to node j; otherwise, it does not need to take any action. As node i requests node j to transmit its required data, one problem may occur. Since transmission prices are dynamically updated, by the time a downstream node i decides to contact an ideal upstream node j to receive data, node j's price may have changed. In order to eliminate such disputes and ensure that node i is still eligible for the previous price, we propose a simple signature-based solution using the public key infrastructure (PKI), as follows.

After receiving the PP message from node i, node j replies with a PR message that contains its *signed* ID, price and current local time: $\langle p_j, t_j, j\rangle_{priv_j}$. Node i, upon receiving the

signed message, is able to decrypt it using the public key of node j and view the price: $\langle p_j, t_j, j\rangle_{priv_j} \overset{pub_j}{\rightarrow} \langle p_j, t_j, j\rangle$. If it decides to take node j as its upstream node, node i sends to node j a *start request* (SR) message that includes the following signed component: $\langle\langle p_j, t_j, j\rangle, \langle p_j, t_j, j\rangle_{priv_j}\rangle_{priv_i}$. By decrypting the component, encrypting the first part $\langle p_j, t_j, j\rangle$ using $priv_j$, and comparing the result with the second part, node j is able to verify that node i quoted an authentic price issued by itself when its local time was t_j. If the price is no older than one time slot, node j will proceed to transmit to node i and still use the previous price; otherwise, it simply sends another set of PR messages, without starting the data transmission.

For convenience, Table II lists all the messages employed by our proposed mechanism, where nodes i and j are assumed to be a downstream and an upstream node, respectively.

VI. Performance Evaluation

Given the market-based mechanisms proposed in the paper, the question that remains to be answered is whether these iterative selfish decisions will lead to an overall outcome that is comparable to the situation that all nodes are responsibly maintaining the shared bandwidth resource. Our simulation-based results show the validity of the proposed mechanism, as well as its performance under various simulated scenarios. In particular, we show that the proposed mechanism is able to generate bandwidth allocations comparable to or better than cooperative situations.

A. Simulation methodology

In our simulations, topologies of the underlying backbone IP-layer network are randomly generated by the BRITE universal topology generator [26], and overlay nodes are randomly connected to backbone routers in the IP network through *last-mile* access links. In all our experiments, the backbone IP network consists of 512 routers and 1024 backbone links. Capacities of the backbone links follow a heavy-tailed distribution between $10Mbps$ and $1024Mbps$. The bandwidth capacities of the *last-mile* links were exponentially distributed with an expectation of $10Mbps$. The overlay network contains 256 overlay nodes. We model background traffic as random noise that is independently generated for each link, with the magnitude uniformly distributed from 0 to a small value, *e.g.*, 5% of the link capacity. The *widest* routing algorithm is used to select a IP-layer path of the highest bandwidth between two overlay nodes.

In the overlay application being simulated, overlay nodes query for large data items, then directly download from the upstream node that it has selected. While downloading a data item, a node probes all eligible upstream candidates every 50 time slots, and attempts to switch to another upstream node if it helps to increase its utility.

2000 items of identical sizes (300 Mb) are randomly distributed among overlay nodes, each having 3 separate copies. Experiments were run for 4000 time slots, with each time slot interpreted as 3 seconds in reality. During the first half of simulation time, overlay nodes sequentially establish data transmissions in randomly chosen time slots. At any time, a node may request for a data item with probability $\lambda = 0.1$. We assume service discovery mechanisms exist, such that a downstream node is able to locate all the upstream nodes that can provide the requested item. All nodes are assumed to be stateless, *i.e.*, we do not consider the case where nodes cache downloaded copies and become eligible upstream candidates in the future.

For fair comparisons, we seek to keep the simulation environment consistent across different schemes under investigation. Our simulation environment include the physical network topology, the assignment of link capacities, background traffic, as well as events of node participation and downloading requests.

