SOFTWARE DEFINED RADIO BASED WIRELESS GRIDS

Xuetao Chen (Wireless@VT, Virginia Tech, Blacksburg, VA, USA; chenxt@vt.edu);
S.M. Hasan (Wireless@VT, Virginia Tech, Blacksburg, VA, USA; hasan@vt.edu);
Tamal Bose (Wireless@VT, Virginia Tech, Blacksburg, VA, USA; tbose@vt.edu);
and Jeffrey H. Reed (Wireless@VT, Virginia Tech, Blacksburg, VA, USA;
reedjh@vt.edu).

ABSTRACT

Wireless grids group devices to share hardware and software resources through different types of wireless connections. Due to the heterogonous wireless link and different communication protocols among nodes, reconfiguration capability is essential. Although software defined radio (SDR) may leverage this problem to some extent, how to synchronize the reconfiguration process in a dynamic environment with heterogonous nodes remains an open problem. This paper discusses the reconfiguration requirements of wireless grids. Then, the design challenges for logic clock synchronization are introduced and analyzed in terms of the lower bound and practice design. A synchronization scheme is proposed for multi-hop multi-channel wireless grids. The simulation results show that the proposed method can be used for SDR based wireless grids without a significant increase of time complexity compared with the reference algorithms.

1. INTRODUCTION

Wireless grids can group different types of resources and share them within the grids through wireless links. These resources, such as mobile smart phones, laptops, and even some dummy devices like sensors, printers and cameras, could be attached to the heterogeneous devices. To share resources, these devices need to communicate with each other through wireless links. Since these devices may have different wireless links, a reconfiguration of PHY/MAC layers is needed to maintain a ubiquitous connection. The reconfiguration of the SDR enables the wireless grids to connect to different types of mobile platforms. In addition, resource sharing within a grid is inherently a wireless distributed system. Depending on the application, the specifications of resource sharing may vary. For example, the synchronization and timing schemes for sensor sharing and CPU sharing are different from each other. These differences may also require the reconfiguration of higher layer protocols in addition to PHY/MAC.

In this paper we introduce the structure of the wireless grids based on SDR. Then we will discuss the challenges for designing the SDR based wireless grids, which include RF reconfiguration and synchronization schemes. An example is given at the end of the paper to verify the proposed synchronization scheme for SDR based wireless grid.

2. WIRELESS GRIDS

Wireless grids are defined as ad-hoc dynamic sharing of physical and virtual resources among heterogeneous devices. In a typical wireless grid, a service requesting node initials a resource sharing request to the grid. A node with the requested resource becomes the service node for the requesting node. It will respond to the service request and provide service to its requesting node. All the communication among nodes goes through wireless links, which may be different from each other.

The typical software architecture of a wireless grid based on SDR is shown in Figure 1 [1]. The SDR provides the foundation for integrating different wireless standards. The grid specifications define the interfaces, function requirements and integration processes when mapping the upper layer protocols to the wireless links underneath. It also provides the regulation and sanity check to guarantee the wireless links' reconfiguration works in an expected manner. The resources shared within a grid could be content, software, services, and even hardware devices. In addition, the two API layers support their upper layers with functionalities. The edgeware is the access point when forming a wireless grid, which provides the initialization and tearing down functionality, network monitoring and management, and the "language" when nodes talk to each other.

Wireless grids can reduce the distance between services and their potential users, which reduces the traffic through backhaul networks and servers. This may also lead to

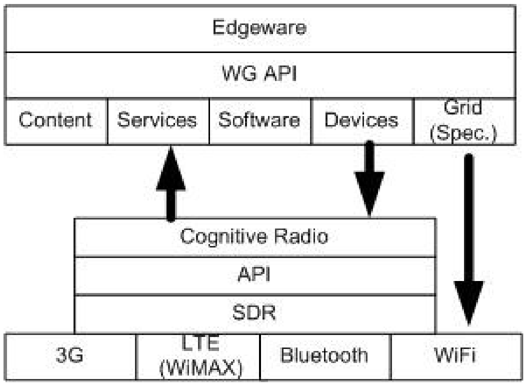


Figure 1. Software architecture of WG nodes.

reduction of network power consumption and response time (RT). Moreover, wireless grids provide ubiquitous access to the desired resources and services. This loose coupling nature leads to a more flexible grouping strategy.

