Access Control in Heterogeneous Multichannel Wireless Networks

Michael P. McGarry and Martin Reisslein
Department of Electrical Engineering
Arizona State University
Tempe, Arizona 85287
Email: {michael.mcgarry,reisslein}@asu.edu
Phone: +1-480-965-8593

Violet R. Syrotiuk
Department of Computer Science and Engineering
Arizona State University
Tempe, Arizona 85287
Email: syrotiuk@asu.edu
Phone: +1-480-965-7034

***Abstract*— We propose a multichannel wireless network architecture that supports nodes with differing multichannel capabilities. Both contention and contention-free medium access control (MAC) protocols are proposed for this heterogeneous multichannel network. In the contention MAC protocol, a new LRN control packet is defined that allows a pair of nodes to re-negotiate channel selection. This relaxes the assumption that nodes can sense the carrier on all supported channels in a short period of time. We model the scheduling problem for our contention-free MAC protocol as a weighted bipartite matching problem that minimizes the sum of the completion times with polynomial time complexity. We close with an illustration showing how the weighted bipartite matching generates a transmission schedule.**

I. Introduction

Driving the need for more efficient use of available spectrum are software defined radios (SDRs) [1] that allow for software control of certain parameters of the transceiver operation in the interest of dynamic reconfiguration (e.g., carrier frequency, modulation scheme, etc.) [2], [3]. SDRs will be a key enabler for networks with multichannel capabilities. This necessitates the need for MAC protocols that can take advantage of capacity available across a wide spectrum such as that proposed in DARPA's Next Generation (XG) [4] vision. Another important characteristic for future MAC protocols is the need to interoperate with legacy (i.e., non-multichannel) nodes with such emerging multichannel nodes. These multichannel networks can provide a significant increase in available bandwidth as compared to single channel networks that are constrained to the bandwidth available on a single communication channel.

Multichannel networks can also provide benefits beyond additional bandwidth. For networks employing a contention medium access control (MAC) protocol, multiple channels can reduce the probability of collisions and thereby improve throughput as a result of fewer back-offs and reduced interference [5], [6]. This is true even if the aggregate capacity of the multiple channels is the same as the capacity of the single channel network. Multiple channels can also aid in providing Quality of Service (QoS). Channels can be allocated for certain traffic classes. As well, admission control can be implemented on some channels, while other channels are allocated for best effort traffic. Further, a contention MAC protocol could operate on some channels, while others operate via a contention-free MAC protocol.

In this paper, we describe a multichannel wireless network where nodes are heterogeneous with regard to their multichannel support (e.g., multichannel capable as well as legacy single channel nodes). In other words, each node has its own set of the multiple channels that it supports. We then propose multichannel MAC protocols for this network. In our proposed MAC protocols we assume symmetric channel support (i.e., if a channel is supported then it is supported for both reception and transmission). However, minor extensions to the protocols can be made to provide asymmetric channel support.

To our knowledge, our protocol is the only multichannel MAC protocol to support a heterogeneous multichannel environment. This flexibility allows for nodes to have different transceiver capabilities.

This paper is organized as follows. In Section II we discuss related work. Section III describes our proposed architecture. In Sections IV and V we propose contention and contention-free media access protocols, respectively. Finally, Section VI concludes our work with a discussion of future research directions.

II. Related Work

We will now outline the existing research work related to multichannel wireless networks.

Jung et al. [7] do not make the assumption that all nodes can always listen to the control channel. Therefore, a multichannel hidden terminal problem arises. To deal with this all nodes are forced to use the control channel during a special window, modeled after the Power Saving Mode (PSM) of 802.11; this requires node synchronization.

The multichannel hidden terminal problem is illustrated in Figure 1. We can see that because nodes X and Y are tuned to channel 2 for their data transmission, they did not hear the Request to Send (RTS) and Clear to Send (CTS) message exchange between nodes Z and W on the control channel. This results in a collision on channel 2.

