



# Timing Matters: Online Dynamics in Broadcast Games

Shuchi Chawla<sup>1</sup>, Joseph (Seffi) Naor<sup>2</sup>, Debmalya Panigrahi<sup>3</sup>(✉), Mohit Singh<sup>4</sup>,  
and Seeun William Umboh<sup>5</sup>

<sup>1</sup> Computer Sciences Department, University of Wisconsin - Madison, Madison, USA  
shuchi@cs.wisc.edu

<sup>2</sup> Department of Computer Science, Technion, Haifa, Israel  
naor@cs.technion.ac.il

<sup>3</sup> Department of Computer Science, Duke University, Durham, USA  
debmalya@cs.duke.edu

<sup>4</sup> H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute  
of Technology, Atlanta, USA  
mohitsinghr@gmail.com

<sup>5</sup> School of Information Technologies, The University of Sydney, Sydney, Australia  
william.umboh@sydney.edu.au

**Abstract.** This paper studies the equilibrium states that can be reached in a network design game via *natural* game dynamics. First, we show that an arbitrarily interleaved sequence of arrivals and departures of players can lead to a polynomially inefficient solution at equilibrium. This implies that the central controller must have some control over the timing of agent arrivals and departures in order to ensure efficiency of the system at equilibrium. Indeed, we give a complementary result showing that if the central controller is allowed to restore equilibrium after every set of arrivals/departures via *improving moves*, the eventual equilibrium states reached have exponentially better efficiency.

## 1 Introduction

In multi-agent systems where different agents have competing objectives, it is well-known that selfish behavior leads to suboptimal system performance at *equilibrium*. The *Price of Anarchy* (POA) and the *Price of Stability* (PoS), which respectively correspond to the worst and best equilibrium states, are widely used in the literature to quantify this suboptimality relative to an optimal solution designed by a central authority. If these two measures are close to each other, they

---

Part of this work was done when all the authors were visiting Microsoft Research - Redmond. Partial support for this work was provided by the following grants: S. Chawla from NSF grants CCF-1101429 and CCF-1320854; S. Naor from ISF grant 1585/15 and BSF grant 2014414; D. Panigrahi from NSF grants CCF-1527084 and CCF-1535972, an NSF CAREER Award CCF-1750140, and faculty research awards from Google and Yahoo; M. Singh from NSF grant CCF-1717947; S. Umboh from ERC consolidator grant 617951 and NSF grant CCF-1320854.

© Springer Nature Switzerland AG 2018

G. Christodoulou and T. Harks (Eds.): WINE 2018, LNCS 11316, pp. 80–95, 2018.

[https://doi.org/10.1007/978-3-030-04612-5\\_6](https://doi.org/10.1007/978-3-030-04612-5_6)

provide a satisfactory understanding of the quality of stable states the system is expected to reach. However when these measures differ significantly, the system can exhibit multiple equilibria with highly varying performance. But, which of these equilibria can be achieved in actual game dynamics? More generally, what is the minimal guidance by a central authority that can guarantee near-optimal system performance in equilibrium?

In this paper, we study these questions in the context of a game that exhibits a particularly rich set of equilibria, namely the *broadcast game*. A broadcast game is defined on a rooted undirected graph with costs on edges. Every vertex in the graph has an agent whose goal is to select a routing path to the root that minimizes her own cost. The cost of every edge is shared equally among all agents using it, and the cost of an agent is the sum of her *cost shares* along the edges in her path to the root. The system is in Nash equilibrium (or NE) if no agent can lower her own cost by unilaterally changing her routing path. The cost of an equilibrium is the total cost of all edges used by at least one agent. The quality of equilibria is measured with respect to the social optimum, which for broadcast games is the minimum spanning tree (MST) of the graph.

Broadcast games are a kind of potential games and the existence of NE in any instance can be proved through a potential function argument, originally given by Rosenthal [34] (see also Monderer and Shapley [30]). Anshelevich et al. [4] observed that different NE in broadcast games can exhibit vastly different performance: the POA can be as large as  $\Omega(n)$  whereas the POS (a concept they introduced to show this gap) is bounded by  $O(\log n)$ ; here  $n$  denotes the number of vertices in the graph.<sup>1</sup> A long line of work (e.g., [10, 19, 27, 28]) subsequently improved the POS bound to  $O(1)$ .

Given this divergence of bounds, Chekuri et al. [13] posed the question of analyzing the quality of equilibria that are actually *reachable* via natural dynamics—a sequence of single agent moves where the moving agent always chooses a new path that strictly decreases her cost relative to her current path. We call such moves “improving moves” or “best response moves”, depending on whether they merely lower the agent’s cost or are optimal for the agent given the current state of the system. It follows from the potential function argument of Rosenthal [34] that *any* such sequence of moves will eventually converge to NE.

Chekuri et al. [13] considered the following *restricted* two-stage process: in the first stage, starting with an empty graph, agents arrive sequentially in arbitrary order and choose their respective best response paths upon arrival; in the second stage, agents make improving moves<sup>2</sup> in arbitrary order until they reach equilibrium. They showed that the equilibria reachable through this process have a cost of  $O(\sqrt{n} \log^2 n)$  times the MST, a significant improvement over the POA bound. This bound was subsequently improved to  $O(\log^3 n)$  for the same two-stage process by Charikar et al. [11].

<sup>1</sup> The full version of this paper [12] provides examples illustrating these bounds.

<sup>2</sup> Observe that when an agent arrives or makes an improving move, paths of other agents may become suboptimal for them.

