A Code Allocation Protocol for Maximizing Throughput in CDMA based Ad Hoc Networks

Saurabh Srivastava
Dept. of CSE, IIT Kanpur
Kanpur, INDIA - 208016
saurabh.srivastava@ieee.org

Sachee Tripathi
Dept of EE, IIT Kanpur
Kanpur, INDIA - 208016
sachee-tripathi@lycos.com

Dheeraj Sanghi
Dept. of CSE, IIT Kanpur
Kanpur, INDIA - 208016
dheeraj@cse.iitk.ac.in

Ajit K. Chaturvedi
Dept of EE, IIT Kanpur
Kanpur, INDIA - 208016
akc@iitk.ac.in

Abstract— **We consider here the problem of allocating variable length orthogonal codes in an ad hoc network based on CDMA. We consider a snapshot version of the problem at some instant. It had been proved earlier that even for a static set of communications and topology the problem is intractable. A greedy algorithm is stated to provide a bounded approximation to the throughput maximizing optimal allocation. We present a simple distributed code allocation protocol based on the greedy approximation. Simulation experiments show the enhanced throughput obtained by our protocol as compared to other code allocation schemes.**

I. Introduction

Multi-access schemes allow multiple transmitters to share the spectrum by proper allocation. Code Division Multiple Access allows all terminals to use the entire bandwidth at the same time and provides flexibility and graceful degradation to the system. If orthogonal variable spreading factor codes are being used for spreading then the use of smaller length codes provides larger data rates and hence should be preferred to achieve greater aggregate throughput. But the number of such mutually orthogonal codes decreases with decreasing length of the codes. Hence, there is a need to *spatially reuse* codes, i.e., codes need to be reused at terminals which are not in hearing range of each other. But allocation of the same code to multiple terminals gives rise to the possibility of space and time overlap of two or more packet receptions on the same code, called a *collision or interference*, which results in corrupted bits being received at the receiver.

Collisions increase the number of retransmissions and decrease throughput. In [1], interference has been described as either being *direct* or *secondary*. Direct interference occurs when two nodes simultaneously initiate transmissions to each other. Secondary interference occurs at a receiver due to simultaneous transmissions by two transmitters who cannot hear each other. They proposed a protocol which allocates a code to each terminal which is different from the codes allocated to its two-hop neighbors. This can be seen to eliminate collisions but requires a large number of codes in the worst case.

The code allocation problem for eliminating secondary collisions has been shown to be NP-Complete [2] and therefore several heuristics solutions have been presented [2], [3]. Due to the known intractability of the problem there has been little work on bounding the worst case behavior of code allocation techniques and none on the increased throughput obtained by using codes of smaller length while doing code allocation.

In fact, the use of variable length codes allows optimal use of the radio spectrum allowing the code allocation to be independent of worst case scenarios that might exist locally. If we were to use fixed length codes, the smallest length of the code would be dominated by the minimum required at any place in the network. Suppose some part of the network had an abnormally high density of terminals. Then the length of the code used here would be extremely large and since the codes are fixed length, the throughput in the whole network (which is related to the length of the codes), would be low. On the other hand, if we were to use variable length codes then we could allocate larger length codes in this part of the network and use shorter lengths in parts where the density was low, thus achieving greater throughput. The use of orthogonal variable length codes (OVSF codes) in W-CDMA has been considered in [4], [5], [6], but with an emphasis on reducing the *blocking probability*[4] in *single* hop networks and not for increasing throughput.

In this paper, we propose a distributed allocation protocol that achieves an approximate solution to the throughput maximization problem in *multihop ad hoc* networks. Using a graph theoretic approach we have shown the equivalence of this problem to a variant of graph coloring which we call the *range sum* of a graph. A greedy algorithm for the range sum is used as the basis for designing the protocol.

