On Mechanism Design Without Payments for Throughput Maximization

Thomas Moscibroda
Systems & Networking Research Group
Microsoft Research, Redmond, USA
moscitho@microsoft.com

Stefan Schmid
Chair for Efficient Algorithms
TU Munich, Munich, Germany
schmiste@in.tum.de

***Abstract*—It is well-known that the overall efficiency of a distributed system can suffer if the participating entities seek to maximize their individual performance. Consequently, mechanisms have been designed that force the participants to behave more cooperatively. Most of these game-theoretic solutions rely on payments between participants. Unfortunately, such payments are often cumbersome to implement in practice, especially in dynamic networks and where transaction costs are high. In this paper, we investigate the potential of mechanisms which work without payments. We consider the problem of throughput maximization in multi-channel environments and shed light onto the throughput increase that can be achieved with and without payments. We introduce and analyze two different concepts: the worst-case *leverage* where we assume that players end up in the worst rational strategy profile, and the average-case *leverage* where player select a random non-dominated strategy. Our theoretical insights are complemented by simulations.**

I. Introduction

Non-cooperative and selfish behavior in networks and large-scale distributed systems is an important challenge to the efficiency of such systems. And as these systems are becoming ever more decentralized, complex and heterogenous, this trend is only going to increase further in the future. Driven by this observation and the goal to contain the inefficiencies caused by selfish behavior, researchers from various disciplines have investigated means to foster cooperation among autonomous participants. One common tool towards this aim is *mechanism design*, which attempts to provide incentives to users such that they behave in more socially beneficial ways. In a world of money, in order to implement such behavior, numerous mechanisms have been designed which are based on payments. In many situations, such payments are natural and can be distributed (or, in case of negative payments, *collected*) in an efficient manner. For instance, most governments today use a wide range of taxes and subsidies to achieve their goals, e.g., to limit inequalities within a society or to support certain industry sectors.

The problem is that in large-scale networks and highly-distributed systems, employing such mechanisms that are based on payments or that include the transfer of money are often impracticable or inefficient. To give just one example, in systems such as multi-hop wireless networks, payment-based incentives have for instance been proposed to encourage nodes to relay other nodes' packets [15], [16]. However, it is challenging to determine what such payments could be and how, in practice, they could be enforced.

For these reasons, it would therefore be ideal for networks and distributed systems if mechanism design schemes could be implemented *without any payments*, and without any kind of monetary transfer. Unfortunately, there are strong theorems in economic literature that show that in general, the power of mechanisms without money is severely limited [2], [12].

The key observation that motivates our paper is that in certain practical cases, the mechanism designer is nonetheless capable of improving the social welfare *without making any payments at all*. Specifically, we show for a basic network flow optimization scenario, there is a form of mechanism design that does neither involve monetary instruments nor payments of any sort. Intuitively, we show that there are situations in which a trusted entity can improve the social welfare of the system, simply by making appropriate *promises of payments* to the participants in case certain outcomes occur. In a way, these promises then work like insurances for the players, hedging them against unfavorable outcomes. This gives these rational players more flexibility in their (selfish) decision which, as we show in this paper, can ultimately lead to a higher social welfare.

Formally, we define and study the concept of *leverage*, which captures how much increase in social welfare a mechanism designer can achieve for a certain amount of payment. As a particularly interesting and potentially important special case in practice, we study the *0-leverage*, which describes how much social welfare can be improved without any payments at all; simply by making appropriate insurance promises to players. The 0-leverage is thus a concept of potentially great relevance in networks and distributed systems in which mechanisms based on monetary transfer are undesired.

Before being able to use these concepts in actual system design, we seek to understand their possibilities and limitations. For this purpose, we study in this paper a simple throughput optimization game that is both sufficiently simple and concise to allow for stringent reasoning and analysis, but still captures a typical, generic scenario that arises (in similar fashion) in many networks and systems. Specifically, we study the leverage in a *throughput maximization game* in which there is a set of channels (of potentially different capacity), and a set of players each of which wants to find a route for its own flow with maximum capacity. For this game, we prove bounds on the achievable leverage with and without payments. We also corroborate our findings using simulation results.

The remainder of this paper is structured as follows: After a motivating example in Section II, we review related work in Section III. We formally define both the throughput maximization setting, as well as the framework for our study of *leverage* in Section IV. After deriving some general results on our game in Section V, Sections VI and VII provide the analytical results on mechanism design with and without payments, respectively. Section VIII presents simulation results, before the paper concludes in Section IX.

II. Motivating Example

The question that we seek to shed light on in this paper is the impact of payments, and to what degree such payments are indeed necessary. The key intuition for why this question is interesting is best conveyed with the following simple example using the classic prisoner's dilemma.

$U(x)$	l	t
l	3/3	0/4
t	4/0	1/1

$Q(x)$	l	t
l	1/1	2/0
t	0/2	0/0

Example: In this game, two players, A and B can either say the truth (t) or lie (l). Whereas the social optimum strategy would be if both players lied (both players get utility of $U_i(l,l) = 3$, see the table for $U(x)$ above), the only Nash equilibrium is if both players say the truth in which case, $U_i(l,l) = 1$ for both players. That is, the social welfare in the optimum and in the Nash equilibrium are $U(Opt) = 6$ and $U(NE) = 2$, respectively.

Now, assume that there exists a global entity that has the power to make financial promises (i.e., insurances) to the two players in case certain outcomes occur. Further, assume that this mechanism designer promises payments as indicated in the table $Q(x)$. That is, if both players lie, the mechanism designer will pay 1 to both players. The thing to recognize is that with the addition of the insurance payments, the player's utility in a given outcome x has now changed from originally $U_i(x)$ to $U_i(x)+Q_i(x)$. Because of this, the players now have a utility of 4 if both players lie, and the utilities in outcomes (l,t) and (t,l) are $(4,2)$ and $(2,4)$, respectively. Hence, it is now *always* in both players' best interest to lie, i.e., both players lying has become a Nash equilibrium in the game.

The intriguing observation is that by making payments of $1+1=2$ for the outcome (l,l), the mechanism designer was able to improve the social welfare by $6-2=4$ and implement (l,l) as a Nash equilibrium. That is, even if we subtract the payments made by the mechanism designer, there is still a net-gain of social welfare of 2 in the system, simply because the mechanism designer made the right set of promised payments.

The example shows that with well-placed payments, a global coordinator is capable of significantly improving the social welfare. The increase in social welfare can even exceeds the cost invested by the global coordinator. The most startling thing, however, is that—as we show in this paper—there are cases in which the central coordinator does not end up paying *anything* at all.

Implications: If a global trusted entity can improve the social welfare simply by making the right set of promises, but without actually making any payments, some benefits of mechanism design could be achieved without having the practically troublesome implications of money transfer. Such a scheme could enable new approaches towards designing distributed systems that need to cope with selfish behavior. In this paper we therefore seek to shed light on what can be achieved with and without payments under such circumstances.