In [9], Choi et el. propose another solution to the multichannel hidden terminal problem. They propose setting a maximum transmission time (MTT) on each channel. This guarantees

InterSense '06. Proceedings of the First International Conference on Integrated Internet Ad hoc and Sensor Networks, May 30-May 31 2006, Nice, France

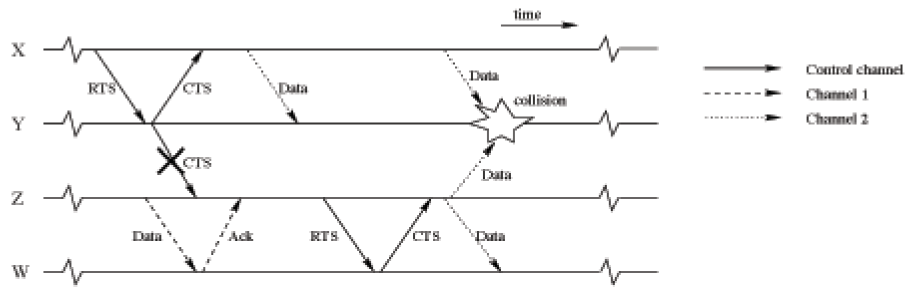

Fig. 1. Multichannel Hidden Terminal Problem [8]

that a data channel is not busy for longer than this MTT. Thus allowing nodes to estimate the free time of a channel for which they have no up to date Network Allocation Vector (NAV) due to the multichannel hidden terminal problem. Setting MTTs of unique values for each channel has the added benefit of aiding QoS support. Channels allocated for delay sensitive traffic can have MTTs set to small values to guarantee a channel is not blocked for long periods of time. Channels allocated for best effort data traffic can have very large MTT values, so as to not cause fragmentation for large packets.

In our proposed MAC protocols we avoid the multichannel hidden terminal problem by requiring that all nodes support a “base” channel that is used as the sole control channel and further that this control channel can be continuously monitored. In our conclusion we discuss briefly how we can augment our proposal to support a network where the multichannel hidden terminal problem arises.

Nasipuri et al. [10] propose a contention MAC for a multichannel network with one control channel and N data channels. All nodes continuously monitor the control channel, eliminating the multichannel hidden terminal problem. The protocol selects the idle channel (assuming it can sense carrier on all channels quickly) that has been used most recently.

An extension to the above work is proposed in [11] where the channel selection is now performed at the receiver as well. A free channel in common between transmitter and receiver is selected.

Wu et al. [12] also propose a multichannel MAC that performs channel selection at both the receiver and the transmitter.

Garces and Garcia-Luna-Aceves propose a multichannel network in which each node has its own receiving channel within a two-hop neighborhood [13]. Time is then divided into reception and transmission periods. Nodes compete using RTS packets with a tree-splitting back-off algorithm to avoid collisions. This architecture does not scale well because each node in a two-hop neighborhood must have a unique channel. As well, since a channel is fixed to a node this protocol can not take advantage of statistical multiplexing across nodes.

Two thorough surveys on multichannel MACs for wireless networks are available [6], [8].

All of the multichannel MAC protocols proposed in the research literature assume all nodes support the same set of channels (homogeneous multichannel environment). We propose an architecture that, to our knowledge, is the only multichannel MAC protocol to support a heterogeneous multichannel environment.

III. Heterogeneous Multichannel Wireless Network Architecture

We propose a multichannel wireless network that supports heterogeneity with regard to each node’s support of multiple channels. We envision a heterogeneous network consisting of single channel legacy nodes along with new emerging multichannel nodes. The emerging multichannel nodes will have a diverse set of multichannel transceivers. Some multichannel nodes can be implemented with a single transceiver, utilizing Software Defined Radio (SDR) technology, thereby enabling the node to “tune” to any of a set of multiple channels by changing some “soft” parameters in the transceiver. Existing alongside these can be nodes that support multiple channels by having transceiver arrays that are made possible by RF circuit integration advances.

This heterogeneous multichannel network will not impose any particular multichannel transceiver architecture on any node. The only requirement in our heterogeneous network architecture is that all nodes support a “base” channel that acts as the control channel and default data channel. This channel can be thought of as a “lobby” where nodes meet by default to negotiate a place to conduct their transmission. All nodes must be able to continuously monitor this channel to avoid the multichannel hidden terminal problem. For legacy single channel nodes, the “base” channel would be the only channel supported.

A method is needed to enumerate the channels. This enumeration scheme needs to be known by all nodes, so a standard should be developed. A detailed discussion of this is outside the scope of this paper. We simply assume that a common enumeration scheme exists.

