

Distributed Scheduling for Wireless Mesh Networks In Realistic RF Environments

Copyright © December, 2006, IEEE. Reprinted from the proceedings of Interservice/Industry Training Simulation and Education Conference 2006 (I/ITSEC '06).

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Argon ST's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Distributed Scheduling for Wireless Mesh Networks in Realistic RF Environments

Hua Zhu, David Tang, Bo Ryu
San Diego Research Center, Inc.
San Diego, CA
{ hua.zhu, david.tang, bo.ryu }@sdrinc.net

Ajay Gummalla
Member, IEEE

ajayg@ieee.org

1. INTRODUCTION

Wireless mesh networking has recently attracted lots of interests in both academia and industry, and may soon become one of the essential technologies in our daily life. While there are many different types of wireless mesh networks serving for various purposes and user scenarios, such as short-range low energy low duty cycle sensor networks and home networks, highly dynamic combat or tactical networks, in this paper, we focus on IEEE 802.16 (IEEE Standard 802.16-2004, 2004, IEEE Std 802.16-2004/Cor1, 2005) mesh, which is the enabling technology to provide wireless mesh connectivity in the metropolitan area network scenarios.

IEEE 802.16 can operate in either a cellular-like PMP (Point-to-Multipoint) mode or a Mesh mode. The Mesh mode aims to support arbitrary multi-hop communications, which places a challenge on network capacity. Because based on the well-known theoretic work of Gupta and Kumar (Gupta, P., Kumar, P.R., 2000), the network capacity decreases substantially as the size (in terms of number of hops) of the network increases. In fact, in many cases, the network capacity is strictly limited by interference. Consequently, how to use the already limited network capacity more efficiently has become one of the most important problems for wireless mesh networks or other types of ad hoc networks. In Time Division Multiple Access (TDMA) based IEEE 802.16 networks, the design of the TDMA scheduling algorithms has significant impact on the performance of individual users (nodes) across the entire network. IEEE 802.16 provides three scheduling algorithms, i.e., centralized scheduling, coordinated distributed scheduling and uncoordinated distributed scheduling. There are tradeoffs among these algorithms, e.g., distributed algorithms are more suitable for multi-hop peer-to-peer communications whereas the centralized algorithm may be more attractive for star topology. However, this is not the interest of this paper. The scope of this paper is limited to address the problem and propose solution regarding the coordinated distributed scheduling algorithm, or CDS for simplicity in this paper.

We found that the major weakness of the CDS algorithm is its claim of “collision-free” transmissions.

The CDS algorithm is based on an easily overlooked modeling approximation that interference in wireless networks can be modeled via wired-like logical connectivity graphs, which we term quasi-interference modeling. Although the CDS algorithm can successfully achieve “collision-free” scheduling in quasi-interference environments, our simulations clearly show that it experiences substantial collisions in realistic non-quasi-interference environments, which will significantly impair the performance of 802.16 mesh networks. More details will be elaborated in Section 3. Given its open-loop scheduling approach, i.e., each node speculates the schedules of other nodes merely based on a commonly agreed random generator, collisions in non-quasi-interference environments cannot be avoided fundamentally.

To address the problem of CDS, we propose Link State Protocol (LSP), a novel distributed scheduling algorithm being designed for OT-TES¹. LSP shares the same goal of achieving efficient collision-free TDMA scheduling with CDS, but it is based on a very different paradigm of node coordination and thereby, it can truly achieve “collision-free” scheduling under the real-world non-quasi-interference RF environments. LSP takes a close-loop scheduling approach, in which all collisions can be detected and resolved. Our comparative simulation results clearly show that LSP successfully achieves true collision-free transmissions while CDS fails to do that. Results on scheduling intervals and communications overhead are also provided.

It is worth noting that, although LSP chooses a very different scheduling approach, the required modifications on existing 802.16 message format is at the minimal. Since both algorithms need to exchange neighborhood information, the only necessary change is to replace all 8-bit-per-entry scheduling fields on CDS with a single scheduling bitmap required by LSP.

The remainder of the paper is organized as follows. An overview of IEEE 802.16 Mesh mode and details of CDS algorithm are described in Section 2. In Section 3, we discuss CDS’s quasi-interference approximation that undermines the “collision-free” scheduling

¹ Operational Test-Tactical Engagement System for the U.S. Army

performance. Simulation observations are also provided in this section. In Section 4, we propose the LSP algorithm that guarantees collision-free scheduling of control messages, provided with comparative simulation results. After summarizing related works in Section 5, we conclude the paper in Section 6.

2. OVERVIEW OF IEEE 802.16 MESH MODE AND CDS ALGORITHM

2.1 General Description of IEEE 802.16 Mesh Mode

802.16 offers both Point-to-Multipoint (PMP) and Mesh (MSH) modes. The main difference between PMP and Mesh modes is that in the PMP mode, traffic only occurs between the BS (Base Station) and SSs (Subscriber Stations), while in the Mesh mode, traffic can be relayed via other SSs, and also can occur directly between SSs. It is worth noting that a concept of relay stations (RSs) and tree-based mobile multi-hop relay (MMR) networking has also been introduced for PMP mode, which may expand the coverage area and enhance throughput (Nohara, M., 2005, Asa, M., Chen, D.T., and Natarajan, N., 2005) of the PMP mode. Comparing with the moderate tree-based multi-hop network topology of 802.16mmr, 802.16 mesh mode places more challenges on the link scheduling algorithms. In order to achieve efficient collision-free multi-hop data transmissions, the Mesh mode defines three scheduling schemes, i.e., centralized, coordinated distributed, uncoordinated distributed scheduling, to resolve wireless interference occurred in the 2-hop neighborhood or 3-hop extended neighborhood of a node. According to the specification, 3-hop extended neighborhood can be used in the environment that is close to free-space.

