Market-Based Self-Optimization for Autonomic Service Overlay Networks

Weihong Wang, Baochun Li

***Abstract*— Rather than managing their heterogeneity and dynamic behavior through centralized intervention, overlay nodes can be programmed to self-organize and self-manage the network. To achieve the highest performance within a service overlay, they are further expected to *self-optimize* the network, by cooperatively providing and allocating resources in an optimal manner. However, since nodes are inherently selfish about resources they contribute or consume, self-optimization could not be achieved if they are not given the correct *incentives*. In this paper, we investigate the effectiveness of a market-based incentive mechanism in directing nodes' behavior and enabling self-optimizations.**

We have designed an *intelligent market* model for a service overlay network, based on which individual nodes, being service *producers* and *consumers*, determine their own resource contributions, consumptions, or service prices based on their own utility maximization goals. We also propose optimal decision making solutions for nodes to achieve their self-interests; in particular, service providers are provided with a *control-based* pricing solution based on *system identification* techniques.

With the multicast streaming application as an example, we show through simulations that, even when selfish nodes all seek their maximal utilities, the resulting network still achieves close-to-optimal performance in both steady and dynamic states. The results also indicate that, by encouraging nodes to behave selfishly and intelligently in a designed market, self-optimization in other autonomic systems may be facilitated in the presence of node selfishness.

I. Introduction

Participants in overlay networks reside in geographically dispersed locations, access the Internet via heterogeneous access technologies, and belong to different administrative domains with different policies. They may join or leave the network at any time, leaving the composition of an overlay network highly dynamic. Due to these characters, it is nearly impossible to manage an overlay network with centrally coordinated intervention, especially as the network becomes large. Therefore, overlay networks are a natural form of autonomic systems. It has been a well-known design philosophy to distribute to individual nodes the functionalities of organizing, controlling and managing an overlay network.

At the topological level, there exist overlay structures (*e.g.*, Chord [1] and Pastry [2], *etc.*), that provide the basic functionalities for nodes to *self-organize* into an overlay network, and to *self-heal* at times of arbitrary node participation and departures. At the service management level, it has been further studied how nodes should self-optimize towards certain global optimal objectives. As an example, the overlay multicast protocol *Narada* [3] aims to minimize end-to-end delays while avoiding high link stress, with nodes choosing parent nodes[1] on their own.

However, in overlay networks consisting of independent and heterogeneous nodes, achieving self-optimization is non-trivial, due to the critical but often overlooked observation: Nodes are inherently *selfish*. The selfishness is caused by the fact that overlay nodes belong to different administrative domains and users, who enjoy the complete freedom to choose the best courses of action that maximize their utilities. They may not follow any externally dictated global optimization algorithms, if their self-interests are not satisfied.

In this context, the critical question is the following: how should we influence the inherent behavior of selfish nodes using certain *incentives*, so that the collective outcome of individual nodes behaving towards their own self-interests still leads to the desirable system optimality?

Game theoretic models [4], [5], [6] have been employed as incentive mechanisms to model selfish nodes, and the steady-state properties of these mechanisms have been widely studied in previous literature. Distributed pricing models [7], [8] have also been proposed to regulate the behavior of service providers and consumers, under the goal of social welfare maximization. However, a common drawback of previous work is that, they are mostly theoretical in nature, and are usually subject to strict assumptions that do not hold in realistic overlay networks.

In contrast, this paper seeks to propose a resource allocation framework for realistic overlay networks composed of selfish nodes. Similar to previous studies on resource allocation in communication networks [9], [10], we take the viewpoint that overlay nodes should be allowed to behave selfishly, and that the optimality of a network should be evaluated from the point of view of the entire system. We aim to achieve two objectives. First, we seek to propose an incentive mechanism that promotes resource contribution and prevents resource overuse, not only at the steady state, but also at times of network dynamics where the supply and demand relationship changes. Second, we seek to design an appropriate software agent that best delegates the selfish user under the proposed incentive mechanism. With these two building blocks, the network performance parameters resulted from *individual* decisions may approach those determined by *global* optimization methods.

The authors are affiliated with the Department of Electrical and Computer Engineering, University of Toronto. Their email addresses are {*wwang,bli*}*@eecg.toronto.edu*.

[1]Henceforth in this paper, a parent of an overlay node is referred to as an *upstream node*, whereas a child is referred to as a *downstream node*.

Our proposed incentive mechanism builds upon an *intelligent market* model, which encourages both service providers and consumers to pursue their highest possible utilities with intelligence. In this paper, we choose an overlay media streaming application as a running example, where upstream nodes that forward media streams are treated as service providers, and downstream nodes as consumers. Each service provider maintains a dynamic price for the service it delivers, which is periodically adjusted for its highest level of utility. Each service consumer, out of multiple service provider candidates, selects the ones that best balance its attainable QoS parameters and economic costs.

Through extensive comparison studies with a well-known approximately optimal overlay multicast protocol, *Narada* [3], we have shown that our market-based incentive mechanism improves the average throughput in the multicast tree topology, and efficiently adapts the topology and bandwidth allocation to network dynamics, while only incurring minor communication and computation costs. We also believe that, with minor extensions, the proposed market mechanism may serve as a general framework for achieving self-optimization for other autonomic systems that consist of selfish and intelligent components.

The remainder of the paper is organized as follows. Sec. II describes the overlay streaming application, and defines our market model. Sec. III form the models on node selfishness and formulates the local optimization problems to be solved by individual nodes. In particular, a novel decision making solution for upstream nodes based on optimal control and system identification is proposed in Sec. IV. Sec. V discusses distributed protocols that facilitate the self-optimization process, and Sec. VI presents our simulation-based evaluations. Related work on autonomic overlay networks is discussed in Sec. VII, and Sec. VIII concludes the paper.

II. Intelligent Market Model

Throughout this paper, we use an overlay media streaming application as an example. As illustrated in Fig. 1, overlay media streaming is an application that multicasts streaming media in an overlay network, from a source node to a set of receiver nodes, that together form the multicast group. Rather than relying on IP multicast, overlay nodes serve as application-layer switches and forward received data to downstream nodes via unicast connections. Overlay multicast topologies may take the form of a single tree [3], multiple trees [11] or a mesh [12]. In some of the designs, receivers in a multicast group may receive media content at different rates compared to their upstream nodes. This can be realized by the use of *multiple description coding*. In this application, the design objective is to achieve *optimal topology formation*, which includes the construction of the overlay topology, and the subsequent bandwidth allocation on overlay links. A topology is considered desirable, if it leads to high average end-to-end throughput, low average end-to-end delay, and low average packet loss rate for all overlay links.

In previous work, distributed protocols are proposed to construct overlay multicast topologies [3], [11], [13], in which nodes are only considered as agents to execute the prescribed protocols. Given the topology, bandwidth shares allocated for multicast traffic are determined by the source rate and the available bandwidth along all relevant physical links.

In comparison, node selfishness is acknowledged in this paper, where we evaluate the optimality of network performance with the total satisfaction perceived by all the nodes. Individual nodes are given complete freedom to determine their connections with other nodes, based on their own utility evaluations. The original problem of optimal topology formulation and resource allocation is therefore turned into a collection of localized decision problems, within which nodes determine how to make the best use of their bandwidth resources to improve their utilities.

In the media streaming application, an overlay node may be seen as providing the *media delivery service* its downstream nodes, if it is serving active media streams. When such a service is treated as a *product* traded within the multicast group, we propose the following *intelligent market model*. In the context of multicast bandwidth allocation, it associates each multicast group with a market.

Within a multicast topology, an upstream node and its immediate downstream node are identified as the *seller* and the *buyer*, respectively. For example, in the simple multicast example of Fig. 1, on the overlay link from node 1 to node 2, node 1 acts as the seller, and node 2 as a buyer. A node is a seller *and* a buyer if it both sends and receives in the multicast group.

The media delivery service is quantified based on the *amount of bandwidth* the upstream node contributes to the downstream node, or the end-to-end throughput the upstream node delivers. Products traded between different pairs of sellers and buyers are further differentiated by other QoS metrics, such as end-to-end delay and end-to-end packet loss rate. Apparently, these metrics are specific to the pair of seller and buyer: they change with the underlying physical path it goes along, and are subject to relevant network dynamics.

Naturally, we require the service to be priced by the end-to-end throughput it provides; the economic revenue or payment regarding an overlay link is determined as the product of throughput and price. We further assume that multiple prices co-exist on the market — each seller determines its charge for *per-unit* of throughput it delivers, and prices are dynamically adjusted by the sellers.