*The Dynamic Price of Stability.* These previous works motivate extending the notion of the price of stability to online dynamics. In the static (or “one shot”) version of our problem, in which players are initially in an empty configuration, the central planner can force the players into any configuration, in particular the one realizing the price of stability. In the dynamic case, however, the central planner cannot do so since some players have already chosen a route. Thus, the central planner has to offer existing players a better strategy, so as to incentivize changes. Informally, the *dynamic price of stability* is the cost of a solution in equilibrium resulting from online dynamics, while allowing for algorithmic intervention by the central planner. The notion of dynamic price of stability can be applied to any game in which one needs to characterize which equilibria can be reached via online dynamics, while minimizing the power of intervention of the central planner. It would be very interesting to find further applications of this new notion.

One way to restate Charikar et al.’s result is that the dynamic POS is polylogarithmic when all arrivals happen before any improving moves. But, what if some agents make improving moves before all of the other agents have arrived, i.e., the sequence of improving moves is interleaved with arrivals? Unfortunately, the analyses presented in [13] and [11] strongly build on the fact that all agents arrive upfront and remain in the system thereafter, and agents must wait for everyone to arrive before making any changes to their strategies. Charikar et al. posed the question of analyzing dynamics in which arrivals and improving moves are interleaved as a “tantilizing and difficult” open problem. In the decade following their work, in spite of tremendous progress in POS bounds for broadcast games, no progress has been made on understanding more general game dynamics.<sup>3</sup>

*More General Dynamics.* Since the work of Chekuri et al., our work is the first to study more general dynamics of the broadcast game. We consider two kinds of extensions to the two-stage process. First, we consider systems with churn where agents arrive as well as depart over time. Second, we allow multiple interleaved stages of arrivals, departures, and improving moves. Our first result shows that if we make a minimal change to the two-stage dynamics studied above, namely adding departures to the first stage, then it is possible to reach an equilibrium that is polynomial (in  $n$ ) worse than the social optimum, placing it in the same regime as the POA bound. To the best of our knowledge, this is the first polynomial lower bound for *any* game dynamics for broadcast games.

**Theorem 1.** *For any large enough integer  $n$ , there exists an instance of the broadcast game with  $n$  vertices and a sequence of arrivals and departures that terminates in an NE of cost  $\Omega(n^{1/3})$  times that of the minimum spanning tree on all the vertices.*

<sup>3</sup> Charikar et al. also studied a variant where arrivals happen in uniformly random order and are interleaved with adversarially ordered best response moves. For this setting, they were able to prove an upper bound of  $O(\sqrt{n} \text{ polylog } n)$  on the quality of the equilibria reached, but did not present any lower bounds.

It is important to observe that since we allow departures, not all vertices have agents at the end of the game. This creates two candidates for OPT: the optimal Steiner tree on the remaining agents, or the MST on all vertices.<sup>4</sup> The former leads to trivial and uninteresting lower bounds (see the full version); so, we use the MST as OPT in this paper. This choice of a weaker optimum makes for a stronger lower bound.

*The Power of Intervention.* Given the above lower bound, a natural question is whether some limited intervention from a central planner can lead to a better outcome for the game. At one extreme, if the central planner is allowed to suggest a strategy to every player simultaneously, then any NE, in particular the best one corresponding to the POS of  $O(1)$ , can be achieved. This level of control is unrealistic. A more reasonable level of control is for the central planner to suggest improving moves to players one by one; importantly, any such move should lower the corresponding agent's current cost share, otherwise the player has no incentive to follow the planner's suggestion.

What about the timing of such interventions? As our lower bound demonstrates, if the timing of interventions is completely adversarial, in particular if no interventions are allowed during the initial arrival/departure phase, the system can end up in a poor NE. To get around this lower bound, we consider dynamics where the central planner is allowed to make a series of improving moves after every adversarial arrival/departure. Observe that the sequence of arrivals and departures can still be ordered adversarially, and indeed can depend on the previous algorithmic interventions. We call such dynamics *equilibrium-preserving* (EQ-P) dynamics because the central planner restores the system to a good equilibrium after every adversarial arrival/departure.

Specifically, the EQ-P dynamics starts from an empty configuration and continues in epochs. At the beginning of each epoch the system is at equilibrium. The epoch begins with an arrival or departure, followed by a series of improving moves to restore equilibrium. Once equilibrium is restored, the epoch ends. Our analysis, in fact, allows for multiple simultaneous arrivals<sup>5</sup> at the beginning of an epoch, and multiple departures at any point of time during the epoch. Formally, we define three different kinds of moves within an epoch:

1. (**Arrivals.**) A set of new players arrive and each picks a best response path with respect to the configuration reached at the end of the previous epoch. (The choice of the set of arrivals is adversarial.)
2. (**Departures.**) A set of players departs the system. (Choice of departing players is adversarial.)

<sup>4</sup> Another bound is the optimal Steiner tree on all vertices for which an agent arrived at some point in the dynamics. Since we can assume metric costs, we can restrict our attention to these vertices and then MST cost is within a factor of two of the cost of optimal Steiner tree.

<sup>5</sup> Note that although arrivals within a single epoch are simultaneous in that every arriving player picks a best response path with respect to the equilibrium state at the beginning of the epoch, arrivals in different epochs are sequential. In this sense our model captures sequential arrivals with interleaved improving moves.

3. (**Equilibrium Restoration.**) The central authority offers players strategies that can improve their (shared) connection costs. This step continues till equilibrium is restored to the system.