Figure 1 shows the frame structure of the Mesh mode. Each frame is divided into two parts: i) a control sub-frame consisting of MSH_CTRL_LEN (0-15) transmission opportunities (termed $XmtOps$ by the

standard) and ii) a TDM data sub-frame consisting of up to 256 minislots. For high reliability, all the $XmtOps$ are in fixed length of 7 OFDM symbols (T_S). The duration of a minislot is often of 4 OFDM symbols. The number of minislots can be derived given the frame duration (T_F), $XmtOp$ duration, minislot duration, and the number of control $XmtOps$. There are two types of control sub-frame, i.e., schedule control sub-frame and network control sub-frame. The latter reoccurs once every $Scheduling_Frames$. The network control sub-frame provides basic functionality of network entry and topology management, while the schedule control sub-frame is to resolve the transmission schedule of data sub-frame. A node will not transmit in any minislot that is not reserved for its use. While reserved, each data transmission, i.e., a physical layer (PHY) burst with one or more MAC layer packet data units (MAC PDUs), may take multiple minislots, which is shown in Figure 2.

The network control sub-frame serves primarily for new nodes that want to gain access to the network. In each network control sub-frame, the first $XmtOp$ is dedicated to network entry message (MSH-NENT) in an unreliable contention-based access. A successful network entry relies on additional handshake messages between the new node and the sponsor node. The remaining ($MSH_CTRL_LEN - 1$) $XmtOps$ are dedicated to network configuration messages (MSH-NCFG). The schedule control sub-frame is used to schedule minislots for TDM transmissions in the data sub-frame. However, the access of $XmtOps$ in the schedule control sub-frame itself requires certain rules, which could be centralized, distributed, or a combination of both scheduling methods. If both scheduling methods co-exist, the first ($MSH_CTRL_LEN - MSH_DSCH_NUM$) $XmtOps$ within the control sub-frame are allocated for centralized scheduling messages (MSH-CSCH/MSH-CSCF), while the remaining ones are for distributed scheduling messages (MSH-DSCH).

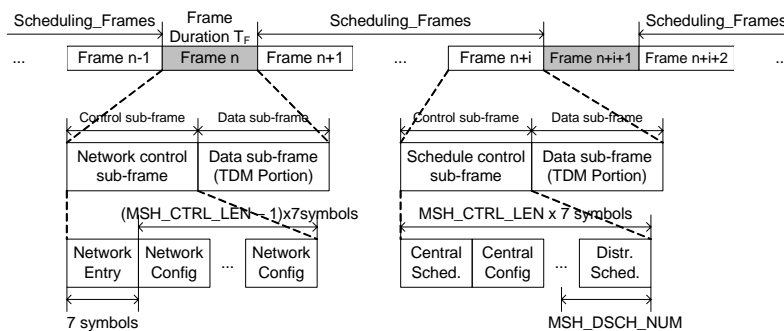


Figure 1. Mesh frame structure

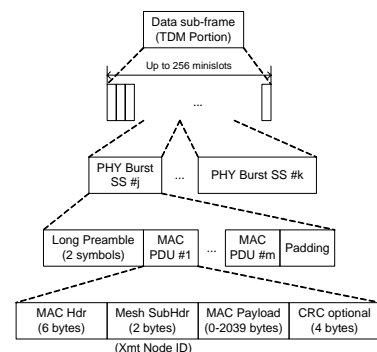


Figure 2. Data sub-frame structure

2.2 Coordinated Distributed Scheduling (CDS)

Let us first give a general picture of all specified scheduling algorithms. The centralized scheduling relies on a centralized coordinator, i.e., Mesh BS (MBS), to schedule MSH-CSCH/MSH-CSCF packets in a collision-free manner, which is the best for links supporting persistent data traffic. The distributed scheduling can be further divided into coordinated distributed scheduling and uncoordinated distributed scheduling. The former, which is used by MSH-NCFG and MSH-DSCH, accesses *XmtOps* in a “collision-free” manner. On the other hand, the latter, which can be used only by MSH-DSCH, adopts a simple contention approach in which collisions may occur if multiple nodes are transmitting in the same *XmtOp*. Uncoordinated distributed scheduling is only best for links with occasional or brief traffic needs. Coordinated distributed scheduling, which is designed for general Mesh operations with substantial peer-to-peer traffic needs, is the main focus of this paper. Due to the limit of space, we will not elaborate other scheduling algorithms in this paper. Interested readers may refer to standard specification (IEEE Standard 802.16-2004, 2004) and the tutorial for more details on Mesh model (Beyer, D., van Waes, N., and C. Eklund, 2002).

Coordinated distributed scheduling (CDS) is designed to achieve collision-free periodical transmissions for two types of control messages, i.e., MSH-NCFG and MSH-DSCH, respectively. Since the exact same algorithm is used independently for these two types of messages in separated *XmtOps*, we can simply analyze one, and results are applicable to the other.

The general concept of CDS is to let nodes running the scheduling algorithm independently derive pseudo-random but predictable behaviors by exchanging 2- or 3-hop neighborhood schedule information with each

other. Both the randomness and predictability are achieved by dynamically constructing random generator seeds for each node according to a common rule. The seed for a node is constructed based on its unique node ID and the index (or timestamp) of a candidate *XmtOp*. Given the neighborhood information, the random number generated locally will be the same with the corresponding one generated at a neighboring node. Therefore, predictability is achieved. In detail, by using previously scheduled transmission opportunities, nodes compute and exchange their next collision-free *XmtOps*, as well as any available schedules of their 2- or 3-hop neighbors, which is in the format of the following scheduling related parameters

$$\begin{cases} NextXmtXm :: 5bits \\ XmtHoldoffExponent :: 3bits \end{cases}$$

Given these two parameters of a specific neighbor, a node can determine a bounded interval for *NextXmtTime* as well as *EarliestSubsequentXmtTime* of the neighbor as the following:

$$NextXmtXm \cdot 2^{XmtHoldoffExponent} < NextXmtTime \leq (NextXmtXm + 1) \cdot 2^{XmtHoldoffExponent}$$