On joining the multicast group, a potential buyer identifies its seller candidates, and evaluates each of them by their prices *and* deliverable Quality of Service (*e.g.*, maximum throughput). Once the buyer has selected a candidate as a seller, a corresponding *link* will be added to the multicast topology. The share of bandwidth to be allocated to the overlay link, henceforth referred to as *reserved bandwidth*, is negotiated by the two nodes.

The reserved bandwidth differs from the actual end-to-end throughput on an overlay link, in the sense that the former is agreed upon by the two nodes before the transfer begins, and the latter is supposed to approximate the former, although its value is affected by flow control and congestion control performed in the network under realistic traffic situations. Payments are to be calculated based on the reserved bandwidth.

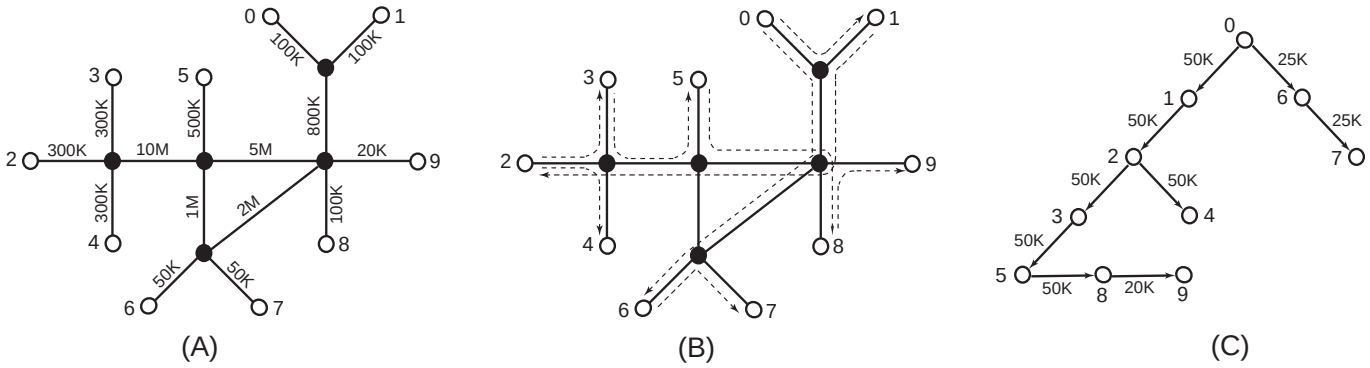

Fig. 1. (A) Overlay nodes (nodes $0 \sim 9$, represented by hollow circles) interconnected by core nodes (represented by solid circles). (B) The match between upstream and downstream nodes, node 0 is the source. (C) The corresponding multicast topology and a possible bandwidth allocation.

Since the network may undergo unpredictable dynamics at any time, we allow buyers in a multicast topology to periodically re-examine their sellers and seller candidates, and to either switch to other alternative sellers, or readjust its reserved bandwidth to the optimal value if its current upstream node remains to be the best seller. Therefore, bandwidth allocation in the multicast topology is adaptively updated as overlay links are established, disconnected and adjusted with respect to bandwidth.

Finally, it is worth noting that our intelligent market model does not require any actual monetary flows between overlay nodes, but may take the form of "virtual currency" that circulates within the network.

III. Modeling Node Selfishness

With economic factors as external incentives, there are a number of ways of modeling the decisions of selfish overlay nodes, each corresponding to a different formulation of optimization problems. For example, one may suggest that we maximize the downstream node's empirical benefit given its economic budget, or to maximize the upstream node's economic profit while delivering services at a fixed quality level. In this paper, we combine the empirical and economic concerns, and assume that nodes always make decisions that best balance the two aspects.

Mathematically, any selfish decision of a node is driven by its *utility function*, which summarizes its inherent preference over its experiences in the network. We further model the selfish nodes as *utility maximizers*, making all their decisions towards maximizing their utilities.

Since the concrete forms of utility functions are essentially unknown *a priori*, we aim to "design" the formulation of such utility functions, such that they represent the best interests and selfishness of the overlay nodes. By designing the utility functions, we may examine the effects of the proposed market model and incentive mechanism by emulating the most likely behavior of selfish nodes.

A. Utility functions

We consider the discrete time domain where time is divided into *slots*, and introduce the following notations. For each time slot t, a node i keeps a price $p_i(t)$ for each unit of bandwidth it reserves for its immediate downstream nodes, which form the set $R_i(t)$. At the same time, it receives streams from a set $S_i(t)$ of upstream nodes. The stream from node j to node i has a reserved bandwidth of $b_i^j(t)$, an end-to-end delay $d_i^j(t)$ and a loss rate $l_i^j(t)$ as perceived by node i. $B_i(t)$ denotes the local available network bandwidth of node i, and $m_i(t)$ is the economic budget maintained by node i itself. In addition, we denote the local bandwidth capacity of node i as C_i, the maximal tolerable delay as D_i, and the maximal tolerable loss rate as L_i.

We assume that the utility function of node i, either as a downstream or an upstream node, takes a *quasi-linear* form: the utility equals the sum of an *empirical* and an *economic* component. The former accounts for the node's empirical benefit (or loss) for receiving (or providing) certain services, which may be characterized by various quality metrics of the services received, *e.g.*, $b_i^j(t)$, $d_i^j(t)$, or $l_i^j(t)$. The latter equals the revenues (or costs) due to the delivery (or consumption) of services. We choose the quasi-linear form of utility functions, since any equilibrium solutions to utility maximization problems are independent of the initial economic funds of market participants, if the economic funds constitute an additive term in each market participant's utility function [14].

The economic component can be simply expressed as the product of the corresponding price and bandwidth value. However, the formulation of the empirical component needs to satisfy a few mathematical properties: In order to present a reasonable preference relation, it has to be *monotonic* and *concave* with respect to each variable it takes; and it usually needs to be twice differentiable for an optimal point to exist analytically [14].

As a possible formulation, we propose the utility functions, $u_{i,D}(t)$ and $u_{i,U}(t)$, for node i, in the form of Eq.(1.1) and Eq.(1.2), as it acts as downstream and upstream nodes, respectively. In both expressions, the last term represents the economic component, and the remaining terms represent the empirical component. By Eq. (1.1), we assume that an end user may simultaneously evaluate throughput, delay and loss rate when receiving streams, though any of them can be omitted by setting the corresponding coefficient to zero. By considering different subsets of the upstream node set $S_i(t)$,

$$\begin{cases} u_{i,D}(t) = \epsilon_{i,1}\log\left(1+\frac{\sum_{j\in S_i(t)} b_i^j(t)}{C_{i,D}}\right) - \epsilon_{i,2}\log\left(1+\frac{\max_{j\in S_i(t)} d_i^j(t)}{D_i}\right) \\ \qquad -\epsilon_{i,3}\log\left(1+\frac{\max_{j\in S_i(t)} l_i^j(t)}{L_i}\right) - \sum_{j\in S_i(t)} p_j(t) b_i^j(t) \qquad (1.1) \\ u_{i,U}(t) = \epsilon_{i,4}\log\left(1-\frac{\sum_{k\in R_i(t)} b_k^i(t)}{C_{i,U}}\right) + \sum_{k\in R_i(t)} p_i(t) b_k^i(t) \qquad (1.2) \end{cases} \quad (1)$$

the equations can cover any topological cases for multicast, *e.g.*, single tree, multiple trees or mesh. In Eq. (1.2), we assume that the network bandwidth is the main resource constraint that each upstream node considers in our example of a streaming application.

Coefficients $\epsilon_{i,l}$ $(l = 1, 2, 3, 4)$ are positive weights that indicate the relative importance of the three metrics — and the relative importance of the empirical and economic components — for the end user. All the parameters $C_{i,D}$, $C_{i,U}$, D_i, L_i and $\epsilon_{i,l}, l = 1, 2, 3, 4$ are inherently node-specific and application-specific, and may be configurable by a node for each multicast group it joins. However, a correct setting of parameters should guarantee that, a node would be willing to take an action, *i.e.*, receiving a stream from a upstream node at certain QoS levels and charges, or sending a stream to a downstream node at certain throughput and earnings, only when the corresponding utility is above zero.

B. Decision problems

Under the prescribed market model, the decision problem of a downstream node is straightforward: It periodically updates its best choices of upstream nodes for receiving the streams, or equivalently, the best combinations of $b_i^j(t)$, $d_i^j(t)$, $l_i^j(t)$ and $p_j(t)$, in the changing environment. As an upstream node, however, it is presented with two decision-making problems. First, upon being requested by any potential downstream node, the upstream node should decide the best throughput that maximizes its own utility. Second, it needs to periodically update the optimal price $p_i(t)$ that induces the highest future utility for itself.

