Congestion Games in Caching Enabled Heterogeneous Cellular Networks

Vineeth S. Varma and Tony Q. S. Quek
Singapore University of Technology and Design, Singapore

***Abstract*—In this work, a heterogeneous cellular network with caching enabled at the small cell aggregators is analyzed. The caching technology is enabled to help offload some of the popular traffic requests onto the cache. In this framework, a network congestion game is formulated to study the evolution of traffic when every user as an independent decision maker decides its choice of communication, i.e, via the macro cell or the caching enabled small cells. To study the effect of the cache on the network delay, a numerical simulation is implemented. The file popularity is modeled using the well known Zipf-distribution and the network delay is studied as a function of the cache buffer size and network traffic. In summary, these results can be utilized by network operators to deploy infrastructure efficiently.**

I. Introduction

Motivation: The increased demand of high-data-rate services and requirement of ubiquitous access calls for innovative solutions that can drastically improve the data rates offered by networks. Augmenting an existing macro-cell by deploying a high density small cell (SC) network, i.e., deploying a heterogeneous cellular networks (HCNs), is considered as an efficient solution to resolve the problem of heavy data demands [1]–[3]. While the deployment of an HCN potentially allows for a higher data rate on the wireless end, the data rate is often restricted through the backhaul links and more significantly through the data server links. Data caching, i.e., storing frequently accessed data within each SC or SC aggregator, is a novel approach that could significantly reduce backhaul bandwidth required and also improve the quality of service to users [4]–[6].

Typically data caching in the network edge can be implemented at either the SC base stations or at the SC aggregator[1]. As the small cell network is highly dense, it is potentially more efficient to implement the cache at the aggregator end, in terms of cost efficiency. In this work, the focus is on the caching system implemented at the aggregator. When an HCN is deployed, the users have a choice on which network they utilize in order to access their targeted data. Users can decide between the caching enabled SC network or the macro network. This choice results in a competitive game between all the users involved. In this work, the resulting game is modeled as a congestion game, while taking into account the possibility of data being cached at the SC aggregator.

Related works: Congestion games, first introduced by Rosenthal (1973) [7], have been extensively studied in literature and applied to various scenarios like traffic control problems and information networks. In the wireless literature, several works use game theory to predict user behavior and model systems [8]–[10]. However, to the best of our knowledge, studying the competition between users in a caching enabled HCN is a novel problem. The results of this work can be used to design HCN infrastructural and networking parameters in a more efficient manner when caching is enabled at the SC aggregator.

Organization: This paper is structured in the following format. In Section II, we present the system model and define the proposed performance metric. In Section III, we conduct an analytic study of the proposed congestion game while Section IV provides many numerical results to sustain the proposed approach. Finally, we conclude the paper and some possible extensions to this work are provided.

II. System Model

We consider a HCN with a macro BS working in parallel with several SCs. The SCs are connected to an aggregator where data can also be cached. We assume a simple file model where all files are of the same size ϕ. The file popularity of file n is given by $q(n)$ where $n \in \mathcal{N} = \{1, 2, ..., N\}$. We are interested in the situation where the independent reference model (IRM) can be

[1]The aggregator is the node where multiple neighboring SC base stations are connected via weaker backhaul links (for cost efficiency). The aggregator is subsequently connected to the core network through a stronger backhaul link.

ISBN 978-3-901882-68-5

applied. Adopting the IRM implies that this popularity stays a constant through a large period of time. Each of these files are stored in servers connected to the internet. We assume that each server is independently connected to the internet with links that offer peak rates of B_{S_n} (files per second). As each server is accessed by several such HCN's simultaneously, the B_{S_n} that we use for a single HCN (as studied in this work) has to be modified based on the total number of HCNs using the same server given by N_c. For example, a video file stored in the Singapore server of the respective website would be accessed by hundreds of networks deployed all over the country.

Let the total traffic generated by users in a cycle be given by T in unit files. This traffic is assumed to be generated by the users periodically every τ seconds which is the cycle duration. Since each SC serves very few users, the transfer time from the SC aggregator to the users is a constant δ_F seconds independent of the traffic T_F. As the macro serves many users, the transfer time from the BS to the user depends on the traffic through the macro BS T_M and the peak rate offered B_M (files per second). We assume the backhaul from the macro BS to the internet to be quite large and so the transfer time is again assumed to be a constant δ_M seconds. However, the backhaul from the aggregator to the internet (or core network) is slower and is given by B_F. To compensate for this, we also assume that the aggregator can store data by caching.

