Dynamic Open Spectrum Sharing MAC Protocol for Wireless Ad Hoc Networks

Liangping Ma, Xiaofeng Han, Chien-Chung Shen

Abstract— **In legacy wireless communication systems, spectrum allocation is static, resulting in extremely low spectrum utilization. On the other hand, there is little spectrum left for allocation to an increasing number of emerging wireless applications. A promising approach to solving this paradox is opening up most of the spectrum for unlicensed spectrum users in ways that co-exist with legacy users. Following this direction, this paper proposes the Dynamic Open Spectrum Sharing (DOSS) MAC protocol. This protocol allows nodes to adaptively select an arbitrary spectrum for the incipient communication subject to spectrum availability. It offers real-time dynamic spectrum allocation and high spectrum utilization without relying on any infrastructure. It also coexists with legacy wireless applications, while avoiding the hidden and exposed terminal problems. We conduct theoretical analysis of the protocol, and study its performance via simulations. In addition, we covers the radio receiver design.**

Index Terms— **Medium Access Control (MAC), ad hoc networks, open spectrum, dynamic spectrum allocation, radio receiver.**

I. INTRODUCTION

In legacy wireless communication systems, spectrum allocation is static, and thus could be wasteful. A recent study by the Federal Communications Commission (FCC) shows that even in major urban areas, the allocated spectrum may be under utilized [1]. On the other hand, there is little spectrum left for allocation to an increasing number of emerging wireless applications. One approach to solving this paradox is opening up most of the spectrum for unlicensed spectrum users in ways that co-exist with legacy users. The potential of this approach has been demonstrated by the success of the IEEE 802.11 standard, which opens up from about 75 MHz to about 300 MHz (depending on specific sub-standards and countries) of the spectrum for unlicensed users to share with legacy users. However, what the 802.11 standard does is not enough. First, the spectrum allocation is still based on the traditional notion of fixed channels, which is not as flexible as is required by efficiency. Even if interference is only partially present over a channel, the entire channel is considered unusable, and is thus wasted. Second, the spectrum used for opportunistic sharing is insignificant compared to the entire spectrum that is suitable for wireless communications. With more spectrum opened up for opportunistic sharing, we can take full advantage of the more technically attractive wideband spread spectrum technologies, such as Ultra Wide Band (UWB) and CDMA [2]. In addition, the spectrum application-and-approval process for legacy communication systems is time consuming, while an automated and dynamic approach offers real-time spectrum allocation and access.

In this paper we consider MAC protocols for wireless ad hoc networks in the context of opportunistic open spectrum sharing. Such protocols are important since the network performance of wireless ad hoc networks is severely constrained by limited spectrum availability. Gupta and Kumar [3] have shown that the capacity of each node in a wireless network is on the order of $W/\sqrt{n}$ bits/sec under optimal circumstances, where W is the capacity of the wireless medium in bits/sec and n is the number of nodes. By Shannon's channel capacity formula [4], W is proportional to the bandwidth of the channel B, i.e., $W = B\log_2(1 + \text{SNR})$, where SNR is the signal to noise ratio. Thus, with more open spectrum available to opportunistic sharing, wireless ad hoc networks will perform better and become a more viable technology. Before presenting our proposed MAC protocol, we briefly review some related work in the literature.

Some Medium Access Control (MAC) protocols have been proposed in the literature to improve the performance of wireless ad hoc networks through utilizing more spectrum. Examples include the Slotted Seeded Channel Hopping (SSCH) algorithm [5], where a number of channels are available for use and nodes exchange pseudo-random schedules for accessing the medium in a time-slotted manner, and the Dynamic Channel Assignment (DCA) algorithm [6], where control messages (RTS/CTS) are exchanged over a control channel and data transfer takes place over a number of data channels. These algorithms, however, break up a certain spectrum band into a number of *fixed* channels, which may result in low spectrum

Liangping Ma is with San Diego Research Center, Inc., San Diego, CA 92121 USA (e-mail: lma@sdrcinc.net), Xiaofeng Han and Chien-Chung Shen are with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, (e-mail:{han,cshen}@eecis.udel.edu).

1-4244-0013-9/05/$20.00 ©2005 IEEE

utilization because of the notion of unbreakable channel quantum. To see this, first note that only one of the data channels can be used by a node at any time (assuming that there is only one interface for the data channels) even though there are several contiguous data channels available. Second, a channel is considered busy even if a small fraction of it is being occupied (by legacy spectrum users or hostile interferences). Therefore, for efficient spectrum utilization, nodes shall be flexible in selecting the spectrum so as to take full advantage of all spectrum opportunities. To be specific, the nodes shall be able to (1) use multiple channels simultaneously, (2) use part of a channel, and (3) combine the available parts of adjacent channels into a single wide channel. With these functions, the center frequency and the bandwidth of the RF signal may be different from packet to packet. In fact, they may take arbitrary values within certain ranges. There is also some work on generic spectrum management, such as the Common Spectrum Coordination Channel (CSCC) protocol [7], where a common control channel is set aside for wireless devices to continuously broadcast their spectrum parameters for mutual observability. However, it is based on the notion of a fixed control channel, which may cause the *control channel saturation* problem.

To overcome the shortcomings of the above-mentioned MAC protocols and to take full advantage of the open spectrum, we propose the Dynamic Open Spectrum Sharing (DOSS) MAC protocol for wireless ad hoc networks. This protocol allows nodes to adaptively select an arbitrary spectrum for the incipient communication subject to spectrum availability. It offers real-time dynamic spectrum allocation and high spectrum utilization without the aid of any infrastructure. Aside from fixed spectrum allocation, another inherent problem with wireless ad hoc networks is the hidden and exposed terminal problems, which have been solved nicely with the busy tone approach for the case of a single fixed channel [8]. In this paper, we extend this busy tone approach for the case of open spectrum.

The remainder of the paper is organized as follows. Section II presents the DOSS protocol, Section III conducts theoretical analysis, Section IV provides simulation results, Section V covers the current radio receiver design, Section VI concludes the paper, and the Appendix sketches two of our previous radio designs and explains why we do not adopt them.