More specifically, to choose the best upstream node, a downstream node i evaluates each upstream candidate j by first computing the optimal throughput $b_{i,D}^{j*}(t+1)$ from Eq. (2). In this equation, $\Delta u_{i,D}(t)$ denotes the expected utility improvement if node i were to receive a flow from node j, assuming that the prices and transmissions from all its other upstream nodes remain unchanged. If node j is one of the current upstream nodes of node i, both $d_i^j(t)$ and $l_i^j(t)$ are measurable from past transmissions; otherwise, since $l_i^j(t)$ would be missing, the third $\log(\cdot)$ term needs to be removed from the expression. If there is no solution to Eq. (2), node j will be excluded from consideration for the tth time slot. Once $b_{i,D}^{j*}(t)$ is determined for each eligible node j, node i then chooses the one, if it exists, with the highest non-negative $\Delta u_{i,D}(t)$.

Condition (2.1) implies the *utility constraint*. Intuitively, node i would not choose to be served by node j if, by doing so, node i's utility decreases. Condition (2.2) is the *budget constraint*: the sum of the anticipated payment should not exceed the current economic budget $m_i(t)$ of node i, and $m_i(t)$ is dynamically updated as node i pays charges or earns revenues. Condition (2.3) represents the *physical constraint*, where $b_{i,\min}$ and $b_{i,\max}$ are determined by the local available bandwidth $B_i(t)$ of node i, and the utility-restricted outgoing throughput bounds of node j, as will be discussed in Sec. V.

On the side of the upstream node, node j evaluates its future behavior of delivering a stream at throughput $b_i^j(t+1)$ by the corresponding utility increment $\Delta u_{j,U}(t+1)$. This is expressed in Eq. (3), where $B_j(t)$ is node j's local available bandwidth, and $\bar{b}_j(t)$ corresponds to the data generation rate if j is the original source, or the input rate if j is a branch node in the multicast topology. Node j would not be sending the stream at $b_i^j(t+1)$ if the resulting $\Delta u_{j,U}(t+1)$ is negative. Clearly, for a pair of nodes to establish a connection, some negotiation on $b_i^j(t)$ is necessary to reconcile the two selfish entities. A viable way of conducting such negotiations is described in Sec. V.

From the economic perspective, the decision of node i on price $p_i(t)$ aims to maximize its revenues to be made in time slot t:

$$p_i^*(t) = \arg\max_{p_i(t)} \left[p_i(t) \sum_{k\in R_i(t)} b_k^i(t) \right] \quad (4)$$

We devote the next section to an in-depth discussion towards an intelligent solution for such a pricing problem.

To summarize, we have proposed a mechanism that incentivizes selfish nodes with prices and in the context of markets, and have modeled selfish nodes as utility maximizers. The utility functions Eq. (1.1) and Eq. (1.2) are formulations reflecting the preferences of selfish nodes over different empirical and economic factors, and we believe that there exist many other eligible forms. An alternative solution is to give end users even more flexibility in determining their own utility functions through online identification, as has been proposed by Courcoubetis *et al.* [15]. As nodes adjust their behavior based on the utility maximization goal, the overlay network is expected to be *self-optimizing*: multicast topology

$$
\begin{aligned}
b_{i,D}^{j*}(t+1) &= \arg\max_{b_i^{j*}(t+1)} \Delta u_{i,D}(t+1) \\
&= \arg\max_{b_i^{j*}(t+1)} \left\{ \epsilon_{i,1} \log \left(\frac{C_{i,D} + \sum\limits_{j' \in S_i(t) \setminus j} b_i^{j'}(t) + b_i^j(t+1)}{C_{i,D} + \sum\limits_{j' \in S_i(t)} b_i^j(t)} \right) - \epsilon_{i,2} \log \left(\frac{D_i + \max\limits_{j' \in S_i(t) \cup j} d_i^{j'}(t)}{D_i + \max\limits_{j' \in S_i(t)} d_i^{j'}(t)} \right) \right. \\
&\quad \left. - \epsilon_{i,3} \log \left(\frac{L_i + \max\limits_{j' \in S_i(t) \cup j} l_i^{j'}(t)}{L_i + \max\limits_{j' \in S_i(t)} l_i^{j'}(t)} \right) - p_j(t)[b_i^j(t+1) - b_i^j(t)] \right\}
\end{aligned} \tag{2}
$$

s.t.

$$
\begin{cases}
\Delta u_{i,D}(t) \geq 0 & (2.1) \\
\sum_{j' \in S_i(t) \setminus j} p_{j}^{'}(t) b_i^{j'}(t) + p_j(t) b_i^j(t) \leq m_i(t) & (2.2) \\
b_{i,\min} \leq b_i^j(t) \leq b_{i,\max} & (2.3)
\end{cases}
$$

$$
\begin{cases}
\Delta u_{j,U}(t+1) = \epsilon_{j,2} \log \left(\dfrac{C_{j,U} - \sum\limits_{k \in \mathcal{R}_j(t) \setminus i} b_j^k(t) - b_i^j(t+1)}{C_{j,U} - \sum\limits_{k \in \mathcal{R}_j(t) \setminus i} b_j^k(t)} \right) + p_j(t) \cdot b_i^j(t+1) \\
b_i^j(t+1) \leq \min(B_j(t), \bar{b}_j(t))
\end{cases} \tag{3}
$$

and bandwidth allocation are automatically adapted to network dynamics, which include node joining and departures, as well as variations in cross traffic. The performance metrics under concern, *e.g.*, total throughput, average delay, and average loss rate, are maintained at acceptable levels in all situations.

IV. Optimal control based pricing decisions

Since making decisions on downstream nodes based on Eq. (2) is rather straightforward, in this section, we focus on the decision problem from the point of view of an upstream node, and propose a solution to make pricing decisions. Given the operation of the intelligent market model, an upstream node i's utility is dependent on its own price, the remaining bandwidth capacity, and the following factors: (1) the set of nodes that compete with i as upstream nodes; (2) the performance measurements on overlay links from these competitors to any potential downstream node k; (3) the utility function of node k; and (4) the prices of the competitors. However, these factors are essentially unknown to node i, since the propagation of global information cannot be assumed in autonomic systems, and is practically infeasible.

A. Optimal control formulation

Since service prices influence the topology formed and bandwidth allocated, each node may be considered to be applying a *control input* to a *plant*, which is the entire system consisting of all the sellers and buyers. In system control terms, we denote $q_i(t)$ as the control input supplied at the beginning of time slot t, and $\sum_{k \in R_i(t)} b_k^i(t)$ as the total amount of bandwidth consumed by i's buyers as the *system output* obtained at the end of the slot. With an appropriate system equation, we may mathematically represent the dependence of output on control input, as well as the *noise input* $\omega_i(t)$ representing dynamic factors that are usually stochastic and hard to model, including the network topology, storage access patterns, background traffic, and the effects of upstream competitions. An illustration of such a dynamic system is shown in Fig. 2.

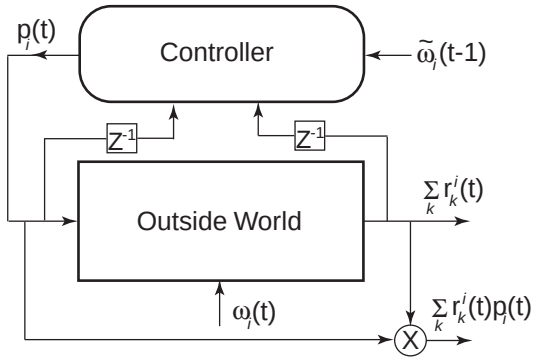

Fig. 2. A diagram of the optimal control system. Z^{-1} represents a *one-step* delay in the time scale. Past values $p_i(t-1)$ and $\sum_{k \in R_i(t)} b_k^i(t)$, as well as an estimate $\tilde{\omega}(t-1)$ of the past noise input, are used to identify the mathematical model of the outside world, and to decide the new price $p_i(t)$ for the optimization goal.

Since each node determines its price according to its utility maximization goal, we may transform the original decision problem Eq. (4) into an optimal control problem based on the system view: node i decides $p_i(t)$ as an *optimal control signal* to the system, so that the *control objective*, *i.e.*, node i's utility $\sum_{k \in R_i(t)} b_k^i(t) p_i(t)$ in the tth time slot, is maximized.