In our market-driven bandwidth allocation mechanism (referred to as *market* in simulation results), upstream nodes update Q-values every 20 steps, coefficients ϵ_1 and ϵ_2 are set to $300C_i$ and $0.5C_i$, respectively, where C_i is the bandwidth capacity of node i. Transmission prices are initialized to 5.

B. Strategies used in comparisons

To be used as control in our comparisons, we have implemented four other fully decentralized strategies that determine the bandwidth allocations for one-hop flows.

- The *GreedyCoop* strategy. In this strategy, a downstream node greedily chooses the upstream node that can deliver the data item at the highest throughput, which is the available bandwidth between the selected upstream node and itself. In *GreedyCoop*, the downstream nodes are greedy to exploit available resources, but the upstream nodes always choose to be cooperative, and to provide the requested data item.
- The *GreedySelf* strategy. In this strategy, the downstream node is still greedy, but the selected upstream node may choose to deny the request for service at a particular transmission rate with a probability τ, which represents the degree of selfishness at the upstream node.
- The *CoopCoop* strategy. In this strategy, downstream nodes are cooperative, and only ask for half of the maximum available bandwidth in data transmission, and avoid transmissions with rates below a certain threshold, such as 20 Kbps in our experiments. The upstream nodes are also cooperative, satisfying all requests at the requested transmission rate
- The *CoopSelf* strategy. In this strategy, downstream nodes are still cooperative as in the *CoopCoop* strategy, but upstream nodes may choose to deny service requests with probability τ.

In all these strategies, a downstream node may switch to a different upstream node if it perceives a higher receiving throughput from the new upstream.

C. Evaluation metrics and simulation results

For applications involving one-hop overlay data transmissions, we are most concerned with the total end-to-end

throughput of the entire overlay, how many transmission requests can be successfully handled, and bandwidth utilization.

1) Total end-to-end throughput: We first analyze the total end-to-end throughput of all active one-hop flows in the overlay with the market-based bandwidth allocation mechanism, as compared to the other decentralized strategies. As shown in Fig. 3, as time progresses, total throughput rises initially in all the strategies, and then stabilizes as all the nodes have joined the overlay. The *GreedyCoop* strategy leads to the highest total end-to-end throughput due to the greedy nature of downstream nodes and the altruistic nature of the upstream, which can be treated as the ideal case when evaluating total end-to-end throughput. In contrast, the *GreedySelf* strategy emulates an unregulated selfish network. When the degree of selfishness at upstream nodes is moderate ($\tau = 0.5$), its total throughput is slightly below the market-based mechanism. We have also tested the *GreedySelf* strategy with other τ values. As intuitively expected, lower τ values (less than 0.5, not shown) lead to a total end-to-end throughput very similar to the *GreedyCoop* strategy, and higher τ values (> 0.5) will produce significantly worse total throughput. The *CoopCoop* strategy emulates a cooperative overlay, and the result is slightly higher than the market-based mechanism. With $\tau = 0.5$, the total end-to-end throughput with the *CoopSelf* strategy is evidently lower than that of the other strategies. From these results, we observe that the total throughput of proposed market-based mechanism approaches the ideal case of *GreedyCoop*, and matches or exceeds the throughput achieved by cooperative strategies.

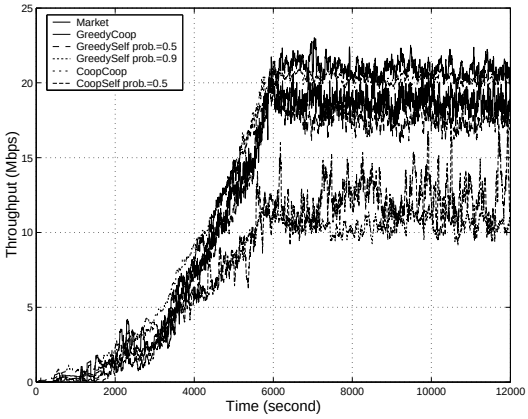

Fig. 3. Total end-to-end throughput of the market-based mechanism, compared to other decentralized strategies with different degrees of selfishness.