II. THE DOSS ALGORITHM

In this section we first give the details of the Dynamic Open Spectrum Sharing (DOSS) protocol, and then discuss its pros and cons.

The DOSS protocol consists of five steps: (1) detection of primary users' presence, (2) set-up of three operational frequency bands/channels: a busy tone band, a control channel, and a data band, (3) spectrum mapping, (4) spectrum negotiation, and (5) data transmission.

A. Detection of Primary Users' Presence

DOSS nodes are only secondary spectrum users, and they can use the spectrum only when the primary licensees are not using it. The critical design issue is the need for a radio that is capable of processing potentially multi-gigahertz-wide bandwidth over the open spectrum and reliably detecting the presence of primary users, which is a challenge to current wireless communication technologies. It is shown, however, that wideband spectrum sensing like this is feasible when using a digital signal processing technique called cyclostationary feature detection [9].

With the wideband spectrum sensing capability, a wireless ad hoc network running the DOSS protocol continuously monitors the open spectrum over the geographic area of deployment for some time, mark the spectra being used by the primary spectrum licensees, and decide on the spectrum that is available for opportunistic sharing.

B. Set-up of Three Operational Frequency Bands/Channels

It is obvious that we need a frequency band[1] for actual data transmissions. Strictly speaking, this data band is not a channel in legacy communication systems, but a range for dynamic channels.

It is undesirable to have a centralized controller for spectrum access in a wireless ad hoc network, which is distributed in nature. However, in the context of the open spectrum, the channel used for a communication is dynamic with arbitrary channel parameters (center frequency, bandwidth, signaling format, etc.), which must be known to the radio receiver. Noting that the design of the radio receiver is much more difficult than that of the radio transmitter, we opt to a simple design approach where the receiver does not need to figure out the channel parameters by itself. This justifies the need for a common control channel, by which each radio receiver knows where to listen to. The drawback of introducing a common control channel is the possibility of the *control channel saturation* problem [5]. However, our radio design experience suggests that designs without a common control channel may not be feasible under current technologies. We have tried two of such designs: one is based on spectrum analysis using Fast Fourier Transform (FFT), and the other is based

---

[1] The concept of frequency band is different from channel. A frequency band may consist of multiple channels. During a communication, either the totality of a channel or nothing of the channel is used, while it is possible only part of a frequency band is used.

on noncoherent modulation/demodulation. For more details, refer to the Appendix.

To alleviate the *control channel saturation* problem, we adopt three techniques:

1. Limit the traffic going through the control channel.
2. Set the bandwidth ratio of the control channel and the data band such that the control channel is not the bottleneck of the network. This ratio can be set just above the ratio of control traffic and data traffic necessary for transmitting a data packet.
3. Allow slow migration of the common control channel based on the traffic load over the current control channel.

To make the initial communications among the nodes possible, we set an initial control channel at the time the network is set up, and this control channel may migrate slowly over time to a better channel that may have a different central frequency and a different bandwidth.

Lastly we want to eliminate the hidden and exposed terminal problems, which are another major cause of the poor performance of wireless ad hoc networks. The use of busy tones has proven to be an effective approach to solving these problems for the single fixed channel case [8]. We extend the concept of busy tones and relate a data channel to a busy tone, leading to the set-up of a third band exclusively for busy tones: the busy tone band.

In summary, we have three channels/bands: a data band, a control channel, and a busy tone band. In what follows we discuss how to relate a data channel to a busy tone.

C. Spectrum Mapping

We establish a one-to-one mapping between the narrow band (low bit rate) busy tones and the wide band (high bit rate) data channels. This way, by observing narrow band receiver-initiated busy tones, a node knows all the data receiving activities in its neighborhood. Denote the busy tone band as $[f_l, f_u]$ and the data band as $[F_l, F_u]$. The mapping is illustrated in Fig. 1. A simple realization of the mapping is through a linear function g

$$g(x) = \frac{1}{F_u - F_l}\left[(f_u - f_l)x + f_l F_u - f_u F_l\right], \quad (1)$$

where $x \in [F_l, F_u]$.

With the spectrum mapping, a receiver only needs to convert the spectrum over which it is receiving to a busy tone and send the busy tone in order to inform other neighbors not to send.

D. Spectrum Negotiation

The next step is spectrum negotiation, by which the sender and the receiver agree on the dynamic channel for the incoming data transmission. The negotiation is done in five steps.

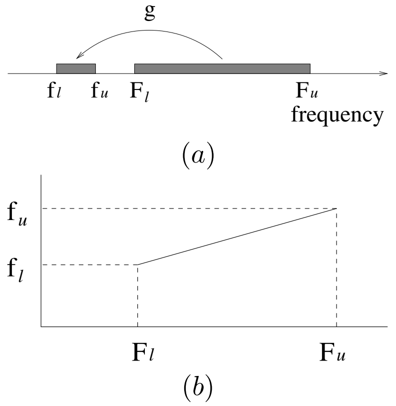

Fig. 1. Spectrum mapping in the *Dynamic Open Spectrum Sharing* (DOSS) protocol.

1. Each node monitors its own available spectrum, which for the sender is the spectrum not being used for receiving in its neighborhood, and for the receiver is the spectrum not being used by itself for receiving.
2. The sender sends a REQ packet (over the control channel) to the intended receiver. A REQ packet contains the channel parameters (center frequencies, bandwidths, etc.) for all available data channels of the sender. To avoid over-fragmentation of the spectrum, those available data channels that are too narrow to be useful are excluded. By listening to busy tones and referring to the spectrum mapping, the sender has full knowledge of the spectra being used for data receiving within its neighborhood, thus being able to avoid disrupting ongoing data receiving activities.
3. The receiver compares the sender's available channels with its own available channels, and picks up an intersection that is available to both. The receiver then replies with an acknowledgement (called REQ_ACK), which contains the channel parameters of the negotiated common channel, over the control channel.

Note: (1) The REQ_ACK is necessary for avoiding ambiguity on the sender side. To know the negotiated common channel, the sender can not solely rely on the reception of the busy tone, since it is possible that another pair of nodes generate an identical busy tone, in which case the sender can not tell whether the received busy tone is destined for it. (2) If there are multiple dynamic channels available, which one does the receiver choose? We argue that it should choose the largest one. This will not make the largest data channel overcrowded, since if the largest one is being used, its busy tone will be heard by other nodes, which will not compete for the largest one but the second largest one.

