

Implementation of a Distributed Joint Routing and Dynamic Spectrum Allocation Algorithm on USRP2 Radios

Pradeep B. Nagaraju*, Lei Ding*, Tommaso Melodia*, Stella N. Batalama*, Dimitris A. Pados* and John D. Matyjas†

*Department of Electrical Engineering, The State University of New York at Buffalo, Buffalo, NY 14260, USA

†Air Force Research Laboratory, Rome, NY 13441, USA

Abstract—A cognitive radio network with decentralized control (i.e., a cognitive ad hoc network) is considered in this demonstration. The demo implements a decentralized and localized algorithm for throughput maximization through joint routing and interference-avoiding waveform selection. The algorithm adapts to time-varying traffic demands, interference profile, and network topology to locally maximize the achievable data rate while avoiding harmful interference to co-located primary or secondary users. The prototype is based on a cross-layer protocol stack implemented in Python, which leverages GNU Radio for adaptive signal generation on a USRP2 software-defined-radio platform.

Index Terms—Cognitive ad hoc networks, dynamic spectrum access, power allocation, software defined radio.

I. INTRODUCTION

This abstract describes our initial efforts in implementing a decentralized and localized algorithm for throughput maximization through joint routing and interference-avoiding waveform selection on a software defined radio (SDR). Our prototype is based on GNU Radio [1] and USRP2 (universal software radio peripheral version 2) [4]¹.

In [3], we developed a framework based on nonlinear optimization for joint routing and dynamic spectrum allocation for cognitive ad hoc networks with decentralized control. Based on this framework, a localized algorithm was designed to jointly select the next hop and spectrum opportunistically depending on queueing and spectrum dynamics with the objective of maximizing the network throughput (see [3] for details). A *common control channel* (CCC) is used by all secondary users for spectrum access negotiation, and is assumed to be time slotted. A *data channel* (DC) is used for data communication. All secondary users exchange local information on the common control channel.

For this demo, we specialized the framework in [3] to the case of joint routing, channel, and transmission rate selection based on an adaptive modulation scheme. Every backlogged node i , once it senses an idle common control channel, performs the following main steps:

- 1) For each neighboring candidate next hop, node i estimates the maximum achievable link capacity by jointly selecting the transmission channel and power;

- 2) For each neighboring candidate next hop, node i calculates the so-called *spectrum utility factor*, defined as the product between the differential backlog of the maximum-differential-backlog session, and the maximum achievable link capacity as estimated in step 1.
- 3) The session corresponding to the optimal spectrum utility factor is scheduled for transmission with corresponding next hop, optimized spectrum and power allocation.
- 4) The signal waveform is selected based on the estimated SINR of the link. In this implementation, we consider BPSK and QPSK as the waveform set. The transmitter compares the expected SINR with a set of pre-defined thresholds to choose the transmission waveform.

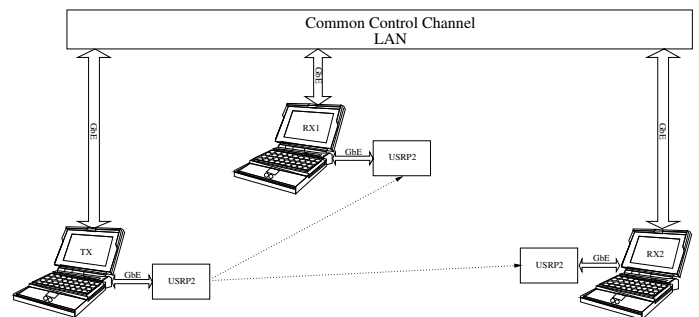


Fig. 1. Experimental Setup.

Once waveform selection, session to be scheduled, and next hop have been determined, node i negotiates channel access with the desired next hop over the CCC. After successful negotiation, the SDR at transmitter and receiver are configured to the selected channel on the DC to complete the data transmission.

II. DEMO TESTBED SETUP

In our test-bed setup, illustrated in Fig. 1, each host node is based on an Intel core 2 duo 3.0 GHz processor and 4 GByte RAM with two Gigabit Ethernet network interface cards (NIC). All systems run an Ubuntu Karmic Koala 9.10 operating system. The CCC is currently implemented through the Gigabit Ethernet backbone. We use one NIC to connect the

¹This material is based upon work supported by the US Air Force Research Laboratory under Award No. 45790. Approved for Public Release; Distribution Unlimited: 88ABW-2010-3085 dated 08 June 2010.

host to the USRP2 SDR, and another NIC to interface the host with a US-Robotics, *USR997932A* 16-port gigabit switch, to implement the wired CCC.

