Throughput-Delay Trade-off for Hierarchical Cooperation in Ad Hoc Wireless Networks

Ayfer Özgür, Olivier Lévêque
Faculté Informatique et Communications
Ecole Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland
{ayfer.ozgur, olivier.leveque}@epfl.ch

***Abstract*—Hierarchical cooperation has been shown recently to achieve better throughput scaling than classical multihop schemes in static wireless networks, under certain assumptions on the channel model. However, the end-to-end delay and the complexity of the scheme turns out to be significantly larger than those of multihop schemes. A modification of the scheme is proposed here that reduces complexity and achieves a throughput-delay trade-off $D(n) = (\log n)^2 T(n)$ for $T(n)$ between $\Theta(\sqrt{n}/\log n)$ and $\Theta(n/\log n)$, where $D(n)$ and $T(n)$ are respectiveley the average delay per bit and the aggregate throughput in a network of n nodes. This trade-off complements the previous results on the throughput-delay trade-off $D(n) = T(n)$ of multihop schemes that achieve throughput scaling between $\Theta(1)$ and $\Theta(\sqrt{n})$.**

I. Introduction

Scaling laws offer a way of studying fundamental trade-offs in wireless networks as well as of highlighting the qualitative and architectural properties of specific designs. Such study has been initiated by the work [1] of Gupta and Kumar in 2000. Their by now familiar model considers n nodes randomly distributed on a unit area, each of which wants to communicate to a random destination at a common rate $R(n)$. They ask what is the maximally achievable scaling of the aggregate throughput $T(n) = nR(n)$ and show that cooperation between nodes can dramatically improve performance. Instead of using the simple non-cooperative scheme of time-sharing between direct transmissions from source nodes to destinations, which only achieves aggregate throughput $\Theta(1)$, the nodes can cooperate and relay the packets by multihopping from one node to the next, in which case an aggregate throughput scaling of $\Theta(\sqrt{n})$ is achieved. The price to pay, however, is in terms of delay. In the multi-hop scheme, the packets need to be retransmitted many times before they reach their actual destinations, which results in larger end-to-end delay. More precisely, as shown later in [2], [3], in a multi-hop scheme, bits are delivered to their destinations in $\Theta(\sqrt{n})$ average time after they leave their source nodes, while the average delay for the simple TDMA scheme remains only $\Theta(1)$. Note that this accounts only for on-the-flight delay; the queuing delay at the source node is not considered.

The problem of delay in networks is much more involved compared to point-to-point communication. In the point-to-point scenario illustrated in Fig. 1, the transmission delay introduced by the channel is usually fixed by the nature of the problem whereas, as we have seen above, it is determined by the particular scheme used in the case of networks. The only design parameter that influences the overall delay of the communication in Fig. 1 is the bulk-size B, where bulks of B bits are mapped into blocks of N channel symbols. It is well known that in this setup, for a given probability of error, the rate of communication R can be increased up to the capacity of the channel if larger bulk size B is allowed, see for example [4, Ch.5]. However, increasing the bulk-size B or equivalently increasing the block-length N results in larger delay in communication and also increased complexity.[1]

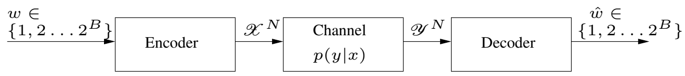

Fig. 1. Block diagram of a point-to-point

Recently, it has been shown in [5] that under certain assumptions on the channel model, a much better aggregate throughput scaling is achievable in wireless networks than the one achieved by classical multi-hop schemes. The authors exhibit a hierarchical cooperation scheme that uses distributed MIMO communication to achieve aggregate throughput scaling arbitrarily close to linear, i.e. $T_h(n) = \Theta(n^{\frac{h}{h+1}})$ for any integer $h > 0$. The parameter h corresponds to the number of hierarchical levels used in the scheme and by increasing h, one can get arbitrarily close to linear scaling. A natural question is whether there is a price to pay for this superior scaling of the throughput. In particular, where is the scheme located on the throughput-delay/complexity trade-off discussed earlier? In this paper, we reanalyze the scheme presented in [5] and show that better throughput is achieved at the expense of increased bulk-size. More precisely, we show that the bulk-size used by the scheme scales as $B_h(n) = \Theta(n^{\frac{h}{2}})$; in other words, it grows arbitrarily large as the throughput approaches linear scaling. Note that the bulk-size immediately imposes a lower bound on the end-to-end delay of the communication; even if there is no transmission delay from the source node to the destination node, receiving a bulk of $B(n)$ bits will take at least $\Theta(B(n)/\log n)$ channel uses for a destination node, since a simple application of the cut-set bound upper

[1]Note that for point-to-point communication, B and N are simply related as $R = \frac{B}{N}$.

bounds the rate of reception by (or transmission from) a node with $\log n$ bits per channel use. The same argument applies to complexity since in the best case, complexity is approximately linear in block-length, which is lower bounded by $\Theta(B(n)/\log n)$.

In the rest of the paper, we present a modification of the hierarchical cooperation scheme that achieves the same aggregate throughput $T_h(n) = \Theta(n^{\frac{h}{h+1}})$ by using a much smaller bulk-size of $B_h(n) = \Theta(n^{\frac{h}{h+1}})$ bits. The key idea in [5] that yields the hierarchical architecture is to set up the receive and transmit cooperation for the distributed MIMO transmissions as multiple problems of the original kind, that is of communicating between n source-destination pairs in a network of n nodes. Any known solution to the original problem can then be used for cooperation, eventually yielding a better solution for this problem. However, if the scheme to begin with uses large bulk-size, using it for cooperation yields a scheme with even larger bulk-size. This is the reason for the increase in bulk size $\Theta(n^{\frac{h}{2}})$ with the number of hierarchical levels h. In this paper, we study the problem of cooperation more carefully. We pose it as a network multiple access problem, where each of the n nodes in the network is interested in conveying independent information, say L bits, to each of the other nodes in the network. We propose a two-phase hierarchical scheme that solves this problem in $\Theta(n^{\frac{h+1}{h}})$ time-slots for any $h > 0$. Using this scheme for cooperation, we achieve a bulk-size scaling $B_h(n) = T_h(n)$ for the original problem. We show that reduced bulk size consequently reduces the delay to $D_h(n) = n$.