4. The receiver refers to the spectrum mapping to find and

turn on the corresponding busy tone, telling its neighbors not to send over this data channel.

5. Upon receiving the REQ_ACK, the senders knows the dynamic data channel over which the receiver is waiting for the data packet, and tunes its data transmitter to that channel for data transmission.

Figure 2 shows an example in which node A is the sender and node B is the receiver. Node A has two available channels F_1, F_2. Node B has three available channels F_3, F_4, F_5. Node A sends its channel availability information through a REQ packet to node B. Node B realizes that channel F_6 is common to both, finds f_6 using the spectrum mapping, sends a REQ_ACK packet, and turns on busy tone f_6.

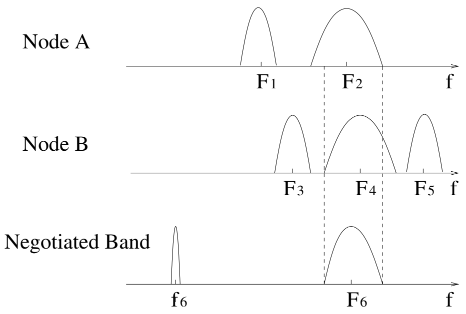

Fig. 2. Spectrum negotiation in the DOSS protocol.

E. Data Transfer

The sender sends a data packet over the negotiated dynamic data channel to the receiver. If the packet is correctly received, the receiver replies with another acknowledgement packet (called DATA_ACK) over the negotiated data channel and turns off the busy tone. Upon receiving the DATA_ACK packet, the sender realizes the transmission is successful. If the sender does not receive the DATA_ACK within a timeout (WF_DATA_ACK timeout), which is longer than the maximum RTT, the sender retransmits the data packet.

Note: The sender can not rely on the turn-off of a busy tone, because it is possible that another pair of nodes turn off an identical busy tone. Thus a DATA_ACK packet is necessary.

F. State Description of DOSS

The DOSS protocol can be formally described by a state diagram shown in Fig. 3, where the notations are:

- IDLE: the state where a node is not currently involved in any MAC layer communications.
- ready: a node receives data from its upper layer protocols.
- REQ: a communication request and channel negotiation packet sent over the control channel by the sender.
- WF_REQ_ACK: wait for the REQ_ACK packet, which contains the channel parameters of the the negotiated data channel.
- WF_DATA: wait for data.
- WF_DATA_ACK: wait for data acknowledgement. If the sender does not receive the DATA_ACK for some time, a WF_DATA_ACK timeout (a duration greater than the maximum RTT), the sender regards the transmission failed and retransmits the data packet.

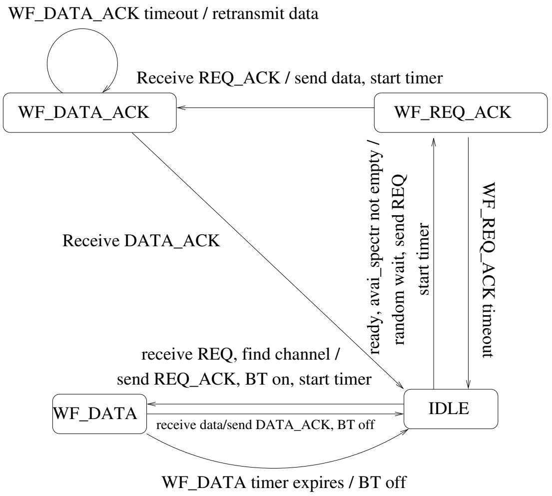

Fig. 3. State diagram for the DOSS algorithm.

G. DOSS and Multicast

The DOSS protocol described above has only concerned about unicast. The support for multicast can be easily added. A primitive design works as follows. In addition to the REQ used for unicast, we now have a new type of REQ created for multicast. To multicast a data packet, the sender declares the spectrum to be used (as a subset of its available spectrum) by transmitting a multicast REQ over the control channel. The neighbors hearing the message adjust their radio receivers to the declared spectrum. Then the sender transmits the data packet over the data channel. Note that no busy tone is used in multicast. Also, it is possible that some neighbors do not receive the REQ for such reasons as collision and thus cannot successfully adjust their radios as intended. This is the tradeoff between reliability and efficiency. This multicast capability naturally supports broadcast since the latter is a special case of the former.

H. The Pros and Cons of the DOSS Protocol

The Pros of the DOSS protocol: (1) it yields real-time and efficient spectrum allocation, (2) it is scalable, (3) it eliminates the hidden and exposed problems. The first pro is evident in the preceding description of the DOSS protocol, and the second one results from the distributed nature of the DOSS protocol. Here we elaborate on the third one by comparing DOSS with the Distributed Coordination Function (DCF) scheme of the IEEE 802.11 standard. IEEE 802.11 uses RTS/CTS to relieve the hidden and exposed terminal problems. However, if the RTS or the CTS packet is not decodable (e.g., the received signal power is just below what is needed for decoding), the RTS/CTS mechanism simply does not work. Alternatively, if we take a different approach by using a busy tone as done in DOSS, only sensing (instead of decoding) is needed and that makes DOSS more robust to signal degradation. We first consider the hidden terminal problem by examining a scenario shown in Fig. 4(a), where node A is responding with a CTS packet after receiving an RTS packet from node B. If node C is located between node A's transmission range (R_t) and sensing range (R_s) [10], node C can sense but can not decode the CTS packet, which includes the information about the duration during which node C should remain silent. As a result, node C may transmit and cause a collision at node A. This example shows that IEEE 802.11 cannot avoid this kind of hidden terminal problems, which, in contrast, can be eliminated in DOSS. To be specific, if node A turns on a busy tone and keeps it on while receiving, node C will be able to sense the busy tone and will not transmit until node A finishes receiving, thus avoiding the collision that would have happened in IEEE 802.11.

