On File Sharing Over a Wireless Social Network

Yi-Ting Chen, Constantine Caramanis, and Sanjay Shakkottai
Department of Electrical and Computer Engineering
The University of Texas at Austin
Email: {yiting.chen, caramanis, shakkott}@mail.utexas.edu

***Abstract*—We consider the problem of broadcasting a large file over a wireless network (e.g., students in a campus). If each user who wants the file must download it from the carrier's WAN, dissemination time scales linearly. Two often-occurring facts suggest we can do better: (a) the demand for the file often spreads via a social network (e.g., Facebook); and (b) the devices predominantly used are GPS enabled, and equipped with a peer-to-peer (ad hoc) transmission mode. The premise of this paper is that (a) and (b) are often the case. Starting from here, we consider this coupled-network problem (demand on the social network; bandwidth on the wireless network) and taking advantage of the fact that the two networks have different topologies, we propose a file dissemination algorithm. In our scheme, users query their social network to find geographically nearby friends that have the desired file, and utilize the underlying ad hoc network to route the data via multi-hop transmissions. We show that for many popular models for social networks, the file dissemination time scales sublinearly with the number of users.**

I. Introduction

The proliferation of mobile devices that can stream video (laptops, smartphones, tablets) has marked a dramatic increase in demand for streaming video. At the same time, content generation and dissemination has become dramatically easier – most phones have installed video-cameras, and knowledge of a video can spread extremely rapidly to vast numbers of people, through social networks including e-mail, Facebook, Twitter, and the like. As deployed capacity approaches saturation, we need new transmission architectures to guarantee our wireless networks continue to deliver traffic effectively and efficiently.

This paper addresses precisely this problem. More specifically: we consider the simple, yet increasingly common setting, where a user (e.g., a student on a college campus) generates a large file (a short video, for example) and wants to spread it to her social network – her friends, their friends, and so on. In the current paradigm, the file creator uploads the file to a central server (e.g., YouTube) and then spreads word of its existence through Facebook, Twitter, etc. Upon learning of the file's existence, interested (we call them "eager") users then download the file from the server, using their provider's wide area network (WAN). Since the WAN has bounded bandwidth, the file dissemination time will necessarily scale linearly in the number of users who ultimately receive the file. Particularly in a dense setting like a college campus, this inherently limited centralized scheme for file dissemination may be highly suboptimal. The central question in this paper is: how much better can we do?

Increasingly, smartphones and similar technology, are equipped with both GPS and peer-to-peer transmission modes. In dense environments, this opens the possibility of forming a wireless ad hoc network in which users communicate with each other through several hops of short distance transmissions. As shown in Gupta and Kumar's seminal work [1], the spatial capacity of a wireless ad hoc network scales as $\sqrt{n}$ – a sharp contrast to the fixed capacity of a WAN. While this scaling spatial capacity of ad hoc networks provides a potential way forward, naive implementation presents severe problems that may leave us worse off than the currently implemented WAN solution. We may have severe congestion caused by subsets of users getting a high number of requests, hence resulting in hot-spots in the network. This will occur, for instance, if users request the file from neighbors on their social network, as most social networks exhibit the presence of super-nodes with very high degree. This is particularly true in the broadcast setting we have here, when we expect there to be such hot spots, which can potentially reduce network capacity by a significant factor [2].

A. Main contributions

In this paper we propose a simple and distributed file dissemination algorithm that works by passing messages through the social network, and requires limited communication and computation overhead. In particular, we verify our algorithm through analysis on popular models for social networks (power law graphs). The main features of our algorithm are as follows:

1) Load balancing: users receiving a large amount of requests distribute them to nearby users on the social network, in such a way that we can guarantee no user has to serve more than six other users. Our algorithm achieves $\sqrt{n}$-scaling with the number of users receiving the file – sublinear, in sharp contrast to the linear scaling required in the WAN file dissemination architecture.
2) Exploiting geographic proximity: We extend our load-balancing algorithm to exploit geographic proximity. Because of the structure of the social network, we show that by searching a few hops deeper in their social network, most users are able to download the file from another user at close range. This idea allows us to further reduce the scaling below $\sqrt{n}$, depending on the depth of the social-network a user may search.