III. Related Work

Most of the classic literature on load-balancing and throughput maximization is based on the assumption that either, there exists a centralized controller that manages the flows efficiently, or that the distributed entities coexisting in the system altruistically collaborate and follow their assigned protocol. The question of how to devise algorithms and mechanisms to optimize system performance in the face of such *selfish participants* is the focus of *algorithmic mechanism design* [10].

In recent years, many mechanism design results involving payments of money, stamps, points or similar objects of value have been proposed for distributed systems, e.g., the work on routing and multicast in wireless multi-hop networks [15], [16], network formation [1], or quota-based spam control [14]. More generally, approaches that are based on the celebrated Vickrey-Clark-Groves mechanism (VCG) involve money transfer in one form or another. The fundamental problem with these schemes is that in practical distributed systems, relying on monetary transfers often imposes a high implementation barrier [6].

Unfortunately, the fundamental *Arrow's Theorem* [2], [12] shows that the power of mechanisms without money is severely limited in general. However, there are instances that show how in certain scenarios monetary payments *can* sometimes altogether be avoided [7]. There have been at least three general approaches to mechanism design without payments. Arguably the most famous such approach is the barter-based *tit-for-tat mechanism* in BitTorrent. However, such barter systems come with their own set of problems (e.g. reliance on altruistic seeders to solve the bootstrapping problem, and are thus exploitable [8], [11]). Secondly, interesting results have also been obtained for *inter-domain routing* [7]. Distributed algorithmic mechanisms have been designed which achieve incentive compatibility in a collusion-proof ex-post Nash equilibrium without payments. Interestingly, the Border Gateway Protocol—the standard inter-domain routing protocol—is an example of such a mechanism. And third, there are systems that are based on the observation that computer systems typically have the ability to arbitrarily reduce service quality (e.g., by dropping messages or insert delays). This has given raise to *money-burning mechanisms* that demand payments in the form of computation or bandwidth (e.g., in the context of email spam or denial of service attacks) [13]. Optimal money-burning mechanisms are studied in [6].

Compared to these works, we pursue a different approach. We assume the existence of a trusted mechanism designer that can make *payment promises* to participants, in case certain unfavorable outcomes occur. Our paper builds upon the k-implementation model introduced by Monderer et al. [9], which has subsequently also been considered in [4], [5]. These papers give possibility and impossibility results for general games, and provide algorithms to compute optimal payments. Monderer et al. also point out an intriguing connection between 0-implementations and *correlated equilibria* [3]. In a correlated equilibrium the players act according to the publicly known random distributions over the strategies specified by the mechanism designer. The authors prove that every correlated profile is 0-implementable with an appropriate implementation device. In this paper, we study a deterministic environment and explore k-implementations for a specific networking application. We seek to shed light on the question of whether and to what extent mechanism design without money can be used to improve throughput in selfish networks. As we will see, high throughput implementations can often be computed efficiently. We compare the obtained throughput increase to mechanisms implementing a socially *maximum* throughput *with payments*, and discuss the "price of free mechanisms". Besides worst-case analyses, we provide average-case bounds obtained from *in silico* experiments.

IV. Model

In this section, we formally define the specific setting in which we study the relative capabilities and limitations of mechanism design with and without payments.

A. The Throughput Game: Definitions & Notations

We consider an abstract networking setting in which there are n players $P = \{p_1, \dots, p_n\}$ and m parallel channels $E = \{e_1, \dots, e_m\}$. Each channel $e_j \in E$ has a certain *capacity* $c_j = c(e_j) > 0$, which indicates how much flow it can handle. In order to simplify the presentation, we assume that the channels are ordered with respect to their capacities: $c_1 \geq c_2 \geq ... \geq c_m$. (If this was not the case, an initial sorting operation would yield an additional additive $O(m \log m)$ term in our time bounds.) On the other hand, each player $p_i \in P$ is associated with a *demand* $d_i = d(p_i)$, which is the amount of flow that the player wants to send through the network. In this paper, we will primarily consider the case in which all players have equal demands, i.e., $d_i = D$, for all $p_i \in P$. As for notation, we say that a channel e_j is *larger* than channel e_k if $c_j > c_k$, *equal* if $c_j = c_k$, and *smaller* otherwise.

Each player chooses a single channel to route its flow. Let $e(p_i)$ denote the channel selected by player p_i, and let $P(e_j)$ be the set of players that have selected channel e_j as their channel, i.e., $P(e_j) = \{p_i | e(p_i) = e_j\}$. The *throughput* a player obtains is determined by its channel and the number of other players that have selected the same channel. Specifically, if several players use the same channel, its capacity is distributed evenly among the players. Concretely, let $n_j = |P(e_j)|$ be the number of players that select a channel e_j. The *throughput* $T(p_i)$ of a player p_i that has selected channel e_j is defined as $T(p_i) = \min\{c(e_j)/n_j, d(p_i)\}$. That is, a player's throughput is either its fair share of the channel capacity or, in case the channel has sufficient capacity, its full demand. Every player attempts to select its channel in a utility maximizing manner, i.e., it chooses the channel which maximizes its throughput without taking into consideration the other players' throughput.

Notice that this abstract model captures a wide variety of natural problems that arise in practical networking and distributed systems scenarios. To give just one example, the different channels in our setting can correspond to actual wireless channels in a wireless networks, or to different network interfaces (say, Bluetooth, Wi-Fi, etc). Each player has to choose on which such channel or interface it wants to transmit its data (flow).

Let X denote the set of all possible strategy profiles.[1] Given a specific outcome of the game $x \in X$, a player p_i's utility $U_i(x)$ is its own throughput, i.e. $U_i(x) = T(p_i)$ in this outcome. The sum of all these utilities is the *social utility* denoted by $U(x) = \sum_{p_i \in P} U_i(x)$.

In this paper, we deal with two notions of rationality, *Nash equilibria* and *non-dominated strategies*. A strategy profile x is called a *(pure) Nash equilibrium* if no player can unilaterally improve its utility (throughput) given the strategy of the other players. In this paper, we primarily look at the second concept of rationality: *non-dominated strategy profiles*. Informally, a non-dominated strategy is a strategy for which there is no alternative strategy which is *always* better for a player, i.e., for any strategy choice of the remaining players. Non-dominated strategies do not assume anything about the behavior of other players and are thus a very general notion of rationality, making significantly weaker assumptions on rational behavior than Nash equilibria. Formally, let $x_i, x_i' \in X_i$ be two strategies available to player p_i. Strategy x_i *dominates* x_i' iff $U_i(x_i, x_{-i}) \geq U_i(x_i', x_{-i})$ for every possible strategy profile by the other players x_{-i}, and if there exists at least one x_{-i} for which a strict inequality holds. A strategy (channel) x_i is the *dominant* strategy for player p_i if it dominates every other strategy $x_i' \in X_i \backslash \{x_i\}$. x_i is a *non-dominated* strategy if no other strategy dominates it. We denote the set of strategy profiles which are non-dominated by $X_{UDom} \subseteq X$.