Our second result shows that this limited level of central intervention is sufficient to guarantee a NE with exponentially better performance:

**Theorem 2.** *Every instance of the broadcast game using EQ-P dynamics converges to an NE of cost  $O(\log n)$  times that of the minimum spanning tree on all the vertices.*

Observe that, as for our lower bound result, we compare the performance of EQ-P dynamics to the MST on all vertices<sup>6</sup> and not the optimal Steiner tree on the vertices that remain in the system. The two benchmarks are identical when there are no departures but the MST benchmark can potentially be much weaker when there are many departures. However, as mentioned earlier, the Steiner tree benchmark is not interesting because it leads to trivial polynomial lower bounds (see the full version [12]). Furthermore that the guarantee provided by the above theorem holds at the end of *every* epoch as compared against the MST over vertices that have arrived up to the end of that epoch, not including future arrivals. A natural open question is whether a polylogarithmic dynamic POS can be achieved through less algorithmic intervention relative to EQ-P dynamics, for example, by allowing players to make best response moves instead of improving moves.

*Technical Challenges.* The broadcast game exhibits a rich set of equilibria and a far richer set of intermediate states of the system. For example, whereas the set of agent strategies (paths) in any equilibrium of the game always forms a tree,<sup>7</sup> intermediate states, even those reached by a series of best response moves, can contain a complex structure of interconnected cycles. A major impediment to analyzing dynamics is that it is extremely challenging to maintain *any* structural invariant on intermediate states. Our work overcomes this challenge by algorithmically maintaining such a structural invariant. Whenever the structural invariant is broken by arrivals or departures, we restore it algorithmically. Importantly, we show that this can always be achieved through a sequence of *improving* moves.

Our structural invariant is a charging of the cost of a state (i.e. collection of paths) against a family of partitions of the underlying graph. Each partition corresponds to a solution to the dual of the standard MST linear program. As such, our charging scheme can be interpreted as a *dual fitting* approach. One challenge in carrying out this approach is that as agents arrive and leave, our analysis must allow for the dual to become grossly infeasible at intermediate states, which in turn results in very expensive intermediate (non-equilibrium)

<sup>6</sup> Because of this comparison against the MST, we prefer the term “broadcast game” for this setting, rather than the “multicast game”.

<sup>7</sup> The existence of a cycle would imply that one of the agents can improve her cost share by switching to a different path and the current state is not an equilibrium.

states. At the crux of our argument is a careful construction of improving moves that ensures that the system cycles between a small set of states of which the stable ones correspond to feasible duals.

*Related Work.* We have already mentioned the long line of work on improving PoS bounds for broadcast games [4, 10, 19, 27], and the game dynamics studied by Chekuri et al. [13] and Charikar et al. [11]. A different approach was taken by Balcan et al. [6], who considered the problem of influencing the dynamics of broadcast games so as to achieve socially efficient equilibria. In their model, players use expert learning, choosing between a best response expert and a central authority expert suggesting (near-)optimal global behavior. Broadcast games belong to a broader class called network design games (see, e.g., [2, 4, 9, 10, 14, 15, 18, 20, 26, 28]), which in turn, are a special case of the widely studied congestion and potential games (see, e.g., [1, 7, 17, 24, 29, 30, 33–35]).

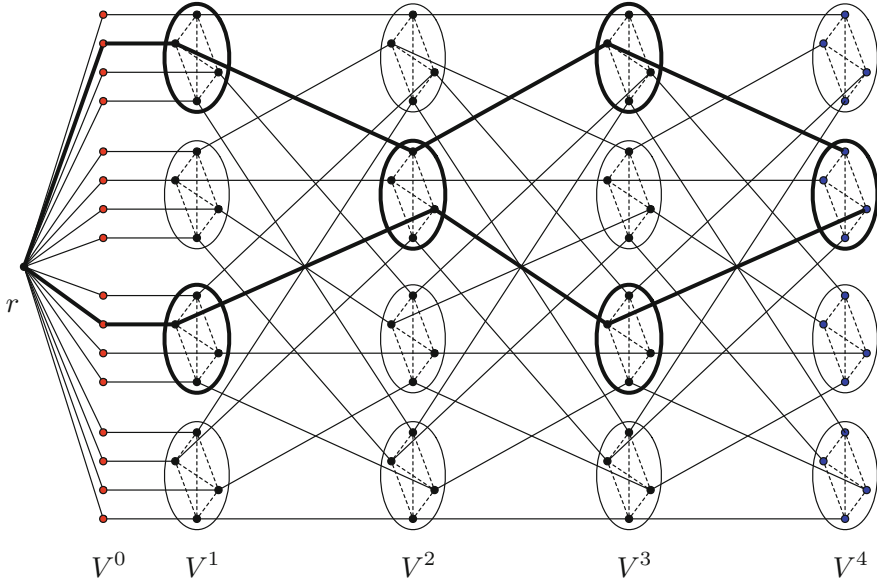
The analysis of game dynamics in this paper crucially relies on the construction of a hierarchical family of multiple dual solutions. This method of analysis has been highly influential in designing online algorithms for network design problems. Implicit use of this method dates back to the work of Imase and Waxman [25] on online Steiner trees and a subsequent line of work of Awerbuch et al. [5], Berman and Coulston [8], Naor et al. [31]. More recently, this method has been explicitly employed in solving a range of node and edge-weighted Steiner network design problems in the online setting [3, 16, 21–23, 32]. In terms of the exact techniques, perhaps the closest to our work is that of Umboh [36], who uses hierarchical tree embeddings to analyze greedy-like online algorithms for network design problems. In contrast to these applications in competitive analysis where decisions are irrevocable, game dynamics allows temporary overcharging of dual solutions, which we crucially use in this paper.

*Organization of the Paper.* We present a proof of our lower bound (Theorem 1) in Sect. 2, and a proof of our upper bound (Theorem 2) in Sect. 3. Due to lack of space, we defer most of the proofs to the full version [12].

## 2 Lower Bound

In this section, we will show that if arrivals and departures are allowed at non equilibrium states, then no dynamics can lead to a good equilibrium (Theorem 1).