(1)

$$EarliestSubsequentXmtTime = NextXmtTime + 2^{XmtHoldoffExponent+4}$$

(2)

Since the exact scheduled *XmtOp* of the neighboring node is unknown, as an implementation issue, one may define *NextXmtTime* to be the last *XmtOp* within the interval when calculating *EarliestSubsequentXmtTime*. As shown in Figure 3, at the previously scheduled *XmtOp*, a node (namely the local node) will run an election algorithm to find its next collision-free *XmtOp*. Based on the calculated *NextXmtTime* time interval and *EarliestSubsequentXmtTime*, a node can

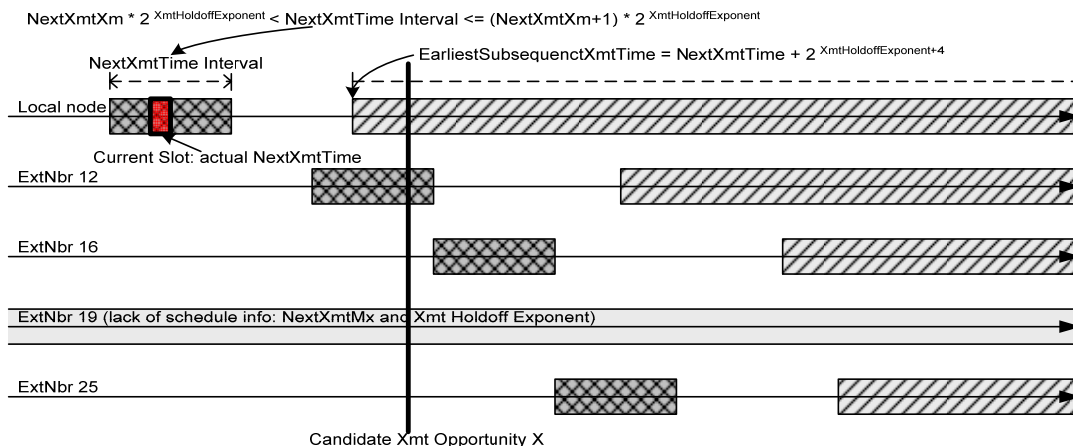


Figure 3. Coordinated distributed scheduling (CDS) – Election-based approach

exclude a subset of neighboring nodes (e.g., Nbr 16 Figure 3) from the competing neighbors of a particular candidate *XmtOp* *X*, which reduces the number of unnecessary idleness in the schedule and improves the utilization of *XmtOps*. For a particular candidate *XmtOp* starting from the *EarliestSubsequentXmtTime* of a local node, if it generates the largest random number amongst all the eligible competing nodes, it wins this *XmtOp*. Otherwise, it will keep incrementing the candidate *XmtOp* and running the same election algorithm until it wins the election (Beyer, D., van Waes, N., and C. Eklund, 2002).

3. QUASI-INTERFERENCE RF MODELING

Although the CDS algorithm intends to provide collision-free transmissions by resolving collisions incurred by concurrent transmissions in a 2-hop or 3-hop neighborhood, it does rely on an easily overlooked modeling approximation. The CDS algorithm is mainly based on a simplified quasi-interference RF model in which wireless networks are approximated with wired-like logical connectivity graphs. In such a model, collisions will not take place at the receiver if there is no connectivity edge to link the receiver to the concurrent senders. It appears like that RF signals are completely cut-off beyond all 1-hop neighbors with zero energy propagated further.

Evidently, this idealized assumption cannot be held true in a realistic non-quasi interference environment, where the accumulative interference from multiple senders far away from the receiver may damage the transmission from a direct neighbor of the receiver. Figure 4 shows two examples in which interference may still cause collision even though the concurrent senders are 3-hop or 4-hop away. In Figure 4(a), node 1 and node 4 are technically 3-hop away. However, since two receivers (node 2 and 3) are very close, concurrent transmissions (1-to-2, and 4-to-3) generate substantial interference to both receivers such that neither one has enough SINR for correct reception. In Figure 4(b), the accumulative interference contributed by multiple senders (node 11, 12 and 13) may reduce SINR at receiver node 2 to an unacceptable level as well. It is worth noting that the number of hops between two nodes in the logical connectivity graph may not necessarily reflect the actually distance between two nodes. The number of hops is highly topology dependent. For example, the distance between node 1 and 13 is far shorter than the one between node 1 and 11, although they are both 4-hop away.

Aware of this modeling discrepancy, CDS adopts a “quick fix” in that scheduling can be done in extended

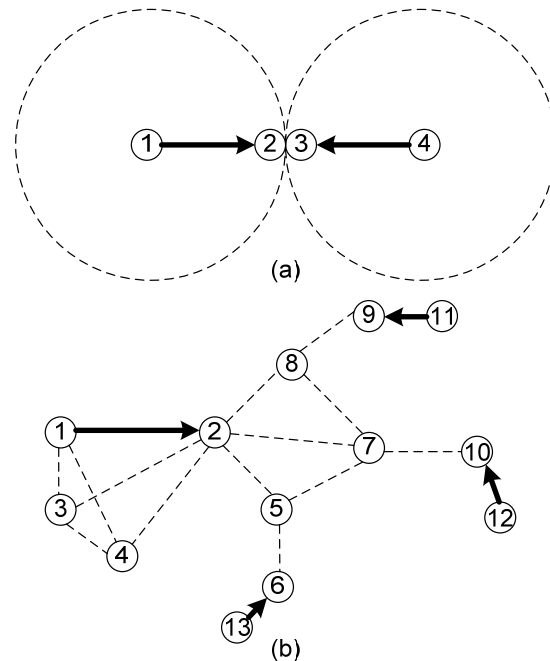


Figure 4. Collisions under non-quasi-interference model

3-hop neighborhood (specified by *ExtendedNeighborType*) when necessary (in Free-Space like environments). However, the paradigm of distributed node coordination used by the CDS algorithm has not been changed. Such paradigm fundamentally cannot prevent such collisions from happening. Furthermore, unconditionally applying 3-hop scheduling to all the senders makes the CDS algorithm less efficient in terms of spatial re-use since some of them may actually operate well with 2-hop scheduling. Nevertheless, based on our extensive simulations, unfortunately we find that the collisions involved in the CDS algorithm are substantial, even for 3-hop scheduling in certain non-free-space-like RF environments. We have observed that, in specific scenarios, the collision ratio can be up to 20% for 2-hop scheduling, and 7% for 3-hop scheduling. Evidently, such high percentage of collisions will significantly impair the performance of 802.16 MAC since many control messages, which rely on the CDS algorithm to access the channel, may not be received by certain neighbors, if not all of them. Moreover, since the scheduling of data transmissions has to rely on the control message handshakes, the actual performance experienced by upper layer will be varied significantly, if not unpredictable.