The rest of the paper is organized as follows: In Section II, we present the network model and assumptions. This is followed by the problem formulation in Section III. Then in Section IV we introduce the range sum of a graph. In Section V we present techniques to allocate codes for maximizing aggregate throughput based on the approximate solution to the range sum problem. In Section VI we present the results of the simulation experiments to demostrate the superior throughput performance of our protocol. Section VII concludes the paper.

II. Network Model and Assumptions

Consider a multihop ad hoc network consisting of portable terminals communicating through a common wide-band transmission channel using CDMA. A symmetric connectivity channel is assumed. Receiver terminals are equipped with the facility to broadcast the code on which they are receiving their transmission. This is done on a broadcast channel (time slot) during the Medium Access Control layer ACK phase.

0-7803-7700-1/03/$17.00 (C) 2003 IEEE

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

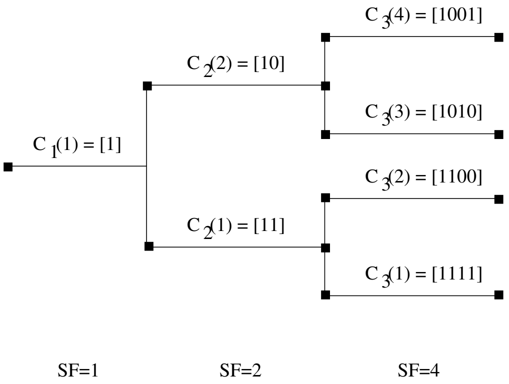

Fig. 1. Tree structured generation of Orthogonal Variable Spread Factor codes

A CDMA system with code reuse is prone to the problem of secondary interference. Two possible situations in which such interference occurs, as observed in [1], is when there exists a receiver B which can hear transmissions from two transmitters A, C, which cannot hear each other and both are transmitting on the same code c. In both situations (one in which A and C are transmitting to B and the other when the receivers are different), a collision occurs at B. In a dynamic code allocation scheme if each transmitter were to observe the codes being used in its neighborhood before deciding its own code then there would have been no collisions.

A. Spreading Codes:

We assume that Orthogonal Variable Spread Factor (OVSF) codes are being employed for variable rate spreading. A tree structured method for generating such codes was proposed in [7].

The ith code at level $N-1$, $C_{N-1}(i)$, generates two codes, $C_N(2i) = C_{N-1}(i) \oplus \overline{C_{N-1}(i)}$ and $C_N(2i-1) = C_{N-1}(i) \oplus C_{N-1}(i)$, at level N. Here $\oplus$ and $\overline{(\cdot)}$ denote the concatenation and bitwise inversion operators respectively. OVSF codes allow us to allocate variable spread factor codes, resulting in variable data rate allocation to the terminals, while still maintaining orthogonality. A pair of (possibly different length) codes in this set are orthogonal to each other except when one is derived from the other.

III. Problem Formulation

An ad hoc network can be modeled as an undirected graph $G_{topo} = (V, E)$, called the *topology graph*, where the set of vertices $V = \{1, \ldots, n\}$ represent the set of terminals with an edge between two vertices iff the corresponding terminals can receive a transmission from each other. To analyse the optimal code allocation problem we consider the network at some instant and call the topology at that instant a snapshot. We construct a new graph called the communication graph which captures the set of communicating entities and the interfering sets for a particular snapshot. We will later show that even for this static instance the code allocation problem is intractable. Therefore an approximate solution is required.

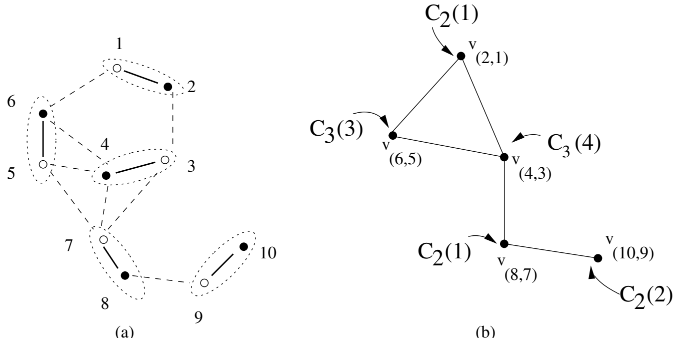

Fig. 2. (a) A snapshot and a set of communications. The transmitters are shown by filled black nodes and their corresponding receivers are hollow and connected by bold black lines. Dashed lines denote entities within radio range. (b) The communication graph. It can be easily verified that the assignment of OVSF codes as shown in the figure maximizes throughput.