We proceed by optimizing the scheduling in this scheme to further reduce the end-to-end delay. To do this, we need to consider a generalized version of the multiple access problem where each node in the network is interested in conveying independent information, say again L bits, to each of the nodes in a subset of $A(n)$ nodes, where the $A(n) < n$ nodes are chosen unformly at random among the n nodes in the network. We show that this task can be accomplished in $\Theta(\frac{A(n)}{n} n^{\frac{h}{h+1}} \log n)$ channel uses for any $h > 0$ if $A(n) \geq n^{\frac{h}{h+1}}$. This allows us to achieve a throughput delay trade-off of $(T(n), D(n)) = (n^b/\log n, n^b \log n)$ for any $0 \leq b < 1$. This new result is depicted in Figure 2, together with previous results from the literature.

A related line of research (see e.g. [6], [7], [2], [8]) is the characterization of the throughput-delay trade-off for mobile networks, where nodes move over the duration of communication according to a certain mobility pattern. In general, mobility schemes achieve an aggregate throughput scaling comparable to that of hierarchical cooperation (i.e. up to linear in n), but the delay scaling performance of such schemes may vary significantly, depending on the chosen mobility model. For instance, under the classical random walk mobility model considered in [2], the performance is quite poor, as illustrated in Figure 2. But from the delay point of view, a more prominent disadvantage which is common to all mobility schemes and which does not appear on the graph in

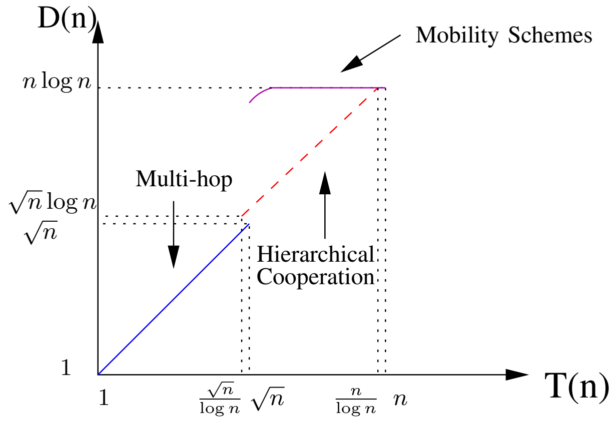

Fig. 2. Throughput-delay performance achieved by hierarchical cooperation together with known results from the literature.

Figure 2, is the constant that precedes the delay scaling law. This pre-constant relates to the speed of nodes in the case of mobility schemes, whereas it relates to the speed of light in the case of hierarchical cooperation.

II. Setting and Main Results

There are n nodes uniformly and independently distributed in a square of unit area. Every node is both a source and a destination. The sources and destinations are paired up one-to-one in a random fashion without any consideration on respective locations. Each source has the same traffic rate $R(n)$ to send to its destination node.[2] The aggregate throughput of the system is $T(n) = nR(n)$.

We assume that communication takes place over a flat channel of bandwidth W Hz around a carrier frequency of f_c, $f_c \gg W$. The complex baseband-equivalent channel gain between node i and node k at time m is given by:

$$H_{ik}[m] = r_{ik}^{-\alpha/2} \exp(j\theta_{ik}[m]) \qquad (1)$$

where r_{ik} is the distance between the nodes, $\theta_{ik}[m]$ is the random phase at time m, uniformly distributed in $[0, 2\pi]$ and $\{\theta_{ik}[m], 1 \leq i \leq n, 1 \leq k \leq n\}$ is a collection of i.i.d. random processes. The $\theta_{ik}[m]$'s and the r_{ik}'s are also assumed to be independent. The constant $\alpha \geq 2$ is called the power path loss exponent of the environment.

Note that the channel is random, depending on the location of the users and the phases. The locations are assumed to be fixed over the duration of the communication. The phases are assumed to vary in a stationary ergodic manner (fast fading). We assume that the channel gains are known at all the nodes. The signal received by node i at time m is given by

$$Y_i[m] = \sum_{k=1}^{n} H_{ik}[m] X_k[m] + Z_i[m]$$

[2]In the following text, we will sometimes refer to this traffic pattern as the unicast problem in order to distinguish it from the multicast problems that will be discussed in Sections IV and V-B.

where $X_k[m]$ is the signal sent by node k at time m and $Z_i[m]$ is white circularly symmetric Gaussian noise of variance N_0 per symbol. Every node is subject to a transmit power constraint that we denote by P.[3]

Following [3], the delay $D(n)$ of a communication scheme for this network is defined as the average time it takes for a bit or packet of constant size to reach its destination node after it leaves its source node, where the averaging is over all bits or packets traveling in the network. So defined, the delay of a scheme quantifies the average time spent by the bits inside the network while operated under this scheme.[4]

The following theorem is the main result of this paper.

Theorem 2.1: Using hierarchical cooperation, the following points are achievable on the throughput-delay scaling curve,

$$(T(n), D(n)) = \Theta\left(n^b/\log n, n^b \log n\right)$$

where $0 \leq b < 1$ (see Figure 2).

III. Overview of the Hierarchical Cooperation Scheme

In this section, we give a brief overview of the hierarchical cooperation scheme as presented in [5] and establish the throughput-delay trade-off for this scheme. Some of the discussions presented here directly build on results already established in [5].