This work was supported in part by NSF Grants CNS-0721380, EFRI-0735905, and CNS-0831580, DTRA grant HDTRA1-08-0029, and the DARPA ITMANET program.

B. Related work

Analyzing the capacity of an ad hoc network has drawn much attention since Gupta and Kumar's work [1]. Since then, much literature, e.g. [3]-[9], has studied this issue for more general settings. Indeed, [3][4] have studied the multicast capacity, [5][6] have considered mobility models, and [7]-[9] have worked on different attenuation regions and/or different topologies. Different from the above, our paper focuses on the effect of the social network on the scaling of the file dissemination time in wireless ad hoc networks.

Exploiting characteristics of social networks to enhance system performance has been suggested for resource scarce networks such as delay-tolerant networks which often suffer intermittent connectivity and long delays. Papers in this field, e.g., [10]-[12], show that social-aware algorithms are efficient to solve broad problems including routing and information propagation. However, most papers in this field only provide simulation based verification. In contrast, our results are supported by quantitative analysis, and our algorithm has provable guarantees in a scaling-law sense.

II. System description

In this section we describe the basic system model, including the model for the wireless network and the placement of the nodes, and the model for the social network.

A. Random wireless network and Gaussian channel model

We model our network as n static nodes, placed independently and uniformly on a square of width $\sqrt{n}$. Thus the (expected) density of the network stays constant. Each node has a transmitter and a receiver. All nodes can communicate with each other with fixed power P. The interference model is described by a Gaussian channel model defined below [7].

Definition 1: (Gaussian channel model [7]) Index nodes by 1, 2, $\ldots, n$. Let x_i be the location of node i. Let $\mathcal{A}$ be the set of active transmitters at this time instant. The transmission rate $R(x_i, x_j)$ from node i to node j is

$$R(x_i, x_j) = \log\left(1 + \frac{P\ell(x_i, x_j)}{N_0 + \sum_{k\in\mathcal{A}\setminus\{i\}} P\ell(x_k, x_j)}\right). \quad (1)$$

Here, $\ell(x, y)$ represents the power attenuation function between points x and y on the square, and is given by

$$\ell(x_i, x_j) = \min\left\{1, \frac{e^{-\gamma||x_i-x_j||}}{||x_i - x_j||^\alpha}\right\} \quad (2)$$

where as usual, $||x - y||$ is the Euclidean distance between x and y.

In this paper, as in [7], we consider either $\gamma > 0$ or $\gamma = 0$ and $\alpha > 2$.

B. Model for social networks

As we identify users with their devices (e.g. cell phones/ PDA), the n nodes in the wireless network also form a social network. A social network is described as a graph $G = (V, E)$ where V is the set of nodes with cardinality n and E is the set of edges. Two nodes are joined by an edge if (and only if) the corresponding users are friends in the social network. The distance between two nodes x and y on the social-graph G is the minimum number of hops between x and y in the social network. Thus a node's neighbors are the nodes one hop away on the social graph, and its k-neighborhood are the nodes within k hops on the social graph. A key property we exploit is that distance between two nodes on the social network is generally unrelated to geographic distance between the corresponding users in the wireless network.

We consider social networks generated by random power law graphs [14]. A graph G is called a power law graph with parameter β if the number of nodes with degree k is proportional to $k^{-\beta}$. Empirical studies of many social (and other) networks have shown them to satisfy so-called power law graph structure (see e.g. [15] [16]). These random graphs satisfy an important property: with overwhelming probability, each node has only a small number of neighbors, i.e., small degree, (small relative to the size of the overall network) and the diameter of the random graph (the maximum number of hops between the vast majority of the nodes) is also small. This property is consistent with properties of most social networks, and in particular, with the famous observation known as the small world phenomenon, first discussed in [13].

Following standard practice, we generate random graphs and in particular random power law graphs, according to expected degree sequences [14].