The buffer at the cache has a maximum capacity given by C (files) or $C\phi$ bits. We assume that the cache operates with the least recently used (LRU) replacement policy. That is, when a request for a file absent in the cache arrives, the least recently used file in the buffer is replaced by the new request (after being downloaded through the backhaul). In this work, since the **file popularity is assumed to be unknown** (to the users and to the network), the LRU caching mechanism is a natural choice. If the requested file is available in the cache, the request can be granted with a very small delay of δ_C [2]. The net delay over a path is calculated as the maximum delay incurred across all the elements in the path (this model is most suitable for file streaming or large data downloads where the file is split into smaller parts and is transferred).

[2]These delays δ_F,δ_S and δ_M are due to network architecture that cause small delays even when the backhaul is large.

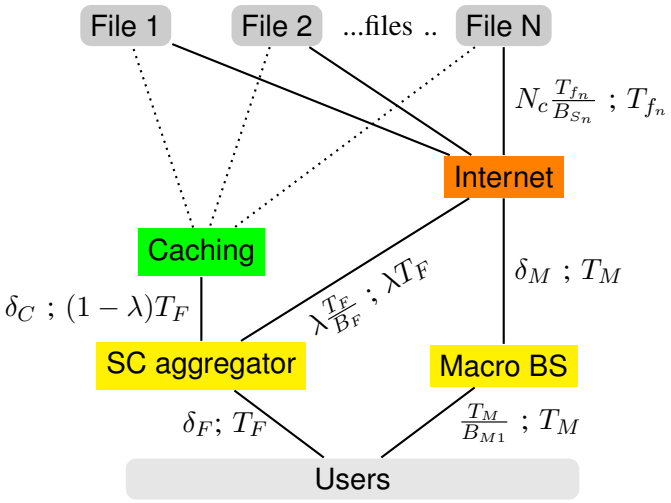

Fig. 1. Congestion game with caching: Information on the delay and traffic through a link are labeled below it in the format (delay;traffic).

III. The Congestion Game

In Fig. 1 we present the schematic of the proposed congestion game. Here, $1-\lambda$ is the average hit rate for requests arriving at the cache. We also take the traffic through the server corresponding to file n to be T_{f_n}. If we assume that the hit rate of file n is $h(n)$ and $\sum_m q(m)h(m) = 1-\lambda$, then we can calculate the traffic through server n as:

$$T_{f_n} = q(n)N_c(T_M + T_F(1-h(n))) \tag{1}$$

where N_c is the number of cells in the region accessing the server for file n.

We now make use of the popular Che approximation for the hit rate in LRU caching buffers [11]:

$$h(n) = 1 - \exp(-q(n)\kappa_C) \tag{2}$$

where κ_C is the unique solution of:

$$\sum_n (1 - \exp(-q(n)\kappa_C)) = C \tag{3}$$

In a link X with backhaul B_X and traffic T_X, the delay is given by $\frac{T_X}{B_X}$. Now, we can proceed to define the congestion game by defining:

1) The set of congestible elements $\mathcal{E} = \{U \to M, U \to F, F \to C, F \to I, M \to I, f_1, f_2, \ldots, f_N\}$ where $X - Y$ represents the congestible element between node X and node Y. The nodes are as defined in the schematic: U the users, M the MC, F the SC aggregator, C the cache, I the internet node and f_n the server with file n.

2) The players are the users generating each traffic file, i.e $\mathcal{T} = \{1, 2, ...T\}$.
3) The set of strategies which are subsets of $\mathcal{E}$. Here clearly there are only two strategies possible $\mathcal{S} = \{\alpha, \beta\}$ where α corresponds to download via the SC and β to download via the MC.
4) For each element $e \in \mathcal{E}$ the traffic or load is given by T_e.
5) For the strategy α the net delay for file type n can be calculated as:

$$d_\alpha(n, \mathbf{T}) = h(n) \max(\delta_F, \delta_C) + (1 - h(n)) \max\left(\delta_F, \frac{\lambda T_F}{B_F}, \frac{T_{f_n}}{B_{S_n}}\right)$$

and

$$d_\beta(n, \mathbf{T}) = \max\left(\frac{T_M}{B_M}, \delta_M, N_c \frac{T_{f_n}}{B_{S_n}}\right) \quad (4)$$

It is a well known result that congestion games being potential games have at least one pure Nash equilibrium (NE). Hence, the next task is to study the NE and its efficiency.