The hierarchical cooperation scheme is based on clustering the nodes in the network and performing long-range MIMO transmissions between the clusters. The long-range MIMO transmissions should be proceeded and followed by cooperation phases establishing transmit and receive cooperation respectively which yields three successive phases in the operation of the network. If simple TDMA is used for establishing cooperation in phase 1 and 3, the overall scheme achieves a $\sqrt{n}$-scaling of the aggregate throughput. This is the three phase scheme discussed in Section III-A. Higher throughputs can be achieved by setting the cooperation problem as multiple communication problems and using the three phase scheme as a solution to each of those communication problems. This yields the idea of recursion and results in a hierarchical architecture, where increasing the number of levels in the hierarchy yields an aggregate throughput scaling arbitrarily close to linear. The hierarchical cooperation scheme is discussed in more detail in Section III-B.

[3]We present the low-level assumptions on the channel and network model in this section for the sake of completeness. However, most of the discussions in the following sections will rely on intermediate results established in [5], hence the dependence of the results on the low level assumptions might not be always evident.

[4]Note that this definition does not account for the source delay. The delay at the source node can be accounted for if the delay of a scheme is defined to be the average time it takes for a bit to reach its destination node after it arrives at the source node. However in order to work with this definition of delay, one needs to make particular assumptions on the arrival process. Note that how larger is this delay from the results we establish in this paper depends on how well the arrival process at the source nodes matches the scheduling in the scheme.

A. The Three Phase Scheme

The network is divided into clusters of M nodes and a particular source node s sends M bits to its destination node d in three steps:

(1) Node s first distributes its M bits among the M nodes in its cluster, one for each node;
(2) These nodes together can then form a distributed transmit antenna array, sending the M bits *simultaneously* to the destination cluster where d lies;
(3) Each node in the destination cluster gets one observation from the MIMO transmission, and it quantizes and ships the observation back to d, which can then do joint MIMO processing of all the observations and decode the M transmitted bits.

From the network point of view, all source-destination pairs have to eventually accomplish these three steps. Step 2 is long-range communication and only one source-destination pair can operate at a time. Steps 1 and 3 involve local communication and can be parallelized across source-destination pairs. Combining all this leads to the following three phases in the operation of the network:

Phase 1: Setting Up Transmit Cooperation Clusters work in parallel. Within a cluster, each source node distributes M bits to the other nodes, 1 bit for each node, such that at the end of the phase, each node has 1 bit from each of the other nodes in its cluster. (Recall our assumption that each node is a source for some communication request and a destination for another.) Thus, since there are M source nodes in each cluster, this gives a traffic demand of exchanging $M(M-1) \sim M^2$ bits. Using TDMA, one-at-a-time transmission between pairs of nodes, these M^2 bits can be exchanged in M^2 time slots.

Phase 2: MIMO Transmissions Successive long-distance MIMO transmissions are performed between source-destination pairs, one at a time. In each one of the MIMO transmissions, say the one between s and d, the M bits of s are simultaneously transmitted by the M nodes in its cluster to the M nodes in the cluster of d. Each of the long-distance MIMO transmissions are repeated for each source-destination pair in the network, hence we need n time-slots to complete the phase.

Phase 3: Cooperate to Decode Clusters work in parallel. Since there are M destination nodes inside the clusters, each cluster received M MIMO transmissions in phase 2, one intended for each of the destination nodes in the cluster. Thus, each node in the cluster has M received observations, one from each of the MIMO transmissions, and each observation is to be conveyed to a different node in its cluster. Nodes quantize each observation into fixed Q bits, so there are a total of QM^2 bits to be exchanged inside each cluster. Using TDMA as in Phase 1, the phase can be completed in QM^2 time slots.

In [5], it is shown that each destination node is able to decode the transmitted bits from its source node from the M quantized signals it gathers by the end of Phase 3. The throughput achieved by the scheme can be calculated as follows: each source node is able to transmit M bits to its destination node, hence nM bits in total are delivered to their destinations in M^2+n+QM^2 time slots, yielding an aggregate throughput of

$$\frac{nM}{M^2 + n + QM^2}$$

bits per time-slot. Choosing $M = \sqrt{n}$ to maximize this expression yields an aggregate throughput $T(n) = \frac{1}{2+Q}\sqrt{n}$.

Note that as opposed to multihop, this three phase scheme allows only bulk transmission between any source-destination pair in the network; i.e. one can not arbitrarily communicate one bit (or L bits with L constant) using the three-phase scheme, but $M = \sqrt{n}$ bits per source-destination pair should be transferred with each use of the scheme.

The end-to-end delay of this scheme is simply the total time for the three phases, since the bits are leaving their source nodes at the beginning of the first phase and are only decoded by their respective destination nodes at the end of the third phase. With the choice $M = \sqrt{n}$, we see that the delay of the three phase scheme is $D(n) = (2 + Q)n$. Note that this delay scaling is much worse when compared to the delay of the multi-hop scheme achieving same aggregate throughput.

B. The Hierarchical Cooperation Scheme

Higher aggregate throughput scaling can be achieved by using a better network communication scheme than TDMA to establish the transmit and receive cooperations in phase 1 and phase 3. Note that there are M^2 and QM^2 bits that need to be exchanged in phase 1 and 3 respectively. This traffic demand of exchanging M^2 bits (or QM^2 bits) can be handled by setting up M sub-phases, and assigning M pairs in each sub-phase to communicate their 1 bit (or Q bits). The traffic to be handled at each sub-phase now looks similar to the original network communication problem (the unicast network problem defined in Section II), with M users instead of n. Any scheme suggesting a good solution for the original problem can now be used inside the sub-phases as an alternative to TDMA; for example, the multi-hop scheme and the three-phase scheme given in Section III-A would be two alternatives both achieving an aggregate throughput scaling $\Theta(\sqrt{M})$ (in a network of size M) as opposed to the $\Theta(1)$ scaling achieved by TDMA. In general, if a scheme achieving aggregate throughput scaling M^b is used to handle the traffic in each sub-phase, the total completion time for phase 1 and 3 becomes $M \times M^{1-b}$ and $M \times QM^{1-b}$ respectively. This in turn yields an aggregate throughput