B. System identification

To determine the optimal control signal, node i needs to first identify the system equation, in order to predict the system output based on any input. However, for this particular system, we do not have any specific insights into the underlying mechanism except its nonlinearity: when the external world is relatively stable, $p_i(t)$ is small, and the remaining bandwidth of node i is sufficiently high, $\sum_{k \in R_i(t)} b_k^i(t)$ may increase even when $p_i(t)$ increases; while after $p_i(t)$ or the level of

remaining bandwidth reaches some point, $\sum_{k\in R_i(t)} b_k^i(t)$ may decrease significantly as $p_i(t)$ increases.

We hence take the *nonlinear black-box* parameterization method [16], which is an established way of emulating any system model about which little *a priori* knowledge is known, and we identify the involved system parameters by the *least squares estimation* method.

With the nonlinear black-box method, the system output is expressed as a weighted sum of *basis functions*, which are mathematical expressions of past and present system input, past system output, related state variables, and noises. In our problem, we express the system output $\sum_{k\in R_i(t)} b_k^i(t)$, hereafter denoted as $g(t)$, as a function of $p_i(t)$, $\omega_i(t)$ and $g(t-1) = \sum_{k\in R_i(t-1)} b_k^i(t-1)$. Taking the sigmoid basis function:

$$\begin{aligned}\kappa_m(p_i(t),\omega_i(t),g(t-1)) &= \frac{1}{1+e^{-\beta[p_i(t)-\gamma m]}} \\ &\cdot\frac{1}{1+e^{-\beta[\omega_i(t)-\gamma m]}} \\ &\cdot\frac{1}{1+e^{-\beta[g(t-1)-\gamma m]}} \quad (5)\end{aligned}$$

we may obtain $\tilde{g}(t)$, an estimate of $g(t)$, from the following system model:

$$\tilde{g}(t) = \sum_{m=1}^{n} \alpha_m(t)\kappa_m(p_i(t),\omega_i(t),g(t-1)) = \boldsymbol{\alpha^T}(t)\boldsymbol{\phi}(t) \quad (6)$$

In this equation, β and γ are positive constants determined by the requirements imposed on the approximation accuracy of $\tilde{g}(t)$ and the upper bound of the derivative of $g(t)$. $\alpha_m(t)$, $m = 1,\ldots,n$ are parameters of the system model that needs to be identified from historical data $p_i(t)$, $\omega_i(t)$ and $g(t-1)$. The constant n is dependent on the ranges of $p_i(t)$, $\omega_i(t)$ and $g(t-1)$, and determines the modeling capacity of the expression. $\boldsymbol{\alpha^T}(t)$ and $\boldsymbol{\phi}(t)$ are two vectors that consist of $\{\alpha_m(t), m = 1,\ldots,n\}$ and $\{\kappa_m(p_i(t),\omega_i(t),g(t-1)), m = 1,\ldots,n\}$, respectively. Moreover, the variable $\omega_i(t)$ is stochastic in nature and unobservable by node i, thus, it need to be estimated from the historical data of control input and output.

At the end of time slot t, an iteration round is carried out in two steps. First, since the values of $g(t)$, $g(t-1)$ and $p_i(t)$ are known, the value of $\omega_i(t)$ may be estimated as the minimizing point of prediction error of the system model identified thus far, and we denote the estimated value as $\tilde{\omega}_i(t)$:

$$\tilde{\omega}_i(t) = \arg\min\{g(t) - \boldsymbol{\alpha^T}(t-1)\boldsymbol{\phi}(t)\} \quad (7)$$

To reduce the effect of randomness on system parameters, we *smooth* the estimate $\tilde{\omega}_i(t)$ with a weighted sum: $\tilde{\omega}_i(t) \leftarrow (1-\lambda)\tilde{\omega}_i(t) + \lambda\tilde{\omega}_i(t-1)$, and λ is a constant within $(0,1)$. This smoothed $\tilde{\omega}_i(t)$ will be used in the second step of updating system parameters $\{\alpha_m(t), m = 1,\ldots,n\}$. Since the system model is stochastic and inherently *time-varying*, we may update the parameters online by performing a round of recursive least squares estimation at the end of each slot t. In our simulations, we have adopted a simplified algorithm called *stochastic approximation* algorithm [17], whose updating rule can be expressed as:

$$\boldsymbol{\alpha}(t) = \boldsymbol{\alpha}(t-1) + \sigma(t)\boldsymbol{\phi}(t)\nu(t) \quad (8)$$

where $\nu(t) = g(t) - \boldsymbol{\phi^T(t)\alpha}(t-1)$, and $\sigma^{-1}(t) = \sigma^{-1}(t-1) + \boldsymbol{\phi^T(t)\phi(t)}$, with $\sigma^{-1}(0)$ being a very small positive value.

As Eqs. (6), (7), (8) have shown, the system identification procedure is computationally cheap, since, at each time slot t, a node only needs to (1) locally keep a few values: $p_i(t)$, $g(t)$, $g(t-1)$ and $\boldsymbol{\alpha}(t)$; (2) solve a minimization problem 7 through any efficient numerical method such as the *one-dimensional golden section search* [18]; and (3) perform one step of iteration of Eq. (8). In practice, the length of a time slot may be chosen on the order of a minute.

C. Adjustment of prices

Given the mathematical model of the external world, the new price $p_i(t)$ of a node i should satisfy:

$$p_i^*(t) = \arg\max_{p_i(t)} p_i(t) \sum_{m=1}^{n} \alpha_m(t)\kappa(p_i(t),\tilde{\omega}_i(t),g(t-1)) \quad (9)$$

which is also efficiently solvable by the golden section search method. For more stable behavior of the entire network, we may update system parameters in every time slot, but adjust prices once every few slots.

From the decision problem of the downstream nodes in Eq. (2), one may notice that, the transmission price $p_i(t)$ should remain in the range $\left(0, \frac{\epsilon_{j,1}}{C_{j,D}+\sum_{k\in S_j(t)} b_j^k(t)}\right)$ at least for some node j. Consider the partial derivative of a downstream node j's utility function with respect to $b_j^i(t)$. If $p_i(t) \geq \frac{\epsilon_{j,1}}{C_{j,D}+\sum_{k\in S_j(t)} b_j^k(t)}$, then

$$\begin{aligned}&\frac{\partial u_j(t)}{\partial b_j^i(t)} = \frac{\epsilon_{j,1}}{C_{j,D}+\sum_{k\in S_j(t)\setminus i} b_j^k(t)+b_j^i(t)} - p_i(t) \leq 0 \\ &\forall b_j^i(t) \in [0,\infty)\end{aligned}$$

Therefore, the optimal throughput for node j has to be $b_j^{i*}(t) = 0$. As upstream nodes in a multicast group, their initial prices may be configured based on their own parameters as any values within $\left(0, \frac{\epsilon_{i,1}}{C_{i,D}+\sum_{k\in S_i(t)} b_i^k(t)}\right)$, and then adjusted according to Eq. (9). In addition, to cope with some unavoidable inaccuracy in the identified system model, especially in the initial stage of iterations, we have applied a simple rule to assist the adjustment of prices. We let each node i memorize its estimated revenue $p_i^*(t)\tilde{g}(t)$ for time slot t. By the time the real revenue $p_i^*(t)g(t)$ is available, if $p_i^*(t)g(t) < 0.5p_i^*(t)\tilde{g}(t)$, the new price $p_i^*(t+1)$ is directly set to be $0.5p_i^*(t)$, otherwise, the new price will be derived from Eq. (9) and be smoothed based on its previous value: $p_i^*(t+1) \leftarrow \lambda p_i^*(t+1) + (1-\lambda)p_i^*(t)$.

V. Distributed protocols

The previous two sections have formulated the per-upstream dynamic-price market model based on which a multicast topology is gradually evolved and the bandwidth resource is dynamically allocated. We have also identified the local optimization problems to be solved by individual nodes that induce network-wide optimality. This section further addresses the necessary protocols that facilitate the self-organization, self-healing and self-optimization functionalities.

A. Upstream probing and valuation

In order to maintain the highest level of utility, any node i is allowed to periodically inspect each of its upstream candidates in terms of prices and deliverable QoS levels, by probing and observation. In doing probing, the downstream node sends a *price and bandwidth probing* (PBP) packet to node j, which then responds with its price $p_j(t)$ and relevant bandwidth information within a *price and bandwidth reply* (PBR) packet.

We follow the *receiver-only packet pair* method [19], [20], which is widely adopted in Internet measurement studies. *Four* identical PBR packets are sent back to back to node i, so that node i not only receives the response from node j, but also an estimate on the available bandwidth $B_i^j(t)$ from j to i: each pair of consecutively sent packets gives an estimate, and the three pairs give an average.