Recently, several wireless grids applications are beginning to emerge, such as cloud printing, home group, distributed content grid, and Global Information Grid [2]. Researchers in the cellular industry are also discussing the possibility of a "cloud cell phone" with LTE technologies [3]. Moreover, Virginia Tech and Syracuse University, as well as several other universities and organizations, are working together on a wireless grid innovation testbed (WiGiT) in order to find some insights into the potential applications of wireless grids and to generate open specifications [1]. In a word, wireless grid is a promising research area that combining wireless communication technologies and distributed computing.

Before the wide application of wireless grids, there are several important open questions that need to be solved, within which, reconfiguration and synchronization are two important topics and will be discussed in the following sections.

3. RECONFIGURATION REQUIRMENTS

Hardware and software reconfigurations are the most important strengths of SDR. Flexible RF hardware, such as the small wideband antenna, RFIC, low power DSP processors, and matching circuits design, is currently under research and development [4][5][6]. The software communication architecture (SCA) and GNU radio are the two famous software architectures for SDR [7]. The reconfiguration capability determined by both the hardware and software is an essential component for wireless grids.

However, reconfiguration capability is also the weakness for a SDR based wireless grid in terms of the synchronization process. A wireless grid has some requirements for the reconfiguration of SDR, such as stability, concurrence, speed, and cost. The reconfiguration itself is actually a disturbance to a wireless grid from the networking point of view. The reconfiguration of different nodes should converge to a certain steady state without manually tuning. Therefore, it also needs to have a global timing reference for all the nodes to behave in an organized way instead of random reconfiguration which hurdles the concurrence. Moreover, the reconfiguration process should be fast enough in order to resume the disrupted service and avoid interference to other nodes. Finally, the lower cost is always preferred considering the device heterogeneity.

The reconfiguration process has to be coordinated among nodes. For example, both WiFi and Bluetooth work in the 2.4 GHz ISM band and even have several channels overlapping with each other [8]. If the reconfiguration process of one node lags behind or progresses much faster than the other nodes, the networks will need more time to be synchronized with each other and make an agreement on the logic clocks. Moreover, the un-synchronized nodes will introduce extra interference and disruption. This is even worse for a wireless grid with dynamic spectrum access capability since the false alarm of channel occupation will increase.

Therefore, a successful reconfiguration of wireless links is critical for SDR based wireless grids. Each reconfiguration process of wireless grids is correspondent to an initialization process, of which synchronization is a building block. A bad synchronization will influence the stability, speed, and concurrence of a reconfiguration directly. A careful design of the synchronization scheme is one of the requirements and the basis of a successful reconfiguration.

4. SYNCHRONIZATION SCHEMES

There are several levels of synchronization for radios, such as carrier, bit, slot, frame/super-frame, and protocol. The rest of this paper discusses the synchronization of the logic clock in the initialization phase of wireless grids. The logic clock of a node depends on the hardware clock and the message it receives, based on which a round number is generated. For example, each node in the wireless grid maintains a round number during communication. This round number defines the timing behavior of protocols. Once they have proper round numbers, synchronization is achieved and the grid is formed.

Synchronization is more important, but difficult for wireless grids, than conventional wired distributed computing networks due to the openness and randomness of the wireless channel, the random locations of nodes, and the ad-hoc nature of wireless grids. The synchronization skew among nodes happens for three reasons: the uncertainty of message delay, the drifts of hardware, and the network diameter and size [9]. Although the uncertainty of message delay can be mitigated with a careful design of clock rate increase or decrease [10], dynamic wireless environments and heterogeneous devices will deteriorate the synchronization performance and even compromise this type of methods. In addition, the wireless protocol parameters, such as re-transmission times, communication frequency band, and node degrees, will also impact synchronization performance. For the SDR platform, the interface between the RF front-end and GPP may add more uncertainty to the process of synchronization [11].

The easiest way to achieve synchronization is GPS [9]. But this solution requires a line of sight reception of the satellite signal. In addition, not every device is capable of embedding a GPS receiver, especially wireless sensors. Moreover, the non-determinism in transmission time caused by the MAC layer leads to difficulties for a coordinated time increment.