$$\frac{nM}{M^{2-b} + n + QM^{2-b}}$$

bits per time-slot, which is maximized by the choice $M = n^{\frac{1}{2-b}}$, yielding $T(n) = \frac{1}{2+Q} n^{\frac{1}{2-b}}$. (Note that plugging $b = 0$ for TDMA yields the aggregate throughput scaling $\sqrt{n}$ derived earlier for the three phase scheme.) Starting with $b = 0$ for TDMA and noticing that $\frac{1}{2-b} > b$ for $0 \le b < 1$, applying the same argument recursively h times, one gets a scheme achieving aggregate throughput scaling $T_h(n) = n^{\frac{h}{h+1}}$. Note that this recursion builds a hierarchical architecture with h levels. At the lowest level of the hierarchy, the simple TDMA scheme is used to exchange bits for cooperation among small clusters. Combining this with longer range MIMO transmissions, one gets a higher throughput scheme for cooperation among nodes in larger clusters at the next level of the hierarchy. Finally, at the top level of the hierarchy, the cooperation clusters are almost of the size of the network and the MIMO transmissions take place at a global scale. See Figure III-B.

For deriving the delay performance of the hierarchical scheme, let us first concentrate on the simplest case $h = 2$. The resultant scheme achieving aggregate throughput scaling $n^{2/3}$ divides the network into clusters of size $M_1 = n^{2/3}$ and uses the three phase scheme inside these clusters for establishing cooperation. More precisely, the traffic of communicating 1 bit (or Q bits) between M_1 source-destination pairs in each sub-phase of phase 1 (or phase 3) is handled by further dividing the cluster into smaller clusters of size $M_2 = \sqrt{M_1} = n^{1/3}$ and using the three phase scheme (TDMA-MIMO-TDMA) given in Section III-A. Note however that the three phase scheme allows only bulk transmissions between source-destination pairs. In this particular case, one will have to communicate M_2 bits between the source-destination pairs assigned at each sub-phase, as opposed to the original requirement of communicating only 1 bit (or Q bits). For the overall scheme, this in turn increases the bulk size to be communicated between every source-destination pair in the network from M_1 bits to $M_1 \times M_2$ bits, resulting also in larger delay. The delay of the two-level hierarchical scheme is given by $M_2 \times n = n^{4/3}$, as opposed to n for the three phase scheme ($h = 1$). Indeed, it can be checked that the aggregate throughput achieved by the two-level scheme is given by the expression

$$\frac{M_2\, M_1\, n}{M_1(M_2^2 + M_1 + QM_2^2) + M_2 n + M_1 Q(M_2^2 + M_1 + QM_2^2)} \quad (2)$$

and the optimal choices of $M_1 = n^{2/3}$ and $M_2 = n^{1/3}$ maximize the aggregate throughput scaling to $T_2(n) = n^{2/3}$, while the denominator dictating the delay of the scheme is of order $D_2(n) = n^{4/3}$. Note that the increase of the communication bulk size does not affect the throughput, since it corresponds to multiplying the numerator and denominator of (2) by the same factor, but it affects the delay.

Extending the argument for larger h and noticing that the cluster size at the k'th level of an h-level hierarchical scheme is given by $M_k = n^{\frac{h+1-k}{h+1}}$, we obtain the bulk-size in an h-level hierarchical scheme as

$$B_h(n) = M_h \times \ldots \times M_1 = n^{\frac{h}{2}}$$

and its end-to-end delay as

$$D_h(n) = M_h \times M_{h-1} \times \cdots \times M_2 \times n = n^{\frac{h^2+h+2}{2(h+1)}}$$

where we observe that for large h, the delay exponent is linear in h. Recall that the aggregate throughput achieved by an h-level hierarchical cooperation scheme is given by $T_h(n) = n^{\frac{h}{h+1}}$.

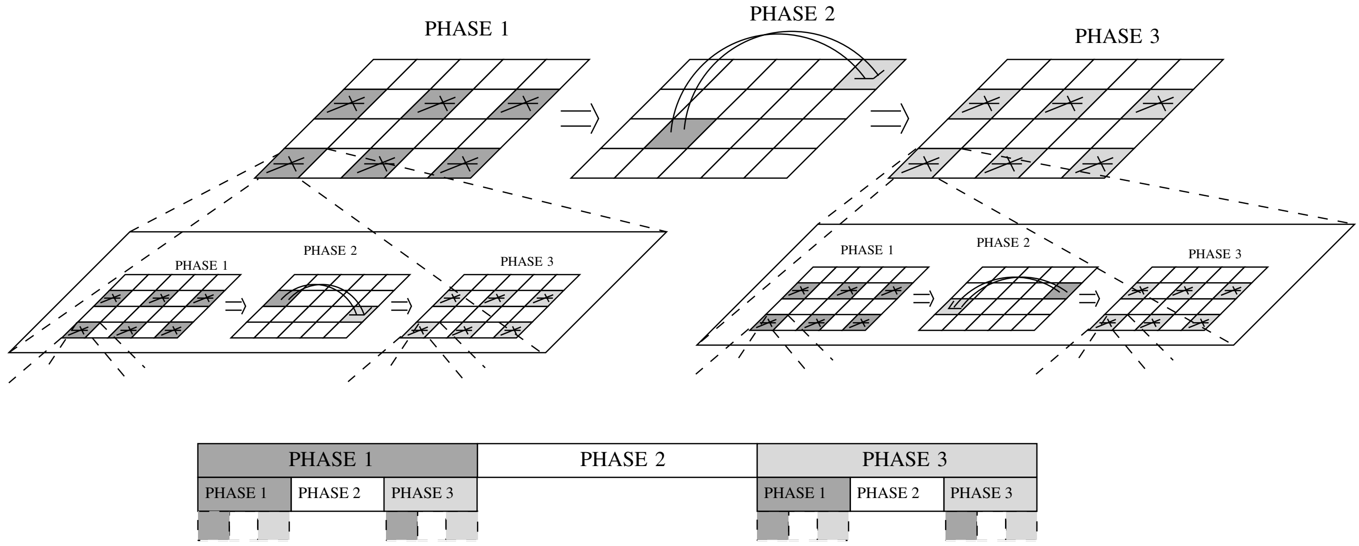

Fig. 3. The salient features of the three phases and the time division in a hierarchical scheme are illustrated. Figure taken from [5].

