Base-Station Assisted Device-to-Device Communications for High-Throughput Wireless Video Networks

Negin Golrezaei, Andreas F. Molisch, Alexandros G. Dimakis
Dept. of Electrical Eng.
University of Southern California
emails: {golrezae,molisch,dimakis}@usc.edu

***Abstract*—We propose a new scheme for increasing the throughput of video files in cellular communications systems. This scheme exploits (i) the redundancy of user requests as well as (ii) the considerable storage capacity of smartphones and tablets. Users cache popular video files and - after receiving requests from other users - serve these requests via device-to-device localized transmissions. We investigate what is the optimal collaboration distance, trading off frequency reuse with the probability of finding a requested file within the collaboration distance. We show that an improvement of spectral efficiency of one to two orders of magnitude is possible, even if there is not very high redundancy in video requests.**

I. Introduction

One of the main drivers of wireless (cellular) data traffic in the near future will be mobile video. According to recent estimates [1] the traffic generated by video delivery requests will quickly outpace mobile web content and lead to an increase in wireless data traffic by two orders of magnitude. This in turn will put enormous strain on already-overburdened cellular networks.

Traditional methods for increasing cellular capacity are (i) improvement of the physical-layer link capacity between transmitter and receiver, (ii) use of additional spectrum, and (iii) decrease of the cell size (including the use of pico- and femto-cells), to improve the area *spectral efficiency*. However, the first of these methods is hampered by the fact that fourth-generation cellular systems use a physical layer (MIMO-OFDM with near-capacity-achieving codes) that is already close to the theoretical limits; the second method suffers from a dearth of available spectrum, and the last one from the high costs of establishing new cell sites and providing the associated backhaul capacity [2].

In this paper, we describe a novel architecture to improve the throughput of video transmission in cellular networks, based on caching of popular video files in cellphones and base station controlled device-to-device communications.[1] Our architecture exploits the large storage available on modern smartphones to cache video files that might be requested by other users. Base stations keep track of the cache content and direct requests to the nearest smartphone that has the desired file, which is then transmitted via a device-to-device link. Since the distance between requesting user and smartphone with the stored file will be small in most cases, multiple device-to-device links can be operated on the same time/frequency resources within one cell. This in turn leads to a dramatic increase in spectral efficiency.

We introduce the fundamental ideas and furthermore provide an approximate analysis of such a system, based on a subdivision of a macrocell into virtual clusters, such that one device-to-device link can be active within each cluster. The cluster size is a key parameter of the system, and can be controlled by the transmit power of the mobile terminals. Finding its optimum value involves a tradeoff between two counter-running effects: (i) a small cluster size means a high frequency reuse, i.e., more D2D links can be operated simultaneously within one cell (ii) a large cluster size increases the probability that a user actually finds the file it requests in its vicinity, and thus a D2D can meaningfully established, i.e., the cluster is *active*.

We also provide an experimental evaluation section based on real-world video popularity distributions taken from [5]. From these results, we find that improvements of the video throughput by one to two orders of magnitude are possible. Our conclusion is that the proposed scheme is a promising way to alleviate capacity bottlenecks in cellular systems when there is high demand for few popular video files.

II. A new architecture for cellular video

We propose a radical new approach to solving the video bottleneck in cellular systems that is based on the following two key observations: a large amount of video traffic is caused by a few, popular, files and storage is a quantity that increases faster than any other component in communications/processing systems:

- The popularity of video files is very unevenly distributed. ”Viral” YouTube videos, movies that are newly available for rental, and reports from recent sports events are typical examples of highly popular videos, which account for a considerable percentage of all video traffic. In current cellular networks, the video is downloaded by each requesting user via the base station of the cellular network, which wastes precious spectral resources.
- Recent years have seen an enormous proliferation of smartphones and tablets that have anywhere between 10 and 64 GByte of storage (not to mention the 500 GByte on typical laptop harddisks), which is commonly under-utilized.

Based on these observations, we introduce *a device-to-device (D2D) architecture where the mobiles are used as caching storage nodes*. Our architecture relies on using this storage capacity to cache video files that might be requested

[1] An alternative, complementary, approach to dealing with the backhaul bottleneck was recently proposed by us in [3] [4], where femto-base stations are replaced by small base stations with high storage capacity but low backhaul capacity.

by other users. Storage of the content could occur either when the smartphone requests this file for its own user; when it overhears a transmission to another user, or as dedicated ”background” downloading at times with low traffic load in the network.

