

COLLISION-FREE TOPOLOGY-DEPENDENT CHANNEL ACCESS SCHEDULING

Lichun Bao and J.J. Garcia-Luna-Aceves

baolc@cse.ucsc.edu, jj@cse.ucsc.edu

Computer Engineering Department

University of California, Santa Cruz, CA 95064

ABSTRACT

We propose a topology-dependent TDMA algorithm that automatically schedules collision-free channel access at each node without repetitious contention phase(s) before channel access once all nodes obtain topology information regarding their two-hop neighbors. Bandwidth of the channel is completely devoted for data traffics except for synchronizations and occasional neighbor updates. Unique qualitative features of the protocol are presented and its performances are analyzed quantitatively by simulations. Computation-oriented nature provides the protocol many flexibilities for quality assurance in channel scheduling.

I. INTRODUCTION

Scheduled access to a common channel in multi-hop packet radio networks has received considerable attentions recently. A scheduled-access approach (also known as transmission scheduling, node/link activation, allocation, or scheduling) consists of establishing transmission schedules, i.e., allocating stations to different times and data channels (e.g., frequencies, spreading codes, or their combination) in a way that no collisions (conflicts) occur and efficient spatial reuse of the available bandwidth is achieved. Efficient scheduling with channel reuse can render much higher channel utilization than such fixed assignment approaches as TDMA and FDMA.

In this paper, we focus on the problem of scheduling collision-free broadcasts by a node (also called node activation) to all its neighbors without any other packet interfering its transmission. The problem has been formulated in such a way that only one node is assigned to a time slot within its two-hop neighborhood. Optimal TDM (time division multiplexing) scheduling for broadcasting channel access in multi-hop packet radio networks has been shown to be an NP-complete problem even with the global topology knowledge by all nodes within the network [3] [9]. Splitting out a second contention-based control segment from the channel for out-band signaling of time slot allocation information, Cidon and Sidi [2], Pond and Li [8] proposed distributed algorithms that eliminate direct interference (primary conflict) and hidden terminal interference (secondary conflict) [12] within the transmission segment of the channel, however, under a considerable overhead due to the control segment.

Several channel scheduling and reservation protocols based

on in-band signaling (phased dialogs or RTS/CTS handshakes) before transmissions have been proposed in the past [11] [14]. The limiting factor with this class of scheduling protocols is that they spend a portion of the channel for channel contention, and such contention may result in unused time slots, because of unresolved contentions due to collisions.

A number of topology-transparent scheduling methods have been proposed [1] [5] [6]. The basic idea of the topology-transparent scheduling is for a node to transmit in a number of time slots in each frame. The times (slots) when a node i transmits in a frame corresponds to a unique code such that, for any given neighbor k of i , node i has at least one transmission slot during which k and none of k 's own neighbors are transmitting. Therefore, within any given time frame, any neighbor of i can receive at least one packet from i collision-free. The limitations of the topology-independent scheduling approaches described to date are that: (a) the sender is unable to know which neighbor(s) can correctly receive the packet it sends in a particular slot, which implies that the sender has to send its packet in the various slots it has available in a frame, and (b) the frame length (number of slots) must be larger than the number of nodes in a two-hop neighborhood and depend on the network size, which is less scalable.

All these protocols for TDMA bear some similarities in that they either compete with two-hop neighbors for collision-free channel access or allow collisions and amend with repetitions. We propose a TDMA solution, called CTMA (Collision-free Topology-dependent Multiple Access), which allows a time slot to be shared by two-hop neighbors without conflict and time-consuming contention phases. The only requirement is the knowledge of its two-hop neighbors by each node, which is also a requirement of several other topology-dependent transmission scheduling solutions. For each time slot, nodes run a common algorithm that produces a permutation of their contending two-hop neighbors. A node that finds itself at the head of the permutation is permitted to transmit. As the permutation changes from time slot to time slot, nodes take turns for transmissions. Since the relative order of nodes in the permutation does not conflict at all nodes, the decision for transmission is conflict-free at all times. A key advantage of CTMA is that it inherently supports broadcasting and multicasting, which is critical in multi-hop packet-radio networks, because routing-table updates must be sent on broadcast mode for efficiency, and because multicast

applications are becoming extremely popular in the Internet, of which future packet-radio networks are becoming a seamless extension.

In section II the assumption about network topology is given and rules for channel access are described. Section III shows mechanisms to support network mobility and neighbor maintenance. Section IV illustrates simulation results on a randomly generated network. Section V concludes the paper.