For node i, probes regarding the same node j may be carried out once every few minutes. For a current upstream node of i, end-to-end delay and loss rate are observed between two consecutive probes, and the latest values of $d_i^j(t)$ and $l_i^j(t)$ are obtained by smoothing new observations based on their historical values.

1) Negotiations on bandwidth reservation: For any pair of upstream and downstream nodes, given the service price of the upstream node, both nodes would aim to enhance their own utilities by making the optimal amount of bandwidth reserved for the flow between them. Unless their optimal choices happen to coincide, some negotiation procedures are necessary for a stream transfer to be successfully established between the two.

Considering the fact that downstream nodes have the privilege of evaluating and selecting upstream nodes on the market, we continue to assume that upstream nodes are more concerned with attracting downstream nodes for the purpose of improving self-utilities. Hence, they behave less aggressively in specifying the transfer of streams before a downstream node is secured. We propose a negotiation procedure that proceeds as follows.

As a starting point, a downstream node i probes an upstream candidate j by sending the PBP packet. Upon receiving the probe, node j sends back two values in the PBR packet, $b_{\min}^j$ and $b_{\max}^j$, which form its acceptable range of bandwidth reservations for any node that comes in at this moment. The acceptable range can be derived from the decision problem Eq. (3) with $\Delta u_{i,U} \geq 0$.

Once obtaining the PBR, node i knows the price $p_j(t)$, the available bandwidth on the overlay link from node j to itself, as well as the possible amount of bandwidth to be reserved at node j. The best amount of bandwidth reservation is then computed from the utility maximization problem of Eq. (2), with the physical constraints determined as:

$$\begin{cases} b_{i,\min} = b_{\min}^j(t) \\ b_{i,\max} = \min(B_i^j(t), b_{\max}^j) \end{cases}$$

Knowing the optimal throughput and the corresponding utility increment, it is then up to node i to decide whether to have node j as its upstream node, or to adjust the bandwidth reserved at node j. If the application requires a single multicast tree, node i may choose the upstream node with the highest utility increment as its upstream node, if only the utility increment is adequately high (by a factor of at least 1.2 in our simulation studies). Otherwise, node i can maintain a number of best upstream nodes as upstream nodes, which brings the highest utility increments.

2) Resolution of concurrent requests: Since the same upstream node i may be concurrently probed by multiple downstream nodes, by the time downstream j decides to establish a connection with i, other downstream nodes may have already established theirs, thus the connection request from downstream j, which contains the bandwidth reservation $b_{j,D}^{i*}(t)$, may be turned down due to lack of bandwidth. To prevent such conflicts between competing connections, we have devised the following resolution scheme, as is illustrated by Fig. 3.

Suppose the upstream node receives the first probe from a downstream node A, which may be either a potential downstream node or an existing downstream node, at time t_1. It immediately responds and waits for a constant time period T_U, unless A returns a *connection request* (CR) packet or a *bandwidth adjustment* (BA) packet at time $t_2 < (t_1 + T_U)$. Here, both the CR and BA packets contain the optimal bandwidth reservation $b_{j,D}^{i*}(t)$ computed by the downstream node, and are used to establish a new connection and to update the bandwidth reservation, respectively. Once the upstream node performs the corresponding operation, the processing of the first request is accomplished.

Meanwhile, probes that are received from other nodes during (t_1, t_2) or $(t_1, t_1+T_U]$ are queued at the upstream node, and will be processed in a FIFO fashion. On the other hand, a downstream node waits for the upstream node's reply to its probes for a limited time T_D. If the packets have remained in the queue for longer than T_D, or the queue becomes full, the upstream node removes its most out-of-date request from the queue. In practice, we may assume T_D and T_U to be on the order of seconds.

Another type of request that an upstream needs to handle is the *disconnection request* (DR). We assume that the request from a downstream node for tearing down a connection directly indicates that it has stopped paying the upstream node. Thus, an upstream node always responds to such a request immediately, and its processing of probing packets can be preempted by disconnection requests. For ease of reference, the distributed algorithms an overlay node needs to run periodically are summarized in Table I.

TABLE I

ALGORITHMS FOR AN OVERLAY NODE i

Price adjustment as upstream
if it is the beginning of slot $(t+1)$
  if $p_i^*(t)g(t) < 0.5p_i^*(t)\tilde{g}(t)$
    $p_i^*(t+1) \leftarrow 0.5p_i^*(t)$
  else
    derive $p_i^*(t+1)$ from Eq. (9)
    $p_i^*(t+1) \leftarrow \lambda p_i^*(t+1) + (1-\lambda)p_i^*(t)$

Bandwidth reservation negotiation as upstream
if PBP packet received from node j
  derive $b_{i,\min}$ and $b_{i,\max}$ by Eq. (3)
  send 4 PBR packets
  while timer T_u is not expired
    if CR packet received from node j
      connect to node j at $b_{j,D}^{i*}(t)$ if feasible
    if BA packet received from node j
      adjust reservation for node j to $b_{j,D}^{i*}(t)$ if feasible
    if DR packet received from node j
      disconnect from node l

Upstream selection and rate adjustment as downstream
if it is time to update connections
  for each upstream candidate
    probe with PBP
    and compute utility by Eq.(1.1)
  for each upstream candidate
    if its utility is higher than the worst current upstream
      disconnect from the worst current upstream
      connect to the new upstream
  for each current upstream
    adjust bandwidth reservation if needed

Resolving concurrency as upstream
while request queue is not empty
  if the first one is PBP from node j
    Bandwidth reservation negotiation as upstream
  if the head-of-queue request expires or the queue is full
    remove the request at the head of queue

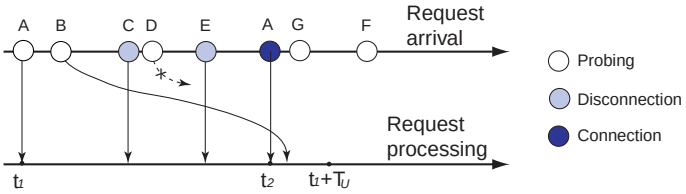

Fig. 3. An example to resolve concurrent connection and disconnection requests. Packets are labeled by their sources (the originating downstream nodes). As the probe from downstream node A is being processed between t_1 and t_2, probing packets from upstream nodes B and D are queued, while disconnection requests from downstream nodes C and E are processed immediately. By time t_2, the probe from downstream D may have been discarded.

VI. EVALUATIONS

The proposed market-based resource allocation mechanism is designed for self-optimization of autonomic service overlay networks, where participants are subject to self-interests and capable of determining their own behavior in the network. Our modeling of node selfishness and the proposed decision making algorithms are meant to emulate the most intelligent and selfish behavior from overlay nodes. In this section, we evaluate the performance of the designed mechanisms via simulations, based on the overlay multicast streaming application as an example. For simplicity of implementation, we consider performance optimality in a network in terms of *average end-to-end throughput* and *average end-to-end delay* from the source to each receiver node, and do not consider packet loss.

A. Simulation methodology

Our backbone connections were generated by the BRITE universal topology generator [21], with 512 routers and 1024 backbone edges, whose capacities are distributed between 10 Mbps and 1024 Mbps, with a heavy-tailed distribution. Overlay nodes are randomly connected to backbone routers through access links, whose capacities are exponentially distributed with an average of 10 Mbps. All experiments are executed for 1000 time slots.

We emulate background traffic as random noise independently deployed on each link, the magnitude of which is uniformly distributed from 0 to a small value, *e.g.*, 5% of the link capacity. The *shortest-widest* routing algorithm is adopted to generate QoS-aware routes, and the transmission delay along a path is approximated as the number of physical links that the overlay path consists of.

All our simulation experiments were performed with a simple example: forming a single multicast tree. Every node maintains up to 10 upstream candidates on the network, with candidates randomly assigned initially. The simulation program guarantees that candidates are properly maintained by each node, so that no loop is caused no matter which candidate a node connects to. Nodes probe for their neighbors' most up-to-date information every 10 time slots, and probes are sent asynchronously at nodes' own paces. Further, we take a simplifying assumption that all the downstream nodes configure their utility functions (Eq. (1.1) and Eq. (1.2)) in the same way:

$$\begin{array}{ll} \epsilon_{i,1} = 2C_i & \epsilon_{i,2} = 1 \\ \epsilon_{i,3} = 0 & \epsilon_{i,4} = 0.5C_i \\ C_{i,D} = C_{i,U} = C_i & D_i = 15 \end{array}$$

Recall that C_i is the local network bandwidth capacity of node i. Finally, each node is issued with an initial fund of 500 at bootstrapping.