The stored files are transmitted, upon request, to a user requesting a particular file. The transmission occurs by D2D communications; since the distance between transmitting and receiving device is much shorter than between device and base station, multiple device-to-device links can be operated on the same time/frequency resources within one cell. This in turn leads to a dramatic increase in spectral efficiency. The base station keeps track of which phone has which files stored. Thus, when a user requests a certain video file, the base station can direct it to the nearest smartphone that has the file stored, which is then transmitted via a device-to-device link, and can optimize the frequency reuse between the devices.

Users can collaborate by caching popular content and utilizing local device-to-device communication when a user in the vicinity requests a popular file. By enabling device-to-device communications, the ensemble of mobile devices can become a distributed cache that allows a much more efficient downloading. Furthermore, storage allows users to collaborate even when they do not request the same content *at the same time*. This is a new dimension in wireless collaboration architectures beyond relaying and cooperative communications.

III. Model and Setup

Assume a cellular network where each cell/base station (BS) serves n users. For simplicity we assume that the cells are square, and we neglect inter-cell interference, so that we can consider one cell in isolation. Users are distributed uniformly in the cell. Every user is assumed to have a storage capacity called cache, which is filled up with some video files. We suppose that every user can store one file. This assumption has the advantage of yielding a clean formulation; however, our work can be easily extended to larger cache size.

The cell is divided into smaller areas called cluster, see figure 1. All clusters are square with equal area. As can be seen from figure 1, the cell side is normalized to 1 and cluster side is equal to r. We call r *collaboration distance*.

Since wireless D2D involves short-range communications, within each cluster, device-to-device (D2D) communications are allowed, i.e., users in each cluster can communicate with each other locally; however to avoid intra-cluster interference, only one such communication per cluster is allowed. We furthermore assume that such communications does not introduce interference for other clusters, and that all clusters have the same (square) size. Clearly this model is oversimplified, as it does not account for the inter-cluster interference, the fact that the pathloss coefficients might be different in different parts of the cell, and fading. However, our results indicate, it captures a fundamental tradeoff in D2D collaboration and as shown in our preliminary experiments can give very high gains. We furthermore assume that the D2D communication does not interfere with communication between BS and users; this assumption is justified if the D2D communication occurs in a separate frequency band (e.g., WiFi). Note that this is a first step to study such a new architecture in cellular network, using a number of simplifying assumptions. In our future work we will relax those assumptions, see, e.g., [7].

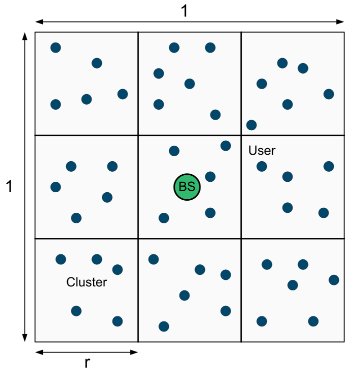

Fig. 1. An example of the single-cell layout.

In the best case, D2D communication takes place simultaneously in all clusters and the number of parallel links is equal to the total number of clusters in the cell, i.e., $\frac{1}{r^2}$. But, in general, not all of these parallel links can be established. A parallel link can be established if and only if any user in a cluster can find its request files in the cache of at least one other users in the cluster.

To be specific, the system works as follows: if a user requests one of the files stored in neighbors’ caches in the cluster, neighbors will handle the request locally through D2D communication; otherwise, the BS should serve the request. As a result, the probability that D2D communication is done in clusters depends on what users store. We can imagine that users in each cluster have access to a central virtual cache (CVC) filled up with the stored files in users’ caches in the cluster. For filling up the CVC, we can exploit the redundancy in requests. More popular files will be requested more. Thus, if users store the most popular files, it is more probable that file requests are serviced locally. In other words, if popular content is stored in that way, most of the traffic for popular content can be relegated to local D2D communication.