2) Percentages of rejected requests: The percentage of rejected requests in all transmission requests also reflects the capability of a strategy to utilize available resources in the overlay. In our experiment, a request is considered to be rejected by the overlay, if the downstream can not find any upstream nodes from which it may download the requested data item. In our market-based mechanism, this may occur when the available bandwidth is low or when the prices are high. In the *GreedySelf* strategy, such denied requests are due to the selfishness of upstream nodes. In *CoopCoop*, the reason may be that the available bandwidth to the downstream node is lower than the minimum threshold. In *CoopSelf*, either of the above two reasons can lead to denied requests. For simplicity, in *GreedySelf* and *CoopSelf* strategies, a request is considered rejected when the best upstream candidate denies the request.

Fig. 4(A)-(E) show that all other strategies stabilizes to a similar mean percentage of rejected requests — around 45%, while the *market-based* mechanism stabilizes to around 30%. This is a very desirable property. It indicates that under the market-based mechanism, downstream and upstream nodes are more likely to be able to manage one-hop flows between them according to their needs. In Fig. 4, the percentage of rejected requests is visibly higher in the first half of time, because many nodes have not joined the overlay, and their data items are not yet discovered. Similarly, Fig. 5 has shown that at any time, *market* and *CoopCoop* have the largest number of active flows being transmitted.

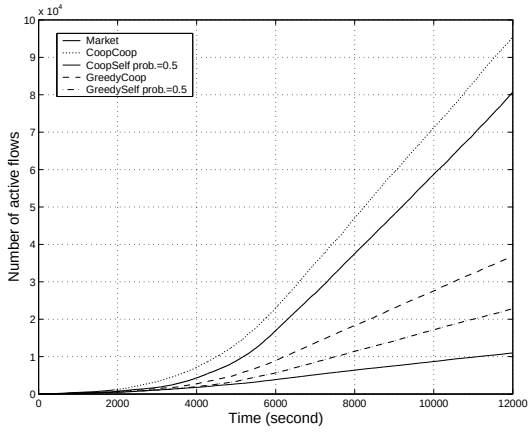

Fig. 5. The total number of active flows in the network.

3) Downloading time distribution: For successfully downloaded items, we record the total downloading time for each item, and plot the cumulative density functions for all strategies in the comparison, as shown in Fig. 6. We have found that the *GreedyCoop*, *GreedySelf* (with $\tau = 0.5$) and *CoopSelf* perform similarly, the market-based mechanism has the lowest average downloading time, and *CoopCoop* has the highest. Combined with previous figures, our results so far indicate that our market-based mechanism delivers comparable or superior performance compared to the cooperative strategies.

4) Bandwidth utilization on last mile links: Fig. 7 shows the bandwidth utilization as a percentage of utilizing last-mile access link capacities at each of the overlay nodes, and Table III lists bandwidth utilization averaged over all overlay nodes, obtained after 5 rounds of simulations.

We observe that in all strategies except *CoopSelf* (with $\tau = 0.5$), bandwidth utilization is quite high, mostly ranging from 50% to 100%. *GreedyCoop* and *CoopCoop* have the highest overall bandwidth utilization, due to the cooperative nature of upstream nodes. Our market-based mechanism performs similarly to the *GreedySelf* strategy.