IV. Contention MAC Protocol

In this section we propose a contention MAC protocol for the heterogeneous multichannel wireless network described in the previous section.

We now detail our proposed method of access arbitration. We provide two modes of operation for access arbitration, brute-force and negotiation-based channel selection. Brute-force channel selection is for networks whose nodes can sense the carrier on all supported channels reasonably fast. The negotiation-based channel selection only requires one supported channel to be sensed at a time. The negotiation-based approach can take much longer to arbitrate access but can statistically provide faster arbitration since nodes do not have to wait until they have sensed all channels. We now detail these two approaches.

A. Brute-Force Channel Selection

The brute-force channel selection scheme works as follows. When a node has a packet to transmit to another node, it transmits an RTS packet on the "base" channel. In this RTS packet it sends a proposed list of channels for communication. This list is the set of channels supported by this node that are available for transmission (i.e., sensed as clear through clear channel assessment), sorted in the order of the sensed level of clarity.

Upon receipt of the RTS packet by the intended receiver, a channel is selected from the proposed list from the sender, and that is supported by and sensed as clear at the receiver. If there is no channel matching this criteria then the receiver does not respond with a CTS, otherwise the receiver sends a CTS with the selected channel noted.

When the sender receives the CTS it transmits its data on the channel specified. All other listening nodes mark this channel unavailable for the time specified in the CTS. Once the data is received by the receiving node, this node will return an ACK to the sender.

B. Negotiation-based Channel Selection

For nodes where sensing all the supported channels is not possible in a reasonable amount of time, another approach can be followed. Instead of preparing a list of channels to go in the RTS, the sender can send a single "proposed" channel that has been sensed as clear. If the receiver agrees with this proposal (i.e., the channel is supported and clear at the receiver), it sends a CTS to acknowledge its agreement and data transmission occurs on the selected channel.

However, if the receiver either does not support the channel or it is not free at the receiver, instead of responding with a CTS the receiver responds with the new LRN ("Let's ReNegotiate") packet that signals a re-negotiation of the channel. The receiver selects a channel that it supports and is currently free at the receiver and inserts it in this LRN packet. When the sender receives this LRN packet it determines if the channel selected is supported and if it is free at the sender. If so, it sends a new RTS with this channel, otherwise it selects a different channel and inserts this in the new RTS. When the receiver node receives this new RTS, if it contains the channel it proposed in the LRN and that channel is still free at the receiver it sends a CTS and the data transmission occurs on the selected channel.

If the originally proposed channel is no longer free or the RTS contains a channel other than the one in the LRN, the receiver goes back to renegotiate. (A finite limit must be set on the number of re-negotiations before the two nodes give.)

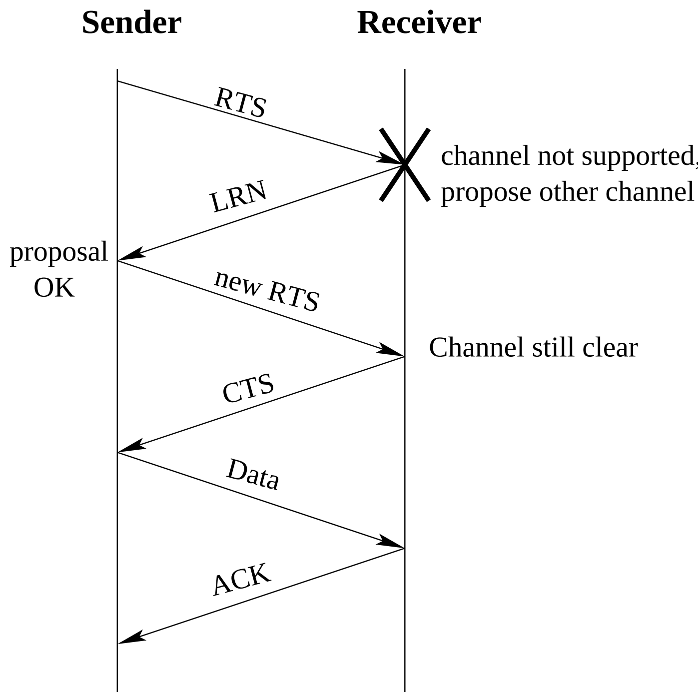

Fig. 2. Illustration of Channel Selection Process

Figure 2 illustrates the channel selection process using the new LRN control packet.