Definition 2: ([14]) Let $w = (w_1, w_2, \ldots, w_n)$ be an expected degree sequence satisfying $\max\{w_k^2\} \le \sum_{1\le k\le n} w_k$. We say $G = (V, E)$ is a random graph generated by the degree sequence w if edge $(i, j) \in E$ is present with probability $w_i w_j / \sum_{1\le k\le n} w_k$. Given a subset $S \subseteq V$, following [14], we define the volume of S to be $\mathrm{vol}(S) = \sum_{i\in S} w_i$, *i.e.*, the sum of weights of nodes in S. Similarly, define $\mathrm{vol}_k(S) = \sum_{i\in S} w_i^k$ and $\tilde{d} = \mathrm{vol}_2(G)/\mathrm{vol}(G)$.

Definition 3: ([14]) A random graph generated by Definition 2 is a random power law graph with parameter β, average degree $\bar{d}$ and maximum expected degree M if w_i is chosen by

$$w_i = c(i_o + i)^{-1/(\beta-1)}, \quad (3)$$

where $c = \frac{\beta-2}{\beta-1}\bar{d}n^{1/(\beta-1)}$ and $i_0 = n\left(\frac{\bar{d}(\beta-2)}{M(\beta-1)}\right)^{\beta-1}$.

C. Assumptions on system parameters

In this paper, we consider social networks with the properties of a random power law graph with $\beta > 3$ (many graphs have this property, see, e.g., the collaboration graphs in [16]). We assume that the minimum expected degree is $m = K\log(n)$ where K is a constant greater than 10, and the maximum expected degree is M, satisfying $\log^2(n) \ll M \ll \sqrt{n}$. Thus, almost all nodes are in the largest component and the diameter of the graph is $D \approx \log_{\tilde{d}}(n)$ [14].

The transmission time consists of two parts: propagation delay and file receiving time. The propagation delay is the time required to receive the first bit since the start of the transmission. The file receiving time is the time required to finish the transmission since then. For simplicity, we assume the file length F is large, and we ignore the propagation delay in the analysis.

III. Algorithm

At some initial time, the file generator (*the source*) creates the file, and advertises it on her social network. At any given time, a node either has the file (*active node*), knows about the file and wants it because a social-network neighbor has it (*eager node*), or is oblivious to its existence (*inactive node*).

The algorithm proceeds in three phases. In the Requesting Phase, eager nodes use their social network to request the file from active nodes – if knowledge of geographic location is available, nodes favor (geographically) nearby active nodes. In the Scheduling Phase, again the social network is used to schedule a sequence of transmissions whereby each eager node is assigned a transmission node from which it will obtain the file. In the Transmission Phase, nodes transmit the file to their appointed requestors, employing established routing techniques [7]. This final third phase is conceptually distinct from the first two phases, and it is important to emphasize this point here. The routing techniques used are independent of the social network structure, and follow the multi-hop ad hoc network protocols described in [7].

A. Algorithm

Our algorithm takes the input as the diameter of the social network, D, as well as two parameters which we specify: ϵ, and $\mathcal{L}$, whose roles are as follows. Nodes are allowed to search for another node in the social network from which to download the file, at a distance of at most $2\epsilon D + 1$ hops away. Thus if $\epsilon = 0$, they cannot look beyond a single hop away, and if $\epsilon = 0.5$, they have access to the entire social network. Thus the parameter ϵ controls the search depth. The parameter $\mathcal{L}$ is used to exploit geographic proximity: most nodes will download the file from nodes that are at a geographic distance of at most $\mathcal{L}$. If nodes have no notion of geography, we set $\mathcal{L} = \infty$, hence all nodes are within $\mathcal{L}$. Otherwise, we set $\mathcal{L}$ to a smaller value.

Given parameters $(\epsilon, \mathcal{L}, D)$ as described above, the algorithm finds active nodes from which eager nodes can download the file. This is accomplished through coordination through the social network. The main idea is the following: eager nodes send requests to one of their social-network neighbors with the file. Since a single node may get many such requests, it does not serve all of them, but rather finds other active nodes nearby in the social network to serve them, and also enlists the receiving nodes themselves to forward along the file.