A. The Communication Graph

Let us denote by $G_{topo} = (V, E)$, the topology graph. Then the vertex, V', and edge, E', sets in the communication graph, $G_{comm} = (V', E', t_0)$, for some set of transmitters and receivers which are active at time $t = t_0$, are defined as follows:

- $V' = \{v_{(u,v)} \mid u, v \in V$ and u is a transmitter and v its corresponding receiver at time $t = t_0\}$
- $E' = \{(v_{(u,v)}, v_{(p,q)}) \mid$ iff $(u, q) \in E$ or $(p, v) \in E\}$

In an ad hoc network, the problem of allocating variable length codes with a throughput maximization criterion can be formulated as follows: Reduce the topology graph of the network, given a set of communicating transmitter-receiver pairs, to the communication graph. Since adjacent vertices in the communication graph represent pairs of entities which have to be assigned different codes, this results in a simple instance of graph coloring. But the length of the code allocated to the nodes in this graph also determines the throughput achieved by using a code allocation scheme at that particular instant. An assignment of codes which results in the maximum possible throughput for this snapshot is called an optimal allocation. Hence the code allocation problem is to assign codes such that adjacent pairs receive orthogonal codes with a throughput maximization criterion.

Example Let us consider the topology graph as shown in Figure 2a. The bold lines connect communicating entities and the dashed lines connect terminals which can hear each other. It can be easily verified that the assignment of orthogonal OVSF codes as shown in Figure 2a will lead to the maximum throughput for this set of communications. If we merge each transmitter-receiver pair into a vertex to construct the communication graph, as shown in Figure 2b, then the problem reduces to a modified instance of coloring. In addition to coloring the nodes with orthogonal codes (different colors), we also require that the codes be assigned in such a manner that the throughput is maximized in the network.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

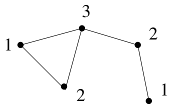

Fig. 3. Example graph and a coloring for its chromatic sum.

IV. The $\sigma-$Range Sum of a Graph

To facilitate the theoretical analysis of the variable length code allocation problem we first define the $\sigma-$range sum of a graph.

A. Graph Theoretic Definitions and Notation

For a given undirected graph $G = (V, E)$, the order and size of the graph are the number of vertices and edges respectively. The degree of a vertex is the number of edges incident on it and let Δ and $\bar{d}$ denote the maximum and average degree of the graph.

A coloring of a graph is an assignment of colors from a specified set to the vertices of a graph. A coloring is called proper if no two adjacent vertices share the same color. The *chromatic number* of a graph is defined to be be the minimum number of colors which are required to achieve a proper coloring of the graph from the set of natural numbers. The *chromatic sum* [8] of a graph is the minimum sum of the colors of the vertices over all colorings of the graph with natural numbers. The graph shown in Figure 3 has been colored with natural numbers and the chromatic number can easily be seen to be 3 and the chromatic sum to be 9 (=1+2+3+2+1) which is obtained by the sample coloring shown in the figure.

B. The $\sigma-$Range Sum

We now introduce a variant of the chromatic sum of a graph. Instead of coloring the nodes with colors from the set of natural numbers, we instead color the vertices with ranges. Given a parameter σ, the ranges used for coloring are obtained recursively, by equally subdividing the line segment $[0, 1)$ into σ equal parts at each step of the recursion. We color the nodes with these ranges in a manner such that no two adjacent nodes are colored with ranges having a non zero intersection. More formally:

Definition Define the sets $R_\sigma(k)$ with parameters σ, k recursively as:

- $R_\sigma(0) = \{(0, 1]\}$
- $R_\sigma(k+1) = \left\{ \left[l(r) + (t-1) * \frac{|r|}{\sigma},\ l(r) + t * \frac{|r|}{\sigma} \right) \middle| t \leq \sigma,\ t \in N,\ \forall r \in R_\sigma(k) \right\}$

  where $l(r)$ denotes the left end of the range r.