There are five requirements for synchronization: validity, synch. commit, correctness, agreement, and liveness [12]. Once synchronized, every node knows it is synchronized and the round numbers can continue increasing without losing the synchronization, which means that each node will always output a consistent round number after synchronization with high probability.

4.1. Performance Bound

The performance bounds for wireless synchronization are listed as follows.

a. A single Channel

It can be shown that no algorithm can prevent a clock skew of $\Omega(\log_b D)$ for neighboring nodes with a diameter of D [13], where the value of b relates to the maximum clock drifting bound. D is the maximum uncertainty between any two nodes. The upper bound has been improved to this value also. A skew between any two nodes is bounded by $\Omega(\sqrt{D})$ with high probability.

b. Multiple Channels with Interference.

For a regular network [12], its synchronization protocol requires at least

$$\Omega(\ln^2 N/((F-t)\ln\ln N)+Ft\ln N/(F-t))$$

where F is the total number of channels. t is the maximum number of the possible disrupt channels, N is the upper bound of participated nodes, or network size, and (F-t) is the number of clear channels.

4.2. Practice Designs

The synchronization performance bounds are based on the worst case analysis. In practice, synchronization may vary according to system assumptions. Optimal synchronization protocols may have better performance than the lower performance bound by adapting to different scenarios instead of only considering the worst case. Different types of synchronization schemes have been suggested for wireless sensor networks in practice [14][15]. The most famous ones are reference broadcasting synchronization (RBS) and the flooding time synchronization protocol (FTSP). RBS uses physical time-stamps to remove the non-determinism of the transmitter and form precise relative timing without an external reference [16]. By utilizing MAC time-stamping and time skew estimation, FTSP can mitigate the uncertainty of message delay to a few clock ticks [17]. These schemes are suitable for synchronization during communications.

Some of the current proposed synchronization algorithms for network initialization are based on leader election to group the nodes since leader election is a better choice for multi-hop and dynamic networks compared with spanning tree [12][13]. Once a leader is elected successfully, all the other nodes within a cluster can synchronize themselves to their leader. If the nodes' location are modeled as a two dimensional Poisson point process, a leader election or synchronization process can be finished in $T=3e\Delta\ln N$ with a high probability no less than $1-N^{-3}$. Here, Δ is the maximum number of a node's neighbors (network degree), and N is the total number of nodes (network size) [18]. For a multi-channel and single hop scenario, an adaptive protocol terminates synchronization within $O(t'ln^3N)$ depending on the actual disrupt frequency number, $t'<t$, instead of the worst case with t disrupted channels [12].

4.3. Proposed Algorithm

SDR based wireless grids require synchronization before and after reconfiguration. Moreover, different wireless links may have different frequency channels. Sometimes, these channels may be overlapped, such as between WiFi and Bluetooth. Therefore, multi- channel is a requirement of the reconfiguration process of SDR based wireless grids. The results of the synchronization process for SDR based wireless grids should be multi-hop and multi-channel with clusters as building blocks.

However, the existing synchronization schemes do not satisfy these requirements. For example, [12] only considered a multi-hop wireless network with a single channel. Although [18] discussed the multiple channel cases, it only considered a single hop scenario.

We assume that the minimum unit for the communication is a slot. Each node can switch frequency channels from slot to slot. The receiver can receive a message successfully when there is only one transmitter transmitting. That is, nothing

Table 1. Frequency Table Update Algorithm

1:	*If (collision happens)*
2:	*Score(i) = Score(i) – (F-1);*
3:	*Score(j) = Score(j) +1; j ≠ i*
4:	*q(k) = Score(k)/sum(p(k)); k = 1,2…F.*
5:	*else if (Rx successfully)*
6:	*Score(i) = Score(i) + 1;*
7:	*Score(j) = Score(j) -1; j ≠ i*
8:	*q(k) = Score(k)/sum(p(k)); k = 1,2…F.*
9:	*end*

can be received if collision happens. Each transmitter transmits according to a probability of p, which is an ALOHA type transmission.

Moreover, each node maintains a frequency table and scores for each frequency channel. The score indicates the channel status. When collision happens, the node will reduce the score for that channel. When the node receives a message successfully on a channel, it will increase its score. This score is normalized to generate the probability for frequency selection. When a node needs to switch the channel, it always chooses the better channel with higher probability. Note that spectrum sensing is not required with this algorithm. This frequency table actually provides channel cognition even without spectrum sensing. The details for constructing the frequency table are shown in Table 1.