II. PROTOCOL DESCRIPTION

A. Topology Assumptions

The topology of the network is abstracted as an undirected graph $G = (V, E)$, where V is a collection of nodes speaking CTMA for communication, and E is the set of links between nodes. Each node is assigned a unique ID number and has a wireless transceiver that uses a single multi-access broadcasting channel to exchange data frames with neighbors. A link between two nodes indicates the two nodes can directly communicate with one another via the common channel, in which the two nodes are called *one-hop neighbors*. Two nodes having a common one-hop neighbor are called *two-hop neighbor* to each other. Usually, when we say two-hop neighbors of a specific node, it also includes one-hop neighbors and itself.

B. Time Division Multiplexing

CTMA is a topology-dependent, distributed TDMA protocol, which schedules channel access according to *time slots* of equal duration. However, we do not deal with synchronization in this paper, but suggest achieving by either: (a) listening to data traffics in the network, and time slot is always aligned to the latest starting point of a complete packet transmission by one-hop neighbors in that time slot, or (b) other means, such as GPS (global positioning systems) timing signals.

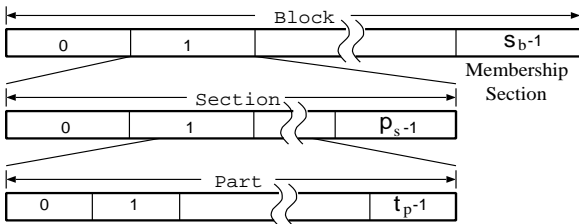


Figure 1. Time Division

A time slot is the smallest time unit for transmitting one or more complete data frames. To utilize the channel, we combine a number of consecutive time slots, say t_p , into a *part*, p_s consecutive parts into a *section* and s_b consecutive sections into the largest chunk of time, *block* (Figure 1). Each node chooses one and only part in which to compete for time slots to transmit, depending on the crowdedness of that part.

For management purposes, the last section of a block is allocated for membership maintenance and is called *membership section*. New neighbors, who did not transmit but listened in

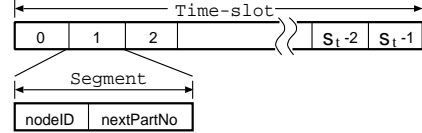


Figure 2. time slot Format in Membership Section

previous sections, announce their existence in the membership section. For this purpose, time slots in the last section are further divided into s_t *segments* of equal duration for sending signals by new members. Each signal contains the node ID and the part number that the node is willing to use in the coming block (Figure 2.)

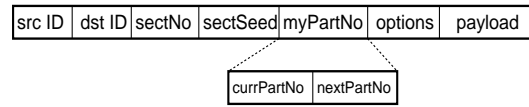


Figure 3. time slot Format in Regular Sections

To schedule topology-dependent channel access, one-hop neighbors are required to exchange complete information about their one-hop neighbor IDs and corresponding part numbers so that every node knows identities of its two-hop neighbors. We provide some fields in the header of each data frame for this purpose. In figure 3, `myPartNo` contains part numbers of the transmitting node in current block and next block. Field `options` may include one-hop neighbor updates about their part number. Depending on the payload of a data frame, neighbors exchange their one-hop neighbor updates in a single or multiple data frames. A data frame also includes current section number and section seed (explained in next section) so that new members synchronize with the scheduling.

Every node in the network keeps rotary counters for current segment number, time slot number, part number and section number within a block, cyclically counting from 0 to $s_t - 1$, $t_p - 1$, $p_s - 1$, $s_b - 1$ respectively.

C. Channel Access Scheduling

CTMA depends on part numbers, IDs of itself and two-hop neighbors and a *section seed* to schedule channel access. The section seed is a number assigned to each section of time divisions, which increments itself after every section, and rolls back to 0 when it reaches its maximum value. New members obtain the section seed by listening to the network traffic. If no neighbor is available, a random seed is chosen. If the section seeds disagree between network nodes, the maximum value is adopted.

Using these pieces of information, a node generates a message digest md out of the concatenation of current section seed s and ID number for itself and each two-hop neighbor. Algorithms generating message digest are found in [10] and [13].

Nodes access the common channel abiding to following rules:

[Rule 1] Within a section, a node transmits in its chosen part

p only.

[Rule 2] Within a part, a node with message digest md for the current section transmits in time slot t , where $t \equiv md \bmod t_p$ (hashing.)

[Rule 3] If two nodes i and j contends for the same time slot t of a part, they concatenate their respective message digests and ID numbers, and get (md_i, i) and (md_j, j) . If $(md_i, i) < (md_j, j)$, i wins the right of transmission, vice versa. When multiple two-hop neighbors are hashed into the same time slot, the above algorithm gives an implicitly sorted sequence of these contending neighbors for every node, which corresponds to a *random-permutation* of these nodes. If a node finds itself the first one of the sorted sequence, the node is the winner for transmission in the time slot. Otherwise, the node waits for the next section which has a different section seed and a different permutation of these nodes.