V. Contention-Free MAC Protocol

In a contention-free MAC protocol we have the problem of generating a transmission schedule based on requirements of the nodes in the network. This transmission schedule would be generated by a central node that arbitrates access for all transmissions within its range. The arbitration mechanism can mirror that already proposed for Ethernet Passive Optical Networks [14], where the central node, called the Optical Line Terminal, implements a polling MAC protocol, where nodes, called Optical Network Units would be polled for both their transmission requirements and to grant exclusive access to the shared medium. For our wireless network equivalent, upon polling all nodes in its range to obtain information regarding all requested transmissions, as well as having prior information about each nodes multichannel transceiver capabilities (provided by some means of network registration), the central node, will produce the schedule and send messages to grant each transmitting node a certain time of exclusive access to a specified channel. We will now explore algorithms for the transmission scheduling.

For a homogeneous multichannel network some scheduling algorithms have been proposed [15]. These scheduling algorithms require nodes to support all channels. In this section we propose scheduling algorithms using results from *scheduling theory* [16] that allow for nodes to support different channels.

The multichannel scheduling problem can be formulated using the scheduling notation defined in [16]. In scheduling notation, a scheduling problem is defined by a triple $\alpha|\beta|\gamma$, where α describes the machine environment (e.g., single machine, parallel machines, etc.), β describes the processing characteristics and constraints, and γ describes the objective to be minimized.

Mapping our problem to a problem in scheduling theory, transmissions from nodes correspond to jobs, their transmission slot requirements correspond to processing times, and the channels used for transmission correspond to the machines. In scheduling notation the formulation for our scheduling problem is:

$$P|M_j|\sum_j w_j C_j$$

or

$$P|M_j|C_{max}$$

. In the above model, P refers to the P identical parallel machines (channels) that defines our machine environment. Our only processing characteristic or constraint is the M_j which refers to machine (channel) eligibility constraints. Specifically, M_j is the set of machines (channels) that job (node) j can be executed (transmitted) on. This is required because each node has its own subset of supported channels. If all nodes supported transmission on all channels we could remove the machine eligibility constraint to obtain models $P||\sum_j w_j C_j$ or $P||C_{max}$, where the β part of the scheduling notation triple is omitted since we have no processing constraints. Finally, we have two possible objectives to minimize: 1) the $\sum_j w_j C_j$ is the sum of the weighted completion times of jobs and 2) the C_{max} is the *make-span* of the schedule. The completion time, C_j, is the time at which the transmission for node j is complete. The make-span, C_{max}, is the maximum completion time or the length of the schedule produced. Minimizing the make-span, maximizes the load balancing. Maximizing the load balance would allow the network to more efficiently utilize the transmission resources, this will indirectly lower queueing delays.

If all nodes support transmission on all channels we can remove the machine eligibility constraints and obtain the following models.

$P||C_{max}$ is NP-hard [16], however LPT (longest processing time first) rule provides a good upper bound on performance. It is $4/3 - 1/3m$ competitive with the optimal, where m is the number of machines. For an algorithm to be ρ competitive means in the worst case this algorithm is ρ times worse than optimal.

$P||\sum_j C_j$ is solved to optimality by SPT (shortest processing time first) rule. However, when we add the weights, the problem can only be solved by the heuristic WSPT (weighted shortest processing time first) rule. This heuristic is $1/2(1+\sqrt{2})$ competitive with the optimal.

If we include the machine eligibility constraints, *Least Flexible Job* (LFJ) first scheduling is proven optimal for $P|M_j, p_j = 1|\sum_j C_j$ and $P|M_j, p_j = 1|C_{max}$ if the M_j have a special nesting structure that may not be the case in our network. The special nesting structure between the machine eligibility constraints for 2 nodes holds if one and only one of the following relationships holds for nodes j and k:

- M_j is equal to M_k
- M_j is a subset of M_k
- M_k is a subset of M_j
- M_j and M_k do not overlap

The $p_j = 1$ (processing time for job j equals 1) component means that the slot requirements of all the nodes would have to be equal, or we would have to schedule individual slots separately. This could produce unwanted fragmentation if the individual slots are not scheduled consecutively (this opens the possibility of them being scheduled during the same time but on different channels; this concurrent transmission on multiple channels may not be feasible). If we remove the $p_j = 1$ requirement and/or the nesting structure of M_j, then LFJ is simply a **heuristic** for the problem. We can augment the LFJ heuristic by breaking ties with SPT for minimizing the sum of completion times and with LPT for minimizing the make-span. We refer to these heuristics as LFJ-SPT and LFJ-LPT.