5) Convergence of Q-value and variation of price: We are further concerned with convergence of Q-values and the

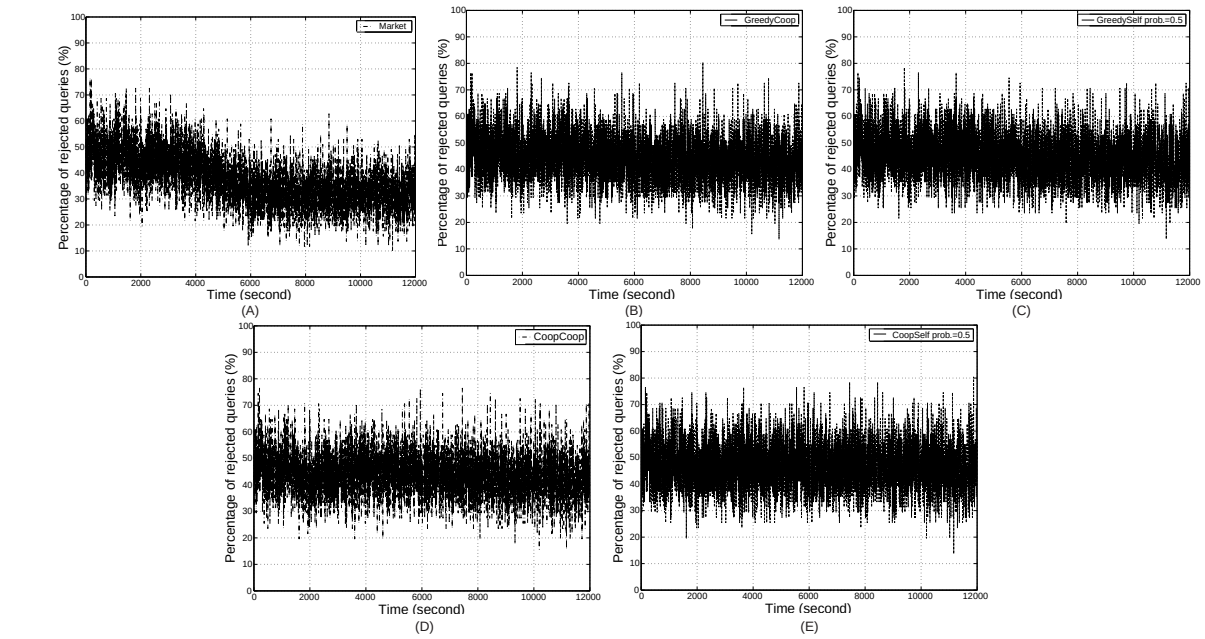

Fig. 4. [illegible] M [illegible]).5.