A. The Nash equilibrium

In a hypothetical case where each user is aware of the popularity of his file request n and the probability of it being cached, the user's would be able to pick the strategy specific to n. However, based on our system model, these values are unknown at the user or network end. Therefore the **users can observe only the average delays** along paths α and β. Hence, the NE will be at the point where both of these paths have the same average delay. First, we will calculate the delay along each path when downloading a file n, when x users out of T opt to use the SC path as:

$$d_\alpha(n, x) = h(n) \max(\delta_F, \delta_C) + (1 - h(n)) \max\left(\delta_F, \frac{\lambda x}{B_F}, \frac{q(n) N_c (\lambda x + T - x)}{B_{S_n}}\right)$$

Similarly, the delay along the macro station path is:

$$d_\beta(n, x) = \max\left(\frac{T - x}{B_M}, \delta_M, \frac{q(n) N_c (\lambda x + T - x)}{B_{S_n}}\right) \quad (5)$$

Let $T_F = x^*$ be a possible NE solution, then x^* can be found by solving (6) [7].

$$\mathbb{E}_n \left[d_\alpha(n, x^*) - d_\beta(n, x^*)\right] = 0 \quad (6)$$

B. Equilibrium analysis

In this subsection we would like to analyze the NE obtained from (6). For this purpose, we make some assumptions on the delay parameters without much loss in generality. Note that these assumptions are just used for the simplification of the calculations and the following results can be proved even without them, but will involve a more convoluted, yet straightforward proof.

1) The primary delay when the file is found in cache is the backhaul; $\delta_C < \delta_F < \frac{\lambda x}{B_F}$.
2) The macro backhaul has a very low delay; $\delta_M < \frac{T-x}{B_M}$.

With these assumptions, we can prove that:

Theorem 1: The NE of the defined congestion game is unique and (6) has a unique solution x^*.

Proof: For this, we re-write (6) as:

$$\mathbb{E}_n : (1 - h(n)) \max\left(\frac{\lambda x}{B_F}, N_c \frac{q(n)(\lambda x + T - x)}{B_{S_n}}\right) - \max\left(\frac{T - x}{B_M}, N_c \frac{q(n)(\lambda x + T - x)}{B_{S_n}}\right) + h(n)\delta_F = 0 \quad (7)$$

Now we study the summation over n which can be split into $(1 - \lambda)\delta_F + P_1 + P_2 + P_3$ defined by these threshold values

$$n'(x) : \frac{\lambda x}{B_F} = N_c \frac{q(n')(\lambda x + T - x)}{B_{S_n}} \quad (8)$$

and

$$n''(x) : \frac{T - x}{B_M} = N_c \frac{q(n'')(\lambda x + T - x)}{B_{S_n}} \quad (9)$$

The first part of the summation is independent of x, and so we focus on the other parts. P_1 is when $n < n', n''$. In this case, the term is (assuming ordered files $q(n) \geq q(n + 1)$):

$$P_1 : \sum_{1}^{\min(n', m'')} -h(n) q(n) N_c \frac{q(n)(\lambda x + T - x)}{B_{S_n}} \quad (10)$$

As $\lambda < 1$ this term is clearly increasing in x.

The term P_3 is when $n > n', n''$. In this case, the term is:

$$P_3 : \sum_{\max(n', n'')}^{N} q(n) N_c \left[\frac{\lambda x}{B_F} - \frac{T - x}{B_M}\right] \quad (11)$$

Clearly, this term is also increasing in x.

Finally, if $n' > n''$, the term P_2 is when $n' > n > n''$. In this case, the term is:

$$P_2 : \sum_{n''}^{n'} N_c q(n) \left[(1 - h(n)) q(n) \frac{q(n)(\lambda x + T - x)}{B_{S_n}} - \frac{T - x}{B_M}\right] \quad (12)$$

Since by definition of $n > n''$, $\frac{T-x}{B_M} > N_c \frac{q(n'')(\lambda x + T - x)}{B_{S_n}}$, we can conclude that this term is also increasing in x by studying $\frac{\partial P_3}{\partial x}$. For the case of $n'' > n'$ it can be trivially seen that all terms increase in x.

Therefore, we have shown that $\mathbb{E}_n \left[d_\alpha(n,x) - d_\beta(n,x) \right]$ is a strictly increasing function of x. Therefore if $\mathbb{E}_n \left[d_\alpha(n,x^*) - d_\beta(n,x^*) \right] = 0$, this point x^* is unique.