Extensive simulations have been conducted to study the performance of the CDS. We implement the IEEE 802.16 Mesh Mode in QualNet (QualNet User’s Manual, 2005), which has the capability of modeling

realistic non-quasi-interference environments with a broad variety of propagation models. An OFDM system profile of Mesh MAC (ProfM3_Mesh) and 7MHz channelization at ETSI bands (ProfP3_7) has been used (for details, please refer to (IEEE Standard 802.16-2004, 2004)). All performance relevant configuration parameters are listed in Table 1.

Table 1. Simulation configuration

PHY Profile: ProfP3_7: WirelessMAN-OFDM PHY profile for 7 MHz channelization	
Carrier frequency	3.5GHz ETSI licensed band
Channel bandwidth	7 MHz
Modulation	64-QAM 3/4 (Assuming fixed rate)
E_b/N_0 (for BER<10 ⁻⁶)	19.0 dB
OFDM raw bitrates	21.60 Mb/s (for $T_g=T_b/4$)
Receiver SNR threshold	23.894 dB
Noise power	-97.073 dBm (2.8027e-17 mW/Hz)
Receiver power sensitivity threshold	-73.179 dBm (minimum performance requirement -70 dBm)
Transmission Power	34.77 dBm (3W)
Antenna height	1.5 m
Antenna gain plus all implementation losses	-5.0 dB
MAC Profile: ProfM3_Mesh: WirelessMAN-OFDM Basic Packet Mesh MAC profile	
Frame duration	code {0x01} $T_F = 4ms$
Symbol duration	$T_S = 3.7037 \mu s$
Minislot size	1PS = 4 symbols
Control opportunity size	7 symbols
Scheduling Frames	8
MSH_CTRL_LEN	8
MSH_DSCH_NUM	8 (all for coordinated distributed scheduling)
Number of minislots	256

The following combinations of propagation models (i.e., Free Space, Two-Ray, ITM²) and multi-hop topologies (grid, uniform distribution, actual terrain sample) with 49 nodes are constructed. Given the fixed high data rate with 64-QAM ³/₄ modulation and relative large channel bandwidth, the radio range is limited. Therefore, we are able to construct multi-hop scenarios with a reasonable test area size.

- Free space model + 49 nodes in grid formation with a grid distance of 300 m (network diameter = 8 hops)
- Free space model + 49 nodes uniformly distributed in 2.5kmx2.5km (network diameter = 10 hops)
- Two ray model + 49 nodes uniformly distributed in 2.5kmx2.5km (network diameter > 10 hops, not fully connected)

² Irregular Terrain Model (ITM)

- ITM model + 49 nodes uniformly distributed in 1.1kmx1.1km mountainous terrain with relative elevation range of approximately 200 m. (network diameter = 10 hops). ITM is a terrain based RF model based on Longley-Rice model.

Identical $XmtHoldoffExponent$ are used for all scenarios in this paper. However, consistent results are observed for non-identical $XmtHoldoffExponent$ cases. As we mentioned in Section 2.2, we only present results for both MSH-NCFG, but the same performance can be observed for MSH-DSCH as well with the exact same scheduling behaviors. The following set of performance metrics is collected:

- Reception collision ratio (%): overall collision ratio seen at receivers (only collisions caused by interference from concurrent transmissions are counted)
- Scheduling interval ($XmtOp$): average number of corresponding transmission opportunities between two consecutive scheduled slots
- Number of extended neighbors: average number of neighbors in the n -hop extended neighborhood

In the simulation, we can identify error receptions caused by interference from concurrent transmissions (i.e., received SINR³<SNR⁴ threshold, but received SNR³≥SNR threshold, and received power≥receiver power sensitivity). Note that the receiver power sensitivity and SNR threshold have been carefully configured based on bit-energy-to-noise-density ratio (E_b/N_0) so that reception error will not occur if the only source of interference is local thermal noise. Therefore, reception error implies external interference, or reception collision. Also note that we only collect statistics in stabilized condition, and therefore we can exclude collisions in transient condition.

Figure 5 shows the non-zero reception collision ratio, which verifies our analysis in this section. With the smallest $XmtHoldoffExponent$, nodes have high possibility to schedule concurrent transmissions beyond 2- or 3-hop scheduling neighborhood. Based on the idealized quasi-interference model, such spatial re-use and concurrent transmissions should not cause any collision. However, we observe substantial amount of collisions for both 2-hop scheduling (up to 20.78%) and 3-hop scheduling (up to 6.76%). The reception collision ratio decreases when $XmtHoldoffExponent$ increases from 0 to 4, for extend neighborhood type of 2 and 3, respectively. Note that it decreases quite fast