We assume that if there are k users in a cluster, each of them caches one of the k most popular files without repetition, which means that CVC is filled up with the k most popular files. Actually, this is the best way of storing files since in each cluster, there is no overlap in users’ caches and users can access the best possible set of files locally. Any other way of assigning files results in worse performance; the performance results obtained here are thus an upper bound [2]. Storing files in this way can be achieved through central control by the BS. Of course caches might need to be updated if users move from one cluster to another; however, if most users are quasi-static, they will not change their clusters often, and the overhead for updating users’ caches becomes negligible.

For the popularity distribution of the video files, we assume

[2] Further work [6], [7] shows that scaling laws (how the capacity improves as a function of the number of users) do not change when random caching is used.

Zipf distribution. This means that the frequency of the ith popular file, denoted by f_i, is inversely proportional to its rank:

$$f_i = \frac{\frac{1}{i^\gamma}}{\sum\limits_{j=1}^{m} \frac{1}{j^\gamma}}, \quad 1 \leq i \leq m, \tag{1}$$

where m is the number of files. Note that our results hold even if m scales as a function of n as one would naturally expect larger populations to have a broader spectrum of requests. The exponent γ characterizes the distribution by controlling the relative popularity of files. The larger γ means that most of popularity weights are concentrated in the first few popular files. The Zipf distribution has been established in numerous studies as being a good approximation to measured popularity of video files [8].

If D2D communication occurs in a cluster, we call it *active*. The problem that we investigate in the next section is the choice of the optimum r or equivalently the optimum number of clusters in the cell such that the expected number of active clusters is maximized. The expected number of active clusters is a suitable measure of spectral efficiency because it indicates the average number of parallel links in the cell. The number of parallel links may be increased by allowing short collaboration distance. Thus, we also have more potential clusters (parallel links). But, it is not always the case, since by having a short collaboration distance, users have small number of neighbours which decreases the chance of finding the request content in the neighbours' caches. Thus, we need to find the optimum tradeoff between the number of possible parallel links in the cell and the probability of finding the requested content within a cluster.

IV. Finding the optimal collaboration distance

In this section, we find the optimum r for given values of size of the library m and the number of users in the cell n.

We define a binary random variable a_j for cluster j such that a_j is equal to 1 if the cluster j is active; otherwise, it is equal to 0. The total number of active clusters in the cell, denoted by A, equals to:

$$A = \sum_j a_j. \tag{2}$$

Since a_j is a binary random variable, the expectation of a_j is the probability that cluster j is active, i.e., D2D communication takes place in cluster j. Since users are uniformly distributed in the cell and all clusters have equal area, the expectation of a_j denoted by $E[a_j]$ does not depend on j, i.e., $E[a_j] = E[a]$ for any j where $E[a]$ is the probability that any cluster is active. Thus, from (2), the expected number of active clusters is given by:

$$E[A] = \sum_j E[a_j] = \frac{1}{r^2} E[a], \tag{3}$$

where $\frac{1}{r^2}$ is the number of clusters in the cell. The probability that a cluster is active depends on number of users in the cluster which is denoted by K. Therefore, $E[a]$ can be written as:

$$E[a] = \sum_{k=0}^{n} E[a|K=k] \Pr[K=k], \tag{4}$$

where $E[a|K=k]$ is the probability that a cluster is active provided that there are k users in the cluster. $\Pr[K=k]$ is the probability that the number of users in the cluster is k. The probability that a user is located in the cluster is ratio of the cluster area to the cell area. Thus, the number of users in the cluster is binomial random variable with parameters n and r^2, i.e., $K = B(n, r^2)$. The probability that there are k users in the cluster equals to:

$$\Pr[K=k] = \left(\begin{array}{c} n \\ k \end{array} \right) (r^2)^k (1-r^2)^{n-k}, \tag{5}$$

where $\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{(n-k)!k!}$.