This multi-channel cognition capability may be integrated with the multi-hop clustering method in [18] in order to form a synchronization scheme for a SDR based wireless grid. The proposed method is included in Appendix A. Theoretical analysis can be performed following the similar steps in [18].

5. AN EXAMPLE

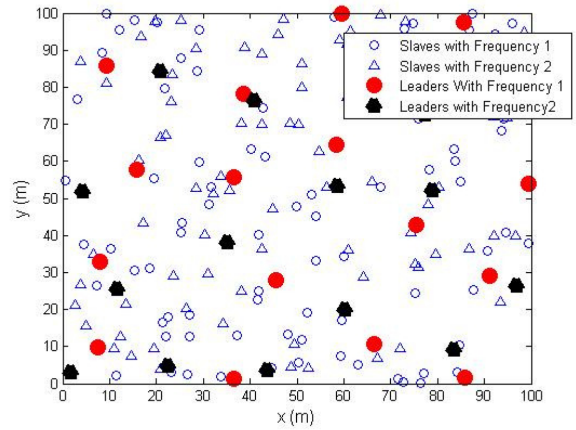
In this section, we will simulate a synchronization process based on the proposed method. In order to compare the proposed method with the method in [18], we follow a similar definition of parameters.

If the nodes are located within an area with a fixed node density of λ, and transmission range of α, the maximal number of a node's neighbors Δ can be derived with an accuracy not lower than $1-\varepsilon$ [19]. The optimal transmission probability can be shown as

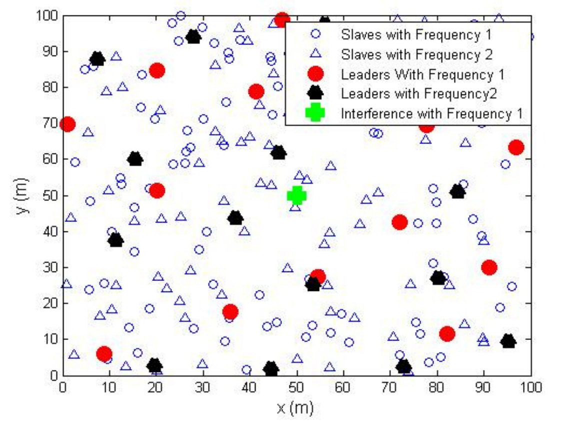
$$p = 1 - \sqrt[1-\Delta]{\Delta}$$

The timer threshold for status transitions are

$T_1 = 3\Delta \ln N$, and $T_2 = 3e\Delta \ln N$



(a) Without interference



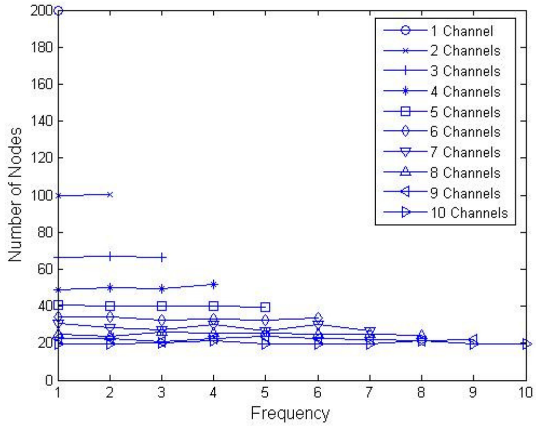
(b) With interference.

Figure.2. Location distribution with proposed method

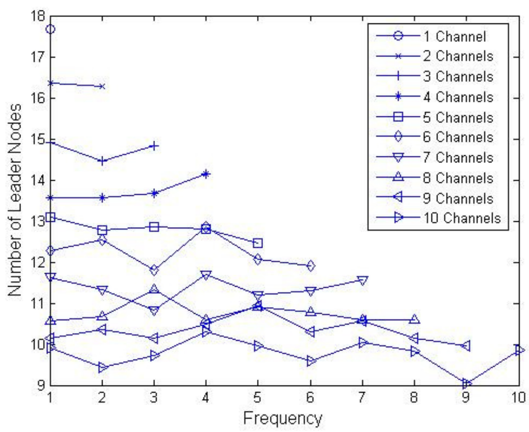
where N is the network size [18].