When $\mathcal{L}$ is set to a non-infinite value, it may not always be possible for nodes to obtain the file from geographically proximate neighbors – for instance, suppose the generator has no neighbors in her geographic proximity. In such cases, we allow file transfers that exceed geographic distance $\mathcal{L}$, and these happen from two or one-hop neighbors on the social network. We call transfers within geographic distance $\mathcal{L}$, $\mathcal{L}$-transfers, and all other transfers S-transfers, since they are near in the social-network distance. Similarly we refer to $\mathcal{L}$-requests and S-requests.

ALGORITHM 1:

Input: parameter ϵ, distance threshold $\mathcal{L}$, and the diameter of the social network D.

Requesting Phase: Consider an eager node, x, at time t.

Step 1: Let $\mathcal{N}_x(t)$ denote node x's $2\epsilon D + 1$-neighborhood in the social-graph at time t. Let $\mathcal{N}_x^{\mathcal{L}}(t) \subseteq \mathcal{N}_x(t)$ be the set of nodes in $\mathcal{N}_x(t)$ that have the file and whose Euclidean (geographic) distance to x does not exceed $\mathcal{L}$.

Step 2: If $\mathcal{N}_x^{\mathcal{L}}(t)$ is not empty, x sends an $\mathcal{L}$-request to a randomly picked node in $\mathcal{N}_x^{\mathcal{L}}(t)$.

Step 3: If $\mathcal{N}_x^{\mathcal{L}}(t)$ is empty and the distance from x to the source on the social-graph is smaller than $\epsilon D + 1$, then x sends an S-request to a one-hop neighbor in the social-graph which has the file.

Step 4: Otherwise, x waits and goes back to step 1 at time $t + 1$.

Scheduling Phase: Consider an active node y. It maintains two balanced binary trees, an $\mathcal{L}$-tree and an S-tree, constructed from its $\mathcal{L}$-requests and S-requests, respectively. It builds these trees by adding requesting nodes sequentially, as the requests arrive.

When node y receives an $\mathcal{L}$-request, node y adds the eager node to the $\mathcal{L}$-tree and asks its parent on the tree to deliver the file, and similarly for S-requests.

Transmission Phase: An eager node waits until the node designated as its transmitting node in the Scheduling Phase has the file. It then sets up a wireless transmission, and routes data through a highway system described in [7]. Note that the transmitter will have to serve at most 6 nodes: 2 from its own $\mathcal{L}$-tree, 2 from its own S-tree, and 2 from the tree it joins when it is an eager node (which could be either an $\mathcal{L}$-tree or an S-tree). Thus, we divide a time slot into six and each transmitter serves all nodes in a round robin fashion.

IV. Performance analysis

In this section, we assume nodes can search for an active node in their $2\epsilon \log_{\bar{d}}(n) + 1$-neighborhood. Here, we set $\epsilon < 1/10$, thus allowing nodes to search a neighborhood that is large, but nevertheless a vanishing fraction of the size of the entire network. Increasing the size of neighborhoods should allow nodes to find geographically proximate active nodes more easily. Accordingly, we set $\mathcal{L} = 8\sqrt{n^{1-\epsilon'} \log(n)/\sigma\pi}$ for any $\epsilon' < \epsilon$. Setting $\epsilon = 0$ reduces to the case where nodes ignore, or are oblivious to, social network geography. For $\epsilon > 0$, nodes take advantage of knowledge of the social network.

In the following two theorems, we provide upper and lower bounds on the file dissemination time, as a function of ϵ. Even for $\epsilon = 0$, we show that our algorithm's load-balancing is enough to achieve dissemination time that scales as $\sqrt{n}$. Allowing $\epsilon > 0$ enables us to further reduce dissemination time by an additional factor of $n^{\epsilon/2}$, by exploiting geography. The sketches of the proofs are presented in Section VI, and the detailed proofs can be found in [17].

Theorem 4: Suppose the source is chosen uniformly at random from the nodes in the largest component, and the file length is F. Suppose the parameter $\mathcal{L}$ is set as described above. Then the file dissemination time under Algorithm 1 with parameter $0 \leq \epsilon < 0.1$ is

$$\mathcal{O}(\sqrt{n^{1-\epsilon'}}\log^{2.5}(n)F), \tag{4}$$

for any $\epsilon' < \epsilon$ with high probability.