For notational convenience we shall refer to the ith element of $R_\sigma(k)$ as $R_\sigma(k, i)$ and denote the set of $R_\sigma(k)$'s as $\Re_\sigma$ i.e. $\Re_\sigma = \lim_{k \to \infty} \bigcup_{i=0}^{k} R_\sigma(k)$.

Example Figure 4 shows the hierarchical organization of the ranges in $\Re_2$. The ith level in this *range tree* corresponds to the set $R_2(i)$. Moreover the parameter σ(=2 in this case) can be visualized as the degree of branching of this range tree.

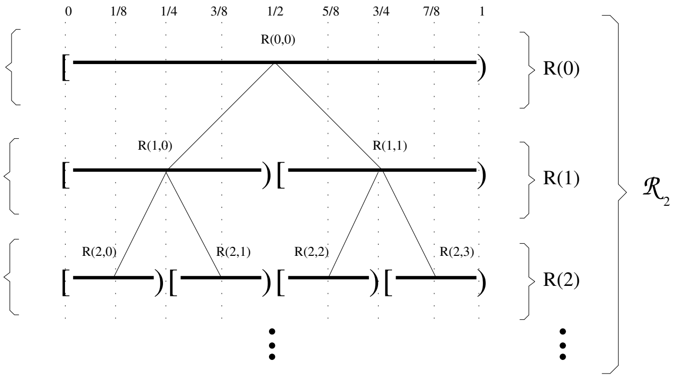

Fig. 4. Graphical illustration of the range set $\Re_2$. Overlapping ranges are shown connected and form a complete binary tree. For simplicity, the subscript 2 has been omitted.

Definition Let $G = (V, E)$ be a graph with vertex set V and edge set E. A proper $\sigma-$range coloring is a coloring of the nodes with adjacent nodes being colored with non-intersecting ranges from $\Re_\sigma$; i.e. $r : V \to \Re_\sigma$ such that $r(u) \cap r(v) = \phi$ whenever $(u, v) \in E$.

We define the *$\sigma-$range sum* of the graph G, denoted by $\Gamma_\sigma(G)$, to be the maximum sum of the lengths of ranges over all proper $\sigma-$range colorings of G.

Example Consider the problem of 2-range coloring the graph shown in Figure 5. For $\Re_2$, the length of a range in $R(1, -)$ (p and q) is 0.5 and in $R(2, -)$ (r and s) is 0.25.

This example illustrates the difference of the range sum problem from the chromatic sum and chromatic number problem. The chromatic number and sum have been observed to be 3 and 9 respectively. Both the assignments in Figures 5b and 5c are proper range coloring with the only difference being in the range assigned to the vertex v_5. Figure 5b shows that the graph can be properly range colored using only three ranges but it can easily be verified that any coloring with three ranges cannot achieve a sum of more than 1.75 $(= 2|r| + 2|p| + |s|)$ in the assignment shown. While on the other hand if a fourth range is used then a larger sum of 2.0 $(= 2|p| + |r| + |q| + |s|)$ can be achieved and it can be verified that this is the range sum for this graph.