The results obtained in this section establish the poor delay performance of hierarchical cooperation. Note that the delay is mostly due to the large bulk-size used by the scheme and this not only imposes large delay but also results in high complexity. This is different from multi-hop schemes since their bulk-size is constant ($\Theta(1)$) and the delay is due to the time spent in relaying the messages inside the network. In the next section, we modify the scheme so that it achieves the same throughput using much smaller bulk-size.

IV. Hierarchical Cooperation with Smaller Bulk-Size

In this section, we treat the problem of cooperation in the three phase scheme with more care. We start by defining the network multiple access problem to be the following.

Definition 4.1 (The Network Multiple Access Problem): Consider the assumptions on the network and channel model given in Section II. Let each node in the network be interested in communicating independent information to each of the other nodes in the network. In particular, let us assume that each node has 1 independent bit (or L independent bits, with L constant) to send to each of the other nodes in the network and the quantity of interest is the smallest time $F(n)$ required to accomplish this task. This problem we refer to be the network multiple access problem.

The following theorem provides an achievable solution to this problem.

Theorem 4.1: For any integer $h > 0$, the network MAC problem can be solved in

$$F(n) \le K\, n^{\frac{h+1}{h}}$$

time-slots, for some constant $K > 0$ independent of n.

Proof of Theorem 4.1: Let us start by assuming that there exists a scheme that solves the multiple access problem in $F(n) = n^b$ time-slots with $b > 1$. Note that one such scheme is simple TDMA that yields $b = 2$. Using this existing scheme, we will construct a new scheme that yields smaller $F(n)$.

As before, let us start by dividing the network into clusters of M nodes. Let us first focus on one specific cluster S and one node d located outside of this cluster. In particular, all nodes in S have 1 bit to send to d. These bits can be communicated to d in two steps:

(1) The nodes in S *simultaneously* transmit their 1 bit messages destined to d forming a distributed transmit antenna array for MIMO transmission. The nodes in the destination cluster which d belongs to, form a distributed receive antenna array for this MIMO transmission.
(2) Each node in the destination cluster obtains one observation from the MIMO transmission in the previous phase; it quantizes and ships this observation to d, which can do joint MIMO processing of all the observations and decode the M transmitted bits from the nodes in S.

As a first step towards handling the whole network problem, note that these two steps should be accomplished between S and all other nodes in the network. This can again be done in two steps:

Phase 1: MIMO transmissions We perform successive long-distance MIMO transmissions between S and all other nodes in the network. In each of the MIMO transmissions, say between S and d, the M nodes in S are simultaneously transmitting the 1 bit messages they would like to communicate to d and the M nodes in the cluster where d lies are observing the MIMO transmission. The MIMO transmissions should be repeated for each node in the network, hence we need n time-slots to complete the phase.

Phase 2: Cooperate to decode Clusters work in parallel. Since there are M nodes inside each cluster, each cluster received M MIMO transmissions from S in the previous phase, one intended for each node in the cluster. Thus, each

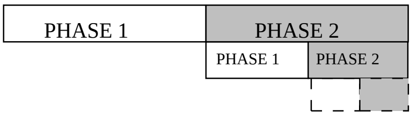

Fig. 4. The figure illustrates the time-division in the hierarchical scheme that solves the network multiple access problem.

node in the cluster has M observations, one from each of the MIMO transmissions, and each observation is intended for a different node in the cluster. Each of these observations can be quantized into Q bits, with a fixed Q, which yields exactly the original network multiple access problem, with M nodes instead of n. Using the scheme we assumed to exist in the beginning of the proof, this task can be completed in QM^b time slots.

The total time we have spent during the two phases for handling the traffic originated from cluster S is given by $n + QM^b$. From the network point of view, the above two steps should be completed for all n/M clusters in the network. Thus, the multicasting task can be completed in $\frac{n}{M}(n + QM^b)$ time slots. Choosing $M = n^{\frac{1}{b}}$ in order to minimize this quantity yields $F(n) = (1 + Q)n^{2-\frac{1}{b}}$.

Note that $2 - \frac{1}{b} < b$ for $b > 1$. In other words, we have established a solution for the multiple access problem that is better than the one we started with. Indeed, the two phase scheme described above can be used recursively yielding a better scheme at each step of the recursion. In particular, starting with TDMA achieving $b = 2$ and applying the idea recursively h times, one gets a scheme that solves the multiple access problem in $\Theta(n^{\frac{h+1}{h}})$ time slots. The operation of this scheme is illustrated in Figure 4. □

The interest in the multiple access problem arises from the fact that it exactly models the required traffic for cooperation in the three phase scheme. Recall the communication requirement inside the clusters in Phase 1 and 3 described in Section III-A. This communication requirement, equivalent to a network multiple access problem, is handled using TDMA in the three phase scheme which has been seen to be suboptimal in the Section III-A. In the hierarchical cooperation scheme described in Section III-B, this multiple access problem is handled by decomposing it into a number of unicast network problems. The resultant scheme is optimal in terms of throughput, but not very satisfying in terms of bulk-size. By using the solution to the multiple access problem suggested in this section, one can modify the hierarchical cooperation scheme, so as to achieve the same throughput with smaller bulk-size and consequently smaller delay. Note that the gain is coming from treating the cooperation problem as it is and not necessarily as multiple network communication problems as was previously done in Section III-B.

Corollary 4.1: A modified hierarchical cooperation scheme can achieve an aggregate throughput $T_h(n) \geq K_1 n^{\frac{h}{h+1}}$ with bulk-size $B_h(n) = K_2 n^{\frac{h}{h+1}}$ and delay $D_h(n) \leq K_3 n$, for any integer $h \geq 0$ and some positive constants K_1, K_2, K_3 independent of n.