$E[a|K=k]$ defined in (4) is the complement of the probability that no D2D communication takes place in the cluster. D2D communication is possible if at least one of k users in the cluster can access to its request file in the cache of other users. Thus, $E[a|K=k]$ can be written as:

$$E[a|K=k] = 1 - \Pr[u_1 = 1 \cap u_2 = 1 \cap ... \cap u_k = 1], \tag{6}$$

where $\Pr[.]$ represents the probability. The u_i for $i = 1, ..., k$ is a binary random variable that is 1 if the user i cannot find its request file in the neighbors' caches in the cluster, i.e., in the CVC excluding the file in ith user's cache. In general, if the CVC of the cluster uses a *random* caching strategy, u_i and u_j for $i \neq j$ will be *dependent* random variables. In that case, given that user i requests according to the some popularity distribution and it cannot find it in the CVC, it is more likely that what is currently in the CVC is not popular files. So, if user j requests independently from user i but according to the same popularity distribution, it is more probable that user j also cannot find its request file in the CVC in the cluster. However, according to our assumptions, the CVC in the cluster is *deterministic* and users request are independent from each other. Therefore, the random variables u_i and u_j for $i \neq j$ are independent. Thus, we can simplify Eq. (6) as:

$$E[a|K=k] = 1 - \prod_{i=1}^{k} \Pr[u_i = 1]. \tag{7}$$

Without loss of generality, we assume that user i caches the ith most popular file. Note furthermore the possibility of *self-requests*, i.e., a user might find the file it requests in its own cache; in this case clearly no D2D communication will be activated by this user. Accounting for these self-requests,

$$\Pr[u_i = 1] = 1 - (P_{CVC}(k) - f_i), \tag{8}$$

where f_i is the request frequency of the ith popular file and is given in (1). $P_{CVC}(k)$ is the probability of hitting the CVC and is given by

$$P_{CVC}(k) = \sum_{i=1}^{k} f_i, \quad 1 \leq k \leq m, \tag{9}$$

It is obvious that $P_{CVC}(k)$ for $k > m$ is equal to 1. From (7) and (8), we find:

$$E[a|K = k] = 1 - \prod_{i=1}^{k} (1 - (P_{CVC}(k) - f_i)). \tag{10}$$

Substituting $E[a|K = k]$ in (4), we get:

$$E[a] = \sum_{k=0}^{n} \left(1 - \prod_{i=1}^{k} (1 - (P_{CVC}(k) - f_i)) \right) \Pr[K = k], \tag{11}$$

where $Pr[K = k]$ is given in (5). From (3), (4), and (11), the expected number of active clusters can be written as:

$$\begin{aligned} E[A] &= \frac{1}{r^2} \sum_{k=0}^{n} E[a|K = k] \Pr[K = k] && \text{(12a)} \\ &= \frac{1}{r^2} \sum_{k=0}^{n} \left(1 - \prod_{i=1}^{k} (1 - (P_{CVC}(k) - f_i)) \right) \Pr[K = k]. && \text{(12b)} \end{aligned}$$

Notice that $Pr[K = k]$ is a function of r. Thus, $E[A]$ is a function of one variable r. To find the maximum expected number of active clusters or equivalently the maximum average spectral efficiency, we should take a derivative of $E[A]$ in respect with r. While finding r_{opt} analytically in closed form does not seem feasible, numerical solutions are possible with very low effort, as we require a root search of a function of one variable within the interval $0 < r < 1$.

In the above, we have optimized the number of active clusters. An alternative criterion is the minimization of the download time. While the active clusters ignore self-requests, the "minimum download time" considers self-requests as positive, since they imply that users get files with zero delay.

$$\text{maximize} \quad (n - n_{self} - E[A])\omega_{BS} + E[A]\omega_{D2D} \tag{13}$$

where n_{self} is the average number of users that get their desired file with zero delay through self- requests and $n_{self} = \frac{1}{r^2} \sum_{k=0}^{n} P_{CVC(k)} \Pr[K = k]$. ω_{D2D} and ω_{BS} are the average download time through D2D communications and the BS, respectively.

V. Experimental Evaluation

In this section, we provide some numerical results how system parameters affect r_{opt}. We consider there are n active users which are randomly distributed across the entire cell. There is a library of size m. As stated before, the popularity distribution of files follows a Zipf's law with exponent γ. In all figures except those we want to consider effects of γ, we assume γ to be 0.6; this value is based on a study conducted on the University of Massachusetts Amherst campus in 2008 [5].