We construct a family of lower bound instances parameterized by an integer  $m \geq 1$ . The  $m$ th instance uses the metric induced by weighted graph  $G_m$  (see Fig. 1). The vertex set of this graph consists of a root  $r$  and  $m + 1$  layers  $V^0, \dots, V^m$ . For  $1 \leq i \leq m$ , layer  $V^i$  consists of  $m$  clusters  $C_1^i, \dots, C_m^i$ , each of which is a clique over  $m$  vertices. We use  $v_{j,k}^i$  to denote the  $k$ -th vertex of  $C_j^i$ ; recall that each of  $i, j$ , and  $k$  take on integral values in  $[m]$ . Layer  $V^0$  also consists of  $m^2$  vertices, which are labeled  $v_{j,k}^0$  for  $j, k \in [m]$ , but there are no edges between these vertices. The vertices of  $V^m$  are called *end* vertices,



**Fig. 1.** Example for  $m = 4$ . Auxiliary vertices are in red, end vertices are in blue. Ovals represent clusters. Intra-cluster edges are shown as dashed edges. The two bold paths starting from the same cluster successively diverge into different clusters and converge into the same cluster on their way to the root.

and those of  $V^0$  are called *auxiliary* vertices. Observe that the graph  $G_m$  has  $n = m^2(m + 1) + 1$  vertices in all.

Next, we describe the edges. Each pair of vertices within the same cluster  $C_j^i$  is connected by an edge of length  $1/m$  for all layers except  $V^0$ . The remaining edges in the graph connect vertices in neighboring layers and are all of length 1. Each auxiliary vertex  $v_{j,k}^0$  in  $V_0$  is connected to the root and to its corresponding vertex  $v_{j,k}^1$  in layer 1. For  $1 \leq i \leq m - 1$ , we have an edge  $(v_{j,k}^i, v_{k,j}^{i+1})$  for each  $j, k \in [m]$ . In other words, the vertices of the  $j$ -th cluster in layer  $i$  are connected to the  $k$ -th vertices of the clusters in layer  $i + 1$ ; in particular, the  $k$ -th vertex of the  $j$ -th cluster in layer  $i$  is connected to the  $j$ -th vertex of the  $k$ -th cluster in layer  $i + 1$ . For example, see the edges leaving the first (top) cluster of  $V_1$  in Fig. 1. Observe that there are exactly  $m^2(m + 1)$  inter-layer edges, and exactly  $m^3(m - 1)/2$  intra-cluster edges.

Observe that each end vertex  $v_{j,k}^m$  has a unique path  $P_{j,k}$  to the root that consists of only inter-layer edges (see Fig. 1). We call these paths *canonical paths*. Note that each inter-layer edge belongs to exactly one canonical path. In other words, the set of inter-layer edges is a disjoint union of all the canonical paths  $P_{j,k}$ .

*The Cost of the Final Equilibrium.* Before describing the sequence of arrivals and departures of terminals, let us analyze the final equilibrium state and its cost

relative to the optimal cost. Let  $\text{OPT}$  denote the cost of the minimum spanning tree over all vertices in  $G_m$ . Observe that this is an upper bound on the cost of any optimal solution at the end. The final state following our sequence of arrivals and departures, denoted  $\mathcal{F}$ , consists of  $m$  players situated at every end vertex  $v_{j,k}^m$  in layer  $m$ ; each player uses the canonical path  $P_{j,k}$  to route to the root. The following lemma shows that this is an equilibrium state with a polynomially larger cost relative to  $\text{OPT}$ .

**Lemma 1.** *State  $\mathcal{F}$  is an equilibrium and the cost of  $\mathcal{F}$  is  $\Omega(m)$   $\text{OPT}$ .*

*Sequence of Arrivals and Departures.* The sequence is constructed in  $m$  phases, each phase consisting of  $m^2$  rounds, one per end vertex  $v_{j,k}^m$ , and indexed by  $(j, k)$ . Informally, the objective of each phase is to add one more terminal at each of the end vertices  $v_{j,k}^m$ . Within round  $(j, k)$  in a phase, we use a set of “temporary” terminals whose sole aim is to force the terminal at  $v_{j,k}^m$  that arrives at the end of the round to choose the canonical path as its best response. The temporary terminals are introduced at intermediate vertices along the canonical path during the round, and removed at the end of the round.

Formally, let  $\prec$  be an arbitrary total order on the pairs  $(j, k)$ . The sequence  $\sigma$  is constructed to maintain the following invariant: at the end of round  $(j, k)$  of phase  $\ell$ , there will be  $\ell$  players on  $v_{j',k'}^m$  for  $(j', k') \prec (j, k)$ , and  $\ell - 1$  players on the remaining end vertices. Furthermore, each player on  $v_{j,k}^m$  uses the path  $P_{j,k}$ .

We now specify the subsequence for each round. Consider round  $(j, k)$  of phase  $\ell$ . For simplicity of notation, we use  $v^i$  to denote the vertex of  $V^i$  on  $P_{j,k}$ . We also use  $P^i$  to denote the segment of  $P_{j,k}$  starting at  $v^i$  and ending at the root. The round consists of  $m + 1$  iterations. In iteration  $0 \leq i \leq m - 1$ ,  $m^2$  players arrive at  $v^i$ . In iteration  $i = m$ , one player arrives at  $v^m$ . Finally, the players on  $v^0, \dots, v^{m-1}$  depart. We can now show using induction over the terminal arrivals, that for every terminal the best-response path on arrival is the segment of the canonical path connecting it to the root.