Remark 5: Significantly, our algorithm can be applied to more general social network structures as long as we set an appropriate parameter $\mathcal{L}$. For example, given a graph G with diameter ℓ_{max} and maximum degree d_{max}. If nodes are only allowed to search for the file from one-hop neighbors, i.e., nodes can not exploit geography, we can set $\mathcal{L} = \infty$. Thus, the file dissemination time is $\mathcal{O}(\sqrt{n}\log(d_{max})\ell_{max}F)$ with a proof following immediately from the proof of Theorem 4.

We next give an *algorithm independent* lower bound in the following theorem. To prove the lower bound, we place no restrictions on computation or communication overhead. Moreover, we make (overly) optimistic assumptions throughout in order to guarantee a bound. For instance, we assume nodes only download from their nearest social-network neighbors. We show in the next theorem that our results are comparable to the lower bound.

Theorem 6: Consider the file dissemination problem under the setting described above. Let F be the file length. Then, for any algorithm that only allows nodes to download the file from their $4\epsilon\log_{\tilde{d}}(n)+2$-neighborhoods, the file dissemination time is lower bounded by

$$\Omega(n^{1/2-2\epsilon-\xi}F), \tag{5}$$

with high probability for any $\xi > 0$.

V. Conclusions

In this paper, we consider a simple, low-overhead file dissemination algorithm that exploits peer-to-peer capabilities of many smartphones and similar devices, and, critically, exploits the social network that spreads knowledge of the file. We give a load-balancing algorithm that uses the social network to schedule transmissions so that spatial-capacity of the ad hoc network is exploited without creating congestion or hot spots. We show that dissemination time scales like $\sqrt{n}$ – significantly slower than the linear time for WAN. Then, we show that if nodes have knowledge of geographic position, this can be exploited to further decrease file dissemination time. Finally, we show in both cases that our algorithm performs close to an algorithm-independent lower bound.

VI. Sketch of proofs

A. Proof for Theorem 4

To show Theorem 4, we need the next two lemmas which characterize the local behavior of random power law graphs. Specifically, we are interested in how the size of neighborhoods of nodes in the largest component grows. We show that for any node in the largest component, the number of nodes in a small neighborhood grows like a factor $\tilde{d}$ if we explore one more step. We prove this by providing upper and lower bounds that only differ by a factor of n^{ξ} for any $\xi > 0$. The detailed proofs can be found in [17] which are based on some results in [14].

Lemma 7: Consider a random power law graph satisfying the assumptions in Section II-C. Then, there are at least $\sigma n^{\epsilon'}$ nodes in a node's $\epsilon\log_{\tilde{d}}(n)$-neighborhood with probability $1 - o(n^{-1})$, for any $\epsilon' < \epsilon < 0.4$ where σ is a constant depending on β and K.

Lemma 8: Consider a random power law graph satisfying the assumptions in Section II-C, and suppose we have $\epsilon < 0.4$. Consider a node either picked randomly or with weight smaller than W. Then, with probability at least $1 - \mathcal{O}(\log^{-1}(n))$, for any $\epsilon' > \epsilon$ and any fixed constant λ, there are at most $2W\tilde{d}^{\lambda}n^{\epsilon'}/\log(n)$ nodes in this node's $\epsilon\log_{\tilde{d}}(n) + \lambda$-neighborhood, where

$$W = \begin{cases} \log^{\beta/\beta-3}(n) & \text{if} \quad 3 < \beta \leq 4 \\ \max\{\log^{5/\beta-4}(n), \log^2(n)\} & \text{if} \quad 4 < \beta \end{cases} \tag{6}$$

Remark 9:
1) These two lemmas show that the size of a node's neighborhood increases by a factor roughly equal to $\tilde{d}$ if we explore the neighborhood one more step. To understand the intuition, we can look at the expected increasing factor. Consider a small set S. Let X_i be the indicator function that node i is a neighbor of the set S. Then, the expected sum weight of neighbors of S is roughly $\sum_i w_i\mathbb{P}(X_i = 1)$. Thus, we have

$$\begin{aligned}\sum_i w_i\mathbb{P}(X_i = 1) &= \sum_i w_i\frac{w_i\text{vol(S)}}{\text{vol(G)}} \\ &= \text{vol(S)}\frac{\text{vol}_2\text{(G)}}{\text{vol(G)}} = \text{vol(S)}\tilde{\text{d}}.\end{aligned}$$