B. Mechanisms in our comparison

We compare the resource allocation outcomes of a few self-optimization mechanisms that assume different behavior of overlay nodes. The baseline mechanism is Narada [3], a well-known multicast tree formation protocol. To ensure a fair comparison, we simulate and compare two variations of the Narada protocol.

- *Augmented Narada*, which is referred to as *ANarada* in our simulation results. In the original Narada protocol, every receiver in the tree receives data at the same rate, and the tree is formed using a minimum spanning tree algorithm that optimizes towards delay. We derive *ANarada* using an all-shortest-widest paths algorithm instead of the minimum spanning tree algorithm; optimizing first bandwidth (*i.e.,* width) and then delay (*i.e.,* distance). Moreover, *ANarada* allows nodes to receive flows as fast as possible at heterogeneous rates, as long as a node does not receive at a higher rate than its upstream node in the tree.
- *Selfish augmented Narada*, which is a special case to *ANarada*, in the sense that upstream nodes (except the source) may deviate from the transmission rates dictated by the *ANarada* algorithm with probability τ, and only transmit at half of their dictated throughput. It is referred to as *SANarada* in the simulation results.

Aside from our proposed mechanism (referred to as *Market* in the simulation results), we also consider its two variants that form topology and allocate bandwidth based on individual decisions:

- *MarketNN*, which is the same as *Market* except no bandwidth negotiation is involved. It emulates the situation that downstream nodes alone determine the optimal throughput and upstream nodes always agree to reserve such amounts. Thus, it represents the optimal deliverable QoS levels subject to the economic constraints of downstream nodes.
- *Market0*, which is the same as *MarketNN* except that all prices are kept as zero. In this case, downstream nodes determine the optimal throughput solely based on their empirical utilities, and upstream nodes always satisfy their requests. This situation actually approximates the optimal deliverable QoS levels without economic constraints, with optimality judged by the empirical utility functions of downstream nodes.

C. Comparisons in the steady state

In the steady state, we study the topology formation and bandwidth allocation capabilities of the five different mechanisms previously presented, by comparing the average end-to-end throughput and the average end-to-end delay from the source to each receiver. Overlay nodes sequentially join the multicast group at randomly selected times during the first half of the simulation time.

1) Average throughput: Fig. 4 shows the average throughput for all the existent receivers over time, where 32 nodes eventually join the multicast group. All three individual decision based mechanisms, *Market*, *MarketNN* and *Market0*, outperform *ANarada* and *SANarada*, due to the following reason. In the former three mechanisms, once nodes choose their best upstream nodes, the corresponding bandwidth shares are reserved, other nodes need to find other suitable upstream nodes elsewhere in the network. However, with *ANarada* and *SANarada*, nodes can more easily choose the same upstream node. Whenever there is any change to its downstream nodes, the upstream node reallocates its bandwidth resource based on the maximum capacity on the underlying paths from itself to each downstream node. Hence, their average throughput is relatively lower.

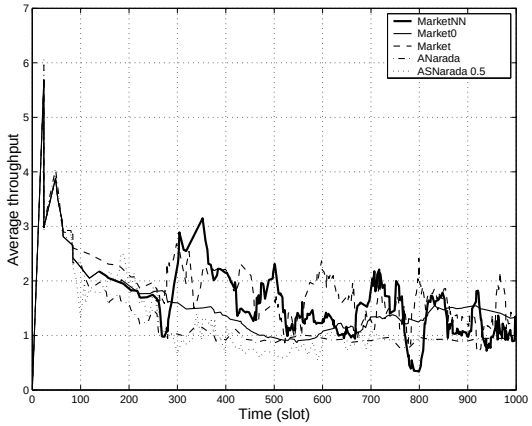

Fig. 4. End-to-end throughput averaged over all existent receivers: a 32-node network.

Our *Market* mechanism achieves slightly lower throughput than that of *MarketNN* and *Market0*, due to the existence of prices in nodes' utility functions. Its performance appears more stable than *MarketNN*, because of bandwidth negotiation.

In all the experiments, we chose the selfishness probability τ of *SANarada* to be 0.5; for clarity, we have also compared the performance difference between *SANarada* settings with different τ values. As Fig. 5 has shown, nodes with selfish probability $\tau > 0.5$ lead to quite unacceptable performance in the multicast tree.

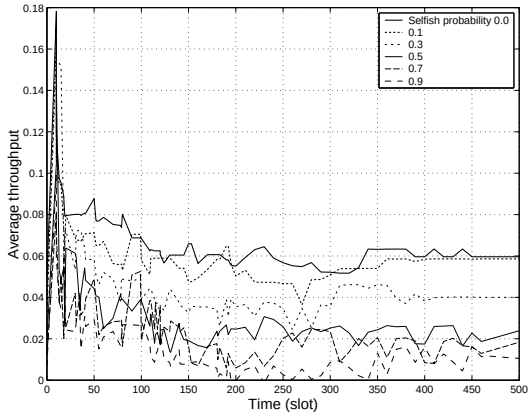

Fig. 5. Average throughput under the SANarada mechanism.

With 64 overlay nodes joining the multicast group, Fig. 6 shows similar throughput comparison, and it is more evident

that the average throughput achieved by *Market* lies between those of *MarketNN* and *Market0* and that of *ANarada*.

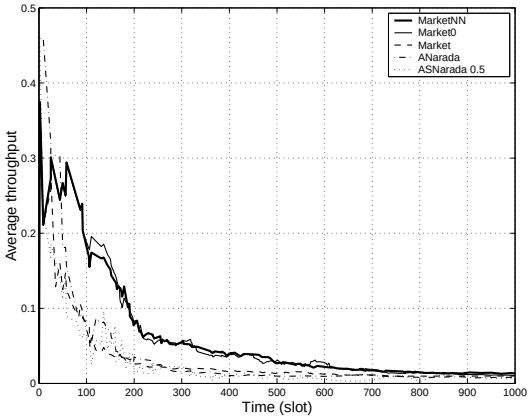

Fig. 6. End-to-end throughput averaged over all existing receivers: a 64-node network.

2) Average delay: Fig. 7 illustrates the average end-to-end delays for the five mechanisms with 32 overlay nodes in the multicast group. All the mechanisms behave similarly in the small network, except that *SANarada* shows much lower delay. The reason is that, with probability τ, upstream nodes contribute only half of the transmission capacity to the network, downstream nodes tend to directly connect to the source node which contributes to its full capacity.

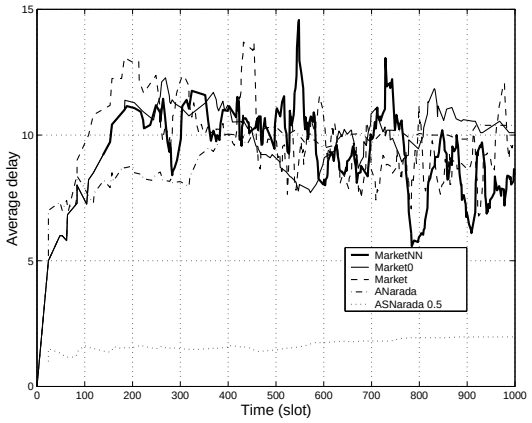

Fig. 7. End-to-end delay averaged over all existent receivers: a 32 node network.

Results from a larger multicast group containing 64 nodes (Fig. 8) show that the three individual decision based mechanisms actually lead to lower average delay than *ANarada*, because every node tends to choose upstream nodes closer to the source to reduce its end-to-end delay. Recall that the *ANarada* mechanism optimizes throughput then delay through cooperative algorithms, the utility-based mechanisms have achieved higher average throughput and lower delay through independent and non-cooperative adjustments.

Further, based on both Fig. 6 and Fig. 8, we may infer that, in comparison to *MarketNN* and *Market0*, the *Market* mechanism tends to form shorter multicast trees. When the local network bandwidth is abundant, nodes closer to the root tend to have lower prices to attract more downstream nodes.

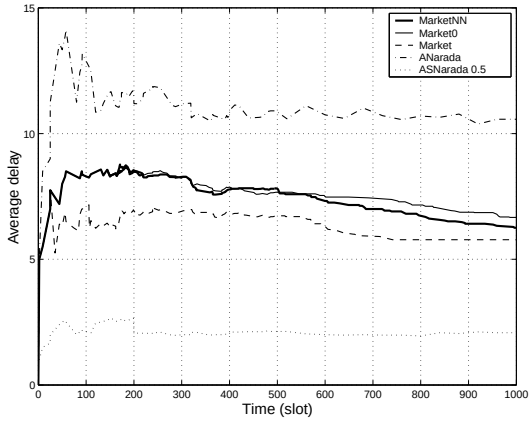

Fig. 8. End-to-end delay averaged over all overlay links: a 64 node network.