**Lemma 2.** *Consider a terminal arriving at vertex  $v^i$  in iteration  $i$  of round  $(j, k)$  in phase  $\ell$ . The best-response path of the terminal to the root is the segment of its canonical path  $P^i$ .*

Lemma 2 shows that the sequence of arrivals and departures above terminates in the final state  $\mathcal{F}$ , which costs  $\Omega(m)$   $\text{OPT}$  by Lemma 1. Since  $m$  is polynomial in the number of vertices, Theorem 1 follows.

### 3 EQ-P Dynamics

In this section and the next, we describe and analyze EQ-P dynamics for the broadcast game. We first set up our notation and terminology, and prove some basic structural properties that are used in the rest of the paper. Let  $G = (V, E)$  be a complete graph,  $|V| = n$ , with metric costs  $c : V \times V \rightarrow \mathbb{R}_+$  defined on the edges. We assume without loss of generality that every vertex has a unique agent (a.k.a. terminal) residing at it. Agents arrive and depart over time. Since



edge costs satisfy the triangle inequality, before an agent arrives, no other agents route their paths via the vertex corresponding to this agent. Indeed, we assume that our intervention algorithm has no knowledge of vertices corresponding to agents that are yet to arrive. However, if an agent already in the system departs, other agents may continue to route their paths via its vertex, and the vertex remains in the graph. At any point of time during the process, our algorithm only considers the subgraph induced over vertices whose agents have arrived prior to that time. We call a vertex active if the agent at that vertex is still in the system.

The graph  $G$  is revealed via an online process that is divided into *epochs* (indexed by time  $t$ ). At the start of epoch  $t$ , the set of vertices in  $G$  that have already appeared is denoted by  $V_t$ . We denote the set of active terminals among them by  $A_t \subseteq V_t$ . Each terminal  $v \in A_t$  has a current routing path  $p_v$  connecting it to the common root  $r$ . The cost share of  $v$  along this path is the sum of  $v$ 's cost share over the edges in the path, where the cost of an edge is equally shared between all terminals currently using it. In the EQ-P scenario, we further enforce the invariant that the set of paths  $p_v$  are in NE, i.e., no terminal has an incentive to unilaterally deviate to a different routing path.

The *routing* at any time  $t$  is defined to be the set of routing paths  $(p_v)_{v \in A_t}$ . A *best response path* of a terminal  $v$  with respect to a routing, denoted  $p_v^*$ , is a path from  $v$  to  $r$  with the minimum shared cost if  $v$  were to move to this path. If there are multiple such paths, we break ties in favor of paths having fewer edges with no terminal other than  $v$  using them. Note that this may not break all ties, in which case, any of these paths can be designated as the best response path. A terminal  $v \in A$  is said to have an *improving move* with respect to a routing if by moving from its current path  $p_v$  to a new path  $q_v$  strictly decreases  $v$ 's cost share. Given a routing, its potential [34] is defined to be  $\Phi = \sum_{e \in E} \sum_{i=1}^{N_e} c_e/i$ , where  $N_e$  is the number of agents using  $e$ . A standard argument shows that any improving move decreases the potential by a value which is uniformly bounded away from zero, resulting in a finite convergence of our dynamics. The following well-known lemma states that in equilibrium, the routing paths form a tree.

**Lemma 3.** *In equilibrium, the routing paths of a broadcast game form a tree.*

Each epoch  $t$  is divided into several phases. The first phase consists of an arrival or departure event. In the former case, a new set of terminals  $U_t \subseteq V \setminus V_t$  arrive, and the cost of all edges incident on terminals in  $U_t$  is revealed. Each new terminal  $u \in U_t$  chooses a *best response* routing path  $p_u$ . In the latter case, a set of terminals leave, thereby removing the corresponding vertices from the set of terminals  $A_t$ . (Note that the corresponding vertices remain in  $V_t$ .) Lemma 6 establishes that the structure of the set of routing paths after arrivals or departures remains a tree.

Both arrival and departure events lead to changes in the cost shares of edges. In the EQ-P scenario, this might lead to a violation of the equilibrium state that was being previously maintained. In this case, the system performs a sequence of *improving moves*, in each of which a terminal changes its routing path in order to reduce its cost share.

Improving moves may temporarily create cycles in the collection of routing paths  $\{p_v\}_{v \in A_t}$ . We order and group improving moves into contiguous blocks or phases such that every phase ends with the routing paths forming a tree. Furthermore, the trees at the beginning and end of the phase differ in a single pair of edges. The collection of moves in each such phase is called a *tree-follow* move.

**Definition 1 (Tree-follow move).** *A tree-follow move from  $u$  to  $v$  in  $T$  is a sequence of improving moves that start with routing tree  $T$  and end with routing tree  $T' = T \setminus (u, \text{parent}(u)) \cup (u, v)$ , where  $\text{parent}(u)$  is the parent vertex of  $u$  in  $T$ . Observe that each terminal in the subtree rooted at  $u$  in  $T$  reroutes its path to the root according to  $T'$ .*

Because of departure events, the routing tree may contain non-terminal vertices as Steiner vertices. It is convenient to extend the notion of an improving move to vertices that are not terminals. Let  $w \notin A$  be a non-terminal vertex. We say that  $w$  has an improving move if the following properties hold: (1) There exists a terminal  $v$  whose routing path  $p_v$  includes  $w$ ; let  $p_w$  denote the segment of  $p_v$  between  $w$  and  $r$ ; (2) There exists a path  $q_w$  between  $w$  and  $r$  such that if  $v$  were to retain its current routing path from  $v$  to  $w$  but move from  $p_w$  to  $q_w$ , then the cost share of  $v$  would strictly decrease.