Figures 2 and 3 show the effects of size of the library on the optimum collaboration distance and the maximum average number of D2D links. The optimum collaboration radius and the maximum average number of active clusters are respectively illustrated in figures 2 and 3. In both figures, $n = 500$. As the size of the library increases, users request from a larger set of files and there is more diversity in requests. The probability that these diverse requests hit the CVC in the cluster decreases. So, by increasing m, users want to be surrounded by more neighbours in the clusters or equivalently to access to more files locally from the CVC. Hence, as it can be seen from figures, by increasing m, the optimum r increases and the maximum average number of active clusters decreases.

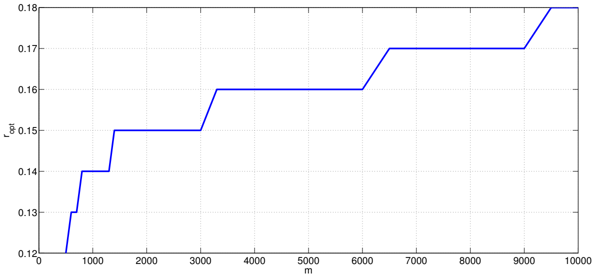

Fig. 2. The optimum radius versus the size of the library m for $\gamma = 0.6$ and $n = 500$.

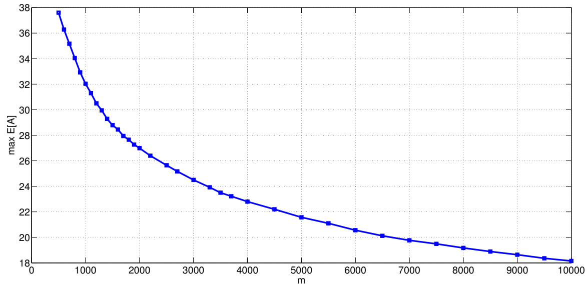

Fig. 3. The maximum average number of active clusters versus the size of the library m for $\gamma = 0.6$ and $n = 500$.

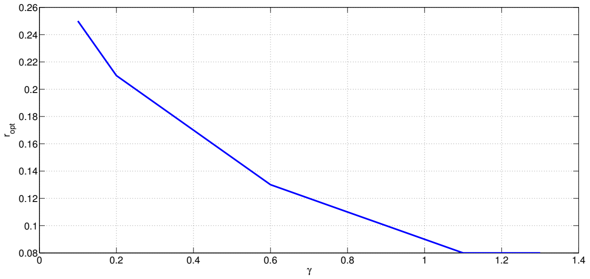

Fig. 4. The optimum collaboration distance versus γ for $n = 500$ and $m = 1000$.

The effects of changing the Zipf distribution exponent are investigated in figures 4, 5 and 6. In all figures, the number of users in the cell is 500 and the number of files is 1000. Figure 4 and figure 5 respectively show the optimum collaboration distance and maximum average number of active clusters versus γ. We can observe from figures that by increasing γ, r_{opt} decreases and the maximum $E[A]$ increases. For the small

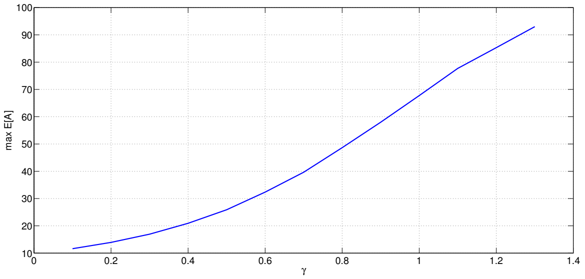

Fig. 5. The maximum average number of active clusters versus γ for $n = 500$ and $m = 1000$.

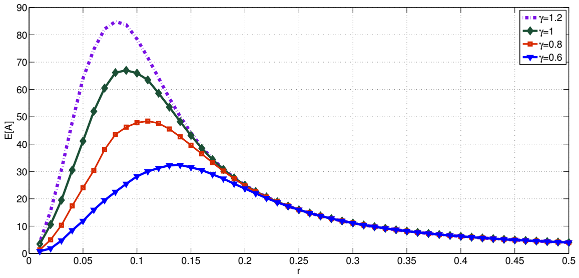

Fig. 6. The average number of active clusters versus r for $n = 500$, $m = 1000$ and different values for γ.