Proof of Corollary 4.1: Consider the three phase hierarchical scheme described in Section III-A. By Theorem 4.1, the required traffic for transmit and receive cooperation in phase 1 and phase 3 can be handled in $KM^{\frac{h+1}{h}}$ and $KQM^{\frac{h+1}{h}}$ time slots respectively. The expression for the aggregate throughput then becomes

$$\frac{Mn}{KM^{\frac{h+1}{h}} + n + KQM^{\frac{h+1}{h}}}$$

which is maximized by the choice $M = n^{\frac{h}{h+1}}$, yielding aggregate throughput $T_h(n) = \frac{1}{1+K+KQ} n^{\frac{h}{h+1}}$, bulk-size $B_h(n) = n^{\frac{h}{h+1}}$ and delay $D_h(n) = (1 + K + KQ)n$. □

V. Hierarchical Cooperation with Better Scheduling

In the previous section, we presented a modified hierarchical scheme that achieves throughput $T_h(n) = \Theta(n^{\frac{h}{h+1}})$ using bulk-size $B_h(n) = \Theta(n^{\frac{h}{h+1}})$. However, the delay of this scheme is still $D_h(n) = \Theta(n)$. In this section, we optimize the scheduling in the scheme to further improve the delay performance to $D_h(n) = \Theta(n^{\frac{h}{h+1}} \log n)$. We first start by improving the scheduling in the three phase scheme with $h = 1$ discussed in Section III-A. We then consider the modified hierarchical scheme with $h \geq 2$ discussed in Section IV .

Before starting, we state the following binning lemma, similar in spirit to Lemma 4.1 and Lemma 5.1 in [5] and can be proven using similar techniques. The lemma will be used repeatedly throughout the rest of the paper.

Lemma 5.1: Let us assume that $f(n)$ balls are thrown into n bins, independently and uniformly at random. The following properties are satisfied with high probability.

(a) If $\lim_{n\to\infty} \frac{f(n)}{n \log n} = \infty$, then there are $\Theta(\frac{f(n)}{n})$ nodes in each bin.
(b) If $\lim_{n\to\infty} \frac{f(n)}{n} = c$ with $c \geq 0$ a constant independent of n, then there are at most $O(\log n)$ nodes in each bin.

A. Better Scheduling for the Three Phase Scheme

Recall the operation of the three phase scheme from the point of view of a single source-destination pair s-d as described in Section III-A: a step (1) where s distributes its M bits among the M nodes in its cluster, followed by a step (2) where these M bits are simultaneously transmitted to the destination cluster via MIMO transmission, and a step (3) where the quantized MIMO observations are collected at the destination node d. These three steps need to be eventually accomplished for each source-destination pair in the network. In this section, we improve the scheduling in accomplishing this task: we organize M successive sessions and allow only n/M source-destination pairs to complete the three steps in each session.

In the beginning of each session we randomly choose one source node from each cluster, thus n/M source nodes in total. In general, the n/M destination nodes corresponding to these randomly chosen source nodes can be located anywhere. However, from Lemma 5.1, we know that no more than $\log n$ of these destination nodes are located in the same cluster with high probability. We proceed by accomplishing the three steps for these chosen source-destination pairs:

Phase 1: Setting Up Transmit Cooperation Clusters work in parallel. The chosen source node in each cluster distributes its M bits to the other nodes by using TDMA, which takes M time-slots in total. Note that as opposed to the scheme described in Section III-A, there is only one source node operating in each cluster.

Phase 2: MIMO Transmissions Successive MIMO transmissions originated from each cluster are performed, transmitting the bits of the active source node in each cluster to its respective destination cluster. Note that in the current case, there is only one MIMO transmission originated from each cluster, so there are only n/M MIMO transmissions that need to be performed in total. This will require total time n/M.

Phase 3: Cooperate to Decode Clusters work in parallel. Each cluster received at most $\log n$ MIMO transmissions in phase 2 by Lemma 5.1-b, each MIMO transmission intended for a different destination node in the cluster. The received observations at each node are quantized into Q bits and are to be conveyed to the actual destination nodes. The traffic inside each cluster is at most of exchanging $QM \log n$ bits and can be completed using TDMA in at most $QM \log n$ time slots. (See Figure 5.)

The operation continues with the next session by choosing a new set of n/M source nodes randomly among the nodes that have not yet accomplished the above steps. Note that all source-destination pairs will accomplish the three steps in a total of M sessions.

With this rather smoother operation on the network level, we accomplish to serve n/M source-destination pairs in each session, that is transfer $M \times \frac{n}{M}$ bits in total to their destinations in $M + \frac{n}{M} + QM \log n$ time slots yielding aggregate throughput

$$\frac{M \times \frac{n}{M}}{M + \frac{n}{M} + QM \log n} \tag{3}$$

which is maximized by the choice $M = \sqrt{n}$ yielding aggregate throughput $T(n) = \frac{1}{2+Q} \frac{\sqrt{n}}{\log n}$. The delay experienced by each bit is now much less compared to the three phase scheme in Section III-A, since it is again dictated by the total time spent in the three phases (denominator of (3)), which is now less than $D(n) = (2 + Q)\sqrt{n} \log n$.

Note that instead of choosing $M = \sqrt{n}$, which is the optimal choice to maximize the throughput achieved by the scheme, one can choose $M = n^b$ with $0 \leq b \leq 1/2$. In this case, we also restrict the number of source-destination pairs to be served in each session to M, which can be less than the total number of clusters n/M. Indeed, we operate one source node in each of the $M (\leq n/M)$ clusters and simply keep the remaining clusters inactive. The expression for the aggregate throughput becomes

$$\frac{M \times M}{M + M + QM \log n}$$

which implies that the scheme achieves aggregate throughput $T(n) = n^b / \log n$ and delay $D(n) = n^b \log n$ for any $0 \leq b \leq 1/2$. Note that this throughput-delay trade-off differs only by $\log n$ from the trade-off achieved by multi-hop schemes.