[Rule 4] When a time slot is not occupied by any node within two hops of a node, the node competes for the spare time slot by running a similar random-permutation algorithm on its two-hop neighbors belonging to the same part. This time, however, the message digest is generated from the concatenation of section seed, node ID, plus the spare time slot number.

Since every node chooses a part in which to transmit, and the random-permutation based on the message digest gives fairness to each node, every node is equally likely to get a chance for channel access within a section. The chances of transmission by a node are dependent on local topologies. Unlike other MAC protocols that resolve contentions every time a node needs channel access, the priority of accessing the channel is precomputed at each node in CTMA.

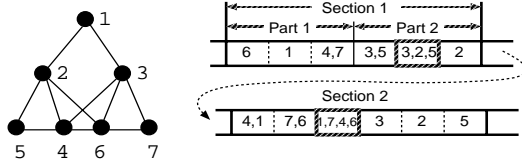


Figure 4. An Example for Channel Access Scheduling

Figure 4 shows a sample network topology and corresponding channel scheduling. The parameter t_p (time slots within a part) is set to 3 and p_s (parts within a section) is 2. The figure exemplifies two continuous sections, where node 1, 4, 6, 7 choose part 1, and the others choose part 2. In each section, nodes are hashed into different time slots as shown in the figure. Some time slots (shaded boxes) are not occupied by any node, for which nodes belonging to corresponding part compete [Rule 4]. The head of each sequence is justified for transmission in that time slot.

III. NEIGHBOR PROTOCOL

In mobile networks where network topology changes frequently from nodes' movements, each node becomes closer or farther to one another. Since collision-free transmission scheduling is tightly related with two-hop neighborhood, it

is critical for a node to realize and incorporate neighborhood changes promptly, and reharmonizes itself with channel access scheduling. The neighbor protocol handles these changes in a reliable fashion; we describe briefly the mechanisms provided in CTMA that deal with topology changes.

A. Joining The Network

We rely on the membership section of each block to allow new members add themselves into channel access schedules. For this purpose, we distinguish nodes in terms of those that are already in the network topology and those that are new. All nodes already in the topology listen during the membership section of a block, when new members are set to announce their existence.

A new member first listens to the network traffic for at least a complete block before it tries to participate in the network topology. The duration of a block, which is s_b sections, is derived such that we have high certainty (we choose $> 99\%$) that every node in the network transmits at least once in the block. We leave out the computation of s_b and only use the result in this paper. A data frame (Figure 3) provides the part number for current and next block of the transmitting node, which supplies enough information about one-hop neighbors of a new member. Meanwhile, the new member also synchronize time slot and section-seed with the rest of the network. The section-seed is included in the header of each data frame (Figure 3). If a new member cannot find any neighbor, it decides a time division according to its own clock.

In the membership section, a new member selects a random part number and hashes itself into a time slot of that part using the common algorithm described in [Rule 2]. The new member also chooses a random segment within the time slot to transmit its signal, which contains its ID number and part number (Figure 2). It is expected that all one-hop neighbors of the new member hear the signal and incorporate the new member in their one-hop neighbor set.

There could be collisions when multiple new members within two hops try to notify the network in exactly the same part, time slot and segment of the membership section; furthermore, the new member wants all two-hop neighbors to know its existence to avoid collisions. Therefore, we require that new members be quiet in the following block and listen to one-hop neighbor updates transmitted by its one-hop neighbors to ensure that joining request are received correctly by these nodes. If some neighbors do not acknowledge the status of the new member in the next block, the new member tries to send other signals in the following membership section until it receives acknowledgments from all of its one-hop neighbors.

In addition, the new member and its one-hop neighbors need to exchange their one-hop neighbor information. Once synchronization is achieved, the new member is able to for transmit.

When network density is moderate and new members show up infrequently, the network can incorporate a new member within a finite time because the randomness of CTMA assures

that the same collision does not occur persistently.

B. New Connections

When two nodes become one-hop neighbors, they need to firstly recognize each other and secondly to synchronize their one-hop neighbor information. We name them a and b respectively and consider the recognition of b by a in two cases.

1. a receives a complete data frame from b . a sends out a neighbor update in its next data frame regarding the status of b . Unless b sends back a neighbor update about a , a will send out signal in membership section like a new member does and wait for acknowledgment from b . The process is repeated until b recognizes a , when a and b exchange complete one-hop neighbor information.