If there are 200 nodes located within an area of 100×100 square meters, λ =0.02, α = 20 meters, and $\varepsilon = 10^{-3}$, then $\Delta = 60$, p = 0.0263, T_1=954, and T_2=2592 [18]. We assume that the total number of the frequency channel is F=2.

Figure 2 shows a simulation result with the above parameters. Notice that the uniform distribution of the leader nodes and the slave nodes are within the transmission range of the leader nodes. Figure 2(b) shows a result when an interference source is located in the center of the area with a transmission range of 20 meters. The nodes within the inference's neighborhood can avoid the disrupted frequency channel that the interference is using. Moreover, the number of nodes in channel 1 equals that of channel 2. Both the number of nodes and the number of leaders are almost evenly distributed among frequency channels when the number of frequency channels increases, as illustrated in Figure 3(b).

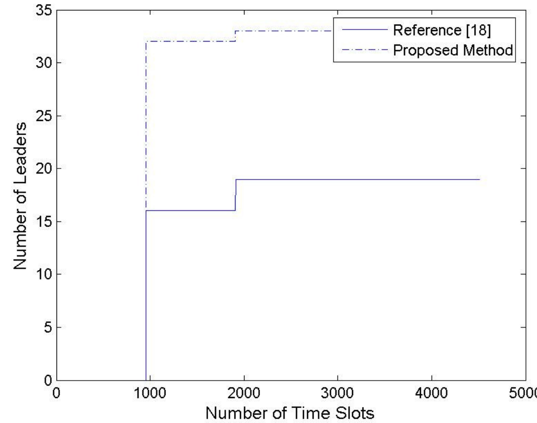


(a) Nodes distribution

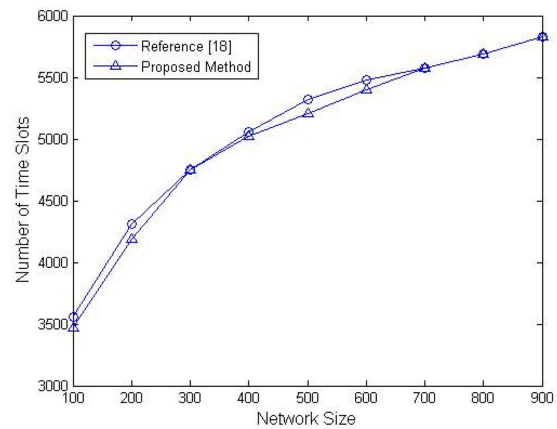


(b) Leader distribution

Figure 3. Nodes distribution among frequency channels.



(a) Convergence comparison



(b) Time complexity comparison

Figure 4. Performance comparison.

Figure 4(a) compares the convergence performance of the proposed method with the method in [18]. The modified method does not increase the running time too much. It only increases the number of leaders. Figure 4(b) shows the running time for different network sizes. The proposed method is robust to the network size since the running time is a concave function of network size.

6. CONCLUSION

This paper discusses the function requirements of SDR based wireless grids, especially reconfiguration and synchronization. A synchronization method with frequency channel cognition is proposed to satisfy the multi-hop and multi-channel requirements of SDR based wireless grids. The simulation examples show that the proposed method can help form the wireless grid within a limited running time. The nodes distribution amongst different frequency channels is even.

ACKNOWLEDGMENT

This work was supported in part by NSF Grant # 0917973 and ONR Grant # N00014-07-1-0536.

APPENDEX

The proposed algorithm adds frequency channel cognition into the reference method [18]. The details of the algorithms are summarized in Table 2.