C. Computing the Range Sum for arbitrary graphs

In [9], we show that computing the range sum for arbitrary graphs is an NP-Complete problem. Hence an approximation algorithm is required. The following is a simple greedy algorithm to proper range color a graph. The algorithm processes vertices in an arbitrary order and assigns a range with the maximum possible length such that all adjacent vertices which have not been assigned a range are left with some subset of $(0, 1]$ which can be used.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

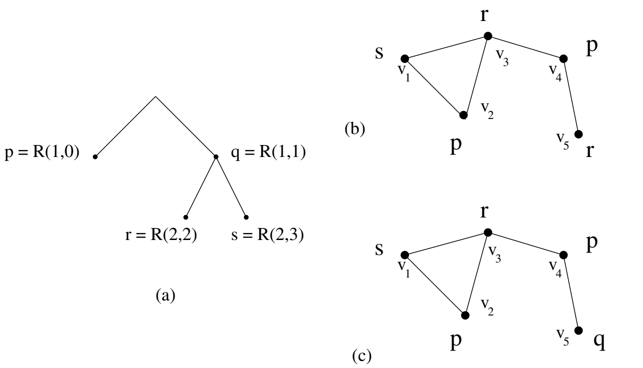

Fig. 5. (a) The range tree for $\sigma = 2$ upto the relevant levels. (b) A proper range coloring using just 3 colors. (c) A range coloring which uses 4 ranges and achieves the 2-range sum for this graph.

Algorithm **Greedy-Range-Color**(G, σ):

Input: A undirected graph $G = (V, E)$ and integer parameter σ.

Output: A proper $\sigma-$range coloring $r : V \rightarrow \Re_\sigma$ of G.

For each node $v \in V$ maintain:

- *r(v)*, the range(*color*) assigned to v, initialized to ϕ
- *Unused(v)*, the subset of $[0, 1)$ not being used to color any node u adjacent to v; or more formally

$$Unused(v) = [0,1) - \bigcup_{\forall u;(u,v) \in E} r(u)$$

Algorithm:

Step 1: Determine an ordering of the vertices $v \in V$, $v_1, v_2, \ldots, v_n$.

Step 2: Consider the vertices in the order determined above. Let the current vertex be v_r. Find the least value of t such that

- $Range_\sigma(t) \cap Unused(v_r) = Range_\sigma(t)$ and
- $Unused(u) - Range_\sigma(t) \neq \phi, \quad \forall u$ satisfying $(u, v_r) \in E$ and $r(u) = \phi$

Color v_r with $Range(t)$, i.e $r(v_r) = Range_\sigma(t)$ and update $Unused(u) = Unused(u) - r(v_r) \quad \forall u, (u, v_r) \in E$. Repeat until there is some uncolored vertex.

In [9] we show that the coloring computed by Greedy-Range-Color for any G is a $\sigma^{\frac{\sigma+\bar{d}}{\sigma-1}}$-approximation to $\Gamma_\sigma(G)$.

Example Consider again the graph shown in Figure 5. The lexicographic order of the ranges in $\Re_2$ is R(0,0), R(1,0), R(1,1), R(2,0), R(2,1), R(2,2), R(2,3), $R(3,0)\ldots$. And consider the ordering of the vertices as shown in the figure itself. It can be easily verified that the range coloring that is produced by Greedy-Range-Color is the same as shown in Figure 5b which is coincidently optimal.

V. Code Allocation for Maximizing Throughput

As discussed in Section II, the problem of code allocation such that there are no collisions reduces to coloring a node (representing a pair of communicating entities in the topology graph) in the communication graph, such that their immediate neighbors are not using the same color (code). For codes taken from the OVSF code set, the following lemma holds:

Lemma 5.1: The problem of maximizing the aggregate throughput by optimal code allocation is equivalent to finding the 2-range sum of the communication graph, G_{comm}.

A. Distributed Code Allocation using Greedy Range Coloring (DCA-GRC)

In this section we shall refer to codes and ranges ($\subseteq [0, 1)$) interchangeably using the one-to-one mapping from the OVSF code tree to the range tree for $\Re_2$. When the network shows variation in the topology structure, use of OVSF codes allows the use of smaller spreading factor codes in parts with less density and hence increased aggregate throughput for that region. We now present a distributed version of the greedy range coloring algorithm outlined in Section IV-C [9]. By Lemma 5.1, this amounts to finding an allocation which maximizes throughput.