A priori, it is not clear whether improving moves can always be grouped into tree-follow moves. In Lemma 7, we show that in every routing tree  $T$  which is not in equilibrium, there exists a sequence of improving moves that collectively form the tree-follow move from  $u$  to  $v$  for some vertices  $u$  and  $v$ . When there are multiple such moves, we use a careful charging scheme to identify the order in which tree-follow moves should be implemented. (See Algorithm SELECT TREE MOVE defined at the end of this section.)

Since every vertex in a tree has a unique path to the root, it suffices to specify the tree itself in lieu of all of the routing paths. Henceforth, we will use  $T_t$  to denote the tree induced by  $\{p_v\}_{v \in A_t}$  without explicitly specifying the paths themselves.

### EQ-P Dynamics

1. *Initialization.*  $t = 1$ ,  $V_0 = \{r\}$ ,  $T_0 = \{r\}$ ,  $A_0 = \emptyset$ .
2. *For  $t = 1, 2, \dots$* 
  - (**Arrivals.**) Let  $U_t$  be the set of terminals arriving. Let  $A_t \leftarrow A_{t-1} \cup U_t$ . For each  $v \in U_t$ , let  $p_v = p_v^*$  where  $p_v^*$  is the best response path of  $v$  with respect to  $T_{t-1}$ . Let  $T_t = T_{t-1} \cup_{v \in U_t} p_v^*$ .
  - (**Departures.**) Let  $D_t$  be the set of terminals departing. Let  $A_t = A_t \setminus D_t$ . Let  $T_t = \cup_{v \in A_t} p_v$ .
  - (**Tree Follow Moves.**) While  $T_t$  is not in equilibrium:  
Use algorithm SELECT TREE MOVE to determine a tree-follow move to implement in  $T_t$ ; let this be a move from  $u$  to  $v$ , and let  $\text{parent}(u)$

denote the parent of  $u$  in  $T_t$ . Implement the sequence of improving moves for this tree-follow move to obtain the new routing tree  $T_t \leftarrow T_t \setminus (u, \text{parent}(u)) \cup (u, v)$ .

*Charging Scheme.* In proving the upper bound for EQ-P dynamics, we use a dual charging scheme to bound the cost of the routing tree. We define the dual and the corresponding lower bound on the optimal cost next. We call a partition  $P = (S_1, \dots, S_m)$  of the vertex set  $V$  a *level- $j$  dual* for an integer  $j$  if it satisfies the following:

- $P$  is a partition:  $\cup_{S \in P} S = V$ , and for any  $S_a, S_b \in P$ ,  $S_a \cap S_b = \emptyset$ .
- The components have bounded diameter: for any  $S \in P$ , and any vertices  $x, y \in S$ ,  $c(x, y) < 2^j$ .
- The components are far from each other: there exists a “center”  $s_i$  in each component  $S_i$ , such that for all  $S_a, S_b \in P$ ,  $c(s_a, s_b) \geq 2^{j-1}$ .

We use the term *cuts* to denote the components  $S$  of the partition. The lemma below follows immediately from the observation that any spanning tree over  $V$  must connect the centers of all cuts in a level- $j$  dual  $P$ .

**Lemma 4.** *For any level- $j$  dual  $P$ , the cost of the minimum spanning tree OPT is at least  $2^{j-1}(|P| - 1)$ .*

In order to bound the cost of an equilibrium resulting from EQ-P, we relate the cost of the edges used in the solution to a family of duals. Let  $\Pi = \{P_j\}_{j \in \mathbb{Z}}$  denote a family of partitions, where  $P_j$  is a level- $j$  dual.

Our charging scheme for routing solutions that form a tree proceeds as follows. Every vertex in the routing tree is responsible for the cost of its parent edge. Consider an edge  $e = (v, \text{parent}(v))$  with length in  $[2^{j+2}, 2^{j+3})$  for some  $j \in \mathbb{Z}$ . We charge the cost of this edge to the cut in the level- $j$  dual that contains  $v$ :  $S \in P_j$  such that  $v \in S$ . Our goal is to show that every cut gets charged a small number of times, and use the following well-known property (see the full version for a proof).

**Lemma 5.** *Suppose that our charging scheme charges each cut in the family  $\Pi$  at most once. Then the cost of the solution is at most  $O(\log n)\text{OPT}$ .*

For much of our analysis, we will assume that the dual family  $\Pi$  is provided to us. In the full version, we discuss how to construct this family algorithmically as terminals arrive online.

*Classification of a Tree Routing.* We classify the tree routings reachable via EQ-P dynamics into one of four states depending on the charging structure defined by the solution. We remark that not all tree routings are reachable via EQ-P dynamics, indeed even the set of equilibria obtained is smaller than the set of

all equilibria. Let  $T$  be a routing tree for some set of active terminals  $A$ . We say a vertex  $u$  is a leaf (non-leaf) if it is a leaf (non-leaf) in  $T$ . Note that all leaves must be terminals, but a non-leaf vertex may or may not be a terminal.

1. **BALANCED-EQUILIBRIUM:** In this state, no terminal (and therefore, no non-terminal vertex in  $T$ ) has an improving move. Furthermore, every cut is charged at most once. (Note that not every NE is a BALANCED-EQUILIBRIUM state.)
2. **BALANCED:** In this state, some terminals (and potentially non-terminals) may have improving moves, but every cut is charged at most once.
3. **LEAF-UNBALANCED:** In this state, every cut is charged by at most one non-leaf vertex (and any number of leaf terminals). (Recall that leaf vertices in the routing tree are necessarily terminals.)
4. **NON-LEAF-UNBALANCED:** In this state, all but one of the cuts are charged by at most one non-leaf vertex (and any number of leaf terminals). The exceptional cut, that we denote by  $S^*$ , is charged by at most two non-leaf vertices, say  $u$  and  $v$  (and any number of leaf terminals). One of these,  $u$  or  $v$ , must be the last vertex to have made a (tree-follow) move.