■

C. Quality of Service

Normally, the quality of service (QoS) criteria of such a network could be considered to be the average expected delay, i.e.:

$$\bar{\delta}(x) = \frac{\mathbb{E}_n \left[x d_\alpha(n,x) + (T-x) d_\beta(n,x) \right]}{T} \tag{13}$$

However, there is also another possible QoS metric that can be studied, which is the expected maximum delay, i.e., the maximum delay encountered on average while downloading a file, maximized over all possible files and all possible paths.

$$\hat{\delta}(x) = \max[\max_n \{ d_\alpha(n,x) \}, \max_n \{ d_\beta(n,x) \}] \tag{14}$$

This metric is highly relevant as this delay can cause the network to become potentially unstable. It can be seen that when $\hat{\delta}(x) > \tau$, new users arrive into the network before the existing T users are finished downloading. As the users downloading this n-th file (with maximum delay), always encounters a higher delay on average, these users can potentially accumulate in the network destabilizing it [3].

D. Price of Anarchy

We can define the global optimum for the average expected delay QoS criteria as $\min_x \bar{\delta}(x)$. Thus the price of anarchy (PoA) is given by:

$$PoA_1 = \frac{\bar{\delta}(x^*)}{\min_x \bar{\delta}(x)} \tag{15}$$

where x^* is the user allocation to the SC path at the NE. Similarly, when the QoS criteria is the expected maximum delay, the price of anarchy can be calculated as:

$$PoA_2 = \frac{\hat{\delta}(x^*)}{\min_x \hat{\delta}(x)} \tag{16}$$

[3]The user data traffic is simply given by $\frac{\phi T}{\tau}$ in bits per second.

IV. Numerical Results

So far we have analyzed the NE and PoA from a mathematical perspective. However, in this section, we are interested in the practical application of our calculations. Specifically, we would like to consider a typical file popularity distributions and study the performance of the NE with various values of key parameters like the cache capacity C and the number of HCNs per server N_c.

Some of the general parameters used are:

1) File size $\phi = 100$ MB and $N = 10^6$ files.
2) The file distribution is a Zipf style distribution with 10 files sharing the same rank, i.e the 11-th to 20th file is half as popular the 1st to 10th file, the 21st to 30th file is one third as popular as the 1st to 10th file etc. (total of N files).
3) Delays $\delta_F = 1$ s, $\delta_M = 1$ s and $\delta_C = 1$ s.
4) Bandwidths $B_M = 10$ files per second or 1 GBps and $B_F = 5$ files per second or 500 MBps.
5) Bandwidth of each server is taken to be a constant 10 files per second. The number of cells sharing the same server $N_c = 1000$ unless otherwise indicated.
6) Number of users per cycle $T = 300$. A higher traffic condition would be represented by a smaller value for τ like 60 seconds, whereas a low traffic condition could be represented by τ = 180 seconds for example.

A. Cache performance

Based on the Che approximation and numerical calculations, we plot the average hit rate of the cache $(1 - \lambda)$ against the Cache Capacity C in Fig. 2. From the figure, it can be seen that an average hit rate of 50% is achievable with $C = 40dB$, i.e, 10,000 files. As each file is assumed to be of 100 MB each, the scale on the X-axis can be seen as from a minimum cache capacity of 1 GB (10 dB) to a maximum at 10 TB (Terabytes).

B. QoS: expected average delay

We plot the average delay $\bar{\delta}(x^*)$ at the NE against the Cache Capacity C in Fig. 3. From the figure, it can be seen that when the servers are under a higher load due to a larger number of cells accessing them (higher N_c), the improvement in the average delay is more pronounced due to a higher cache capacity. In Fig 4., the PoA for this QoS is plotted and it is interesting to see that the PoA is in both cases quite close to unity, i.e., the average delay at the NE is very close to the delay at the global

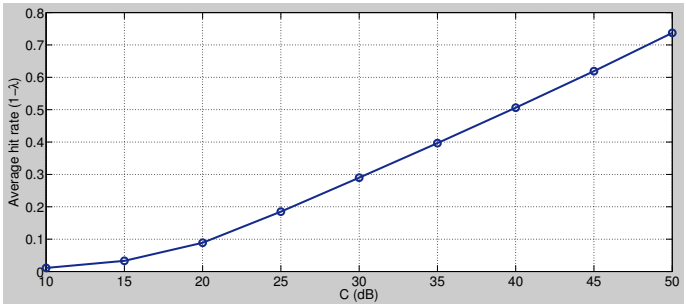

Fig. 2. Average hit rate of the cache $(1-\lambda)$ against the Cache Capacity C in dB.

optimum. Thus, the congestion game can be seen as "efficient" in terms of average delay.