B. Better Scheduling for the Hierarchical Cooperation Scheme

In this section, we adopt the scheduling idea of Section V-A to the modified hierarchical scheme presented in Section IV. However, this modification is not trivial and requires us to consider a generalized version of the network multiple access problem.

Definition 5.1 (The Generalized Network MAC Problem): Consider the assumptions on the network and channel model given in Section II. Let each of the n nodes in the network be interested in conveying independent information to a subset $A(n)$ of the nodes ($A(n) \leq n$), where the $A(n)$ nodes are chosen randomly among the n nodes in the network. In particular, let us assume that each node in the network has an independent 1 bit message (or L independent bits, with L constant) to send to each of these $A(n)$ nodes and the quantity of interest is the minimal time $G(n)$ required to accomplish this task. We define this to be the generalized network multiple access problem.

The following theorem provides an achievable solution to this problem. We skip the proof of the theorem since it is similar in spirit to the proof of Theorem 4.1.

Theorem 5.1: For any integer $h > 0$, if $A(n) \geq n^{\frac{h}{h+1}}$, then the network multiple access problem can be solved in

$$G(n) \leq K \, \frac{A(n)}{n} \, n^{\frac{h+1}{h}} \log(n)$$

time-slots, for some constant $K > 0$ independent of n.

Note that the generalized network multiple access problem contains the network multiple access problem discussed earlier as a special case with $A(n) = n$. Plugging $A(n) = n$ in Theorem 5.1, we recover the result of Theorem 4.1 with an extra $\log n$ factor. Indeed, when the condition $A(n) \geq n^{\frac{h}{h+1}}$ is satisfied with strict inequality in order, the extra $\log n$ factor in Theorem 5.1 is not needed. However, it is needed to account for the case $A(n) = n^{\frac{h}{h+1}}$, in which case it arises due to part-b of Lemma 5.1.

We are now ready to apply the scheduling idea in Section V-A to the hierarchical cooperation scheme. Consider dividing the network into clusters of M_1 nodes and then further divide these clusters into smaller clusters of size M_2. Following the scheduling idea in Section V-A, let us organize M_1/M_2 sessions and for each session randomly choose one small cluster inside every large cluster. Only the source nodes located in the chosen small clusters and their corresponding destination nodes will be served at each session. As usual, we are operating in three successive phases in each session:

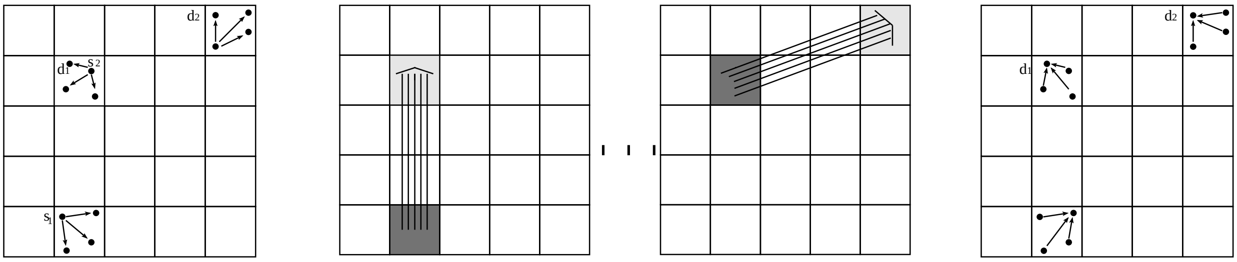

Fig. 5. The three phase scheme with better scheduling. The figure illustrates the operation in one session.

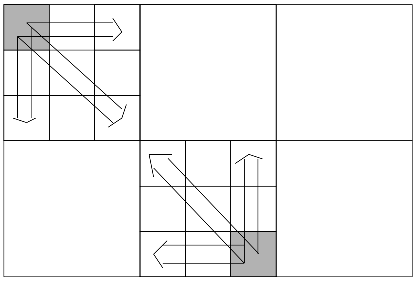

Fig. 6. The figure illustrates sub-phase 1 of phase 1 of the modified hierarchical scheme with better scheduling. Note that there is only one small cluster distributing bits inside every large cluster.

Phase 1: Setting Up Transmit Cooperation The active small clusters operate in parallel. Note that there is a single active cluster of size M_2 inside every large cluster of size M_1. Let S be the chosen small cluster inside the larger cluster L that will operate in the current session. In this phase, each of the M_2 source nodes in S need to distribute their M_1 bits among the M_1 nodes in the larger cluster L, each of the M_1 bits goes to a different node. This can be accomplished in two sub-phases:

- **Sub-Phase 1: MIMO transmissions** Successive MIMO transmissions are performed between nodes in S and each node in L. In each of these MIMO transmissions, say the one between S and a node d in L (located outside of S), the M_2 nodes in S are simultaneously transmitting the 1 bit messages they would like to communicate to d. The M_2 nodes located in the same small cluster with d are acting as a distributed receive antenna array for this MIMO transmission. Since these MIMO transmissions should be repeated for every node in L, this sub-phase takes a total of M_1 time-slots. See Figure 6.
- **Sub-Phase 2: Cooperate to Decode** All small clusters in the network work in parallel. In particular, each small cluster in L has received M_2 MIMO transmissions from S in the previous phase, one MIMO transmission for each node in this small cluster. Thus, each node in the small cluster has M_2 observations, one from each of the MIMO transmissions and each observation is to be conveyed to a different node in the cluster. Quantizing each observation into Q bits, we get the network multiple access problem defined in Section IV in a network of size M_2, and by Theorem 4.1 this problem can be handled in $QM_2^{\frac{h_1+1}{h_1}}$ time-slots for any integer $h_1 > 0$.

Phase 2: MIMO Transmissions At the end of the first phase, all source nodes in the active small clusters have distributed their M_1 bits among the nodes in the larger cluster. Now, successive long-distance $M_1 \times M_1$ MIMO transmissions between large clusters are performed. During each MIMO transmission, the M_1 bits of a particular source node in the active small cluster are transferred to the destination cluster where its destination node is located. The number of MIMO transmissions to be performed in this phase is equal to the total number of source nodes active in this session. Hence the total phase can be completed in $\frac{n}{M_1} \times M_2$ time-slots.