Now consider the exposed terminal problem. IEEE 802.11 MAC can alleviate the exposed terminal problem by allowing a node to send if the node hears an RTS but not a CTS. However, if the RTS is not decodable, this mechanism will fail. This is illustrated in Fig. 4(b), where node C senses (but cannot decode) node A's RTS and does not hear node B's CTS. As a result, node C is not sure whether it senses an RTS packet or a CTS packet. Thus, node C has to refrain from transmitting to node D to avoid possible collision at node A. However, in DOSS, when node A is sending to node B, there will be no busy tone that node C can hear. Node C concludes that no one nearby is receiving and thus is free to transmit to node D.

The cons of the DOSS protocol is the need for multiple radio transceivers. When a receiver receives a data packet, it must turn on the busy tone at the same time, Thus, we need at least two radio transceivers. In a simple design, we need three for the operational channels/bands separately. If we allow the transceiver to be more complex, the number of transceivers can be reduced to two: one for the busy tone band, and one shared by the control channel and the data band. The need for multiple transceivers will increase the device cost. However, a modest increase in device cost is well worth the improvement in network performance.

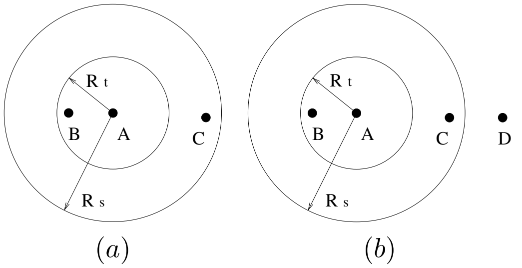

Fig. 4. (a) Hidden terminal with IEEE 802.11: node C may cause a collision at A; and (b) exposed terminal problem with IEEE 802.11: node A may prevent C from transmitting to D.

III. THEORETIC ANALYSIS

In this section, we present analytic results on the DOSS algorithm to gain some insights into it. We first consider the channel capacity of a single wireless link that is achievable by DOSS and compare it with what is achievable by the corresponding static spectrum allocation protocol. Let the spectrum be B Hz wide, a γ fraction of which is taken by primary users and thus not available to DOSS nodes, where $0 \leq \gamma \leq 1$. For the portions of spectrum where there are no primary users, let the signal to noise ratio (SNR) be $\alpha = P_s/P_n$, where P_s is the signal power and P_n is the noise power. For analytic simplicity, we assume that the power spectrum density of the primary users equals that of the signal transmitted by a wireless node, and this gives the power of the primary users γP_s.

For DOSS nodes, only the portions of the spectrum that are free of primary users are chosen for the communications. These portions sum up to $(1-\gamma)B$ Hz. Thus its achievable channel capacity per unit time (in bits/sec) [4] is

$$C_d = B(1-\gamma)\log_2(1+\alpha). \tag{2}$$

For the corresponding static spectrum allocation protocol, a node may take two schemes accessing the spectrum. In the first scheme, the node senses the presence of the primary users and marks the entire channel B as "unusable" and does not access the spectrum. The achievable channel capacity then amounts to zero, i.e., $C_s = 0$. Since $C_d \geq 0$, DOSS always outperforms this scheme. In the second scheme, the node accesses the entire spectrum B anyway, regardless of the presence of the interference from the primary users. This is the best scheme, and its achievable channel capacity is

$$C_s = B\log_2(1+\frac{P_s}{P_n+\gamma P_s}) = B\log_2(1+\frac{\alpha}{1+\gamma\alpha}). \quad (3)$$

To compare the second scheme with DOSS, we consider the capacity gain

$$g = C_d - C_s, \quad (4)$$

which is plotted in Fig. 5 as a function of α and γ. It is seen that the gain is positive except for the extreme cases where γ is close to 1 (too much spectrum is used by the primary users) and α is close to 0 (the SNR is too low). Furthermore, the higher the SNR α, the greater the gain. The reason for the gain is that if a node uses the entire spectrum B, the overall SNR over the entire spectrum B could be very low, leading to low achievable channel capacity. In contrast, DOSS only uses the portions of spectrum that are free of primary users, therefore resulting in a much higher SNR and hence higher capacity despite using less spectrum.

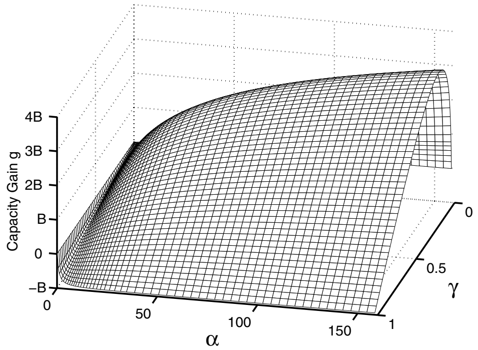

Fig. 5. Gain in achievable channel capacity of DOSS over the best static spectrum allocation scheme.

Next we analyze DOSS over a fully connected wireless network. By fully connected, each node can hear the transmission of any other node in the network. This scenario allows for a tractable analysis and is yet enough to provide insights. We focus on the throughput and the interaction between the control channel and the data channel. We use the following notations:

- r_c: control channel raw data rate,
- r_d: data channel raw data rate,
- t_p: PHY header transmission time,
- $t_r = l_r/r_c + t_p$: transmission time of the REQ packet, where l_r is the length of the REQ packet (at the MAC layer),
- $t_d = l_d/r_d + t_p$: transmission time of the data packet (assuming all data packets have the same length), where l_d is the length of the data packet,
- τ: maximum one-way propagation delay,
- $t_1 = l_1/r_c + t_p$: transmission time of the first ACK, or REQ_ACK, where l_1 is the length of the REQ_ACK packet,
- $t_2 = l_2/r_d + t_p$: transmission time of the second ACK, or DATA_ACK, where l_2 is the length of the DATA_ACK.

Notes: (1) To facilitate synchronization and to increase reliability, the PHY headers of all packets are sent at the fixed rate r_c. (2) The propagation time τ takes into account the time spent detecting an arriving signal.