2. a does not receive a complete data from b , but detects persistent high-level noises (collisions) in some time slots. a will send out signal in the following membership section like a new member does. As long as collisions exist, a keeps sending signals in the membership section. Once a is recognized by b , b follows the process as in the previous case.

In the second case, noise may also be caused by two one-hop neighbors of a not knowing each other, which requires a to send out updates about one-hop neighbors in the colliding part to resolve conflicts.

On the other hand, the disappearance of a one-hop neighbor or a link can be detected by a node when neither data frame nor signal is received from the neighbor for a number of blocks. A node indicates disappearance of a one-hop neighbor by setting the part number of the neighbor to 3 in a neighbor update. If different neighbors report different part numbers for a node, the receiver of the update believes in all valid part numbers of the node, which implies allocation of multiple time slots to the node. Obsolete part number of a node is eliminated by aging.

IV. PERFORMANCE

We simulate the performance of CTMA assuming a static topology. A time unit in the simulation equals on CTMA time slot, and the network topology is randomly generated by placing 100 nodes within an area of 1000×1000 square meters. The topology of the network is based on the transmission range of the transmitter mounted on each node, which is valued from 100 to 600 meters. When two nodes are within the transmission range of each other, they can exchange data frames and have a link between them in the network topology. By controlling the transmission range of these nodes, the network produces different connectivities and topologies. In the simulation, we assume the following parameters:

1. Transmission rate of the wireless channel is 4 Mbps.
 2. A time slot last 4 ms, enough to transmit a 2KB packet.
 3. The number of time slots within a CTMA part is $t_p = 5$.
 4. The number of CTMA parts within a section is $p_s = 3$.
- Thus, a section lasts 60ms.

We ran CTMA for 6000 seconds, which is equal to 100000 CTMA sections, and collected several statistics that reveal the

effects of transmission ranges on contentions for transmission and transmission/reception chances of each node.

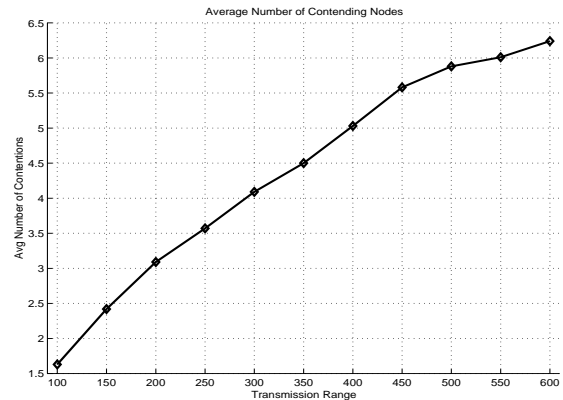


Figure 5. Average Number of Contending Nodes

Figure 5 shows the average number of contending nodes of each node for a time slot. While a node has two or fewer contending nodes on average at transmission range of 100 meters, the average number of contentions extends to 6 when transmission range becomes 600 meters.

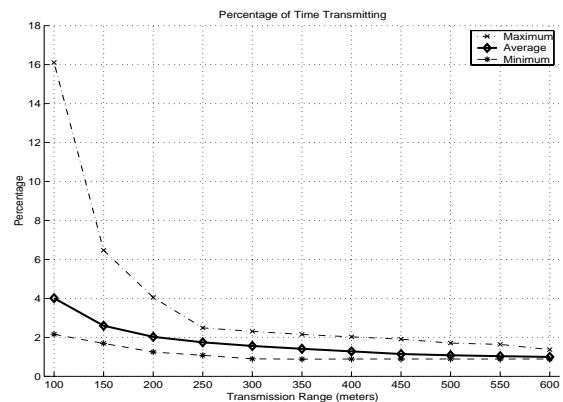


Figure 6. Percentage of Time for Transmission by Each Node

Figure 6 shows that the average percentage of time for each node to transmit is approximately inverse to the average number of contending nodes. On the other hand, Figure 7 indicates that a node receives data frames half of the time.

Because of the randomness of CTMA, we examine the histogram of intervals measured in units of section between successive transmissions by any individual node. Figure 8 shows that all nodes mostly transmit within short intervals. The number of successive transmissions with long intervals drops drastically as the time interval increases. We derive the block size used by CTMA such that the block size approximates the longest interval under certain level of contentions, and new member of the network has enough time to learn about its one-hop neighbors by listening to their transmissions.