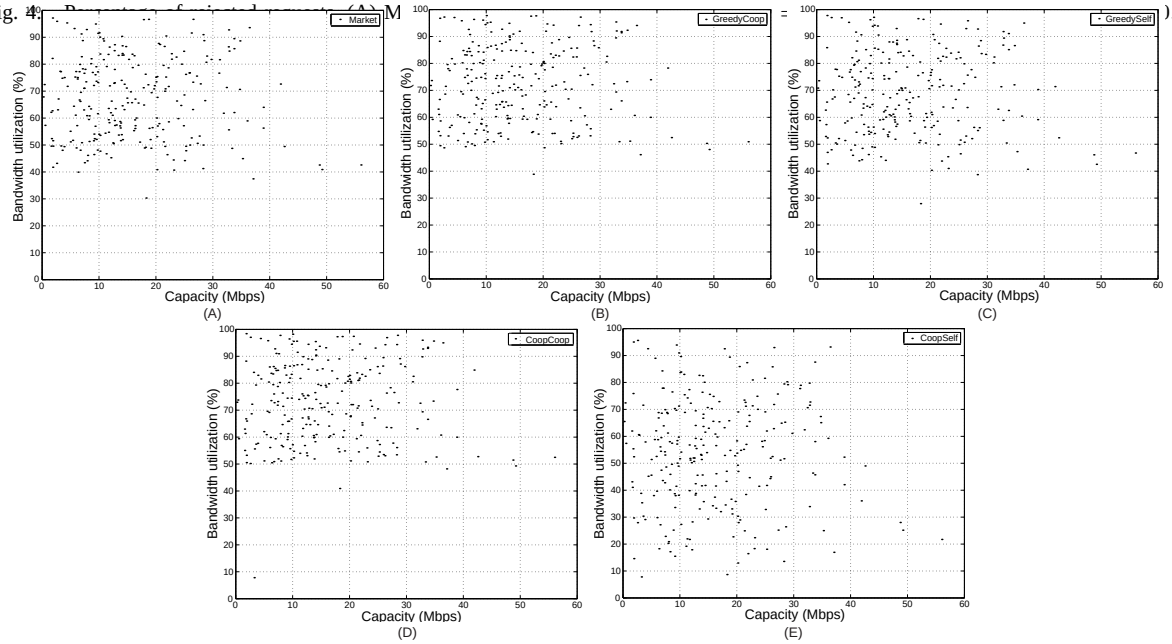

Fig. 7. Bandwidth utilization at different overlay nodes. (A) Market; (B) GreedyCoop; (C) GreedySelf, $\tau = 0.5$; (D) CoopCoop; (E) CoopSelf, $\tau = 0.5$.

resulting transmission prices in our market-based mechanism. In our simulations, we have carefully chosen the number of discrete states (m_b), learning rate (β in Eq. (2)), number of hypothetical iterations (k), and α in Boltzmann distribution, in order to achieve fast convergence and reasonable range of prices. We use the following figures to show the effects of parameter settings, with $m_b = 5$, $\beta = 0.5$, and $\gamma = 0.9$.

Fig. 8 has shown Q-value curves corresponding to all the state-action pairs at an arbitrary node that joins the transmissions at around 1200 seconds after the simulation starts. The number of hypothetical iterations, k, is equal to 5 in Fig. 8(A), and it is 15 in Fig. 8(B). The figures show that Q-values converge quickly after 6000 seconds from the starting point of the simulation, while the learning curves with 15 hypothetical iterations change more sharply.

Fig. 8(C)(D) show Q-values of another node when α was set to 0.001 (Fig. 8(C)) and 0.01 (Fig. 8(D)), respectively. As shown in Fig. 9, the transmission price mostly stays on a positive level when $\alpha = 0.01$, while frequently touching 0 when $\alpha = 0.001$. This is because a very small α offers

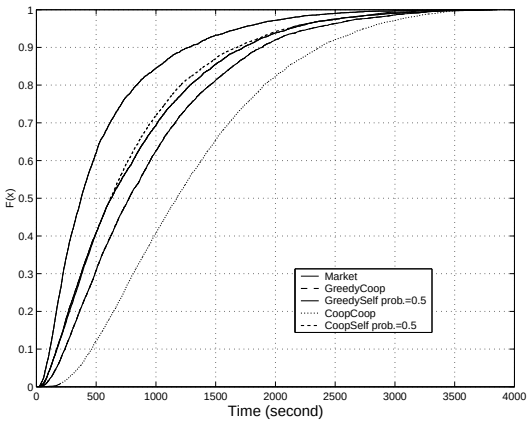

Fig. 6. Cumulative density functions of downloading times.

TABLE III
AVERAGE BANDWIDTH UTILIZATION ON LAST MILE LINKS

Scheme	Bandwidth utilization (%)
Market	70.485
CoopCoop	73.305
CoopSelf	61.31
GreedyCoop	72.93
GreedySelf	70.99

little discrimination among different state-action pairs, so that the price may probabilistically stay at 0, while a relatively larger α may bring the price to a reasonable positive value at the equilibrium. We have also tested even larger α values, *e.g.*, $\alpha = 2.0$, the resulting price may rise without bound or stay infinitely at zero, because a large α essentially prevents reasonable exploration in the state-action space.