³ SINR: Signal-to-Interference-plus-Noise-Ratio

⁴ SNR: Signal-to-Noise-Ratio

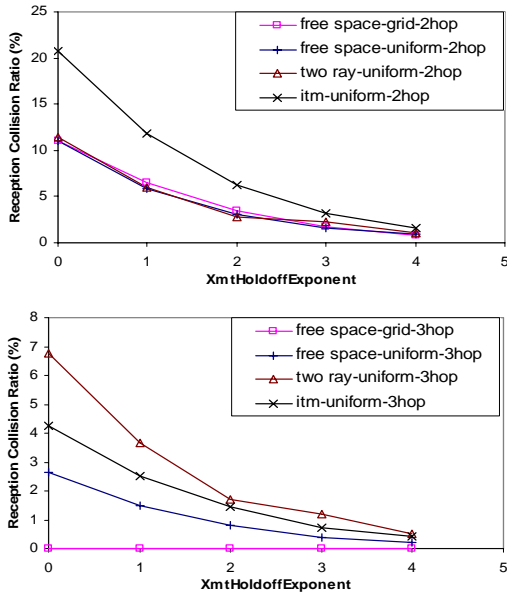


Figure 5. Collision ratio vs. XmtHoldoffExponent
(a) ExtendedNeighborhoodType = 2 hop (b) 3 hop

when $XmtHoldoffExponent$ increases from 0 to 1 and from 1 to 2, but slower for larger $XmtHoldoffExponents$ (3 and 4). This is because, for the latter, the holdoff interval (128 and 256, respectively) is substantially larger than the number of nodes in the affected interference neighborhood, even larger than the total number of nodes in the network. As a result, the probability of concurrent transmissions has already been significantly “diluted”, and keeping incrementing the holdoff interval does not benefit much more. This result may be useful for selecting appropriate $XmtHoldoffExponent$ to alleviate collisions. However, increasing $XmtHoldoffExponent$ also results in longer schedule interval, which may lead to undesirable turn around time for data scheduling handshakes. Furthermore, since each successful data scheduling handshake has to be achieved by successful transmissions of multiple control messages, the compound failure rate of handshakes is even more significant than the collision ratio of individual control messages.

Figure 6 shows the overall average schedule interval with respect to $XmtHoldoffExponent$, for 2- and 3-hop neighborhood, respectively. Since for most cases in our simulation (except for $XmtHoldoffExponent = 0$ in 3-hop extended neighborhood scenarios), the number of neighbors in the extended neighborhood is equal to or smaller than the holdoff interval, therefore, the resulted schedule interval is close to the lower bound specified by the holdoff interval (i.e., the dot lines in Figure 6(a) and (b)), which is equal to $2^{XmtHoldoffExponent}$.

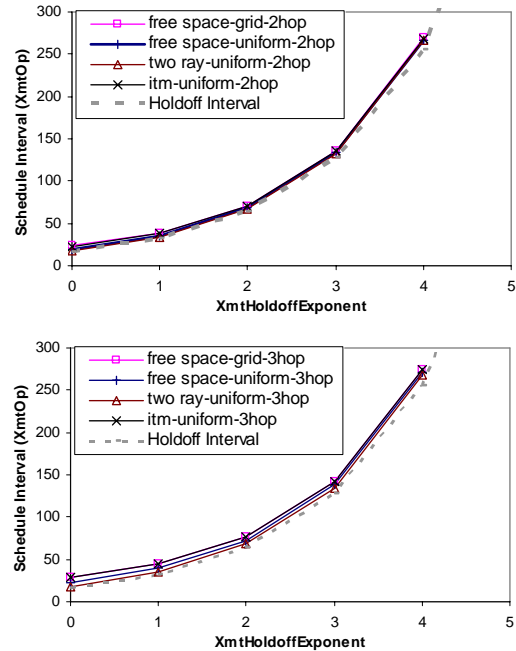


Figure 6. Schedule Interval vs. XmtHoldoffExponent
(a) ExtendedNeighborhoodType = 2 hop (b) 3 hop

4. LINK STATE PROTOCOL (LSP)

Based on the analysis and simulations in Section 3, we recognize that the present CDS algorithm in IEEE 802.16 fundamentally cannot guarantee collision-free scheduling. Note that the collisions we observed are not due to transient states while the algorithm is converging. Instead, those collisions will always exist with certain statistical probability depending on the network topology and RF environment. The main reason is that, CDS solely relies on the predictable pseudo random generator at each node to independently scheduling $XmtOps$. Such type of node coordination paradigm can only prevent quasi-interference collisions within the neighborhood range, but it lacks means to detect collisions due to non-quasi-interference.

To solve this problem we propose LSP, a true collision-free coordinated distributed scheduling algorithm that follows a very different paradigm in terms of how nodes actively interact and coordinate with each other. LSP can be used to replace present CDS for control sub-frame scheduling. Although LSP leverages the concepts of quasi-interference approximation and logical connectivity graph, it adopts a more proactive collision monitoring and adaptation mechanism. Therefore, by using LSP algorithm, collisions, regardless whether it is caused by direct neighbors of the receiver (i.e., quasi-interference) or accumulative interference other than neighbors (non-

quasi-interference), are all tracked and used by the sender to adjust the schedule accordingly. In this way, LSP maintains a high scheduling efficiency by maximizing spatial re-use beyond 2-hop as much as possible, but it is also capable of adjusting schedule appropriately if collisions due to non-quasi-interference beyond 2-hop are detected. Unlike CDS, collisions only occur in the transient state of LSP.

Additionally, given recent development of IEEE 802.16e, the perspective of mobile metropolitan area mesh networks may soon attract more interests. While the current CDS algorithm is not designed to support mobility scenarios, LSP is robust and can be directly applied to mobile mesh networks with only minor modifications, while present CDS has to resolve the additional problems, e.g., how to carefully balance the neighbor lifetime and node mobility. However, we will leave the mobility factor out of the scope of this paper.