We consider the following special social utilities.

- The *social optimum* Opt is the strategy profile with maximum social utility, i.e., $U(Opt) = \max_{x \in X} U(x)$.
- The *worst-case non-dominated strategy profile* $UDom_{wc}$ is the non-dominated strategy profile (outcome of the game) with the worst social utility, i.e., $U(UDom_{wc}) =$

[1] A strategy in our game corresponds to selecting a specific channel. Hence, we use the terminology strategy and channel interchangeably. Similarly, a strategy profile corresponds to an outcome of the game in which every player has selected one specific channel.

$\min_{x \in X_{UDom}} U(x)$. It is the socially worst possible outcome of the game if every player acts rationally and selects a non-dominated strategy.

- It may be overly pessimistic to assume that all rational players select their non-dominated strategy in such a way that $UDom_{wc}$ arises. The *average non-dominated strategy profile* $UDom_{avg}$ is a random variable that denotes an outcome of the game that arises if every player selects one of its non-dominated strategies uniformly at random. The utility $U(UDom_{avg})$ is then defined as the expected social utility of $UDom_{avg}$, i.e., $U(UDom_{avg}) = E[UDom_{avg}]$.

Whereas $UDom_{wc}$ captures the worst-possible outcome of the game if participants behave rationally, $UDom_{avg}$ can be considered as describing the game's "typical" outcome. In the analysis section, we derive analytical upper and lower bounds for $U_{wc}(UDom)$; in our simulation-based evaluation, we show results for both concepts.

B. Implementation Theory & Leverage

The example in Section II shows that it is possible for a mechanism designer which seeks to influence and improve the outcome of the game, to offer payments in such a way as to improve the social outcome of the game in excess of the total amount of payments that it invested. This overall increase of the social good by means of a global mechanism designer is captured by the definition of *leverage*.

The mechanism designer offers a payment $Q_i(x)$ to every player p_i in case the game ends in outcome x. Formally, these payments can be described by a tuple of non-negative payoff functions $Q = (Q_1, Q_2, \dots, Q_n)$, where $Q_i : X \to \mathbb{R}^+$.[2]

With the promised payments, each player now has a utility of $U_i(x) + Q_i(x)$ in outcome $x \in X$. In other words, given an outcome x, each player p_i not only achieves a utility $U_i(x)$ as in the original game, but it also receives the payments $Q_i(x)$ that it was promised by the mechanism designer in case outcome x occurs. Notice that if $Q_i(x) = 0$, i.e., if no payments have been promised to p_i for outcome x, p_i has the exact same utility (i.e., simply its throughput) as it had in the original game, $U_i(x)$. With these promises made by the mechanism designer, the players' choices of strategies (i.e., channels) now change accordingly: each player selects a non-dominated strategy in the game with utility functions $U_i(x) + Q_i(x)$.

Payment Amount: For a specific outcome x of the game, the sum of the payments the mechanism designer makes to all players is denoted by the *payment amount* $Q(x) := \sum_{p_i \in P} Q_i(x)$. Hence, the payment amount $Q(x)$ is the amount that the mechanism designer actually ends up paying given the outcome x.

Implementation: As we have seen in the example of Section III, the key insight is that by making the right promises $Q_i(x)$, it is in fact possible for the mechanism designer to change the game in such a way as to increase the social welfare. Formally, we say that a mechanism designer *implements* a strategy profile (or *outcome*) x if it chooses its payments Q in such a way that it is in all rational players best interest to select the strategy that leads to x. Formally, the mechanism designer implements x iff the only non-dominated strategy in the game is x. In our specific throughput game, if the mechanism designer wants to *implement* a certain assignment of players to channels, it must choose the payment promises in such a way that it is in each selfish player's best interest to choose the corresponding channel.

Cost of Implementation: The *cost* of an implementation (i.e., the payment amount that the mechanism designer actually has to pay in the end) can be very different depending on the game, and how the mechanism designer assigns the payments. We say that if the implementation entails a cost $Q(x) = k$, then outcome x is k-*implementable*. That is, a payment of k is sufficient to make sure that outcome x will occur. If it is possible to implement a strategy *without entailing any payments* at all, the outcome is 0-*implementable*. As discussed in the introduction, 0-implementable outcomes are of particular interest because they can be enforced by the mechanism designer without any actual monetary transfer, even if all players act selfishly.

Leverage: The mechanism designer seeks to improve the social welfare (i.e., the overall throughput) using a certain amount of payment promises that it is willing to make. To quantify this achieved gain we introduce two measures.

The *worst-case leverage* is the absolute improvement by the mechanism divided by the socially optimal welfare. In doing so, we assume a pessimistic view and assume that the players always end up in the *worst* non-dominated strategy profile.

Definition 4.1 (Worst-Case Leverage Φ_{wc}): The worst-case leverage $\Phi_{wc}(x)$ of implementing a strategy profile x in a game is defined as

$$\Phi_{wc}(x) = \frac{U(x) - U(UDom_{wc}) - Q(x)}{U(OPT)}.$$

The worst-case leverage Φ_{wc} of a game is the maximal worst-case leverage over all outcomes x: $\Phi_{wc} = \max_{x \in X} \Phi_{wc}(x)$. The term $U(x) - U(UDom_{wc})$ captures the absolute increase in social welfare that the mechanism designer was able to achieve. Subtracting from this the cost of the implementation—i.e., the payment amount $Q(x)$ that the mechanism designer has to invest to enforce the outcome x—yields the absolute leverage that the mechanism designer can achieve. Finally, we normalize this value by the social optimum $U(OPT)$. Clearly, Φ_{wc} is at most 1. It is positive if the mechanism designer can implement some outcome in which the increase in social welfare exceeds its invested payment amount. At the extreme, if the mechanism designer is capable of achieving the socially optimal outcome at no cost, whereas the worst non-dominated strategy in the original game has social value 0, then the leverage would be $\Phi_{wc} = 1$.

[2]One way to look at it is to consider these payments as a kind of insurance. The mechanism designer is willing to pay a certain amount to protect the players from certain undesirable outcomes. By doing so, it encourages players to act in a socially more optimal fashion, which then results in the leverage.

The worst-case leverage tends to be pessimistic because it assumes that the rational players act in the socially worst-possible way. We therefore also consider the *average-case leverage*, which we define analogously as follows.