Table 2. Network Synchronization Algorithm

Fun_send(): Send packet with prob. p.
Fun_recv():receive packet in current slot.
Freq_update(): update frequency table
Freq_selection(): select frequency according to q.
State of node i: Si = quasi-leader
***While** (1)*
Timer++
***If** Si = quasi-leader then*
Fun_send()
***If** Fun_recv() = quasi_leader*
S_i = quasi_slave; timer = 0;
Freq_update();
***else if** Fun_recv() = leader*
*Si = slave; **Freq_update();** break;*
end
***If** timer = T1*
Si = leader; timer = 0;
end
***If** collision = 1*
Freq_update();Freq_selection();
end
end
***If** Si = quasi-slave*
***If** Fun_recv()=quasi-leader*
Timer = 0;
Freq_update();
***else if** Fun_recv()=leader*
*Si = Slave; **Freq_update();** break*
end
***If** timer = T1*
Si = quasi-leader; timer = 0;
end
***If** collision = 1*
***Freq_update();Freq_Selection();**break;*
end
end
***If** Si=leader*
Fun_send();
***If** timer = T2*
Break;
end
end
If Si = slave
Fun_recv();
If collision = 1
Freq_update();
end
end
end

7. REFERENCES

[1] Wireless Grid Innovation Testbed (WiGiT), online resource: http://wglab.net/

[2] Global Information Grid, Wikipedia, online resource: http://en.wikipedia.org/wiki/Global_Information_Grid.

[3] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-Advanced networks," *IEEE Commun. Mag.*, Vol. 47, No. 12, pp. 42-49, Dec. 2009.

[4] G. Cafaro, T. Gradishar, J. Heck, S. Machan, G. Nagaraj, S. Olson, R. Salvi, B. Stengel, and B. Ziemer, "A 100 MHz 2.5 GHz Direct Conversion CMOS Transceiver for SDR Applications," *IEEE Radio RFIC Symp. 2007*, pp.189-192, June 2007.

[5] K. Lim, and J. Laskar, "Emerging opportunities of RF IC/system for future cognitive radio wireless communications," *IEEE RWS*, pp. 703 – 706, Jan. 2008.

[6] S. E. Sussman-Fort, "Matching network design using non-Foster impedances," *International Journal of RF Microwave Computer-Aided Engineering*, pp. 135–42, March 2006.

[7] J. Hoffman, D.A. Ilitzky, A. Chun, and A. Chapyzhen-ka, "Architecture of the Scalable Communications Core," *NOCS '07.*, pp.40-52, May 2007.

[8] U. Ramacher, "Software-Defined Radio Prospects for Multistandard Mobile Phones," *IEEE Computer Mag.*, Vol. 40, No. 10, pp. 62-69, Oct. 2007.

[9] C. Lenzen, T. Locher, P. Sommer, and R. Wattenhofer, "Clock Synchronization: Open Problem in Theory and Practice," *SOFSEM* 2010, pp. 61-70, 2010

[10] R. Fan, and N. Lynch, "Gradient Clock Synchronization," *PODC'04*, St. John's, Newfoundland, Canada, July, 2004.

[11] T. Tsou, and J. H. Reed, "Software architecture for cooperative applications," *SDR Forum*, Washington DC, Dec. 2009.

[12] S. Dolev, S. Gilbert, R. Guerraoui, F. Kuhn, and C. Newport, "The Wireless Synchronization Problem," *PODC'09*, Calgary, Alberta, Canada, Aug. 2009.

[13] P. Sommer, and R. Watternhofer, "Gradient Clock Synchronization in Wireless Sensor Networks," *IPSN'09*, San Francisco, CA, USA, Apr. 2009.

[14] B. Sundararaman, U. Buy, and A. D. Kshem, "Clock Synchronization for Wireless Sensor Networks: A Survey," *Ad Hoc Networks*, Vol. 3, No. 3, pp. 281-323, May 2005.

[15] S. Fikret, and Y. Bulent, "Time Synchronization in Sensor Networks: A Survey," *IEEE Networks*, Vol. 18, No. 4, pp. 45-50, 2004.

[16] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time: Synchronization using Reference Broadcasts," *OSDI '02*, Boston, MA, Dec. 2002.

[17] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol," *Proc. 2dn ACM Conf. on Embedded Networked Sensor Systems*, pp. 39-49, 2004.

[18] P. Guo, T. Jiang, K. Zhang, and H.-H. Chen, "Clustering Algorithm in Initialization of Multi-hop Wireless Sensor Networks," *IEEE Trans. on Wireless Commun.*, Vol. 8, No. 12, Dec. 2009.

[19] S. Iyer and D. Manjunath, "Topological Properties of Random Wireless Networks," *Proc. Indian Academy Sciences*, vol. 31, no. 2, pp. 117-139, Apr. 2006.