4.1 LSP Algorithm

In current CDS, each transmitted message includes 8 bits scheduling information for itself and 8 bits per its neighbor. In the proposed LSP, each transmitted message includes a scheduling bitmap (named “Transmitted Bitmap”, or TB) of a pre-configured LSP scheduling cycle. Each node running LSP will periodically send one message per scheduling cycle. The number of $XmtOps$ (N_{cycle}) per scheduling cycle closely approximates the role of $XmtHoldoffExponent$ in CDS. Only 1 bit information per $XmtOp$ is needed, therefore scheduling cycle is of N_{cycle} bits. In addition to the TB, each node keeps an addition N_{cycle} -bit long Local Bitmap (LB) to track accumulative scheduling information provided by the scheduling bitmaps inside received messages from its neighbors. To distinguish all the received TBs from the TB a node itself sends out, we denote the former RBs in the context of receivers thereafter. With two bitmaps (TB and LB), nodes can differentiate direct neighbors from indirect neighbors, which prevent flooding scheduling information of indirect neighbors across the entire network and end up with no spatial re-use. Two more local variables, namely $myXmtOp$ and $ifTx$ respectively, are required by LSP. $myXmtOp$ stores the scheduled $XmtOp$ of the node (a value of “-1” stands for “unscheduled”). $ifTx$ is used by the decision making of conflict (collision) detection. The pseudo code of the LSP algorithm is shown in Figure 7.

On the sender side, a node will send out its own TB in its previously scheduled $XmtOp$. At the beginning of an $XmtOp$, if a node has $myXmtOp = -1$, which may occur during the initialization or convergence phase of

the algorithm, a node will randomly pick an idle $XmtOp$ based on its available LB information.

On the receiver side, when a node receives a RB, it updates its TB and LB with different update rules. For TB, it only sets the bit associated with the current $XmtOp$ in which it received the RB to one (refer to Line 31 in Figure 7). For LB, it performs an OR operation with the RB (refer to Line 33 in Figure 7). Relying on these two update rules, TB reflects the direct neighbor environment of the node, and LB reflects the accumulative environment of the node. A “1” in the LB stands for that corresponding $XmtOp$ is taken by a 2-hop neighbor and therefore ineligible for use by the owner of that LB during scheduling. LSP is memoryless in that it only keeps one-cycle scheduling information. Therefore, no additional signaling overhead is needed if any rescheduling occurs.

```

Line1 (curXmtOp and myXmtOp are wrt. XmtOp index)
Line2
Line3 Init{
Line4 TB = 0; // array size = Ncycle
Line5 LB = 0; // array size = Ncycle
Line6 myXmtOp = -1; // my reserved slot
Line7 ifTx = false;
Line8 counter = 0; // number of conflicting neighbors
Line9 curXmtOp = -1;
Line10 set XmtOp timer to zero delay
Line11 }
Line12
Line13 ProcessXmtOpTimerAtSlotBeginning{
Line14 curXmtOp ++;
Line15 curXmtOp % Ncycle;
Line16 set XmtOp timer to a XmtOp duration;
Line17 If (myXmtOp = -1) AND (have listened for one cycle after bootup or net entry)
Line18     random pick myXmtOp among (myXmtOp AND all other i's that LB[i]=0);
Line19 else if (curXmtOp = myXmtOp) AND (counter > 0)
Line20     random pick myXmtOp among (myXmtOp AND all other i's that LB[i]=0);
Line21 counter = 0;
Line22 ifTx = false
Line23 if (myXmtOp)
Line24     TB[myXmtOp] = 1;
Line25 Transmit TB;
Line26 TB = 0; // the whole array is reset
Line27 LB = 0; // the whole array is reset
Line28 ifTx = true;
Line29 }
Line30 Recv_Message_with_RB{
Line31 TB[curXmtOp] = 1;
Line32 // OR for all elements in the array, RB is the received TB
Line33 LB = LB OR RB;
Line34 // Collision Check
Line35 If (RB[myXmtOp] = 0) AND (ifTx = true)
Line36     counter++;
Line37 }

```

Figure 7. LSP algorithm pseudo code

Collision detection is also performed upon the reception of a RB. Inside a RB, if the bit corresponding to $myXmtOp$ is not shown as “1” and $ifTx = true$, a node detects reception failure of its previous transmission at the originator of RB (refer to Line 35 in Figure 7). Note that the reception failure can be caused by 1-hop collision at the receiver, or accumulative interference multi-hop away from the receiver. The latter factor is the fundamentally undetectable case in CDS. A node counts the number of such conflicts for an entire cycle, and then decides if re-scheduling is needed. Current implementation of LSP (refer to Line

19 in Figure 7) enforces a zero-tolerance of such reception failure, which means it strictly ensures the successful delivery of its scheduled message to all its 1-hop neighbors. Nevertheless, the LSP algorithm is generic and can be customized with non-zero thresholds based on specific requirements. Furthermore, an even more powerful decision component can be developed to differentiate receivers such that the scheduling algorithm transforms from a simple broadcast scheduling algorithm to for unicast/multicast oriented scheduling algorithm.

4.2 Simulation Results

Figure 8 shows the performance comparison between current CDS approach and the proposed LSP approach in one of the same scenarios (Free space model + 49 uniformly distributed nodes) described in Section 3. Other scenarios show consistent results though. All the configurations are the same.

In the top figure, it is clear that LSP incurs zero collision for various holdoff intervals except for the cases of 16 and 32 $XmtOps$. For these two cases, LSP only averages an negligible 6.51 and 1.41 collisions per node for the entire simulations, which are all happened during the initial convergence phase. By using LSP, nodes are able to detect collisions caused by accumulative interference (non-quasi-interference) beyond 2-hop neighborhood. Therefore, once they detect such collisions, they were able to re-schedule promptly. It only takes a few rounds of trials (average 6.51 and 1.41 for holdoff interval of 16 and 32, respectively) before all schedules converge. When the number of neighbors is close to the holdoff interval, the algorithm takes more rounds to converge. Of course, if the number of neighbors is larger than the holdoff interval, there will be nodes being blocked without any $XmtOp$ to schedule a transmission. Appropriate configuration of holdoff interval should be larger than the maximum number of nodes in any 2-hop neighborhood.

In the middle figure, one can see that nodes in LSP schedule transmissions at a constant pace of the exact configured holdoff interval (except during the convergence), while the average schedule interval of CDS consists of extra amount of $XmtOps$, which may depends on the number of neighbors of a node.