To reduce collision without incurring significant communication overhead, we use an adapted non-persistent CSMA[11] in the control channel. To be specific,

1. If a node has a packet to send, and its available spectrum is sufficient for an efficient transmission, and it senses the channel idle, then it sends the packet.
2. If the channel is sensed busy, it randomly backs off and will re-sense the channel when the backoff expires.
3. If there is a collision (detected by not receiving the REQ_ACK within a timeout), a node repeats the process in steps 1 and 2.

This non-persistent CSMA variant does not use such control packets as RTS/CTS to reserve the channel, which incurs significant communication overhead for the small REQ/REQ_ACK packets. Instead, it utilizes the lightweight random backoff mechanism, which also reduces the probability of further collision in the event there is a collision.

We assume the attempted channel traffic over the control channel is Poisson with mean arrival rate λ packets/second. Note that this traffic takes into account both the actual transmissions, and the attempted transmissions (i.e., sensing) that do not result in actual transmissions. The attempted channel traffic over the control channel constitutes a renewal process, which includes the Poisson process as a special case. By the renewal theory[11], the PHY layer throughput is

$$S = \frac{\bar{U}}{\bar{B}+\bar{I}}, \quad (5)$$

where $\bar{U}$ is the average duration of a successful transmission, $\bar{B}$ is the average duration of any transmission (whether successful or not), $\bar{I}$ is the average duration of any idle period, and $\bar{B}+\bar{I}$ is the average duration of a renewal interval.

The probability that a single REQ packet is successful is $e^{-\lambda\tau}$, and the probability that a REQ_ACK packet is successful is also $e^{-\lambda\tau}$. Since a request is successful only if both the REQ and the corresponding REQ_ACK are successful. Thus, the probability that a spectrum negotiation is successful is $p_s = e^{-2\lambda\tau}$. Since $\bar{U} = t_r p_s$, we have

$$\bar{U} = t_r e^{-2\lambda\tau}. \tag{6}$$

For analytic simplicity, we assume that $t_r = t_1$. Let the time when a packet is sent be t. Then between time t and $t+\tau$, any attempted transmission will result in an actual transmission and hence a collision. Let $t+Y$ be the time when the last packet arriving between t and $t+\tau$. Then,

$$\bar{B} = t_r + \bar{Y} + \tau. \tag{7}$$

It can be shown that[11] $\bar{Y} = \tau - \frac{1}{\lambda}(1 - e^{-\tau\lambda})$.

The average time between attempted transmissions is $1/\lambda$. But for any successful REQ – REQ_ACK pair, the control channel will be idle for $t_d + 2\tau + t_2$. This effectively extends the average idle time, and thus

$$\bar{I} = \frac{1}{\lambda} + p_s(t_d + 2\tau + t_2) \tag{8}$$

Let S_r denote the throughput of the successful REQs whose corresponding REQ_ACKs are also successful. Substituting (6), (7) and (8) into (5) and replacing S with S_r, we obtain

$$S_r = \frac{\lambda t_r e^{-2\lambda\tau}}{\lambda(t_r + 2\tau + (t_d + 2\tau + t_2)e^{-2\lambda\tau}) + e^{-\lambda\tau}} \tag{9}$$

Note that the throughput in (9) is normalized. The number of successful spectrum negotiations per second is

$$S_r' = \frac{r_c S_r}{l_r + r_c t_p}, \tag{10}$$

where the denominator is the REQ packet length at the PHY layer.

If we ignore such failures at the PHY layer as in synchronization and checksum, the data transmission will be successful as long as the spectrum negotiation is successful. Thus the data throughput (over the data channel) in bits per second is

$$\begin{aligned} S_d' &= S_r' l_d' \\ &= \frac{r_c l_d'}{l_r + r_c t_p} \frac{\lambda t_r e^{-2\lambda\tau}}{\lambda(t_r + 2\tau + (t_d + 2\tau + t_2)e^{-2\lambda\tau}) + e^{-\lambda\tau}} \end{aligned} \tag{11}$$

where l_d' is the packet size of a data packet. If l_d' is measured at the MAC layer, i.e., $l_d' = l_d$, then S_d' is the MAC layer throughput. If it is measured at the application layer, then S_d' is the application layer throughput.

Now we validate (11) via QualNet simulations. The bandwidth of the control channel is 2 MHz, and the raw data rate $r_c = 1$ Mbps. The data band is 10 MHz, and the the raw data rate is 5 Mbps. However, $1/3$ of the data band is taken by primary users. Invoking the assumption that the data rate is proportional to the bandwidth, the maximum raw data rate over the data channel is $r_d = 3.33$ Mbps. The PHY header transmission time $t_p = 0.192$ ms. The MAC payload of the REQ, REQ_ACK and DATA_ACK packets are 80 bytes, 80 bytes, 40 bytes, respectively. The maximum propagation delay is 2μs. In the simulation, 50 nodes send Poisson UDP traffic to a common receiver. Figure 6 shows that the theoretical analysis agrees with the simulation result very well. The minor discrepancy is probably because the attempted traffic process is not exactly Poission.

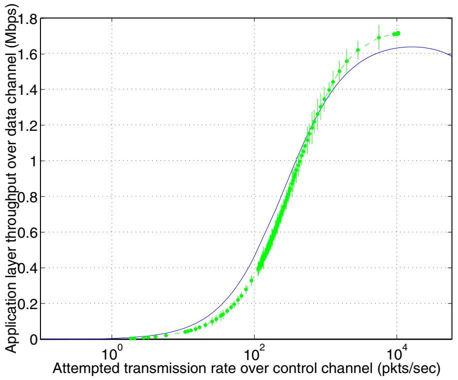

Fig. 6. Application layer data throughput over the data channel vs. attempted transmissions over the control channel. The blue solid line is the theoretic analysis, and the green dashed line is the simulation result, where the vertical hashes represent 95% confidence intervals.

Having validated (11), we now use it to study the interaction between the data channel and the control channel. Recall the definitions of t_r, t_1, t_d, t_2. The data throughput S_d' is not only dependent on the raw data rate over the data channel r_d but also on the raw data rate over the control channel r_c. Figure 7 shows that given the attempted transmission rate $\lambda = 4000$pkts/sec over the control channel, the raw data rate over the data channel r_d becomes the bottleneck of the system for r_c greater than 2 Mbps.