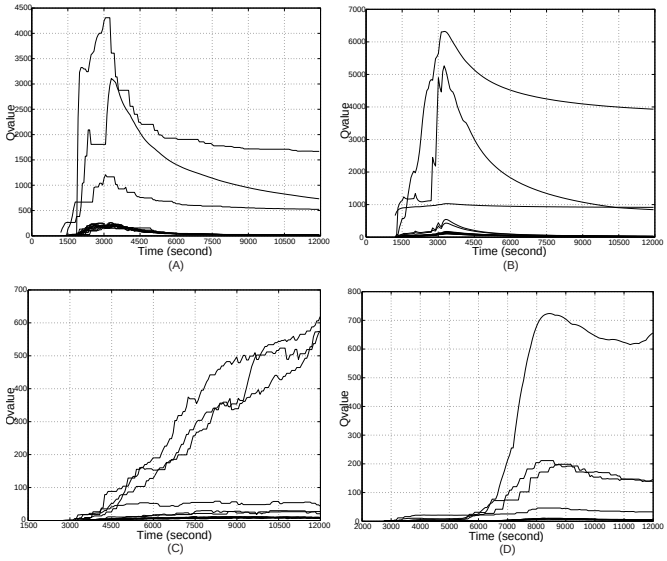

Fig. 8. Convergence of Q-values under different parameter settings.

6) Message overhead: Since a downstream node only needs to send its PP message to a few candidate nodes when it is in need of some data, and each candidate only replies 4 back to back messages, the total number of messages sent in a network

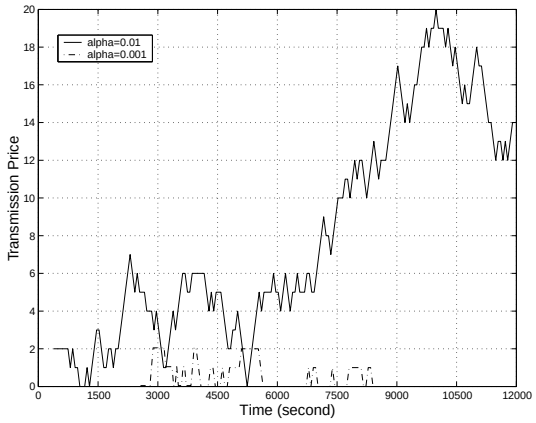

Fig. 9. Variation of transmission price.

increases linearly with the number of overlay nodes and is quite moderate.

To summarize, our simulation results have clearly shown the advantages of our proposed market-based mechanisms as compared to other strategies with different degrees of selfishness. It is also clear that, using reinforcement learning, upstream nodes can efficiently adjust their behavior under system dynamics. For example, learning can be performed while nodes dynamically join the overlay, which gradually leads to better performance. Further, the market-based mechanism leads to a total end-to-end throughput comparable to the *GreedyCoop* strategy, the number of active flows comparable to the *CoopCoop* strategy, as well as the lowest percentage of rejected requests. These results have supported our claim that the market-based mechanism has achieved desirable system-wide properties with respect to bandwidth allocation in selfish overlay networks.

VII. CONCLUSIONS

In this paper, we have addressed the problem of bandwidth allocation in overlay networks comprised of selfish nodes, and designed a market-based mechanism that consists of a pricing game and local utility optimizations at downstream and upstream nodes. We propose distributed solutions that feasibly solve the pricing game, and discuss the local decision problems regarding each one-hop flow. The highlights of this paper are as follows. First, the selfish behavior of overlay nodes is modeled as local maximization. With adequate pricing mechanisms, upstream nodes are obliged to contribute their bandwidth as much as possible, while maintaining sufficient residual bandwidth at the same time; downstream nodes are forced to consume bandwidth wisely, while maintaining a certain level of empirical benefits. Second, we introduce the *learning capability* to overlay nodes, so that they are able to infer the dynamics of the external environment, and to act adaptively and optimally. We believe that the general service provisioning framework used in this paper can be utilized to solve other similar problems that involve the provisioning of services in dynamic settings.

REFERENCES

[1] Y. A. Korlis, A. A. Lazar, and A. Orda, "Architecting Noncooperative Networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, 1995.

[2] R. J. La and V. Anantharam, "Optimal Routing Control: Game Theoretic Approach," in *Proceedings of the 36th Conference on Decision and Control*, 1997.