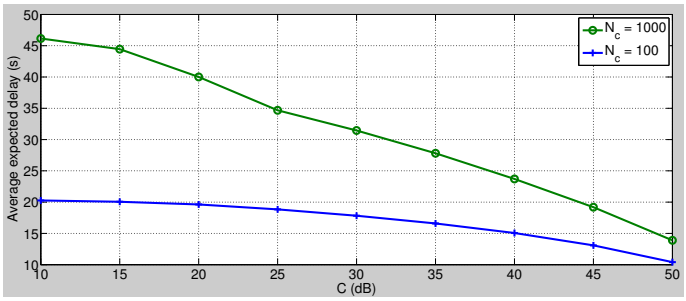

Fig. 3. Plotting the average delay $\bar{\delta}(x^*)$ at the NE against the Cache Capacity C.

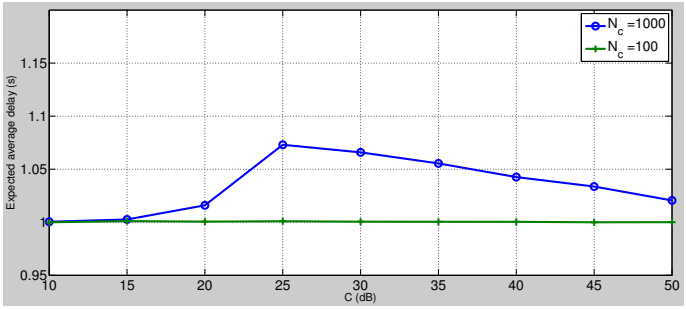

Fig. 4. Plotting the price of anarchy PoA_1 against the Cache Capacity C.

In Fig. 5, the population share for the SC at the NE x^* is plotted against the cache capacity C. From the figure, it can be seen that when the servers are under a higher load (higher N_c), a higher population of users would tend towards choosing the SC, and this population share would grow with the cache capacity.

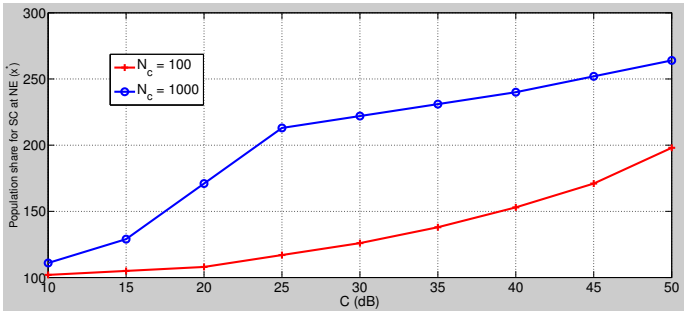

Fig. 5. Plotting the population share at the NE (on the SC) x^* against the Cache Capacity C.

C. QoS: expected maximum delay

We plot the maximum expected delay $\hat{\delta}(x^*)$ at the NE against the Cache Capacity C in Fig. 6. Similar to the the case of average delay, it can be seen that when the servers are under a higher load, the improvement in the maximum delay is also more pronounced due to a higher cache capacity. As seen from the figure, when $N_c = 1000$, the network becomes congested even for the lower traffic case where $\tau = 180$ s. In Fig. 7, the PoA for this QoS is plotted against the cache capacity. Unlike the case of average delay, the PoA is in fact quite large for both cases of high and low load on the servers ($N_c = 100$ or 1000). Surprisingly, it can be seen from the figure that switching off the macro BS can actually help improve the maximum expected delay when ($N_c = 1000$). Of course, in practice this would imply that if the macro-BS stop providing "heavy" data services (like video streaming) and only provide light data traffic (such as messaging), the users will be forced to demand heavy data traffic through the SCs resulting in a lower maximum expected delay.

Remark. 1: Note that, the parameters in this numerical simulation are only approximations or guesses of the real values. The parameters $N-c$, T and τ are all interrelated. A high traffic time (peak hours in the day) could be represented by $N_c = 1000$ or by a large T. A larger N_c implies that the servers are more heavily loaded due to higher global traffic, while a higher T corresponds to a higher local and global traffic.

Remark. 2: The result that actually switching off the macro BS (i.e., the heavy data traffic through the macro BS) can effectively reduce the maximum delay is a result similar to the situation of Braess's paradox in road traffic [12]. Intuitively, this can be explained as users will naturally try to access data through the macro BS when then observe higher average download speeds via the macro network. However, the lower server

speeds cause the congested files to adversely affect the maximum delay.