D. Comparison in dynamics

In the following experiments, 128 overlay nodes sequentially join the multicast group during the first 500 time slots at randomly selected times, then start leaving the group at randomly selected times from the 750th time slot. We study the reactions of these resource allocation mechanisms to such node dynamics.

As shown in Fig. 9, during the node joining phase, more bandwidth resource is being provided to the overlay network. The three mechanisms, *Market*, *MarketNN* and *Market0*, evidently outperform the rest two cases, due to the same reason we have explained for the steady-state difference between them: under the three mechanisms, nodes asynchronously choose their most preferable upstream nodes, and reserve their bandwidth, while in *ANarada* and *SANarada*, nodes might choose the same upstream node and share the common link among them.

The results also show a clear trend that, as nodes join in the network, average throughput will be gradually reduced under all five mechanisms, due to the decrease of the available bandwidth. The trend is more obvious in the *Market*, *MarketNN*, and *Market0* mechanisms. This is because that, as every node tries to connect upstream nodes closer to the source, bandwidth competition will progressively intensify in close proximity to the root of the tree.

During the node departure phase (an enlarged figure is shown in Fig. 10), all the mechanisms show a rising trend in their average throughput, due to the increased availability of bandwidth resource on the network. The *Market*, *MarketNN* and *Market0* mechanisms still outperform the remaining two, even with nodes individually deciding on their incoming and outgoing connections. With respect to the delay metric (shown in Fig. 11), all the mechanisms show some decreasing trend, since nodes tend to form shorter trees as other nodes are leaving. The three individual decision based mechanisms still lead to lower delays than *ANarada*.

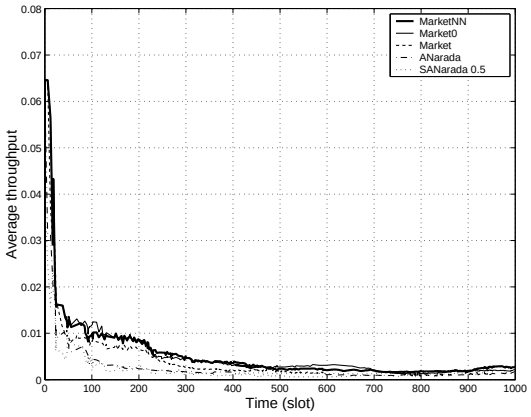

Fig. 9. End-to-end throughput during node dynamics: a 128-node network.

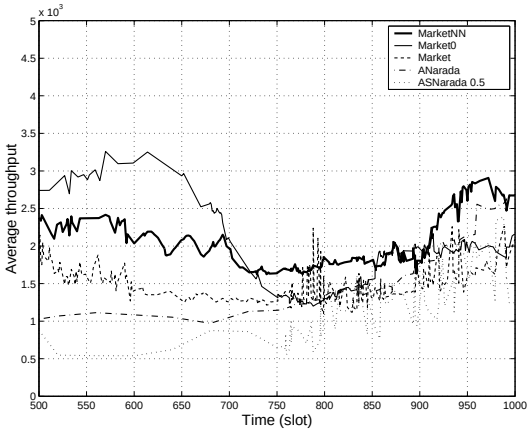

Fig. 10. End-to-end throughput during node departures: a 128-node network.

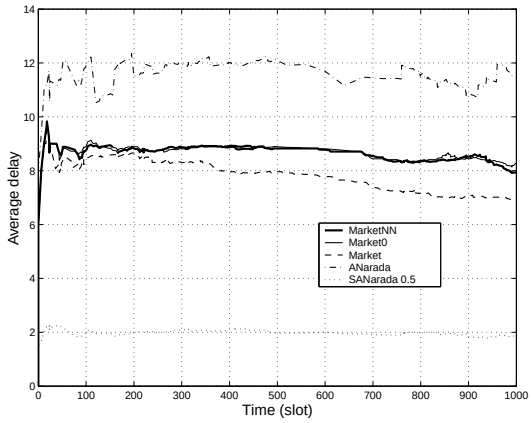

Fig. 11. End-to-end delay during node arrivals and departures: a 128-node network.

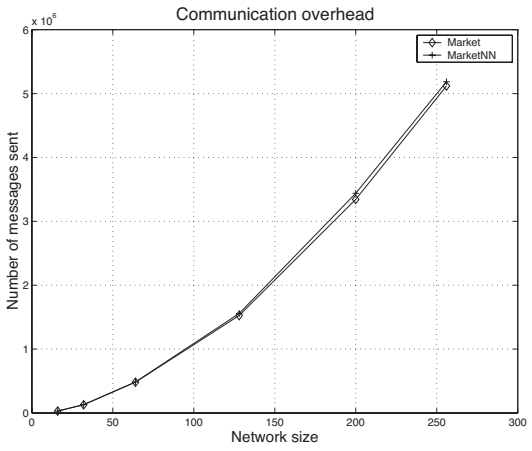

Fig. 12. Communication overhead.

E. Communication overhead

We measure the extra communication overhead brought by the *Market* mechanism by counting the number of requests sent between nodes, regarding probing, probing reply, connection, disconnection and rate adjustment. Since every node only looks at a limited number of upstream (candidate) nodes, the number of messages caused by the mechanism increases linearly with the network size (shown in Fig. 12), which is quite acceptable even for a large network.

Finally, we have investigated the computational efficiency of our system identification methods used in the pricing procedure. In our simulation experiments, nodes recompute their prices every 50 time slots, at their own paces. Fig. 13 depicts the identified model of $g(t)$ with respect to $p(t)$ and $g(t-1)$, for an arbitrary overlay node, in which $\tilde{\omega}(t)$ is kept as a fixed small value. As we can see, the iterations converge rapidly to the steady state after only 5 iterations.

VII. Related Work

There already exists a significant body of research work regarding self-organization, self-healing and self-optimization in overlay networks. Two categories exist: *structured* (*e.g.* in the CAN [22], Pastry [2], and Chord [1]) and *unstructured* (*e.g.*, Gnutella [23], Freenet [24]). They have been commonly adopted to define the basic self-organization and self-healing behavior of overlay nodes. Many distributed algorithms and protocols have been proposed for nodes to manage, maintain and allocate shared resources, so as to self-optimize the network towards global optimality.

Especially, quite a few proposals exist for forming multicast topologies, and for allocating transmission bandwidth under certain optimization objectives. For example, Narada [3] constructs the multicast tree by first building an efficient mesh, and then constructs a minimum spanning tree out of the mesh to minimize the end-to-end delay. SplitStream [11] establishes a forest of multicast trees from a single source, for the purpose of maximizing throughput; and Kostic *et al.* [12] has proposed to construct an overlay mesh of concurrent data dissemination connections, each sending a disjoint set of data, to significantly improve throughput. None of the aforementioned work, however, considers node selfishness, which potentially exists and actually hinders the expected self-optimization mechanisms.

To address node selfishness in overlay networks, some theoretical studies have employed game theory to model overlay nodes as game players, with conflicting interests regarding shared resources [4], [5], [6]. To manipulate their