1) Protocol Idea and Overview: To formulate a distributed version of Greedy-Range-Color observe that in Step 2 of the algorithm only local information is required at each node which is being assigned the code. Therefore it remains to find a distributed implementation of Step 1. Since the ordering of the vertices is not based on any criterion, we can as well allow all nodes to choose their own codes locally, as long as there are no codes being negotiated in the neighborhood. This can be easily ensured by a simple implicit locking phase. We now describe the procedure executed by each node (a pair of transmitter receiver after they have negotiated the codes being used in their neighborhood during the MACK phase) in the communication graph.

Algorithm Code-Negotiation:

1) Construct a tree, T_{code}, of the codes that are being used, i.e. for each code that is being used in the neighborhood construct the appropriate leaf in the tree along with any nodes required in the path to the root code. Label all nodes in T_{code} as *used*
2) Binarize T_{code}, i.e. for each used node in the tree, add *unused* children until all used nodes have two children and all unused nodes are leaves.
3) For each leaf node, l, (which is unused) calculate
$$suitability(l) = \frac{1}{code\_len(l)} - \sum \frac{n}{2 * code\_len(v)} \quad (1)$$
where the sum is taken over all nodes in the tree which are ancestors of l and n is the number of neighbors using the code corresponding to that node in the OVSF code tree.
4) Find the leaf node, l_{max}, with largest value of $suitability(l)$. Resolve ties in a left to right manner. Choose the code corresponding to l_{max}.
When the path from l_{max} to the root has some used codes then SPLIT messages need to be sent to neighbors that are using those codes. A SPLIT message is essentially an instruction to the terminal to start using

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

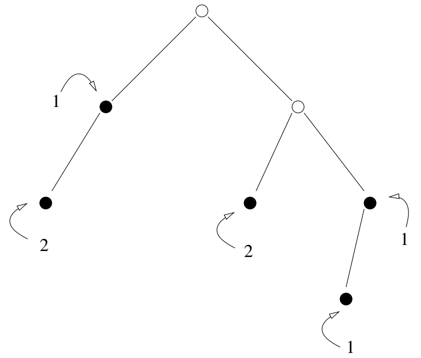

Fig. 6. Construction of T_{code}. The number beside each node denote the number of neighbors using the code corresponding to this node.

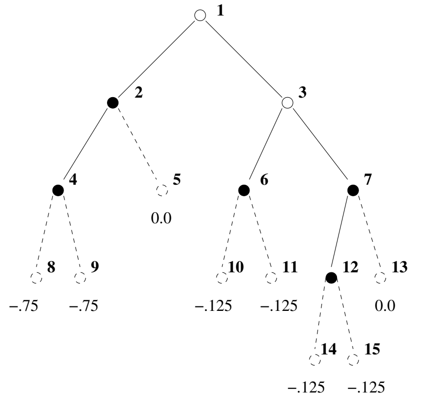

Fig. 7. T_{code} after binarizing. All *unused* leaves have been populated with their *suitability* values.

one of the descendants of the code currently being used, i.e. to start communicating at half the current bit rate. The new code, out of the possible two, that the neighbor will use will depend on the direction of split required to vacate the path to the root.

Example Consider the codes in use to be the ones which are shown as dark nodes in the OVSF code tree in Figure 6. The hollow nodes are the ones added to create a path from each used node to the root. Figure 7 shows the tree after leaf nodes have been added. Node 5 is then chosen as l_{max} and the neighbor using the code corresponding to node 2 will start using 4.

2) Throughput Performance of the Protocol: We now present a bound on the throughput resulting from a code allocation produced by DCA-GRC.

Lemma 5.2: Given a set of communications, algorithm DCA-GRC produces an initial allocation which is proper and an $2^{2\bar{d}-1}$-approximation to γ_{max}, where $\bar{d}$ is the average degree in the topology graph, G_{topo}, and γ_{max} is the maximum possible aggregate throughput for the given set of communications.