[3] E. Altman, T. Basar, T. Jimenez, and N. Shimkin, "Competitive Routing in Networks with Polynomial Costs," in *Proceedings of IEEE INFOCOM*, 2000.

[4] T. Roughgarden and E. Tardos, "How Bad is Selfish Routing," *Journal of ACM*, vol. 49, no. 2, pp. 236–259, 2002.

[5] L. Qiu, Y. Yang, Y. Zhang, and S. Shenker, "On Selfish Routing in Internet-Like Environments," in *in Proceedings of ACM SIGCOMM*, August 2003.

[6] F. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, , no. 49, pp. 237–252, 1998.

[7] J. Shu and P. Varaiya, "Pricing Network Services," in *Proceedings of IEEE INFOCOM*, 2003.

[8] P. Golle, K. L. Brown, I. Mironov, and M. Lillibridge, "Incentives for Sharing in Peer-to-Peer Networks," in *Proceedings of the 2nd International Workshop on Electronic Commerce*, 2001.

[9] T. Alpcan and T. Basar, "A Utility-Based Congestion Control Scheme for Internet-Style Networks with Delay," in *Proc. of IEEE INFOCOM*, 2003.

[10] T. Basar and R. Srikant, "Revenue-Maximizing Pricing and Capacity Expansion in a Multi-User Regime," in *Proceedings of IEEE INFOCOM*, 2002.

[11] N. Nisan and A. Ronen, "Algorithmic Mechanism Design," *Games and Economic Behavior*, vol. 35, 2001.

[12] A. Lazar and N. Semret, "Design and Analysis of the Progressive Second Price Auction for Network Bandwidth Sharing," *Telecommunications Systems: Special Issue on Network Economics*, 1999.

[13] N. Semret, R. R.-F. Liao, A. T. Campbell, and A. A. Lazar, "Pricing, Provisioning and Peering: Dynamic Markets for Differentiated Internet Services and Implications for Network Interconnections," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2499–2513, December 2000.

[14] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta, "Spawn: A Distributed Computational Economy," *Software Engineering*, vol. 18, no. 2, pp. 103–117, 1992.

[15] J. Feigenbaum and S. Shenker, "Distributed Algorithmic Mechanism Design: Recent Results and Future Directions," in *Sixth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial'M 2002)*, September 2002.

[16] J. Feigenbaum, C. Papadimitriou, and S. Shenker, "Sharing the Cost of Multicast Transmission," in *Journal of Computer and System Sciences*, 2002.

[17] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based Mechanism for Lowest-Cost Routing," in *Proceedings of ACM PODC*, 2002.

[18] P. Daras, D. Palaka, V. Giagourta, D. Bechtsis, K. Petridis, and M. G. Strintzis, "A Novel Peer-to-Peer Payment Protocol," in *Proceedings of IEEE EUROCON*, 2003.

[19] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, "Karma: A Secure Economic Framework for Peer-to-Peer Resource Sharing," in *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[20] J. F. Nash, "Non-cooperative games," *Annals of Mathematics*, vol. 54, pp. 289–295, 1951.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[22] R. S. Sutton, "Integrated Architectures for Learning, Planning, and Reacting based on Approximating Dynamic Programming," in *Proceedings of the Seventh International Conference on Machine Learning*, 1990.

[23] Leslie Pack Kaelbling, Michael Littman, and Andrew Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[24] K. Lai and M. Baker, "Measuring Bandwidth," in *Proceedings of IEEE INFOCOM*, 1999.

[25] T. S. E. Ng, Y. Chu, S. G. Rao, K. Sripanidkulchai, and H. Zhang, "Measurement-Based Optimization Techinques for Bandwidth-Demanding Peer-to-Peer Systems," in *Proceedings of IEEE INFOCOM*, 2003.

[26] Boston University, "BRITE: Universal Topology Generator," *available on line at http://www.cs.bu.edu/brite*.