Definition 4.2 (Average-Case Leverage Φ_{avg}): The average-case leverage $\Phi_{avg}(x)$ of implementing a strategy profile x in a game is defined as

$$\Phi_{avg}(x) = \frac{U(x) - U(UDom_{avg}) - Q(x)}{U(OPT)}.$$

The average-case leverage Φ_{avg} of a game is the maximal average-case leverage over all outcomes x: $\Phi_{avg} = \max_{x \in X} \Phi_{avg}(x)$.

Special Leverages: In the context of our work, we specifically distinguish between the following three special cases of the (worst-case and average) leverage:

- The *0-Leverage* Φ^0_{wc} (and Φ^0_{avg}) is the leverage than can be achieved without making any payments at all, $Q(x) = 0$.
- The *Opt-Leverage* Φ^{opt}_{wc} (and Φ^{opt}_{avg}) is the achievable leverage when the mechanism designer implements the social optimum, i.e., the implemented x must be the social optimum.
- The general *k-Leverage*, denoted by Φ^k_{wc} and Φ^k_{avg}, respectively, where we allow the mechanism designer to implement arbitrary profiles and to make arbitrary payments. Because 0-Leverage and Opt-Leverage are special cases of the k-Leverage, it holds that $\Phi^0_{wc} \leq \Phi^k_{wc}$ and $\Phi^{Opt}_{wc} \leq \Phi^k_{wc}$.

In this paper, we will focus on leverages Φ^0_{wc} and Φ^{opt}_{wc}, leaving the study of Φ^k_{wc} as interesting future research.

V. General Results

Before proving the various bounds on the achievable leverage in throughput games, we start by characterizing some general properties of our game. Specifically, in order to compute the worst-case leverage, we need to have bounds on the social optimum as well as on the worst-case non-dominated outcome $UDom_{wc}$. For this purpose, we need some additional definitions. For any channel e_j, define $\eta_j = \lfloor c_j/D \rfloor$ to be the number of players for which it can satisfy their full demands. Furthermore, let $\widetilde{c}_j = c_j - \eta_j D$ be the *residual capacity* of channel e_j that is left after entirely satisfying η_j demands. Finally, let $\phi(1), \phi(2), \ldots, \phi(m)$ denote the indices of channels when ordered in non-increasing order of their residual capacity, i.e., for any two channels e_j, e_k if $\widetilde{c}_j > \widetilde{c}_k$, then $\phi(j) < \phi(k)$. The social optimum is as follows.

Lemma 5.1: The total throughput in the social optimum is given by

$$U(Opt) = \min\{n, \Gamma\} \cdot D + \sum_{j=1}^{\min\{m,n-\Gamma\}} \widetilde{c}_{\phi(j)}, \qquad (1)$$

where $\Gamma = \sum_{j=1}^{m} \eta_j$ is defined as the total number of players whose demand can be fully satisfied in the network. In the backlogged case ($D \to \infty$), this reduces to $U(opt) = \sum_{j=1}^{\min\{n,m\}} c_j$.

Proof: The first summand follows from the fact that the throughput cannot exceed the total demand, hence $U(Opt) \geq nD$, and that $U(Opt) \geq \Gamma D$ holds by the definition of Γ. If $\Gamma \geq n$, the second term evaluates to 0. Otherwise, there are at least $n - \Gamma$ remaining players whose demand D is not fully satisfied. Assuming that Γ players have already been allocated to the channels, the residual capacity of each channel e_j is at most $\widetilde{c}_j < D$. If $n - \Gamma \geq m$ all this residual capacity is used up, i.e., all channels are completely utilized. If $n - \Gamma < m$, the socially optimal solution is achieved if the remaining $n - \Gamma$ players utilize the $n - \Gamma$ channels with largest residual capacity, $\widetilde{c}_{\phi(1)}, \ldots, \widetilde{c}_{\phi(n-\Gamma)}$, because it is always feasible and worthwhile to assign a player to the channel with largest free residual capacity. ■

The following corollary follows immediately.

Corollary 5.2: The socially optimal welfare can be computed in time $O(\min\{n, m\})$.

Definitions 4.1 and 4.2 require knowledge of the utility of the *worst non-dominated strategy* profile in the absence of a mechanism. The following lemma characterizes the structure of $UDom_{wc}$ and shows how its utility can be computed.

Lemma 5.3: The worst-case non-dominated strategy profile is $UDom_{wc} = (e_\gamma, e_\gamma, ..., e_\gamma)$, where γ is the maximum $\ell = \{1, \ldots, m\}$ such that $c_\ell > \frac{c_1}{n}$. Furthermore, it holds that $U(UDom_{wc}) = \min\{c_\gamma, nD\}$.

Proof: When player p_i chooses some channel e_j, the resulting utility is $U_i(e_j, x_{-i}) = \min\{c_j/n_j, D\} \geq \min\{c_j/n, D\}$. Hence, the minimum utility a player can achieve, regardless of the strategies of other players is $U_i \geq \min\{c_1/n, D\}$ by choosing the channel with largest capacity. This strategy profile is dominated by all channels e_k for which $c_k > c_1/n$. By definition, the (not necessarily unique) smallest channel which fulfills this property is e_γ. The proof is concluded by observing that the social welfare is minimized if all players select channel e_γ in which case, either all demand can be satisfied on e_γ or $U(UDom_{wc}) = c_\gamma$. ■

From Lemma 5.3, it follows that the worst non-dominated strategy profile can be found trivially simply by determining channel e_γ. Since we assume that the channels are already sorted in the input, we have the following corollary.

Corollary 5.4: The worst non-dominated strategy profile can be computed in time $O(m)$.

For mechanisms where we seek to entirely avoid payments, Nash equilibria play a crucial role. The following claim is a direct consequence of the analysis by Monderer et al. [9].

Fact 5.5: A strategy profile x can be 0-implemented if and only if x is a Nash equilibrium.

Using this fact, we can derive that in the throughput maximization game, there always exist outcomes that can be implemented without any monetary payments by the mechanism designer.

Lemma 5.6: Every throughput game has at least one outcome x that can be implemented without payments, $Q(x) = 0$.

Proof: We show that a simple best-response strategy efficiently converges to a Nash equilibrium. Together with Fact 5.5, the claim follows. Assume that initially, no player is assigned a channel; and that one player after the other selects a channel in a best response fashion: it will select a channel e_j that maximizes $\frac{c_j}{n_j+1}$.