In the bottom figure, the communications overhead in terms of the length of specific scheduling fields in the control messages is compared. Since we are proposing to use LSP to replace CDS, other control message functionalities and signaling specifications remain the

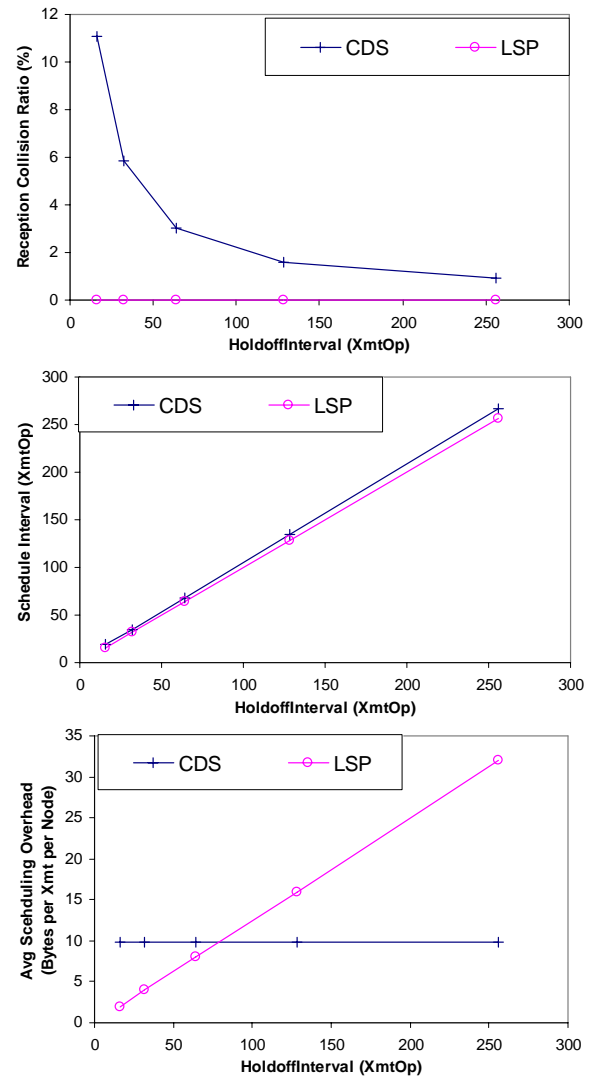


Figure 8. Performance comparison between CDS & LSP

same. Therefore, we can compare the overhead difference of corresponding scheduling fields exclusively. In this particular shown scenario, the overall average number of neighbors is fixed, therefore the scheduling overhead of CDS is constant for all holdoff intervals. Alternatively, the scheduling overhead of LSP increases if the holdoff interval increases. However, as long as the duration of control $XmtOp$, which is designed to be fixed and fairly large, is able to accommodate the size of the control message, the same portion of channel resource is consumed by the control sub-frame in the TDMA-based 802.16 airlink protocol. It is worth noting that theoretically, LSP has higher node density upper bound than CDS. Let N denote the maximum number of nodes in any 2-hop neighborhood. The holdoff interval of LSP, which is also the number of bits of scheduling overhead for LSP, only has to be set to equal to or larger than N . On

the other hand, for CDS, the scheduling overhead is $(8\text{bits} * N)$, which is 8 times in size of LSP scheduling overhead. Practically, the largest holdoff interval that the current CDS can support is constrained by the bits assigned to $XmtHoldoffExponent::3\text{bits}$. Based on (2), it requires a total of 256 bits fixed scheduling overhead for LSP, a CDS scenario with equivalent scheduling overhead will have 31 neighbors for every node. One can argue that this is still within an acceptable range.

5. RELATED WORKS

5.1 IEEE 802.16 / WiMax

As relatively new standard, IEEE 802.16 has been studied much less than access technologies such as IEEE 802.11. Eklund et. al. presented an system level overview of 802.16 standards family in (Eklund, C., Marks, R., Stanwood, K.L., and Wang, S., 2002). Redana and Lott modeled and compared the control message overhead between centralized and distributed scheduling mechanisms in (Redana, S., and Lott, M., 2004). From a different angle, Cao et. al. proposed a theoretic model to compute the schedule interval of 802.16 coordinated distributed scheduling in (Cao M., Ma, W., Zhang, Q., Wang, X., and Zhu, W., 2005). With the algorithm to grant data requests left open in the standard, the schedule interval is an important common performance metric that reflects the scheduling latency of coordinated distributed scheduling. Both general formulation and practical computable approximation under the assumption of geometric distribution of scheduling attempts after *EarliestSubsequentXmtTime* are presented. The model is validated through NS-2 simulation, which matches well with our QualNet results. Other related topics include QoS support in Mesh mode (Liu, F., Zeng, Z., Tao, J., Li, Q., and Lin, Z., 2005), and cross-layer optimization of routing based on MAC layer scheduling behaviors (Wei, H.-Y., Ganguly, S., Izmailov, R., and Haas, Z.J., 2005, Shetiya, H., and Sharma, V., 2005, Tao, J., Liu, F., Zeng, Z., and Lin, Z., 2005). It is worth noting that maximizing spatial reuse and concurrent transmissions is the major idea behind many of such optimization efforts. However, most of proposed optimization schemes are based on the simplified quasi-interference model, in which accumulative interference beyond one hop of the sender is simply ignored. One example is the metric used to select the “best” neighbor to maximize concurrent transmissions at net entry (Tao, J., Liu, F., Zeng, Z., and Lin, Z., 2005). Under realistic non-quasi-interference environment, the actual performance of such schemes needs more careful evaluation.