$P|M_j|\sum_j w_j C_j$ can be viewed as a special case of $R||\sum_j w_j C_j$, where R refers to unrelated machines (machines that have differing processing speeds that depend on the individual job). So, processing time p_j is now extended to p_{ij}, since the processing time depends on the job j and the machine i it is executed on. Accordingly, $P|M_j|C_{max}$ can also be viewed as a special case of $R||C_{max}$. For machines that are in M_j we set the execution time multiplier on these machines to 1, for machines not in M_j, we set the execution time multiplier on these machines to ∞.

$R||\sum_j w_j C_j$ is strongly NP-Hard [16] and can be formulated as an integer program solvable by branch-and-price methods (a form of branch-and-bound) [17], [18].

If we do not require priority weighting, we can reduce the above to $R||\sum_j C_j$. This problem can be formulated as an integer program with a special structure that yields an integer solution under LP-relaxation. A common method used to solve this problem is the *weighted bipartite matching*. A weighted bipartite matching problem in which the number of jobs and number of machines is equal is an *assignment problem*. The time complexity of Weighted Bipartite Matching is $O(n(m + nlogn))$. Where in our case, m is the number of channels and n is the number of nodes.

$$\text{minimize } \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{n} k p_{ij} x_{ikj}$$

subject to

$$\sum_{i=1}^{m}\sum_{k=1}^{n} x_{ikj} = 1, \forall_j$$

$$\sum_{j=1}^{n} x_{ikj} \leq 1, \forall_i, \forall_k$$

where k is the scheduling position, p_{ij} is the number of slots to be transmitted for transmission pair j on channel i, x_{ikj} are binary variables representing whether or not position k

TABLE I

TABLE OF SUPPORTED CHANNELS

Node	Supported Channels
1	1, 2
2	1
3	3
4	1, 2, 3

TABLE II

TRANSMISSION SCHEDULE

Tx Node	Rx Node	Number of Slots	p_{ij}
1	2	2	2, ∞, ∞
2	1	1	1, ∞, ∞
2	4	3	3, ∞, ∞
3	4	5	∞, ∞, 5
4	1	2	2, 2, ∞

on machine (channel) i is selected for job (node) j, m is the number of machines (channels) and n is the number of jobs (nodes).

$R||C_{max}$ is NP-complete [16] and can be solved by a few heuristics proposed by Davis and Jaffe [19]. We have already identified a heuristic for minimizing the make-span, namely LFJ-LPT, so we do not pursue this model any further.

If we do not require priority weighting the most promising model is $R||\sum_j C_j$. Minimizing the sum of the completion times helps to minimize the make-span as well. This in turn minimizes the number of wasted slots.

For channels that are supported by both sender and receiver, we set the processing time per slot of this channel (machine, in our scheduling notation) to 1. For channels that are either not supported by the receiver or the transmitter or both we set the processing time per slot of this channel to ∞; this effectively keeps the weighted bipartite matching algorithm from making a match using this arc.

Figure 3 shows the weighted bipartite graph representing the scheduling problem. The scheduling algorithm needs to know the number of slots required for each sender-receiver pair. There is a node on the left-hand side of the weighted bipartite graph for each of these sender-receiver pairs. On the right-hand side of that graph, there is a node for each position on each channel. There are as many positions as there are transmission pairs; this is for the possibility that all transmissions are assigned to a single channel. The nodes are connected by arcs with a certain weight. These weights are determined by whether a channel is supported as well as the number of slots to be assigned multiplied by the position number.

The weighted bipartite matching algorithm then matches all nodes from the left to a unique node on the right that produces the smallest sum of weights.

We now illustrate this scheduling algorithm using a four node three channel network. Table I shows the list of supported channels for each of the four nodes. Table II shows the transmissions to take place in the next scheduling round.