By induction over the number of players, we show that after the i^{th} player has selected its channel, no player has an incentive to change its choice, that is, the configuration constitutes a Nash equilibrium. For the first player, the claim holds trivially: it will choose the largest channels e_1, which constitutes a Nash equilibrium (and a social optimum). For the induction step, assume that the induction hypothesis is true and that $U_v(x_v, x_{-v}) \geq U_v(x'_v, x_{-v})$, for all of the i players p_v. According to the best response strategy, player $i+1$ selects the channel e_j which maximizes $\frac{c_j}{n_j+1}$. Therefore, for all $k \neq j$, it holds that $\frac{c_k}{n_k+1} \leq \frac{c_j}{n_j+1}$. Thus, no player on channel e_j would be better off by switching to an alternative channel e_k. By the induction hypothesis, it also follows that no player on any channel $e_k \neq e_i$ has an incentive to change its strategy. ■

VI. Without Payments: 0-Leverage Φ^0_{wc}

In this section, we characterize what improvements to the social welfare can be achieved by mechanism design if no monetary payments are to be made at all. Notice that when implementing a strategy profile x, the mechanism designer can (and sometimes does!) promise to each player p_i an arbitrarily large amount of money $Q(x') = \infty$ for all outcomes x' that will eventually be dominated, and hence will not occur.[3] The mechanism designer will only have to make the payments in profile x, and hence, this promised money is not actually spent.

The key ingredient to characterize Φ^0_{wc} is to derive a lower bound on the utility of an outcome that can be 0-implemented. The challenge is that not all 0-implementable outcomes yield the same throughput. For instance, in a game with two players and two channels with $c_2 = c_1/2$, the profiles (e_1, e_1) and (e_1, e_2) are both Nash equilibria, and can hence be 0-implemented [9].[4] However, their utilities are different (c_1 and $c_1 + c_2$, respectively). Thus, in order to maximize the leverage, we need to implement the best Nash equilibrium.

Lemma 6.1: Let $0Best$ denote the best 0-implementable strategy profile. We can bound the utility of $0Best$ as

$$\min\{nD, \sum_{i=1}^{\theta_{low}-1} c_i\} \;\leq\; U(0Best) \;\leq\; \min\{nD, \sum_{i=1}^{\theta_{high}-1} c_i\},$$

where θ_{low} and θ_{high} are defined as

$$\theta_{low} \;:=\; \min\, k \text{ s.t. } \frac{1}{n}\sum_{i=1}^{k-1} c_i > c_k$$

$$\theta_{high} \;:=\; \min\, k \text{ s.t. } \frac{1}{c_k}\sum_{i=1}^{k-1} c_i - k > n-1.$$

In both cases, if there is no such $k \in \{1, \ldots, m\}$, then $k = m+1$.

Proof: We prove the bounds by means of contradiction. First, consider the lower bound. We show that there exists a Nash equilibrium in which players use the first $\theta_{low} - 1$ channels. Assume for contradiction that the first unused channel is $e_{\theta'}$ with $\theta' < \theta_{low}$, such that $\frac{1}{n}\sum_{i=1}^{\theta'-1} c_i < c_{\theta'}$. First, note that no other channel e_k for $k > \theta'$ can be used (up to reorderings of channels of equal capacity), because otherwise, such a player would have an incentive to switch to $e_{\theta'}$. Thus, assume that only the first θ' channels are used. In this case, the average throughput of all players is at most $\frac{1}{n}\sum_{i=1}^{\theta'-1} c_i$ and hence, there must exist a player $p_i \in P$, for which $U_i \leq \frac{1}{n}\sum_{i=1}^{\theta'-1} c_i$. However, because of $\frac{1}{n}\sum_{i=1}^{\theta'-1} c_i < c_{\theta'}$, this player would have an incentive to switch to channel $e_{\theta'}$, which contradicts the assumption that it was a Nash equilibrium. It thus follows that at least the first $\theta_{low} - 1$ channels are used and hence, $U(0Best) \geq \sum_{i=1}^{\theta_{low}-1} c_i$. Clearly, it also holds that $U(0Best)$ cannot exceed nD.

We now prove the upper bound on $U(0Best)$. Again, $U(0Best) \leq nD$ is clear. Assume for contradiction that there is a Nash equilibrium in which some player routes its flow on channel $e_{\theta''}$, for $\theta'' \geq \theta_{high}$. The utility this player obtains is $\min\{c_{\theta''}, D\}$. In order for this player to have no incentive to switch to a higher capacity channel $e_j, j < \theta''$, it must hold that $\frac{c_j}{n_j+1} < c_{\theta''}$ and hence, $n_j > \frac{c_j}{c_{\theta''}} - 1$. Summing up over all channels $e_1, \ldots, e_{\theta''-1}$ yields

$$\begin{aligned}\sum_{j=1}^{\theta''-1} n_j \;&\geq\; \sum_{j=1}^{\theta''-1}\left(\frac{c_j}{c_{\theta''}} - 1\right) \geq \sum_{j=1}^{\theta_{high}-1}\left(\frac{c_j}{c_{\theta_{high}}} - 1\right)\\ &=\; \frac{1}{c_{\theta_{high}}}\sum_{j=1}^{\theta_{high}-1} c_j - \theta_{high} + 1 \;>\; n-1.\end{aligned}$$

The second inequality is due to $\theta'' \geq \theta_{high}$ and the final inequality follows from the definition of θ_{high}. Since there can be most $n-1$ players on $e_1, \ldots, e_{\theta''-1}$ if one player is on channel $e_{\theta''}$ this is a contradiction. ■

We are now ready to prove lower and upper bounds on the 0-leverage.

Theorem 6.2: For $D \to \infty$, the worst-case 0-leverage Φ^0_{wc} can be bounded as follows

$$\frac{\left(\sum_{i=1}^{\theta_{low}} c_i\right) - c_\gamma}{\sum_{i=1}^{\min\{n,m\}} c_i} \leq \Phi^0_{wc} \leq \frac{\left(\sum_{i=1}^{\theta_{high}} c_i\right) - c_\gamma}{\sum_{i=1}^{\min\{n,m\}} c_i},$$

where c_γ, θ_{low}, and θ_{high} are as defined in Lemmas 5.3 and 6.1, respectively. For general D, it holds

$$\frac{\sum_{i=1}^{\theta_{low}} c_i - U(UDom_{wc})}{U(Opt)} \leq \Phi^0_{wc} \leq \frac{\sum_{i=1}^{\theta_{high}} c_i - U(UDom_{wc})}{U(Opt)},$$

with $U(UDom_{wc})$ and $U(Opt)$ as derived in Lemmas 5.1 and 5.3.

Proof: The claim follows immediately by substituting the terms in Definition 4.1 with the results in Lemmas 5.1, 6.1, and 5.3 and the observation that $Q(\cdot) = 0$. ■

[3] Of course, in practice, instead of using infinite payment promises, a mechanism designer will offer a payment which is, e.g., slightly larger than the maximal utility in the game. In this paper, for simplicity, we will denote this value by ∞.

[4] In order to 0-implement, e.g., (e_1, e_2), set $Q_1(e_1, e_i) = \infty$ for $i \neq 2$ and $Q_2(e_j, e_2) = \infty$ for $j > 1$.