Note: BALANCED-EQUILIBRIUM  $\subseteq$  BALANCED  $\subseteq$  LEAF-UNBALANCED  $\subseteq$  NON-LEAF-UNBALANCED, where  $A \subseteq B$  implies that a routing tree in state  $A$  is also in state  $B$ .

*Selecting a Tree-Follow Move.* To define the tree-follow move performed in a non-equilibrium tree state  $T$ , we establish a system of priorities among the improving tree moves based on the current state of the routing tree. A tree follow move of  $u$  to  $v$  is said to be a *leaf move* if  $v$  is a leaf in  $T$ , and a *non-leaf move* otherwise.

**Algorithm** SELECT TREE MOVE

1. **BALANCED-EQUILIBRIUM:** No terminal has an improving move. The system can deviate from an equilibrium state only via arrivals or departure events.
2. **BALANCED:** In this state, for any vertex  $u$  that has an improving tree move, move  $u$  to the closest vertex to which it has an improving move.
3. **LEAF-UNBALANCED:**
  - (a) If there exists a leaf terminal  $u$  with a non-leaf move, then make any such move for  $u$ .
  - (b) Else, if there exists a non-leaf vertex  $u$  with a non-leaf move then move  $u$  to the closest such non-leaf  $v$ .
  - (c) Else, if there exists a non-leaf vertex  $u$  and a leaf terminal  $v$  such that  $u$  and  $v$  are charging the same dual cut, then move  $u$  to  $v$ . If there are multiple such leaf terminals  $v$ , then make any such move.
  - (d) Else, make any improving move. (This will necessarily be a leaf-to-leaf move by exclusion of the previous three cases.)

4. **NON-LEAF-UNBALANCED:** Let  $u$  and  $v$  be the non-leaf vertices that are charging the special cut  $S^*$ . If  $u$  has an improving move to  $v$  then move  $u$ , else move  $v$ , in either case to the closest vertex to which they have an improving move.

The validity of the algorithm depends on two claims. The first shows that whenever a cut is being charged by a leaf and a non-leaf, at least one of these two vertices has an improving move to the other. In this case, we can find a valid tree-move for Step (3c) of SELECT TREE MOVE. The second shows that in a NON-LEAF-UNBALANCED state, whenever a cut is being charged by two non-leaves, at least one of these two vertices has an improving move to the other; we can then find a valid tree-move for Step (4) of SELECT TREE MOVE.

### 3.1 Analysis of EQ-P Dynamics

We now give an outline of the proof of Theorem 2. Our argument hinges on a closure property: the epoch starts with the routing tree being in the BALANCED-EQUILIBRIUM state; Lemma 7 argues that whenever the current routing tree is not in equilibrium, at least one improving move exists, and we can use algorithm SELECT TREE MOVE to make a move; Lemma 8 then shows that for the moves made by algorithm SELECT TREE MOVE, the routing tree remains in one of the four states defined above, in particular, it is always in a NON-LEAF-UNBALANCED state. The epoch ends when the routing tree re-enters a BALANCED-EQUILIBRIUM state. At this point, by definition, each dual cut is charged at most once, and therefore, by Lemma 5 the cost of the routing tree is bounded, and Theorem 2 follows. We must also argue termination of the sequence of moves, but this follows directly from a standard potential argument based on the fact that all our moves are improving moves. The following lemmas capture the essence of our argument.

**Observation 1.** *In EQ-P dynamics the routing paths at the end of every phase form a tree.*

**Lemma 6.** *After the arrival or departure of a set of terminals in an BALANCED-EQUILIBRIUM state, the routing tree  $T$  remains in a LEAF-UNBALANCED state.*

**Lemma 7.** *If the routing tree is not in equilibrium, then at least one improving tree-follow move exists.*

**Lemma 8.** *Let  $T$  be the routing tree for which we make an improving tree-move in Step (3) of algorithm EQ-P.*

- (i) *If  $T$  is in a BALANCED state but not in a BALANCED-EQUILIBRIUM state, then after the move selected in Step (2) of SELECT TREE MOVE, the resulting tree is in a NON-LEAF-UNBALANCED state.*

- (ii) If  $T$  is in a LEAF-UNBALANCED state, then after the move selected in Step (3) of SELECT TREE MOVE, the resulting tree is in a NON-LEAF-UNBALANCED state.
- (iii) If  $T$  is in a NON-LEAF-UNBALANCED state, then after the move selected in Step (4) of SELECT TREE MOVE, the resulting tree is in a NON-LEAF-UNBALANCED state.