Proof: A formal proof is provided in [9]. ■

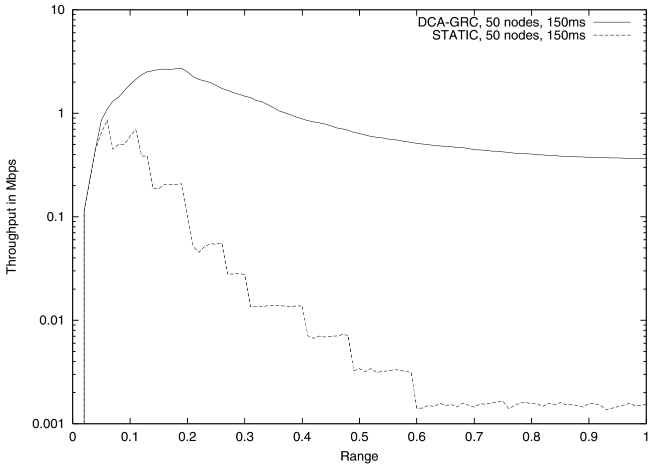

Fig. 8. Throughput performance as the range of the nodes is increased.

VI. Simulation Model and Results

Simulations were done using random topologies to evaluate the performance of DCA-GRC and the throughput obtained was compared to the case when an off-line static code allocation had been done. For the off-line static allocation, we consider the greedy algorithm presented in [3] and assume that a centralized entity decides the codes which are communicated to the nodes without any overhead. For G_{topo} which does not change with time, this means that we are ignoring the network setup time; for a network with mobile nodes, this means that we are ignoring the losses incurred due to collisions. The simulation were done with the following parameters:

- Packet burst arrival process is Poisson with mean of the interarrival times being λ.
- Packet burst size is uniformly distributed between 1 to 5 packets, each of size 1 KB.
- The physical layer is assumed to provide a communication channel with a maximum bit rate of 11Mbps (The base bit rate provided by the IEEE 802.11b standard)
- The nodes are randomly distributed in a square region of unit area and their range of communication is represented as a fraction of the length of the side.

Using an event driven simulator we simulated the packet transmission for the above parameters. For DCA-GRC each packet burst transmission is preceded by a code negotation phase in which the nodes execute *Code-Negotiation* to decide the code to use. No such overhead is incurred in the case of static allocation. A packet is assumed to be corrupted and lost if any other transmitter in the range of the receiver transmitted in the duration of the packet using a non-orthogonal code. For our current simulation we omit the locking procedure that is supposed to be called before code negotiation. This causes a few rare collisions which cause a slight degradation in performance of our protocol.

In Figure 8, we vary the range of the terminals (which is expressed as a fraction of the length of one side the area) such that the network topology varies from a set of isolated nodes

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.

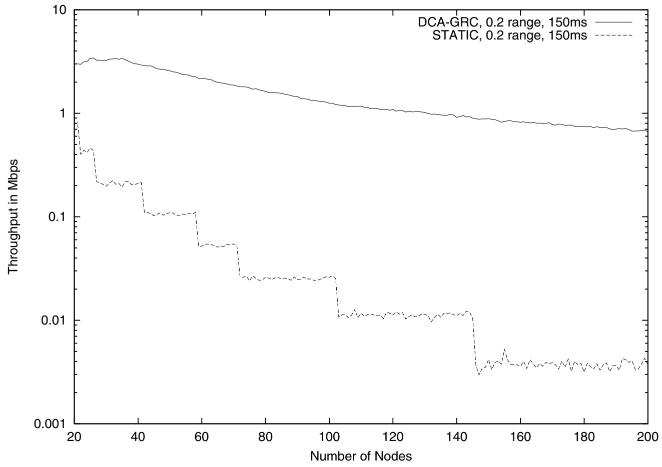

Fig. 9. Throughput performance as the number of nodes in the network increases.