VII. Mechanisms with Payments

While in distributed systems, 0-implementable solutions are often desirable, they constitute only a subset of all possible implementations. In this section—in order to understand the relative possibilities and limitations of implementations with and without payments—we allow payments and see whether larger leverages are possible as compared to Section VI.

A. The OPT-Leverage Φ_{wc}^{opt}

One goal a mechanism designer might pursue is to implement the socially optimal profile (cf Lemma 5.1). Unfortunately, it is often not possible to 0-implement the social optimum or in other words, it is not generally possible to maximize throughput for free. As an example, consider a game with two players and two channels where the channel capacities are $c_1 = 12$ and $c_2 = 4$. In this game, the social optimum (e_1, e_2) (utility of 16) can be implemented at cost 2, while the dominant, 0-implementable Nash equilibrium (e_1, e_1) has utility 12 only. In other words, payments are unavoidable if the social optimum should be implemented in this example. We thus seek to implement the optimal outcome at minimal costs, and study the resulting leverage.

Notice that in contrast to the 0-implementation in Section VI, the OPT-Leverage Φ_{wc}^{opt} can actually be negative in certain cases. To see this, consider a simple network with two links of capacity 1 and ϵ, and two players with infinite demand. In this example, the social optimum is $1+\epsilon$, whereas the best non-dominated outcome is if both players are on the channel with capacity 1. Furthermore, implementing the social optimum requires forcing one of the two players to use the ϵ-channel, which incurs a cost of at least $1/2-\epsilon$. It follows that in this example, $\Phi_{wc}^{opt} = (1+\epsilon-1-1/2+\epsilon)/(1+\epsilon) \approx -1/2$.

We first derive a formula for the implementation cost of an arbitrary strategy profile.

Lemma 7.1: Let $e_{p_i} = e(p_i)$ denote the channel used by player p_i. In order to implement a strategy profile $x = (e_1, ..., e_n)$, a mechanism designer needs to pay exactly

$$Q(x) = \sum_{\forall p_i \in P} \max\left\{\min\{D, \max_{e_j \neq e_{p_i}}\{\frac{c_j}{n_j+1}\}\} - \min\{D, \frac{c_{p_i}}{n_{p_i}}\}, 0\right\}$$

Proof: In order to implement an outcome $x = (e_1, ..., e_n)$, for each $p_i \in P$, the mechanism can promise an arbitrarily large payment amount $P(x') = \infty$ for all profiles $x' \neq x$. As these outcomes will be dominated (and hence, will not occur), no actual payments will result. However, payments made in profile x itself will be substantiated. Consider the payment $Q_i(x)$ that needs to be made to p_i in x. Without any mechanism, p_i obtains a utility of $U_i(x) = \min\{D, c_{p_i}/n_{p_i}\}$. In order to dominate all of p_i's alternative strategies, it must hold that this utility plus the payment is at least $\min\{D, c(e_j)/(n_j+1)\}$ for all alternative channels $e_j \neq e_{p_i}$. Hence, the mechanism designer must pay the difference between $\max_{e_j \neq e_{p_i}}\{\min\{D, c(e_j)/(n_j+1)\}\}$ and $U_i(x)$. The lemma follows by summing up over all players. ■

Observe that in a game with n players and m channels, there are m^n possible strategy profiles. Already for small networks, this number is too large for an exhaustive search of special profiles. Fortunately, the cheapest implementable optimum profile—and hence Φ_{wc}^{opt} can be computed efficiently.

Theorem 7.2: Given a game with n players and m channels, Φ_{wc}^{opt} can be computed in time $O(nm)$.

Proof: In order to calculate Φ_{wc}^{opt} (cf Definition 4.1), we need to compute the utility of the worst non-dominated strategy profile, the socially optimal utility, and the implementation cost. From Corollaries 5.2 and 5.4, we know that the worst non-dominated strategy profile and the social optimum can be computed in time $O(m)$ and $O(\min\{n, m\})$, respectively. Thus, it only remains to study the implementation costs.

The number of players that a channel e_j can hold before it is completely full is $\lceil c_j/D \rceil$. We distinguish between $n \leq \sum_{e_j \in E} \lceil c_j/D \rceil$ and $n > \sum_{e_j \in E} \lceil c_j/D \rceil$. In the former case, the following greedy algorithm yields an optimal channel assignment which constitutes a social optimum with minimal cost: For one player p_i after another, we assign p_i the channel which increases the social welfare the most. Observe that 1) every player can actually improve the social welfare, 2) this is indeed optimal as the choices of different players are independent, and 3) the greedy algorithm terminates in time $O(nm)$. If $n > \sum_{e_j \in E} \lceil c_j/D \rceil$, we apply our greedy algorithm until an additional player cannot improve the social welfare anymore (i.e., to the first $\sum_{e_j \in E} \lceil c_j/D \rceil$ players). Then, we distribute the remaining players among the channels. Note that the social welfare is not a affected by this distribution, but in order to optimize Φ_{wc}^{opt}, we need to minimize the implementation cost. This can be achieved by letting the remaining players choose their channel one after another by a best-response strategy which maximizes the corresponding player's utility. In other words, given the assignments of the first phase of the greedy algorithm, the remaining players compute a Nash equilibrium. By a similar argument as used in the proof of Lemma 5.6, the cost is indeed minimized. The total additional runtime in order to compute this equilibrium is $O(nm)$. ■

We can now derive an expression for Φ_{wc}^{opt} in the case of $n \leq \sum_{e_j \in E} \lceil c_j/D \rceil$.

Theorem 7.3: If $n \leq m$ and for $D \to \infty$, the worst-case Opt-leverage is

$$\Phi_{wc}^{opt} \geq 1 - \frac{c_\gamma + \sum_{e_j \in E^<} \left(\frac{c_1}{2} - c_j\right)}{\sum_{i=1}^{n} c_i} \qquad (2)$$

where $E^< \subseteq E$ is the set of channels e_j for which it holds that $\frac{c_1}{2} > c_j$, and c_γ is as defined in Lemma 5.3.

Proof: The leverage is computed according to Definition 4.1. Lemma 5.1 shows that for $D \to \infty$ and $n \leq m$, the social optimum solution picks the n highest capacity channels, yielding $U(Opt) = \sum_{i=1}^{n} c_i$. In such an optimal configuration x, according to Lemma 7.1, the mechanism designer has to pay each player p_i the difference of its utility $U_i(x) = c_{e_i}$ to the largest possible utility it could get. In x, for $n \leq m$, this best possible alternative is $c_1/2$. Finally, we know from Lemma 5.3 that $U(UDom_{wc}) = c_\gamma$. ■

Note that for smaller D, similar expressions can be derived based on the more exact value in Lemma 5.1.