IV. SIMULATION RESULTS AND DISCUSSIONS

Operating only over a pre-assigned narrow band is not the primary motivation of DOSS. We want DOSS to take advantage of the open spectrum, acquiring as much bandwidth as possible as long as it does not disrupt primary spectrum licensees. Thus, in this section we consider DOSS for a multi-hop wireless ad hoc network over the

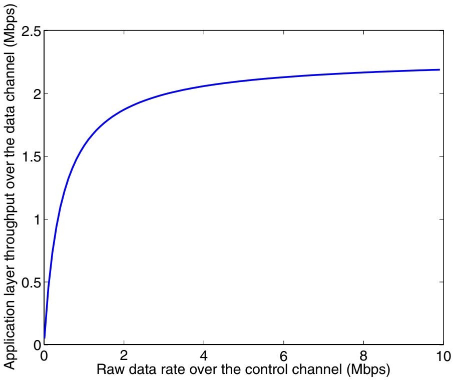

Fig. 7. Application layer data throughput S'_d over the data channel vs. raw data rate over the control channel r_c.

open spectrum. We consider a network of 10 nodes evenly located in a straight line such that a node can only communicate with its immediate neighbors. One of the end nodes send Poisson UDP traffic across the line to the other end. The open spectrum is 110 MHz wide, and the use of the open spectrum is shown in Fig. 8. In the simulation, $r_c = 5.5$ Mbps, $r_d = 29.7$ Mbps, $t_p = 0.192$ ms, and the data packet size is 1000 bytes. The simulation results are shown in Fig. 9.

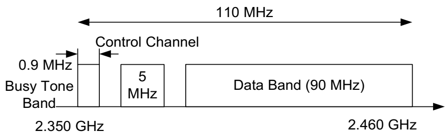

Fig. 8. DOSS finds 110 MHz open spectrum available.

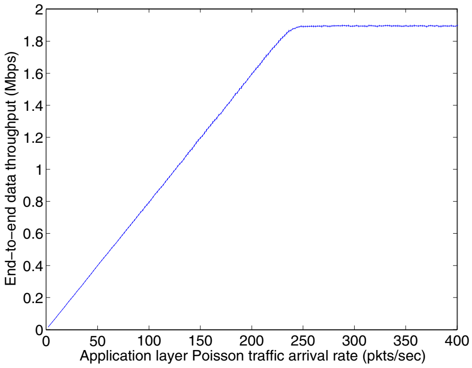

Fig. 9. End-to-end data throughput changes as the transmission rate of the Poisson UDP traffic increases, where the vertical hashes represent 95% confidence intervals (very small).

V. RADIO RECEIVER DESIGN FOR DOSS

The motivation of a simple radio receiver for DOSS is already discussed in Section II-B. This receiver gets the channel parameters of the dynamic data channel over a common control channel. Coherent modulation and demodulation schemes are used. The design is thus along the traditional radio design path except that (1) the receiver is automatically tunable and (2) channel switching is done quickly. Requirement (1) is justified for two reasons: first, the channels used for transmitting the data packets are not fixed and possibly different from packet to packet, and second, the control channel may migrate slowly over time. Requirement (2) provides better time utilization of the data channel. Since the data channel to be agreed upon may be different from packet to packet, there is a need for the sender and the receiver to switch the channel from time to time. During the channel switching, both the sender and the receiver are idle. Therefore the channel switching should be finished quickly.

VI. CONCLUSIONS

We propose a MAC protocol, DOSS, for wireless ad hoc networks operating over the open spectrum. DOSS offers real-time and dynamic spectrum allocation, coexists with legacy wireless communication systems, and eliminates the hidden and exposed terminal problems. We conduct a theoretic analysis of DOSS and validate it, and study the performance of DOSS through simulations.

Acknowledgements: The authors would like to thank Eric Blossom of Blossom Research, Prof. Leonard J. Cimini of the University of Delaware, and Dr. Bo Ryu and Dr. Zhuochuan Huang of San Diego Research Center for their insightful suggestions and comments.

APPENDIX

In addition to the radio design outlined in Section V, we have tried two other designs with the motivation that a receiver can detect the carrier frequency and the bandwidth of the incoming signal on the fly so as to eliminate the need for exchanging control messages for disseminating data channel information. These designs cannot be easily supported by current technologies. Nevertheless, from a theoretical perspective, they are feasible. Thus we describe them here.

A. *FFT Based Radio Design*

The idea behind this design is to down convert a high frequency signal such that a Fast Fourier Transform (FFT) based spectrum analyzer can detect the carrier frequency and bandwidth of the resulting intermediate frequency (IF) signal. The detected RF information is then used to coherently demodulate the IF signal.

In the following we show the entire process from the binary data generating at the transmitter to the binary data decoding at the receiver. The original binary bits, as shown in Fig. 11(a), are first feeded into a BPSK modulator at 2 Mbps. To limit the bandwidth of the BPSK signal, the output of the modulator is shaped by a root Nyquist filter with rolloff factor 0.5 [12], and multiplied by a sinusoidal carrier signal at f_c' (MHz) and then transmitted. In Fig. 11, (b) shows the BPSK modulated signal waveform, (c) shows the pulse-shaped waveform, and (d) shows the transmitted RF signal. The receiver does not know the exact value of f_c' but knows it is within a range $[f_c, f_c + \Delta f_c]$. In the simulation, $f_c = 24$ MHz, $\Delta = 0.3$, and $f_c' = (1+\delta)f_c = (1+0.2)f_c = 28.8$MHz. The transmitted signal is attenuated and corrupted by the Gaussian noise in the channel when it arrives at the receiver, as shown in Fig. 11(e), where the attenuation factor is 1×10^6 and the signal-to-noise (SNR) ratio is 3 dB.