   This shows that the sum of weights of neighbors of S increases by a factor $\tilde{d}$ in expectation.
2) Notice that the result in Lemma 8 is weaker as the failure probability is $\mathcal{O}(\log^{-1}(n))$ compared to $\mathcal{O}(n^{-1})$ in Lemma 7. The result is reasonable, because there exist super nodes (nodes with large expected degree) in power law graphs. Therefore, one can see the neighborhoods of nodes close to super nodes are expected to grow much faster than a factor of $\tilde{d}$.

With the above lemmas of local behavior of random power law graphs, we provide a sketch of the proof of Theorem 4. The full details of the proof can be found in [17].

Sketch of the proof of Theorem 4: The proof consists of three parts. In the first part, we analyze the transmission rates when all nodes can find a proper active node as described in Algorithm 1. We next identify a set of potential active nodes for each node. Finally, we show the theorem by induction.

We first find a lower bound on the transmission rates when all nodes follow Algorithm 1. Instead of determining the transmission rates, we estimate an upper bound on the number of flows through each relaying node. Since each relaying node adopts TDM to serve all flows, the transmission rates are lower bounded by the inverse of that upper bound. As described in Algorithm 1, flows through a relaying node can be classified into $\mathcal{L}$-transmissions and $\mathcal{S}$-transmissions. Note that the distance between a transmitter and a receiver of an $\mathcal{L}$-flow is smaller than $2\mathcal{L}$. This implies any $\mathcal{L}$-flow passing through the relaying node must have either a transmitter or a receiver in a strip of area $\Theta(\mathcal{L})$ which is the responsible service rectangle described in [7] for the relaying node. Therefore, at most $\mathcal{O}(\mathcal{L})$ $\mathcal{L}$-flows pass through the relaying node. On the other hand, the number of $\mathcal{S}$-transmissions through any relaying node is roughly $\mathcal{O}(n^{\epsilon})$, since the receivers of $\mathcal{S}$-transmissions must be within $\epsilon \log_{\tilde{d}}(n)$ hops from the source on the social network, and there are at most $\mathcal{O}(n^{\epsilon})$ such receivers by Lemma 8. Hence, the total number of flows through any relaying node is $\mathcal{O}(\mathcal{L})$, and the transmission rate is greater than $1/c\mathcal{L}$ for some constant c.

Now, we have to show that each node can find an active node satisfying the conditions in Algorithm 1. In fact, we show that any node x with distance to the source greater than $\epsilon \log_{\tilde{d}}(n)$ can request to a node y which is both close in the social-graph and the wireless-square. Specifically, we require that node y satisfies the following

1) y is in the $2\epsilon \log_{\tilde{d}}(n)+1$-neighborhood of x in the social-graph.
2) The distance from y to the source on the social network is smaller than that from x to the source.
3) The Euclidean distance from x to y on the wireless-square is smaller than $\mathcal{L}$.

This holds with high probability, because by Lemma 7 there are roughly $\Theta(n^{\epsilon})$ nodes satisfying 1) and 2) above. Since all nodes satisfying 1) and 2) are placed uniformly on the wireless-square, we can find a node y satisfying the above three. An important consequence of the existence of node y is that nodes at distance k from the source can request to valid active nodes when all nodes at distance at most $k-1$ from the source have the file.

We now show the theorem by induction on k: the distance from a node to the source on the social network. Specifically, let t_k be the time all nodes at distance at most k from the source can receive the file. We show that $t_k \leq ck\log(n)\mathcal{L}F$. Note that the base case clearly holds. We can assume this is true for t_{k-1}. At time t_{k-1} all nodes at distance k from the source are eager and can request to valid active nodes. Therefore, these nodes have to wait at most $\log(n)-1$ successful transmissions before starting to receive the file, because the depth of a binary tree is smaller than $\log(n)$. Hence, $t_k \leq t_{k-1} + c\log(n)\mathcal{L}F$ as the transmission rates are greater than $1/c\mathcal{L}$, and the theorem follows as the diameter of the social network is $\mathcal{O}(\log(n))$. ■

B. Proof for Theorem 6

In the following, we provide a sketch of the proof of the lower bound of file dissemination time (Theorem 6).