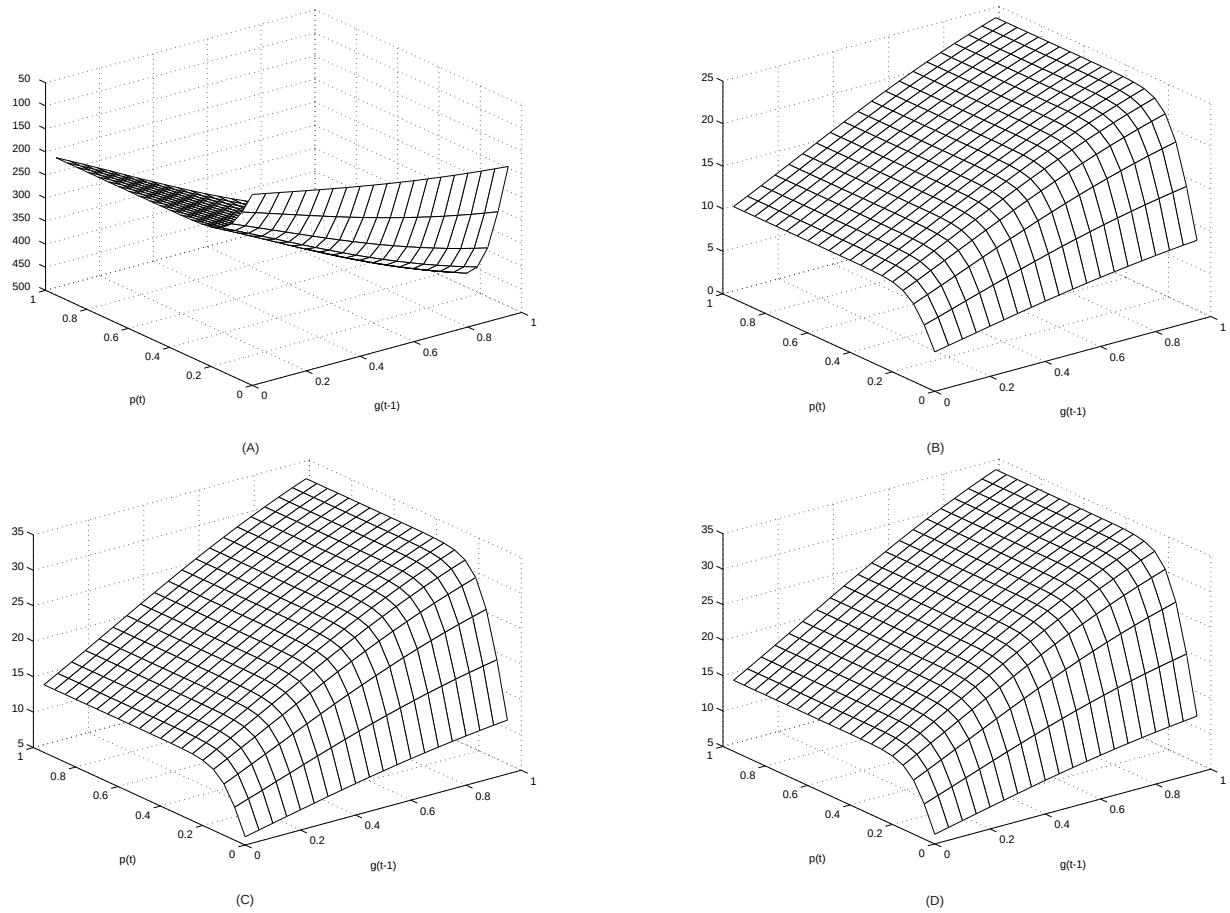

Fig. 13. The identified system model $g(t+1)$. (A) After 1 iteration; (B) After 2 iterations; (C) After 5 iterations; (D) After 10 iterations.

self-interests and lead the system into a desirable equilibrium state, *mechanism design* [25], [26], [27], [28] has been introduced into networking problems. However, due to the inherent limitations of relevant theories, existing literature normally has to unrealistically assume that some global knowledge about the network and other nodes is accessible by the overlay nodes. This may include concrete forms of the utility functions of other nodes, and their discussions have mostly focused on the *steady state properties* of node interactions.

A few other work has also proposed to regulate the behavior of selfish nodes within distributed pricing. Qiu *et al.* [7] (in the context of mobile ad hoc networks) and Cui *et al.* [8] (in the context of overlay multicast) have discussed distributed pricing models based on the classical *social welfare maximization* model, first proposed by Kelly *et al.* [10]. However, the methodologies they have taken are more appropriate for situations where nodes have direct control of the resources to be allocated among selfish users, other than most of the Internet-based service overlay applications, where end systems do not even have any clear view of resources in the physical network [29].

The original contributions of this paper, differing from previous work, are two-fold. First, we present a market-based framework for encouraging selfish and intelligent behavior from overlay nodes, and for decentralizing the resource allocation problem into local decision problems. Second, we propose mathematical models that emulate the selfishness of overlay nodes, captured by the utility functions Eq. (1.1) and (1.2), and a novel solution — based on optimal control and system identification — to the local maximization problems. They have collectively implemented self-optimization in service overlay networks that consist of selfish and intelligent nodes.

VIII. Conclusions

In this paper, we have studied the self-optimization problem for autonomic service overlay networks consisting of selfish and intelligent nodes. We have proposed an *intelligent market model* that manages resource provisioning and allocation, with a goal of maximizing the sum of node utility. Reasonable utility functions have been designed to account for the selfishness of nodes in the context of a multicast streaming application, and appropriate solutions have been proposed for the local optimization problems. In particular, we have adopted a system control point of view, and provided an optimal pricing solution based on system identification techniques.

Under the proposed market model, prices act as control forces in a selfish overlay network: downstream nodes adjust their demands according to changing prices and upstream nodes adjust their prices based on their utilities received on the market. Through extensive simulation studies, we have shown that with the proposed market-based incentive mechanism, even when all nodes behave selfishly towards their own utility

maximization goals, the resulting multicast group can still provide QoS metrics comparable to or better than well-known approximations to optimal outcomes. The intelligent market model, together with the decision making algorithms, may serve as a general incentive-based self-optimization scheme, for any autonomic systems that are built on selfish nodes and provide more than one producer choices for each consumer.

REFERENCES

[1] I. Stoica, R. Morris, D. Liben-Nowell, M. R. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, Febuary 2003.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Systems," in *Proc. IFIP/ACM Middleware 2001*, November 2001.

[3] Y. Chu, Sanjay. G. Rao, S. Seshan, and H. Zhang, "A Case for End System Multicast," in *Proc. pf ACM SIGMETRICS*, 2000.

[4] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for Cooperation in Peer-to-Peer Networks," in *Workshop on Economics of Peer-to-Peer Systems*, June 2003.

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

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

[7] Y. Qiu and P. Marbach, "Bandwidth Allocation in Ad Hoc Networks: A Price-Based Approach," in *Proc. of IEEE INFOCOM*, 2003.

[8] Y. Cui, Y. Xue, and K. Nahrstedt, "Optimal Resource Allocation in Overlay Multicast," in *Proc. of the 11th IEEE International Conference on Network Protocols*, 2003.

[9] Scott Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, 1995.

[10] F. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[11] M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-Bandwidth Multicast in Cooperative Environments," in *Proc. of the 20th ACM Symposium on Operating Systems Principles (SOSP)*, 2003.

[12] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh," in *Proc. of the 20th ACM Symposium on Operating Systems Princples (SOSP)*, 2003.

[13] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and Jr. J. W. O'Toole, "Overcast: Reliable Multicasting with an Overlay Network," in *Proc. of the Fourth Symposium on Operating System Design and Implementation (OSDI)*, 2000.

[14] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*, Oxford University Press, 1995.

[15] C. Courcoubetis, G. D. Stamoulis, C. Manolakis, and F. P. Kelly, "An intelligent agent for optimizing QoS-for-money in priced ABR connections," *Telecommunications Systems, Special Issue on Network Economics*, 2000.

[16] L. Ljung, *System Identification*, Prentice Hall, 1999.

[17] K. J. Âström and B. Wittenmark, *Adaptive Controli*, Addison-Wesley, second edition, 1995.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, second edition, 2002.

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

[20] K. Lai and M. Baker, "Nettimer: A Tool for Measuring Bottleneck Link Bandwidth," in *Proc. of the USENIX Symposium on Internet Technologies and Systems*, 2001.

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

[22] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proc. of ACM SIGCOMM*, 2001.

[23] Clip2 Distributed Search Services, "The Gnutella Protocol Specification v0.4," available on line at "http://dss.clip2.com".

[24] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," in *Proc. of Workshop on Design Issues in Anonymity and Unobservability*, 2000.

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

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

[27] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based Mechanism for Lowest-Cost Routing," in *Proc. of ACM Symposium on Principles of Distributed Computing*, 2002.

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

[29] R. Siamwalla, R. Sharma, and S. Keshav, "Discovering Internet Topology," Tech. Rep., Cornell University, 1999.

Weihong Wang. Weihong Wang received her B.A.Sc. and M.A.Sc. degrees in Electrical Engineering from Tsinghua University, China, in 1998 and 2001, respectively. She is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering, University of Toronto, Canada. Her research interests include the application of microeconomics and game theory on agent-based systems, and decision making strategies based on system control and artificial intelligence.

Baochun Li. Baochun Li received his B.Engr. degree in 1995 from Department of Computer Science and Technology, Tsinghua University, China, and his M.S. and Ph.D. degrees in 1997 and 2000 from the Department of Computer Science, University of Illinois at Urbana-Champaign. Since 2000, he has been with the Department of Electrical and Computer Engineering at the University of Toronto, where he is currently an Associate Professor. He holds the Nortel Networks Junior Chair in Network Architecture and Services since October 2003, and the Bell University Laboratories Chair in Computer Engineering since July 2005. In 2000, he was the recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems. His research interests include application-level Quality of Service provisioning, wireless and overlay networks. He is a senior member of IEEE, and a member of ACM. His email address is `bli@eecg.toronto.edu`.