Referring back to figure 3, the selected arcs that produce a match for every node on the left with the minimum total weight are highlighted. Arcs with infinite weight are infeasible and left out of the figure. The weight of each arc is kp_{ij}, where p_{ij} is the number of slots to be transmitted for transmission pair i on channel j multiplied by either 1 or ∞ depending on whether the channel is supported by both nodes in the transmission pair or not. Finally, k is the position of the transmission on the channel with respect to other transmissions scheduled on the channel.

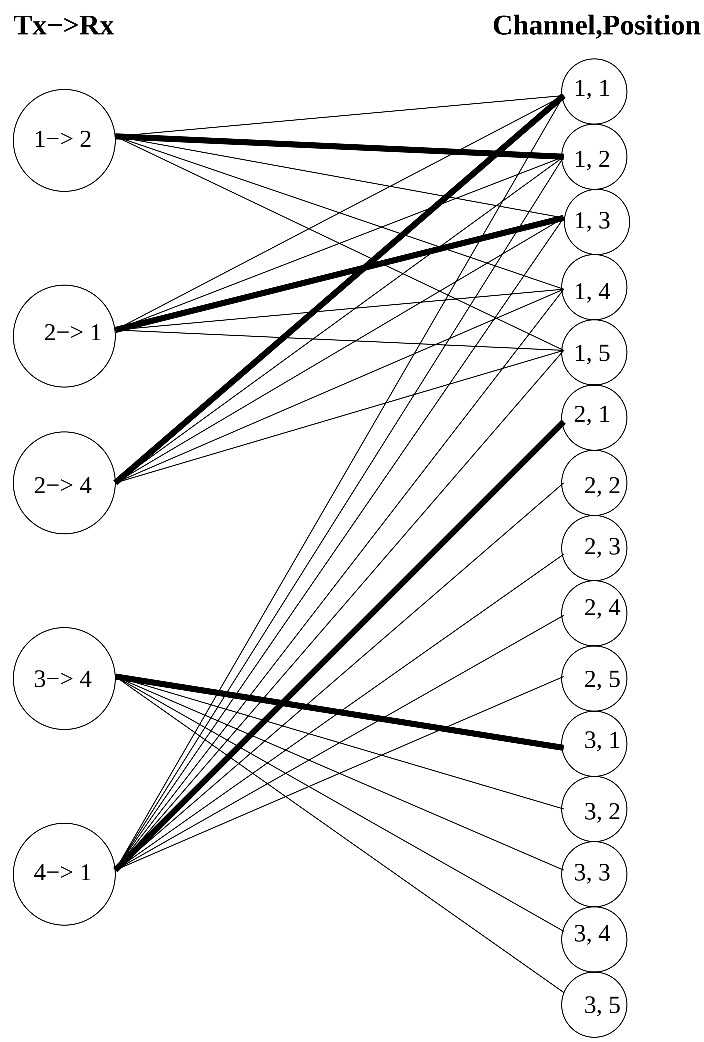

Fig. 3. Illustration of Weighted Bipartite Matching, transmission pairs on left are being matched to scheduling positions on the channels to the right. The matchings produced are indicated by the bold lines.

Figure 4 shows the schedule produced from the weighted bipartite matching.

VI. CONCLUSION

In conclusion, we have proposed both contention and contention-free MAC protocols for heterogeneous multichannel wireless networks. In our contention MAC protocol we have defined a new control packet, LRN, that allows a pair of nodes to re-negotiate channel selection. This relaxes the assumption that nodes can sense the carrier on all supported

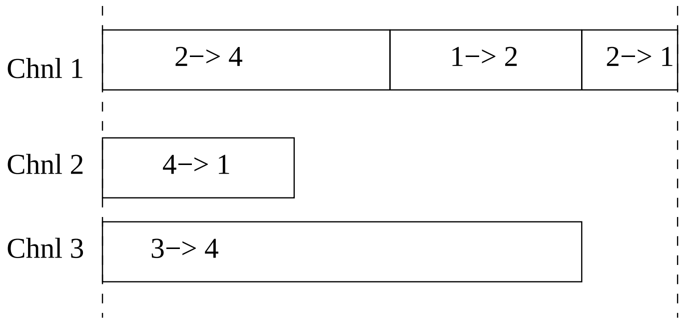

Fig. 4. Schedule Output from Weighted Bipartite Matching

channels in a short period of time. We modeled the scheduling problem for our contention-free MAC protocol as a weighted bipartite matching problem that minimizes the sum of the completion times in polynomial time.