Phase 3: Cooperate to Decode By part-a of Lemma 5.1, there are order M_2 destination nodes located in each of the large clusters. Thus, each large cluster has received M_2 MIMO transmissions in the previous phase, and the quantized MIMO observations spread over the M_1 nodes of the large cluster should be collected at the corresponding M_2 destination nodes. This is the generalized network multiple access problem of size M_1 with $A(M_1) = M_2$. By Theorem 5.1, it can be solved in $\frac{M_2}{M_1} \times M_1^{\frac{h_2+1}{h_2}} \log M_1$ time-slots for any integer $h_2 > 0$ provided that $A(M_1) \geq M_1^{\frac{h_2}{h_2+1}}$.

Gathering everything together, at every session of this modified hierarchical cooperation scheme, we deliver $M_1 \times M_2 \times \frac{n}{M_1}$ bits to their destinations in a total of

$$\left(M_1 + M_2^{\frac{h_1+1}{h_1}}\right) + \frac{n}{M_1} \times M_2 + \frac{M_2}{M_1} \times M_1^{\frac{h_2+1}{h_2}} \log M_1$$

time-slots. The aggregate throughput is given by

$$\frac{\frac{n}{M_1} \times M_2 \times M_1}{M_1 + M_2^{\frac{h_1+1}{h_1}} + \frac{n}{M_1} \times M_2 + \frac{M_2}{M_1} \times M_1^{\frac{h_2+1}{h_2}} \log M_1}$$

which is maximized by the choice $h = h_2 = h_1 + 1$, $M_1 = n^{\frac{h}{h+1}}$ and $M_2 = M_1^{\frac{h-1}{h}}$, yielding aggregate throughput $T(n) = \frac{n^{\frac{h}{h+1}}}{\log n}$ and delay $D(n) = n^{\frac{h}{h+1}} \log n$. Note that these choices for M_1 and M_2 satisfy the constraint $A(M_1) = M_2 \geq M_1^{\frac{h_2}{h_2+1}}$.

Note that at this point, we have proven that all points on the throughput-delay scaling curve $(T(n), D(n)) =$

$(n^{\frac{h}{h+1}}/\log n,\, n^{\frac{h}{h+1}}\log n)$ with h being a positive integer are achievable. In order to show that all points on the line $(T(n), D(n)) = (n^b/\log n,\, n^b \log n)$ with $0 \le b < 1$ are achievable, we can choose $M_1 = n^b$ with $0 \le b \le \frac{h}{h+1}$ in the above discussion, while maintaining the relationships $M_2 = M_1^{\frac{h-1}{h}}$ and $h = h_2 = h_1 + 1$. Extending the argument at the end of Section V-A, we also restrict the number of small clusters to be served in each session to $M_1^{1/h}$ which can now be less than the total number of large clusters $n/M_1\,(\ge M_1^{1/h})$. Indeed, we operate one small cluster in each of the $M_1^{1/h}$ large clusters and simply keep the remaining large clusters inactive. The expression for the aggregate throughput becomes

$$\frac{M_1^{\frac{1}{h}} \times M_2 \times M_1}{M_1 + M_2^{\frac{h_1+1}{h_1}} + M_1^{\frac{1}{h}} \times M_2 + \frac{M_2}{M_1} \times M_1^{\frac{h_2+1}{h_2}} \log M_1}$$

which shows that we can achieve aggregate throughput $T(n) = M_1/\log M_1$ and delay $D(n) = M_1 \log M_1$. Recalling that $M_1 = n^b$, we get the points on the throughput-delay scaling curve $(T(n), D(n)) = (n^b/\log n,\, n^b \log n)$ for any $0 \le b \le \frac{h}{h+1}$ and $h > 0$. This concludes the proof of the main result of this paper. □

VI. Conclusion

The present work shows that hierarchical cooperation not only can lead to higher throughput in ad hoc networks, but also to reasonable end-to-end delay, given that some extra care is taken in setting up cooperation at the lower levels and scheduling communications. Meanwhile, we have discussed the network multiple-access problem in the present paper, which might be of interest in its own right.

References

[1] P. Gupta and P. R. Kumar, *The Capacity of Wireless Networks*, IEEE Transactions on Information Theory, Vol. 42 (2), pp. 388–404, March 2000.

[2] A. El Gamal, J. Mammen, B. Prabhakar, D. Shah, *Optimal Throughput-Delay Scaling in Wireless Networks-Part I: The Fluid Model*, IEEE Transactions on Information Theory 52 (6), pp. 2568–2592, June 2006.

[3] A. El Gamal, J. Mammen, B. Prabhakar, D. Shah, *Optimal Throughput-Delay Scaling in Wireless Networks-Part II: Constant-Size Packets*, IEEE Transactions on Information Theory 52 (11), pp. 5111–5116, November 2006.

[4] Gallager R. G., *Information Theory and Reliable Communication*, Wiley & Sons Inc., 1968.

[5] A. Özgür, O. Lévêque, D. Tse, *Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks*, IEEE Transactions on Information Theory, Vol. 53 (10), pp. 3549–3572, October 2007.

[6] M. Grossglauser and D. Tse, *Mobility Increases the Capacity of Adhoc Wireless Networks*, IEEE/ACM Transactions on Networking, Vol. 10 (4), pp. 477–486, 2002.

[7] M. J. Neely and E. Modiano, *Capacity and Delay Tradeoffs for Ad-Hoc Mobile Networks*, IEEE Transactions on Information Theory, Vol. 51 (6), pp. 1917-1-937, June 2005.

[8] G. Sharma, R. R. Mazumdar and N. B. Shroff, *Delay and Capacity Trade-offs in Mobile Ad Hoc Networks: A Global Perspective*, ACM/IEEE Transactions on Networking, Vol. 15 (5), pp. 981–992, 2007.