VIII. Evaluation

We proceed to investigate the potential for mechanism design without payments in "typical" scenarios. Specifically, given that the worst-case leverage tends to be overly pessimistic as the number of player increases, we are interested in the average-case leverage Φ_{avg}: we assume players end-up in a *random non-dominated strategy profile* (cf Definition 4.2).

Simulation Methodology: We compute the average-case and worst-case leverages ($\Phi_{avg}^{OPT}, \Phi_{avg}^{0}, \Phi_{wc}^{OPT}$, and Φ_{wc}^{0}) using the algorithms described in Sections VI and VII. For each test-run, we create 100 sample points (random non-dominated strategy profiles) and present the average. Error bars are omitted as the amount of variance is generally small. Unless otherwise stated, there are 100 channels each with capacity 100.

Impact of Number of Players: A first set of experiments shows the effect of additional players on the leverage. Figures 1 and 2 show the different leverages in a system with 16 and 32 channels, for both constant and uniformly distributed channel capacities. Player demands are 100. It can be observed that in all cases, the worst-case leverages tend to reach large values quickly. The reason is that in such networks, the worst non-dominated outcome is typically very bad compared to the social optimum, $U(UDom_{wc}) \ll U(Opt)$, especially as the number of players grows. In contrast, the average leverages behave differently. After an initial spike (the maximum being roughly around the point when $|P| = |E|$), the average leverages decay and go to 0 as $|P|$ increases. The reason is that with a large number of channels, the randomly selected non-dominated outcomes become increasingly better, and in the limit reach the social optimum.

An interesting phenomenon can be observed in Figure 2 for uniform channel capacity distributions. Whereas, again, both Φ_{wc}^{0} and Φ_{wc}^{OPT} reach high numbers very quickly, there is a temporary decline when the number of players roughly equals the number of available channels in both Φ_{wc}^{OPT} and Φ_{avg}^{OPT}. This phenomenon can be explained as follows: in order to implement a socially optimal strategy profile, each channel must be occupied by exactly one player. As the weakest channels are taken last, a large cost accrues when the last free channels are assigned. After that, the implementation become cheaper again. In order to substantiate this explanation, Figure 3(a) plots the total implementation costs $Q(Opt)$ as a function of the number of players m. It can be seen that for different m this implementation cost drastically increases at around m.

Impact of Demand: Figure 3(b) shows the impact of the players' demand on the different average leverages. The most interesting take-away is that in all but one case, the leverage increases and reaches a stable point roughly between 0.35 and 0.4. The only exception is Φ_{avg}^{Opt} for uniformly distributed channel capacity. While surging faster than the other three values, it then drops and stabilizes on a significantly lower leverage. The reason is that compared to Φ_{avg}^{0}, Φ_{avg}^{Opt} is much more susceptible to large diversities in channel capacities. If, for instance, there is a large number of channels with very

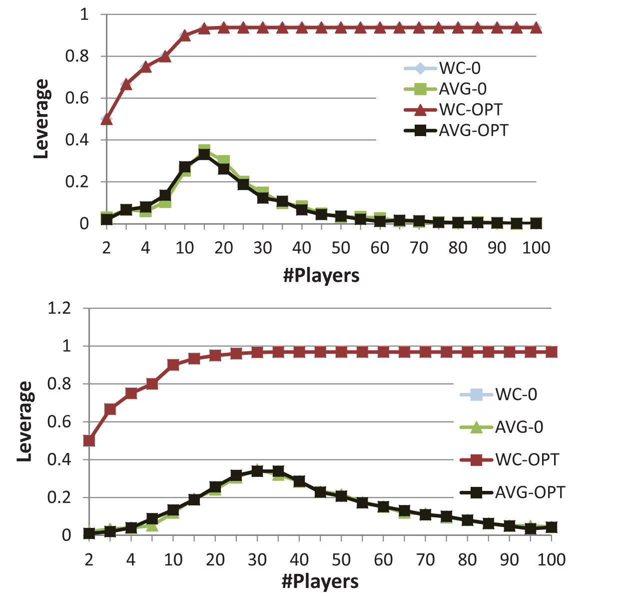

Fig. 1. Leverage as a function of the number of players for (a) 16 and (b) 32 channels. Channel capacities are constant $c_j = 100$ for all $e_j \in E$. Notice that the curves for Φ_{wc}^{Opt} and Φ_{wc}^{0}, and for Φ_{avg}^{Opt} and Φ_{avg}^{0} overlap.

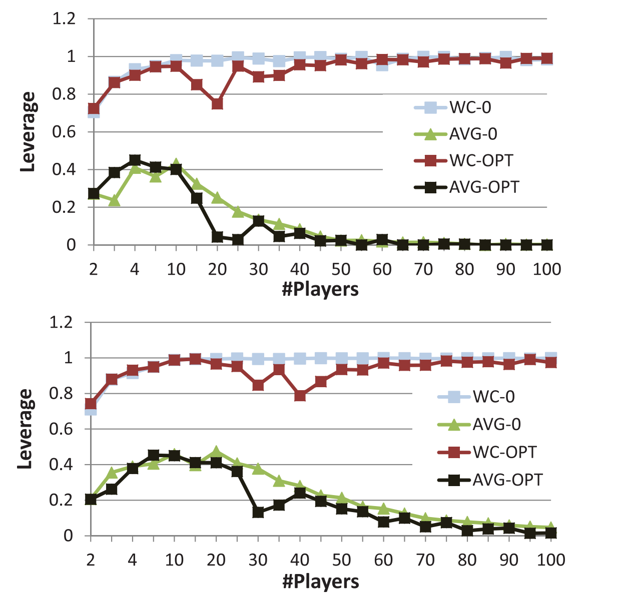

Fig. 2. Leverage as a function of the number of players for (a) 16 and (b) 32 channels. Channel capacities are uniformly distributed in $[0, 100]$.

small capacities, Φ_{avg}^{0} will ignore these channels, whereas Φ_{avg}^{Opt} will still try to implement them, thereby causing an increase in payments $Q(x)$. This also highlights the impact on the channel capacity distributions on the achievable leverages.