The block diagram of the receiver is illustrated in Fig. 10. The filter characteristic of the RF amplifier is shown in Fig. 11(f), and the output is shown in (g). Synchronization, including packet synchronization and symbol synchronization [4][13], is then performed. The signal is next multiplied by a local oscillator at frequency f_c. The resulting signal in Fig. 11(h) consists of two frequency components centered at $(2+\delta)f_c$ and δf_c, and is passed to a lowpass filter (LPF) shown in Figure 11(i) to remove the high-frequency component, while preserving the low-frequency component for inexpensive Analog-to-Digital (A/D) conversion. Figure 11(i) and (j) show the output of the lowpass filter and the A/D output, respectively. Current technologies can easily support precise A/D conversions up to a frequency of 250MHz [14, p. 101].

Now we come to the spectrum analysis part. Figure 11(l) shows the spectrum of the digitized discrete-time signal passed to the FFT device, and the center of the spectrum provides an estimate, which is 4.83MHz, of the carrier frequency difference δf_c or $0.2 f_c = 4.80$MHz. A sinusoidal signal at 4.83 MHz is generated to multiply the digital signal, and the product is input to a lowpass filter shown in Fig. 11(m), to obtain a baseband signal, which is shown in Fig. 11(n). The baseband signal is then sampled at the rate 2MHz to generate an estimate of the BPSK modulated signal, as is shown in Fig. 11(o). The estimate is then thresholded, producing the digital output shown in Fig. 11(p), which is identical to the original digital signal.

Note that the delay unit D in Fig. 10 means temporary storage of the digitized signal. As Δf_c increases, both the memory requirement and the computational complexity at the spectrum analyzer increase quickly, which could be easily beyond the capability of current affordable technologies.

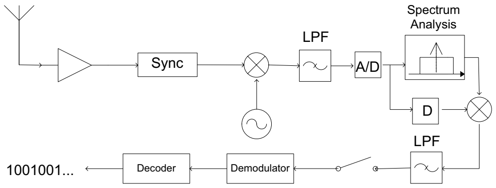

Fig. 10. The block diagram of the FFT-based receiver.

B. Noncoherent Modulation/Demodulation Based Radio Design

To take advantage of coherent modulation/demodulation while eliminating the need for additional control packets, we use in-band non-coherent signaling at the beginning of a packet, while coherent modulation/demodulation for the rest of the packet. A simple non-coherent modulation/demodulation [15, p. 264] scheme is the On-Off Keying (OOK), which switches a carrier sinusoid on for a binary 1 (or 0) and off for a binary 0 (or 1) [15, p. 332]. The information on the carrier frequency and bandwidth could be OOK encoded at the PHY layer header.

The receiver is shown in Fig. 12. The arriving signal first enters an RF amplifier, a coarse dual-band Bandpss filter (BPF) that allows only the desired bands (i.e., $[f_l, f_u]$ and $[F_l, F_u]$) to pass. If the receiver is not expecting a data packet, the output $R(t) = S(t) + N(t)$ enters a noncoherent detector, where $S(t)$ is the signal and $N(t)$ is the noise. In the noncoherent detector, $R(t)$ first enters an envelope detector that outputs $d(t)$, which is then sampled and thresholded to generate a binary output containing the RF parameters of the arriving signal. These parameters are used to tune a BPF, which accurately filters the rest of the packet, and to tune an oscillator, which generates a frequency-coherent sine wave. On the other hand, if the receiver is expecting a data packet, implying the RF parameters are already known, $R(t)$ enters a coherent detector. Besides, a channel scanning device monitors the data band and the control band, and updates an *RF Expected Infomation Base*, which tunes a BPF and an oscillator. In either case, the output of the multiplier is passed onto the next stage for further processing, which includes filtering, sampling, and channel decoding. Figure 13 shows the signals involved in the noncoherent detection of a binary signal (10001001), where the symbol duration is $1\mu s$, the time constant [15, p. 262] of the envelope detector is $0.1\mu s$, the carrier frequency is 24MHz[2]. Figure 14 shows how the bit error rate (BER) changes with the SNR at a noncoherent detector, where the parameters are the same as those in Fig. 13 except that the carrier frequency is 2.4 GHz.

The problem of this design is that a receiver needs to listen to a potentially very large chunk of spectrum, but the total noise power increases linearly with the bandwidth. As a result, the total noise power could be very large, making the SNR in Fig. 13 very low. A filter bank could be utilized to process smaller sub-bands separately, thus alleviating the SNR problem. But then, the computational cost could be forbidding.

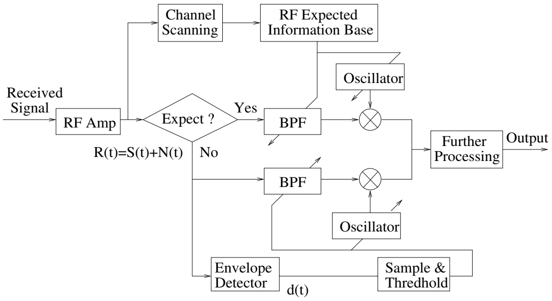

Fig. 12. A non-coherent detection based radio receiver design.

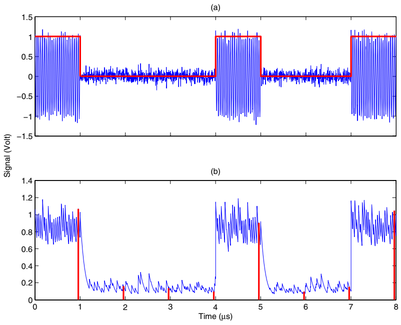

Fig. 13. Noncoherent detection of a binary signal 10001001 (red line in (a)) in DOSS: (a) the output signal from the RF amplifier $R(t)$ (blue line), and (b) the detected envelope $d(t)$ (blue line) and the samples for thresholding(red hashes).

REFERENCES

[1] FCC Spectrum Policy Task Force, "Report of the spectrum efficiency group," Nov., 2002.

[2] R. J. Berger, "Open spectrum: A path to ubiquitous connectivity," *Queue*, vol. 1, pp. 60–68, May 2003.

[3] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. IT-46, pp. 388–404, March 2000.

[4] J. G. Proakis, *Digital Communications*. McGraw-Hill, Inc., 3rd ed., 1995.