5.2 Distributed Scheduling in TDMA-based Multi-hop Ad-hoc Networks

Multi-hop scheduling is one of the fundamental problems of ad hoc networks because the capacity of ad hoc networks degrades substantially when the network size increases (Gupta, P., Kumar, P.R., 2000). Although distributed scheduling algorithms have been studied for other access technologies as upper layer components, slot scheduling for the medium access is a signature problem for TDM-based MAC problem. For a peer-to-peer TDMA system in which there is no logic difference between uplink and downlink, there are a large amount of research work resort to additional mechanisms from PHY, such as introducing CDMA overlay, or emulating full-duplex time slot consisting of paired or more mini slots for signaling back and force, or relying on the aggressive multi-channel network architecture, etc. However, there are a few algorithms that are able to achieve distributed TDMA slot scheduling without significant modifications of simple TDMA radios. We give two representatives as follows.

One such algorithm is SEEDEX, developed by Rozovsky and Kumar (Rozovsky, R., and Kumar, P.R., 2001). SEEDEX’s idea of using common seed construction rule and letting nodes independently generate predictable pseudo random schedules was innovative and very similar to current CDS of 802.16. However, SEEDEX does not intend to create collision-free scheduling, instead it only uses seed information aggressively to minimize the number of active contending 2-hop neighbors in a Slotted-Aloha access method. As a result SEEDEX is expected to have performance improvement over Slotted-Aloha. However, it has been observed that the performance of SEEDEX is subjected to certain parameters very sensitively.

Another class of algorithms includes NAMA, LAMA, PAMA and HAMA, developed by Bao and Garcia-Luna-Aceves (Bao, L. and Garcia-Luna-Aceves, J.J., 2004). Among all four schemes, only NAMA does not require additional CDMA overlay. These algorithms adopt the same idea of seed construction. However, they use the available neighbor information conservatively to avoid concurrent transmissions within 2-hop neighborhood. It is worth noting that, like 802.16 CDS, these algorithms are vulnerable to non-quasi-interference channel as well. Therefore true “collision-free” scheduling is not achieved.

Note that both SEEDEX and NAMA fall into the same paradigm of 802.11 CDS, that letting nodes schedule slots independently with shared random seeds and

predictable pseudo random generator. This type of coordination is not enough to prevent collisions due to accumulative interference in realistic non-quasi-interference environments.

6. CONCLUSION

In this paper, we studied the performance of IEEE 802.16 coordinated distributed scheduling under realistic non-quasi-interference model. We observed that substantial amount of collisions may occur even with 3-hop scheduling, which is designed to alleviate such collisions. We then proposed a LSP algorithm and evaluate its performance through simulations. LSP adopts a very different scheduling paradigm, and thus, it successfully achieves collision-free scheduling under the same scenarios that CDS failed.

REFERENCES

- IEEE Standard 802.16-2004 (2004), *IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, Oct. 2004.
- Beyer, D., van Waes, N., and C. Eklund, (2002), *Tutorial: 802.16 AC Layer Mesh Extensions Overviews*, IEEE 802.16 (document S802.16a-02/30), St. Louis, March 11, 2002.
- Eklund, C., Marks, R., Stanwood, K.L., and Wang, S., (2002), *IEEE Standard 802.16: A Technical Overview of the WirelessMAN™ Air Interface for Broadband Wireless Access*, IEEE Communications Magazine, June 2002, pp.98-107.
- Cao M., Ma, W., Zhang, Q., Wang, X., and Zhu, W., (2005), *Modeling and performance analysis of the distributed scheduler in IEEE 802.16 mesh mode*, in Proceedings of the 6th ACM Int'l Symp. on Mobile Ad Hoc Networking and computing (MobiHoc'05), 2005, pp. 78-89.
- Liu, F., Zeng, Z., Tao, J., Li, Q., and Lin, Z., (2005), *Achieving QoS for IEEE 802.16 in Mesh Mode*, 8th Int'l Conference on Computer Science and Informatics, Salt Lake City, Utah, July, 2005.
- Redana, S., and Lott, M., (2004), *Performance Analysis of IEEE 802.16a in Mesh Operation Mode*, Proc. of the 13th IST SUMMIT, Lyon, France, June, 2004.
- Wei, H.-Y., Ganguly, S., Izmailov, R., and Haas, Z.J., (2005), *Interference-aware IEEE 802.16 WiMax mesh networks*, 61st IEEE Vehicular Technology Conference (VTC 2005-Spring), Vol. 5, 2005, pp. 3102-3106.
- Shetiya, H., and Sharma, V., (2005), *Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks*, in Proc. of the 1st ACM workshop on Wireless multimedia networking and performance modeling, 2005, pp. 140-149.
- Tao, J., Liu, F., Zeng, Z., and Lin, Z., (2005), *Throughput Enhancement in WiMax Mesh Networks Using Concurrent Transmission*, Int'l Conference on Wireless Communications, Networking and Mobile Computing, 2005. pp. 871-874.
- QualNet User's Manual*, (2005), version 3.8, Scalable Network Technologies, Inc. 2005, available at <http://www.scalable-networks.com/>.
- Gupta, P., Kumar, P.R., (2000), *The capacity of wireless networks*, IEEE Transactions on Information Theory 46(2), pp. 388-404, 2000.
- Rozovsky, R., and Kumar, P.R., (2001), *SEEDEX: A MAC protocol for ad hoc networks*, ACM Mobihoc'01, Oct. 2001.
- Bao, L. and Garcia-Luna-Aceves, J.J., (2004), *Distributed Channel Access Scheduling for Ad Hoc Networks*, book chapter in "Algorithms and Protocols for Wireless and Mobile Networks", Edit. A. Boukerche, Publishers: CRC/Hall Publisher, 2004.
- Nohara, M., (2005), *Ad Hoc Meeting Report: Mobile Multihop Relay Networking in IEEE 802.16*, document IEEE 802.16-05/51, 21 July 2005.
- Asa, M., Chen, D.T., and Natarajan, N., (2005), *Concepts for 802.16-based Mobile Multihop Relay Networking*, document IEEE C802.16-05/15, 19 July 2005.
- IEEE Std 802.16-2004/Cor1, (2005), *Corrigendum to IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, Draft 5, 2005-09-12.