γ, there is little redundancy in the users requests; moreover, the popularity weights of few most popular files, i.e., files in the CVC, are not significant. So, to increase the chance of having D2D communication within a cluster, the collaboration distance r should increase. Hence, the maximum expected number of active clusters decreases for small γ. In short, more redundancy in video requests results in higher spectral efficiency. Figure 6 shows the average number of active cluster versus r. It illustrates how selecting the appropriate r is important to have high spectral efficiency for example for $\gamma = 0.6$, if we choose $r = 0.05$, we can have approximately 10 active clusters; however, when $r = r_{opt} = 0.14$, we will have 30 active clusters. Moreover, we can see that r_{opt} decreases as γ increases.

Figures 7 shows the effects of number of users in the cell. For larger n, the cell is more dense and a user is surrounded by more neighbours given a collaboration radius. Thereby, users can find their desired files within a short distance. This allows us to decrease the size of clusters while we are sure that with high probability there will be enough users in each cluster to collaborate with each other. Thus, we see significant improvement in number of active clusters by increasing n.

VI. Summary and conclusions

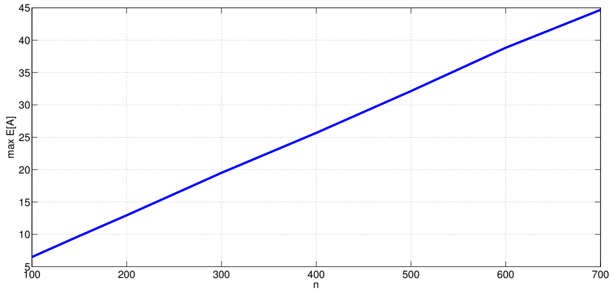

Fig. 7. The maximum average number of active clusters versus n for $\gamma = 0.6$ and $m = 1000$.

We proposed a novel scheme for increasing the efficiency of video content delivery in cellular communications systems. Our scheme exploits (i) the redundancy of user requests as well as (ii) the considerable storage capacity of smart phones and tablets. Users cache popular video files and - after receiving requests from other users - serve them via device-to-device localized transmissions. We optimized the collaboration distance of the D2D communications, and investigated the impact of the popularity distribution of video files. Our preliminary experimental evaluation shows that performance improvements of one to two orders of magnitude are possible.

Since our approach does not require any additional infrastructure it can be easily implemented with minimal costs. The main requirement is creating the incentives of participation for users. One incentive is higher download rates since files obtained from other local devices are transmitted with higher data rates compared to the base station; a tit-for-tat mechanism [9] can be used to motivate collaboration. Further, network operators could provide incentives for users to donate storage and bandwidth, especially during peak demand hours. A more thorough investigation of these issues remains as future work.

Another topic of future work involves lifting the simplifying assumptions of this paper. In particular, we are interested in investigating non-coordinated and randomized (instead of deterministic) caching strategies, as well as take inter-cluster interference into account. Further, a more fundamental understanding of the benefits of D2D collaboration and caching in a scaling law sense is part of on-going work.

References

[1] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537 /ns705/ns827/white_paper_c11-520862.html.

[2] V. Chandrasekhar, J. G. Andrews, and A. Gatherer,"Femtocell networks: a survey," *IEEE Commun. Mag.*, 46(9):59 – 67, Sept. 2008.

[3] N. Golrezaei, K. Shanmugam, A.G. Dimakis, A.F. Molisch and G. Caire, "FemtoCaching: wireless video content delivery through distributed caching helpers", accepted in INFOCOM 2012.

[4] N. Golrezaei, K. Shanmugam, A.G. Dimakis, A.F. Molisch and G. Caire, "Wireless video content delivery through coded distributed caching", accepted in ICC 2012.

[5] *http://traces.cs.umass.edu/index.php/Network/Network*.

[6] N. Golrezaei, A.G. Dimakis and A.F. Molisch, "Asymptotic throughput of base station assisted device-to-device communications", to be submitted for publication.

[7] N. Golrezaei, A.G. Dimakis and A.F. Molisch, "Wireless device-to-device communications with distributed caching", submitted for publication.

[8] M. Cha, H.Kwak, P. Rodriguez, Y.Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system", Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, 1–14, 2007.

[9] B. Cohen, "Incentives build robustness in BitTorrent" *Workshop on Economics of Peer-to-Peer systems*, Vol. 6, 68–72, 2003.

[10] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks", IEEE Communications Magazine, Vol. 47, No. 12, 42–49, 2009.