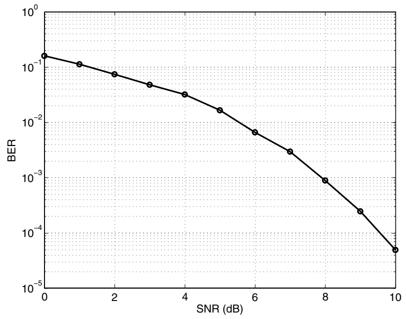

Fig. 14. The bit error rate (BER) vs. the signal-to-noise ratio (SNR), where the signal is $S(t)$ and the noise is $N(t)$ as shown in Fig. 12, for a DOSS noncoherent detector.

[5] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *MobiCom'04*, (Philadelphia, PA), Sept. 2004.

[6] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Int'l Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN)*, 2000.

[7] D. Raychaudhuri and X. Jing, "A spectrum etiquette protocol for efficient coordination of radio devices in unlicensed bands," in *Proceedings of the 14th IEEE International Symposium on Personal Indoor, Mobile Radio Communications*, (Beijing, China), pp. 172–176, September 2003.

[8] Z. Hass and J. Deng, "Dual Busy Tone Multiple Access (DBTMA) - a multiple access control scheme for ad hoc networks,," *IEEE Trans. Communications*, vol. 50, pp. 975–985, June 2002.

[9] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), 2004.

[10] J. Schiller, *Mobile Communications*. Harlow, England: Addison-Wesley, 2000.

[11] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I – Carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.,*, vol. COM-23, pp. 1400–1416, Dec. 1975.

[12] S. Sampei, *Applications of Digital Wireless Technologies to Global Wireless Communications*. Upper Saddle River, NJ: Prentice Hall, 1997.

[13] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*. Indianapolis, Indiana: SAMS Publishing, 2002.

[14] W. Tuttlebee, ed., *Software Defined Radio: Enabling Technologies*. West Sussex, England: John Wiley & Sons, Ltd, 2002.

[15] L. W. Couch II, *Digital and Analog Communication Systems*. Prentice Hall, Inc., 5th ed., 1997.

[16] The Institute of Electrical and Electronics Engineers, *Wireless LAN Medium Access control (MAC) and Physical Layer (PHY) Specifications*, 1999. ANSI/IEEE Std 802.11.

---

[2] We do not use 2.4 GHz only for the purpose of presentation clarity.

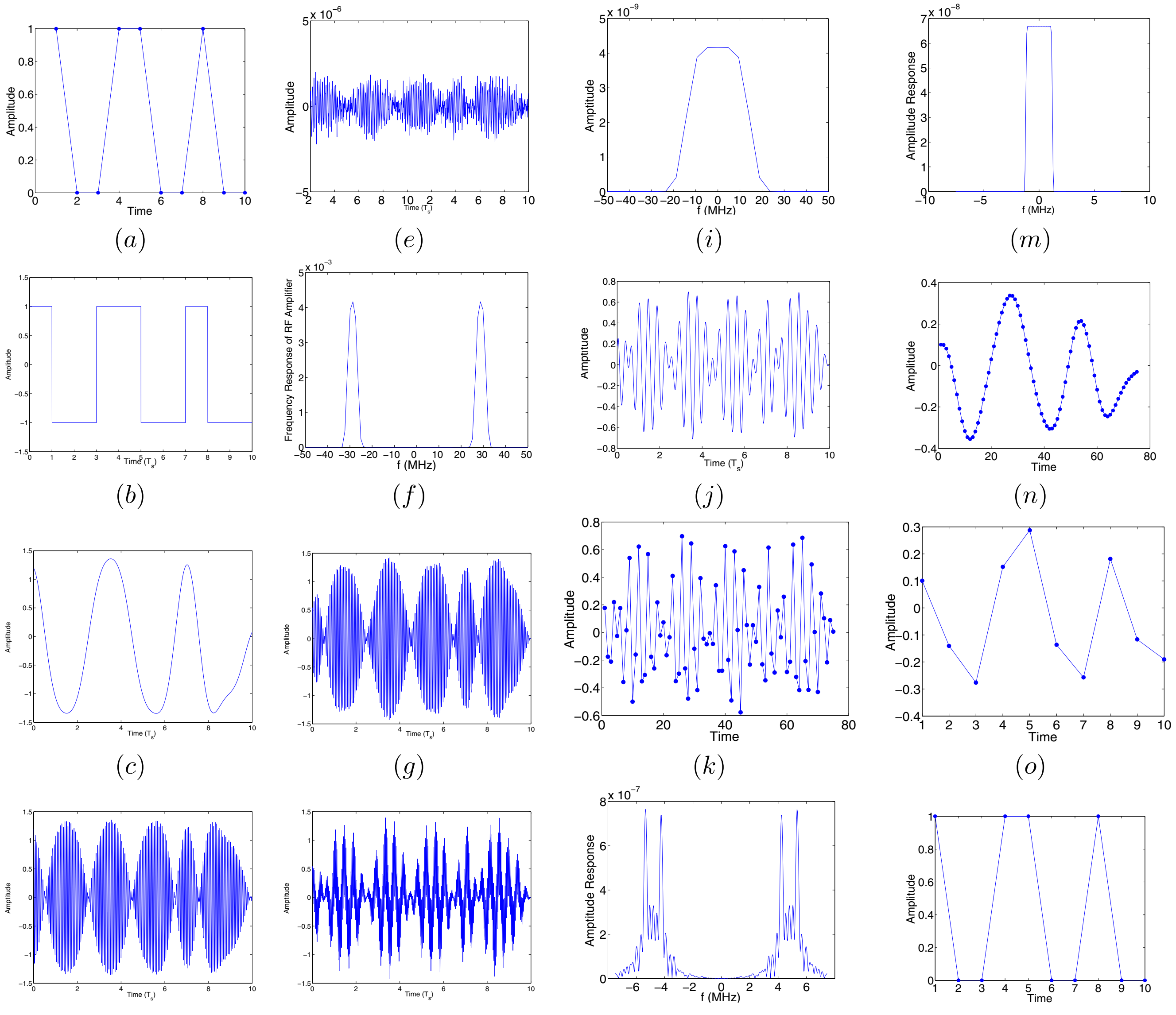

Fig. 11. Signals at various stages for an FFT-based radio receiver design.