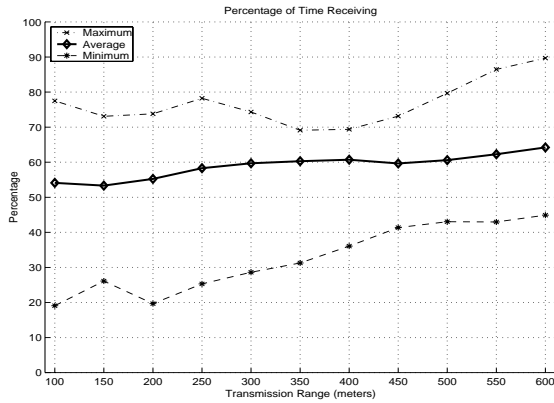


Figure 7. Percentage of Time for Transmission by Each Node

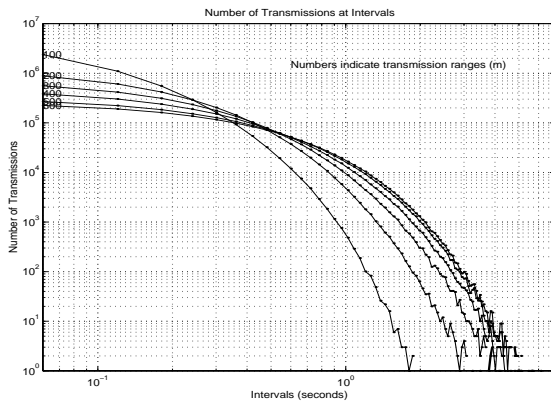


Figure 8. Histogram of Intervals between Success Transmissions

V. CONCLUSION

CTMA is a novel topology-dependent TDMA scheduling protocol that provides collision-free data transmissions without impromptu handshakes or contention phases for channel access and time slot reservations. Contentions for a time slot are resolved by a synchronized random-permutation algorithm that computes priority of each node for transmissions. CTMA also effectively handles network mobility using membership sections in the CTMA blocks.

REFERENCES

- [1] I. Chlamtac and A. Farago. Making transmission schedules immune to topology changes in multi-hop packet radio networks. *IEEE/ACM Transactions on Networking*, 2(1):23–9, Feb 1994.
- [2] I. Cidon and M. Sidi. Distributed assignment algorithms for multihop packet radio networks. *IEEE Transactions on Computers*, 38(10):1353–61, Oct 1989.
- [3] A. Ephremides and T.V. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, 38(4):456–60, April 1990.
- [4] William Feller. *An introduction to probability theory and its applications*, volume 1. New York: John Wiley, 2nd edition, 1957.
- [5] Ji-Her Ju and V.O.K. Li. An optimal topology-transparent scheduling method in multihop packet radio networks. *IEEE/ACM Transactions on Networking*, 6(3):298–306, June 1998.
- [6] Ji-Her Ju and V.O.K. Li. TDMA scheduling design of multihop packet

radio networks based on latin squares. *IEEE Journal on Selected Areas in Communications*, 17(8):1345–52, Aug 1999.

- [7] Richard J. Larsen and Morris L. Marx. *An introduction to probability and its applications*. Number p.165-167 in ISBN-0134934539. Englewood Cliffs, N.J. : Prentice-Hall, 1985.
- [8] L.C. Pond and V.O.K. Li. A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks. In *MILCOM 89*, volume 1, pages 70–4, Boston, MA, USA, 1989.
- [9] R. Ramaswami and K.K. Parhi. Distributed scheduling of broadcasts in a radio network. In *IEEE INFOCOM'89 The Conference on Computer Communications. Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 497–504, Ottawa, Ont., Canada, 23-27 April 1989. IEEE Comput. Soc. Press.
- [10] R.L. Rivest. The MD4 message digest algorithm. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90 Proceedings*, pages 303–11, Santa Barbara, CA, USA, 11-15 Aug 1991. Berlin, Germany: Springer-Verlag.
- [11] Z. Tang and J.J. Garcia-Luna-Aceves. A Protocol for Topology-Dependent Transmission Scheduling. In *Proc. IEEE Wireless Communications and Networking Conference 1999 (WCNC 99)*, New Orleans, Louisiana, September 21–24 1999.
- [12] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution. *IEEE Transactions on communications*, 23(12):1417–33, December 1975.
- [13] J.D. Touch. Performance analysis of MD5 [authentication algorithm]. In *Computer Communication Review*, volume 25, pages 77–86. ACM, Oct 1995.
- [14] Chenxi Zhu and M.S. Corson. A five-phase reservation protocol (FPRP) for mobile ad hoc networks. In *IEEE INFOCOM '98*, volume 1, pages 322–31, San Francisco, CA, USA, 29 March-2 April 1998.