## References

1. Aland, S., Dumrauf, D., Gairing, M., Monien, B., Schoppmann, F.: Exact price of anarchy for polynomial congestion games. *SIAM J. Comput.* **40**(5), 1211–1233 (2011)
2. Albers, S.: On the value of coordination in network design. *SIAM J. Comput.* **38**(6), 2273–2302 (2009)
3. Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., Naor, J.: A general approach to online network optimization problems. *ACM Trans. Algorithms* **2**(4), 640–660 (2006)
4. Anshelevich, E., Dasgupta, A., Kleinberg, J.M., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. *SIAM J. Comput.* **38**(4), 1602–1623 (2008)
5. Awerbuch, B., Azar, Y., Bartal, Y.: On-line generalized steiner problem. *Theor. Comput. Sci.* **324**(2–3), 313–324 (2004)
6. Balcan, M.-F., Blum, A., Mansour, Y.: Circumventing the price of anarchy: leading dynamics to good behavior. *SIAM J. Comput.* **42**(1), 230–264 (2013)
7. Beier, R., Czumaj, A., Krysta, P., Vöcking, B.: Computing equilibria for a service provider game with (im)perfect information. *ACM Trans. Algorithms* **2**(4), 679–706 (2006)
8. Berman, P., Coulston, C.: On-line algorithms for steiner tree problems. In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pp. 344–353. ACM (1997)
9. Bilò, V., Bove, R.: Bounds on the price of stability of undirected network design games with three players. *J. Interconnect. Netw.* **12**(1–2), 1–17 (2011)
10. Bilò, V., Flammini, M., Moscardelli, L.: The price of stability for undirected broadcast network design with fair cost allocation is constant. In: *FOCS*, pp. 638–647 (2013)
11. Charikar, M., Karloff, H.J., Mathieu, C., Naor, J., Saks, M.E.: Online multicast with egalitarian cost sharing. In: *SPAA*, pp. 70–76 (2008)
12. Chawla, S., Naor, J., Panigrahi, D., Singh, M., Umboh, S.W.: Timing matters: online dynamics in broadcast games. *CoRR*, abs/1611.07745 (2016). <http://arxiv.org/abs/1611.07745>
13. Chekuri, C., Chuzhoy, J., Lewin-Eytan, L., Naor, J.(Seffi), Orda, A.: Non-cooperative multicast and facility location games. *IEEE J. Sel. Areas Commun.* **25**(6), 1193–1206 (2007)
14. Chen, H.-L., Roughgarden, T.: Network design with weighted players. *Theory Comput. Syst.* **45**(2), 302–324 (2009)
15. Christodoulou, G., Chung, C., Ligett, K., Pyrga, E., van Stee, R.: On the price of stability for undirected network design. In: *Bampis, E., Jansen, K. (eds.) WAOA 2009. LNCS, vol. 5893*, pp. 86–97. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-12450-1\\_8](https://doi.org/10.1007/978-3-642-12450-1_8)

16. Ene, A., Chakrabarty, D., Krishnaswamy, R., Panigrahi, D.: Online buy-at-bulk network design. In: IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17–20 October, 2015, pp. 545–562 (2015)
17. Fabrikant, A., Papadimitriou, C.H., Talwar, K.: The complexity of pure Nash equilibria. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 13–16 June 2004, pp. 604–612 (2004)
18. Fanelli, A., Leniowski, D., Monaco, G., Sankowski, P.: The ring design game with fair cost allocation. *Theor. Comput. Sci.* **562**, 90–100 (2015)
19. Fiat, A., Kaplan, H., Levy, M., Olonetsky, S., Shabo, R.: On the price of stability for designing undirected networks with fair cost allocations. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 608–618. Springer, Heidelberg (2006). [https://doi.org/10.1007/11786986\\_53](https://doi.org/10.1007/11786986_53)
20. Freeman, R., Haney, S., Panigrahi, D.: On the Price of stability of undirected multicast games. In: Cai, Y., Vetta, A. (eds.) WINE 2016. LNCS, vol. 10123, pp. 354–368. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-54110-4\\_25](https://doi.org/10.1007/978-3-662-54110-4_25)
21. Gupta, A., Ravi, R., Talwar, K., Umboh, S.W.: LAST but not least: online spanners for buy-at-bulk. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, 16–19 January, Hotel Porta Fira, Barcelona, Spain, pp. 589–599 (2017)
22. Hajiaghayi, M.T., Liaghat, V., Panigrahi, D.: Online node-weighted steiner forest and extensions via disk paintings. In: FOCS, pp. 558–567 (2013)
23. Hajiaghayi, M.T., Liaghat, V., Panigrahi, D.: Near-optimal online algorithms for prize-collecting steiner problems. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 576–587. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-43948-7\\_48](https://doi.org/10.1007/978-3-662-43948-7_48)
24. Harks, T., Klimm, M.: On the existence of pure nash equilibria in weighted congestion games. *Math. Oper. Res.* **37**(3), 419–436 (2012)
25. Imase, M., Waxman, B.M.: Dynamic steiner tree problem. *SIAM J. Discrete Math.* **4**(3), 369–384 (1991)
26. Kawase, Y., Makino, K.: Nash equilibria with minimum potential in undirected broadcast games. *Theor. Comput. Sci.* **482**, 33–47 (2013)
27. Lee, E., Ligett, K.: Improved bounds on the price of stability in network cost sharing games. In: EC, pp. 607–620 (2013)
28. Li, J.: An  $O(\log(n)/\log(\log(n)))$  upper bound on the price of stability for undirected shapley network design games. *Inf. Process. Lett.* **109**(15), 876–878 (2009)
29. Milchtaich, I.: Congestion games with player-specific payoff function. *Games Econ. Behav.* **13**, 111–124 (1996)
30. Monderer, D., Shapley, L.S.: Potential games. *Games Econ. Behav.* **14**, 124–143 (1996)
31. Naor, J., Panigrahi, D., Singh, M.: Online node-weighted steiner tree and related problems. In: FOCS, pp. 210–219 (2011)
32. Qian, J., Umboh, S.W., Williamson, D.P.: Online constrained forest and prize-collecting network design. *Algorithmica* **80**(11), 3335–3364 (2018)
33. Law-Yone, N., Holzman, R.: Strong equilibrium in congestion games. *Games Econ. Behav.* **21**, 85–101 (1997)

34. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. *Int. J. Game Theory* **2**(1), 65–67 (1973)
35. Ui, T.: A shapley value representation of potential games. *Games Econ. Behav.* **31**(1), 121–135 (2000)
36. Umboh, S.: Online network design algorithms via hierarchical decompositions. In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1373–1387. SIAM (2015)