Impact of Scale: Figure 4(a) shows the impact of scale on the leverage. In particular, we define the parameter W as the ratio between the number of players and the number of channels, i.e., $|P| = W \cdot |E|$. We keep W fixed and increase the number of channels (and hence also the number of players). It can be seen that for different values of W, the leverage stabilizes on different average 0-leverages Φ_{avg}^{0}. Since the demands equal the channel capacities, we can analyze this experiment

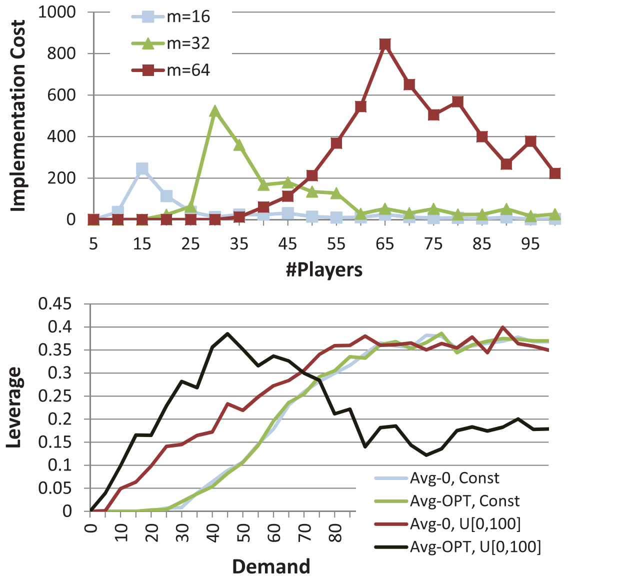

Fig. 3. (a) Implementation cost as a function of the number of players, for different number of channels and uniformly distributed channel capacities. (b) Average leverages as a function of the players' demands with both constant and uniformly distributed channel capacities ([0,100]).

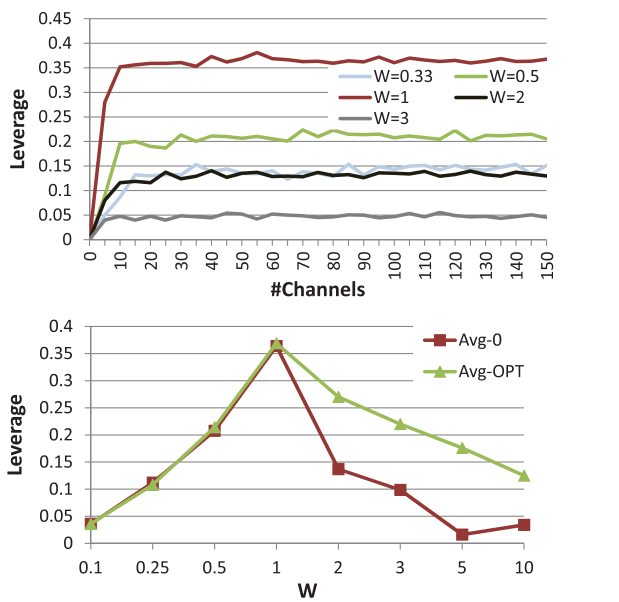

Fig. 4. (a) 0-Leverage Φ^0_{avg} as a function of W, with constant channel capacities. (b) Leverages as a function of W, with demands adjusted appropriately to keep the load at saturation throughput. Channel capacities are uniformly distributed in [0,100].

formally. The social optimum is to have at least one player on every channel and since this is a Nash equilibrium, this can be 0-implemented by Fact 5.5. In an average non-dominated outcome, however, every player will select one of its channels independently at random (if channel capacities are constant. Hence, the expected utility $U(UDom_{avg})$ in this case is the number of channels with at least one player. This can be expressed as $U(UDom_{avg}) = n(1-\frac{1}{m})^n = n(1-\frac{W}{n})^n$. The resulting Φ^0_{avg} is therefore $\Phi^0_{avg} = \frac{n-n(1-\frac{W}{n})^n}{n} \approx 1 - \frac{W}{e^W}$. This is maximized for $W = 1$ leading to $\Phi^0_{avg} = 1 - 1/e$, which is the value obtained in the simulations.

Leverage at Saturation Throughput: Finally, we want to understand the behavior at saturation throughput. For this purpose, again use the above definition of W, but now set the players' demands to $D = Cap/W$, where Cap is the (constant) channel capacity of all channels. That is, we keep the total *load* on the network fixed, but increase/decrease the number of players/demand. Figure 4(b) shows that initially, both Φ^0_{avg} and Φ^{Opt}_{avg} behave similarly. However, after reaching a peak at about $W = 1$, Φ^0_{avg} drops off more sharply. The reason is that because Φ^0_{avg} is not allowed to make any payments, it often loses out on optimizations that can implement Φ^{Opt}_{avg} cheaply as the number of players increases.

IX. Conclusion

In view of ever-growing and ever more highly decentralized networks and distributed systems, there is a growing need to prevent the potentially harmful outcomes of selfish behavior. The key tool that enables such incentives is mechanism design. Unfortunately, monetary payments are often an integral component in such approaches, which severely limits their applicability in many decentralized systems.

This paper takes a different approach. It sheds light on the possibilities and limitations of *leverage*, a mechanism design that does *not* require any payments. We find it intriguing that while often not achieving as good a performance as payment-based schemes, many improvements can theoretically be achieved even without payments in our setting.

References

[1] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. In *45th FOCS*, 2004.

[2] K. Arrow. *Social Choice and Individual Values*. Wiley, 1951.

[3] R. Aumann. Subjectivity and Correlation in Randomized Strategies. *J. Mathematical Economics*, 1974.

[4] R. Eidenbenz, Y. A. Oswald, S. Schmid, and R. Wattenhofer. Manipulation in Games. In *18th International Symposium on Algorithms and Computation (ISAAC)*, 2007.

[5] R. Eidenbenz, Y. A. Oswald, S. Schmid, and R. Wattenhofer. Mechanism Design by Creditability. In *1st International Conference on Combinatorial Optimization and Applications (COCOA)*, 2007.

[6] J. D. Hartline and T. Roughgarden. Optimal Mechanism Design and Money Burning. In *40th STOC*, 2008.

[7] H. Levin, M. Schapira, and A. Zohar. Interdomain Routing and Games. In *40th STOC*, 2008.

[8] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer. Free Riding in BitTorrent is Cheap. In *5th HotNets*, November 2006.

[9] D. Monderer and M. Tennenholtz. k-Implementation. In *4th ACM Conference on Electronic Commerce (EC)*, 2003.

[10] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge, 2007.

[11] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do Incentives Build Robustness in BitTorrent? In *4th NSDI*, 2007.

[12] M. Satterthwaite. Strategy-proofness and Arrow's Condition: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions. *J. Economic Theory*, 1975.

[13] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker. DDoS Defense by Offense. In *ACM SIGCOMM 2006*, 2006.

[14] M. Walfish, J. Zamfirescu, H. Balakrishnan, D. Karger, and S. Shenker. Distributed Quota Enforcement for Spam Control. In *3rd NSDI*, 2006.

[15] W. Wang, X.-Y. Li, and Y. Wang. Truthful Multicast in Selfish Wireless Networks. In *Proc 10^{th} MOBICOM*, 2004.

[16] S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang. On Designing Incentive-Compatible Routing and Forwarding Protocols in Wireless Ad-Hoc Networks. In *Proc 11^{th} MOBICOM*, Cologne, Germany, 2005.