Sketch of the proof of Theorem 6: The main idea of the proof is to show a fraction of the nodes cannot find neighbors which are close both in the social-graph and the wireless-square. We show this holds for nodes with expected degree smaller than $2K\log(n)$. More specifically, for these nodes, we can conclude that almost all of them have small $4\epsilon \log_{\tilde{d}}(n)+2$-neighborhoods, which leads to the claim. Indeed, by Lemma 8, the size of the neighborhoods is roughly bounded by $2W\tilde{d}^2 n^{4\epsilon}/\log(n)$. Moreover, by the fact that nodes are placed uniformly on the wireless-square, with high probability the distance from a node to its closest neighbor should be greater than $\sqrt{n/2\pi W\tilde{d}^2 n^{4\epsilon}}$. We finally show that the above claim holds for almost all nodes simultaneously, by using Chebyshev's inequality to eliminate the dependency between them.

The above analysis suggests that the total sum of bit-meter products required to finish the file dissemination is roughly bounded by $\Omega(n^{3/2-2\epsilon}F)$, since the number of nodes with expected degree smaller than $2K\log(n)$ is $\Theta(n)$. Thus, the theorem follows by observing that the transport capacity (the sum of all bit-meter products the system can transmit in a unit of time) is $\Theta(n)$. ■

REFERENCES

[1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transaction on Information Theory, Vol. 46, No. 2, pp. 388-404, 2000.*

[2] S. Subramanian, S. Shakkottai, and P. Gupta, "Optimal Geographic routing for wireless networks with near-arbitrary holes and traffic," *Proc. IEEE INFOCOM 2008.*

[3] S. Shakkottai, X. Liu, and R. Srikant, "The multicast capacity of ad hoc networks," *ACM MobiHoc 2007.*

[4] S. Li, Y. Liu, and X. -Y. Li, "Capacity of large scale wireless networks under gaussian channel model," *ACM MobiCom 2008.*

[5] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking, vol. 10, no. 4, pp. 477-486, 2002.*

[6] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," *Proc. IEEE INFOCOM 2004.*

[7] M. Franceschetti, O. Dousse, D. Tse, P. Thiran, "Closing the Gap in the Capacity of Wireless Networks Via Percolation Theory," *IEEE Transaction on Information Theory, Vol. 53, No. 3, pp. 1009-1018, 2007*

[8] A. Ozgur, O. Leveque, and D. Tse, "Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks," *IEEE Transactions on Information Theory, vol. 53, no. 10, pp. 3549572, 2007.*

[9] U. Niesen, P. Gupta, and D. Shah, "On Capacity Scaling in Arbitrary Wireless Networks," *IEEE Transactions on Information Theory, vol. 55, no. 9, pp. 3959-3982, 2009.*

[10] E. Daly and M. Haha, "Social network analysis for routing in disconnected delay tolerant manets," *Proc. ACM MobiHoc 2007.*

[11] P. Costa, C. Mascolo, M. Musolesi, and G. -P. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE JSAC, Vol. 26, pp. 748-760, 2008.*

[12] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," *Proc. IEEE INFOCOM 2010.*

[13] S. Milgram, "The small world problem," *Psychology Today, Vol 1, No. 1, pp. 60 V 67, 1967.*

[14] F. Chung and L. Lu, "The average distances in random graphs with given expected degrees," *Internet Mathematics, Vol. 1, No. 1, pp. 91-114, 2004.*

[15] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A, Vol. 311, pp. 590-614, 2002.*

[16] J. Grossman, P. Ion, and R. De Castro, "Facts about Erdös numbers and the collaboration graph," available from the WWW: (http://www.oakland.edu/enp/trivia/), 2003.

[17] Y.-T. Chen, C. Caramanis, and S. Shakkottai, "On file sharing over a wireless social network," *Technical Report, The University of Texas at Austin, May 2011.*