The current setup consists of three nodes. One of the three nodes acts as a transmitter (TX), and the remaining two are receivers (RX1 and RX2). The interference at the RX1/RX2 and primary user power level information are simulated for each of the five considered center frequencies (2.4, 2.408, 2.410, 2.415 and 2.420 GHz) with 20 MHz bandwidth and 200 OFDM subcarriers. The control parameters, i.e., the best transmission channel, transmission power, modulation scheme, and next hop are chosen based on the spectrum utility factor as discussed in Section I. These decision variables are communicated between TX and RX1/RX2 through an RTS/CTS/DTS [3] exchange over the CCC. We implemented the logic in control of the CCC and DC in Python on different threads of the host such that the CCC and DC can run in parallel and independently of each other. We have also developed a graphical user interface based on PyQt4 [2] at the transmitter and at the receivers. The interface graphically shows details of the underlying dynamic optimization of the communication process for the distributed cognitive ad-hoc network.

III. NODE ARCHITECTURE

The architecture of each node in the network consists of three planes, namely *observation* plane, *control* plane and *data* plane, as shown in Fig. 2. The interaction of the three planes allows allocating the best channel and power for the link with the maximum capacity and session backlog while adaptively regulating the modulation scheme.

The observation plane gathers spectrum information through a so-called spectrum observer² and CCC. The CCC gathers spectral information about frequency, power and interference from neighboring devices. This information is relayed to the control plane.

The control plane executes the algorithm for joint routing and spectrum allocation. As discussed in Section I, the algorithm determines the optimal data channel parameters, i.e., frequency and power to be used given the varying channel conditions as perceived by the observation plane. The algorithm is based on solving a sequence of simple optimization problems [3]. We are currently working towards our objective of integrating in the prototype a real-time optimizer to decrease the execution time to the order of a few micro-seconds [6]. The *network controller* (NC) exchanges information regarding next hop selection, best channel and power allocation, with the protocol stack on the data plane. The NC is also responsible for reconfiguring the SDR's central frequency and power for transmission and reception.

The data plane implements the cross-layer protocol stack, including network and link layers. This plane also manages different traffic sessions. The network layer uses the next hop selection parameter from the control plane for constructing and forwarding the data packet to the selected next hop. The link layer selects the waveform modulation scheme based on

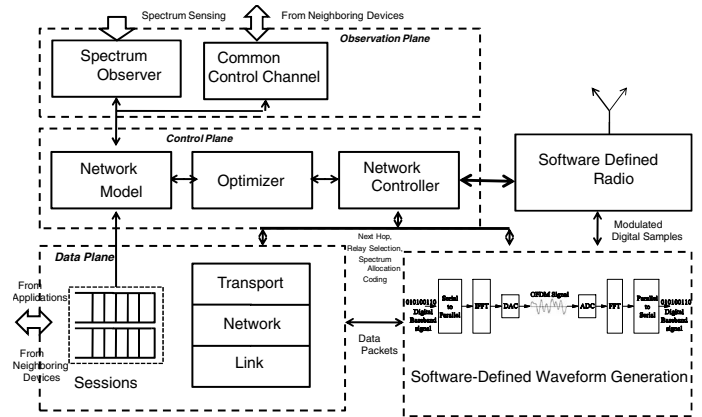


Fig. 2. Node Architecture.

the SINR of the chosen channel that is obtained from the control plane. Baseband digital data is modulated based on an orthogonal frequency division multiplexing (OFDM) scheme. The link layer adaptively selects the modulation scheme of the OFDM signal on the PHY layer based on the estimated channel SINR. OFDM subcarriers are modulated with either BPSK or QPSK based on the SINR of the chosen link. In our current implementation, the SINR on each channel is estimated at the transmitter. If the SINR of the chosen link is between 6dB and 9dB, BPSK is selected, while if it is above 9dB QPSK is selected.

Usually, the baseband digital data is serially arranged in queues. The serial digital data is converted to parallel to map the baseband digital data on each of the OFDM carrier. An IFFT on the parallel data gives the OFDM subcarrier symbols. These OFDM symbols are then converted into analog form through a digital-to-analog converter (DAC). The reverse occurs at the receiver to obtain the baseband digital data. There are two key techniques to implement OFDM on SDR platforms, i.e., maximum likelihood (ML) [5] and pseudo noise (PN) sequence correlation [7]. Based on the current GNU Radio library, we use the PN correlation technique to determine the OFDM signal.

REFERENCES

- [1] GNU Radio-The GNU Software Radio. <http://www.gnuradio.org>.
- [2] Python binding for Nokia Qt. <http://www.riverbankcomputing.co.uk>.
- [3] L. Ding, T. Melodia, S. N. Batalama, J. D. Matyjas, and M. J. Medley. Cross-Layer Routing and Dynamic Spectrum Allocation in Cognitive Radio Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 59(4):1969–1979, May 2010.
- [4] M. Ettus. Building Software Defined Radios: the USRP product family. *Product Brochure*, July 2009.
- [5] P. O. B. J. van de Beek, M. Sandell. ML Estimation of Time and Frequency Offset in OFDM Systems. *IEEE Trans. On Signal Processing*, 45(7):1800–1805, July 2010.
- [6] J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, May 2010.
- [7] T. M. Schmidl and D. C. Cox. Robust Frequency and Timing Synchronization for OFDM. *IEEE Trans. Communications*, 45(12):1613–1621, Dec 1997.

²Currently not implemented in our prototype, can be based on any existing spectrum sensing algorithm.