to fully connected. In effect we are varying the transmit power of the terminals from a very low value to a value which results in a single hop network. In these simulations λ, the mean of the interarrival time, is taken to be $150ms$, which ensures that the nodes almost always have packets in queue. Since the number of terminals is fairly large (50), the static allocation throughput for the fully connected topology is two orders of magnitude lower than what is achieved by DCA-GRC since its performance only depends on the number of active communications in the neighborhood. On the other extreme, both protocols show a steep rise in throughputs obtained when the average degree in the network increases from zero. This is because with an average degree of zero the network consists of a set of isolated nodes and aggregate throughput is zero while the maximum throughput is obtained when the average degree is around 2, since nodes are not interfering with each others' transmission and yet are connected enough to be able to flush incoming packets in time. The static allocation protocol shows a step wise degradation as the maximum degree (which dominates the length of codes being used) descends a level in the code tree.

In Figure 9, we see a similar plot as we increase the number of nodes in the network. Here the range has been fixed at a value of 0.2 while λ is still taken to be $150ms$. DCA-GRC again shows a steady performance while the plot for static allocation decreases in steps. Since the number of codes at some level in the code tree is twice that in the previous level, we see a sharp fall in the throughput for static allocation. This is because an increase in the average degree does not have any effect till the codes are chosen from the same level in the code tree. Whenever the length of the code doubles, which happens on approximately doubling the maximum degree the throughput shows a sharp decrease. On the other hand, our dynamic code allocation scheme is never really affected in its performance since it is dependent on the number of communications in progress which is independent of the maximum degree in the network. The slight downward curve is expected since a larger number of nodes implies larger interference.

VII. Conclusions

In this paper we have introduced the concept of range coloring and the range sum of a graph. Based on a greedy approximation algorithm for the range sum we have presented a distributed code allocation scheme. The performance of this allocation scheme has been evaluated and found to be much better than static allocation schemes which have been proposed earlier.

References

[1] J. Garcia-Luna-Aceves and J. Raju, "Distributed assignment of codes for multihop packet-radio networks," in *Proc. IEEE MILCOM*, Monterey, California, Nov 1997.

[2] A. A. Bertossi and M. A. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 4, pp. 441–449, Aug 1995.

[3] L. Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 6, pp. 668–677, Dec 1993.

[4] T. Minn and K.-Y. Siu, "Dynamic assignment of orthogonal-variable-spreading-factor codes in W-CDMA," *IEEE Journal on Selected Areas in Communications, Special Issue on W-CDMA*, vol. 18, no. 8, pp. 1429–1440, Aug 2000.

[5] M. D. Amico, M. Merani, and F. Maffioli, "Efficient algorithms for assignment of OVSF codes in wideband CDMA," in *Proc. ICC'02*, New York City, Apr 2002.

[6] R. Assarut, K. Kawanishi, R. Deshpande, U. Yamamoto, and Y. Onozato, "Performance evaluation of orthogonal variable-speading-factor code assignment schemes in W-CDMA," in *Proc. ICC'02*, New York City, Apr 2002, pp. 3050–3054.

[7] F. Adachi, M. Sawahashi, and K. Okawa, "Tree structured generation of orthogonal spreading codes with different lengths for the forward link of DS-CDMA mobile radio," *IEE Electronic Letters*, vol. 33, no. 1, pp. 27–28, 1997.

[8] E. Kubicka and A. J. Schwenk, "An introduction to Chromatic Sums," in *Proc. of ACM Computer Science Conference*, Louisville, Kentucky, USA, Feb 1989, pp. 39–45.

[9] S. Srivastava, S. Tripathi, D. Sanghi, and A. Chaturvedi, "Resource optimization in CDMA based ad hoc network," CSE Department, IIT Kanpur, Tech. Rep., 2002, also available from http://www.cse.iitk.ac.in/research/btp2002/98344.html.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on September 30, 2009 at 03:28 from IEEE Xplore. Restrictions apply.