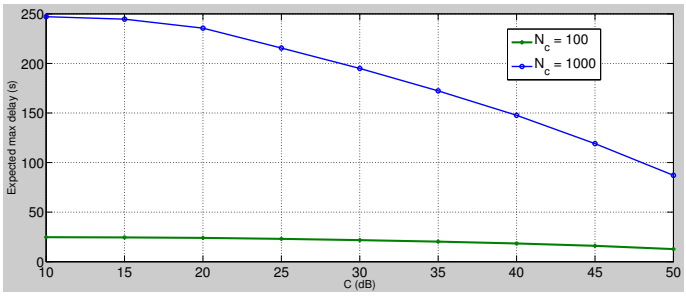

Fig. 6. Plotting the maximum expected delay $\hat{\delta}(x^*)$ at the NE against the Cache Capacity C.

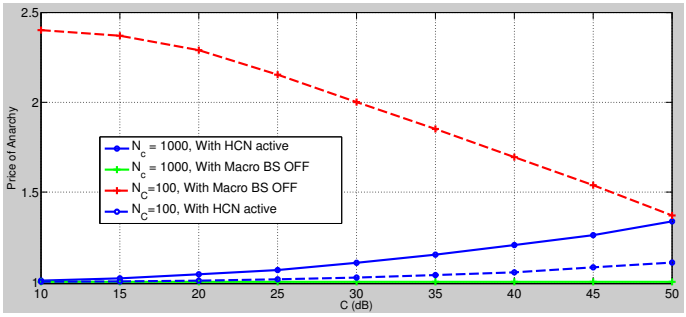

Fig. 7. Plotting the price of anarchy PoA_2 against the Cache Capacity C.

V. Conclusion

This work has studied the congestion game, resulting from the heterogeneous network, where a user can chose to access a file through the caching enabled SC network, or through the macro network. We have shown that the resulting congestion game has a unique Nash equilibrium. This result ensures that the user behavior is predictable once the other parameters are known. Additionally after an extensive numerical analysis, we have studied the impact of the caching mechanism on the delay experienced by users. Surprisingly, it can be seen that in some specific cases, the expected maximum delay can be reduced by actually blocking users from downloading their requested files through the macro network.

This work has shown promising results and is ripe for many extensions. If we abandon the IRF (the assumption that file popularity is independent of time), the caching mechanism will become much more complicated and interesting. The hit rates of the new LRU cache would also depend on the data traffic as having a higher traffic of users would increase the update rate of the cache. This would imply that the resulting game will have a hit rate dependent on the user traffic which in turn depends on the hit rate. Other possible extensions would include accounting for multiple service providers, caching enabled at the SC base station level etc.

References

[1] T. Q. S. Quek, G. de la Roche, I. Güvenç, and M. Kountouris, *Small cell networks: Deployment, PHY techniques, and resource management*. Cambridge University Press, 2013.

[2] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, June 2011.

[3] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. Thomas, J. Andrews, P. Xia, H. Jo, H. Dhillon, and T. Novlan, "Heterogeneous cellular networks: From theory to practice," *IEEE Commun. Magazine*, vol. 50, no. 6, pp. 54–64, June 2012.

[4] T. Imielinski and B. Badrinath, "Mobile wireless computing: challenges in data management," *Communications of the ACM*, vol. 37, no. 10, pp. 18–28, 1994.

[5] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 5, pp. 821–834, 2012.

[6] E. Bastug, J.-L. Guénégo, and M. Debbah, "Proactive small cell networks," in *Telecommunications (ICT), 2013 20th International Conference on*. IEEE, 2013, pp. 1–5.

[7] R. W. Rosenthal, "A class of games possessing pure-strategy nash equilibria," *International Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, 1973.

[8] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, and L. Wynter, "A survey on networking games in telecommunications," *Computers & Operations Research*, vol. 33, no. 2, pp. 286–311, 2006.

[9] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 1, pp. 104–116, 2005.

[10] S. Lasaulce and H. Tembine, *Game theory and learning for wireless networks: fundamentals and applications*. Academic Press, 2011.

[11] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for lru cache performance," in *Proceedings of the 24th International Teletraffic Congress*. International Teletraffic Congress, 2012, p. 8.

[12] P.-D. D. D. Braess, "Über ein paradoxon aus der verkehrsplanung," *Unternehmensforschung*, vol. 12, no. 1, pp. 258–268, 1968.