For future research we can extend our protocols to incorporate methods for dealing with the following:

- Add priority weighting into our schedule generation. Priority weights would require the use of branch and price methods to solve.
- Support tunable transceivers. Tuning time can be factored in when deriving schedules.
- Support networks where multichannel hidden terminal is an issue. (We could probably augment our solution with the solution in [7].)
- Allow for asymmetric channel support (i.e., channels are not necessarily supported for both transmission and reception).
- Support nodes that cannot concurrently transmit on multiple channels.
- Update scheduling algorithm to deal with multi-hop networks.
- Investigate more intelligent channel selection schemes for our contention MAC protocol.

References

[1] "Software defined radio forum," http://www.sdrforum.org.

[2] W. H. W. Tuttlebee, "Software-defined radio: Facets of a developing technology," *IEEE Personal Communications*, vol. 6, no. 2, pp. 38–44, April 1999.

[3] ——, "Advances in software-defined radio," *IEE Electronic Systems and Software*, vol. 1, no. 1, pp. 26–31, February 2003.

[4] "DARPA's XG program," http://www.darpa.mil/ato/programs/xg/.

[5] M. A. Marsan and D. Roffinella, "Multichannel local area network protocols," *IEEE Journal on Selected Areas of Communications*, vol. 1, no. 5, pp. 885–897, November 1983.

[6] A. Nasipuri and S. Das, "Chapter 6: Multichannel MAC Protocols for Mobile Ad Hoc Networks," in *Handbook of Algorithms for Wireless Networking and Mobile Computing*, A. Boukerche, Ed. Chapman & Hall/CRC, 2006, pp. 99–122.

[7] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proceedings of MobiHoc*, May 2004, pp. 222–233, roppongi, Japan.

[8] A. Rangnekar, C. Wang, K. Sivalingam, and B. Li, "Chapter 5: Multiple Access Protocols and Scheduling Algorithms for Multiple Channel Wireless Networks," in *Handbook of Algorithms for Wireless Networking and Mobile Computing*, A. Boukerche, Ed. Chapman & Hall/CRC, 2006, pp. 77–98.

[9] N. Choi, Y. Seok, and Y. Choi, "Multi-channel MAC protocol for mobile ad hoc networks," in *Proceedings of IEEE Vehicular Technology Conference*, October 2003, pp. 1379–1382.

[10] A. Nasipuri and S. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *Proceedings of IEEE WCNC*, September 1999, pp. 1402–1406, new Orleans, LA.

[11] N. Jain, A. Nasipuri, and S. Das, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *Proceedings of IEEE International Conference on Computer Communication and Networks*, October 2001, pp. 432–439, phoenix, AZ.

[12] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A new multichannel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2000)*, September 2000, pp. 232–237.

[13] R. Garces and J. Garcia-Luna-Aceves, "Collision avoidance and resolution multiple access for multichannel wireless networks," in *Proceedings of IEEE INFOCOM*, 2000, pp. 595–602.

[14] M. McGarry, M. Maier, and M. Reisslein, "Ethernet PONs: A survey of dynamic bandwidth allocation (DBA) algorithms," *IEEE Communications Magazine*, vol. 42, no. 8, pp. S8–S15, August 2004.

[15] S. Damodaran and K. Sivalingam, "Scheduling algorithms for multiple channel wireless local area networks," *Computer Communications*, vol. 25, no. 14, pp. 1305–1314, 2002.

[16] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 2nd ed. Prentice Hall, 2002.

[17] Z. Chen and W. Powell, "Solving parallel machine scheduling problems by column generation," *INFORMS Journal on Computing*, vol. 11, no. 1, pp. 78–94, Winter 1999.

[18] J. Van Den Akker, J. Hoogeveen, and S. Van De Velde, "Parallel machine scheduling by column generation," *Operations Research*, vol. 47, no. 6, pp. 862–872, Nov/Dec 1999.

[19] E. Davis and J. Jaffe, "Algorithms for scheduling tasks on unrelated processors," *Journal of the Association of Computing Machinery*, vol. 28, pp. 721–736, 1981.