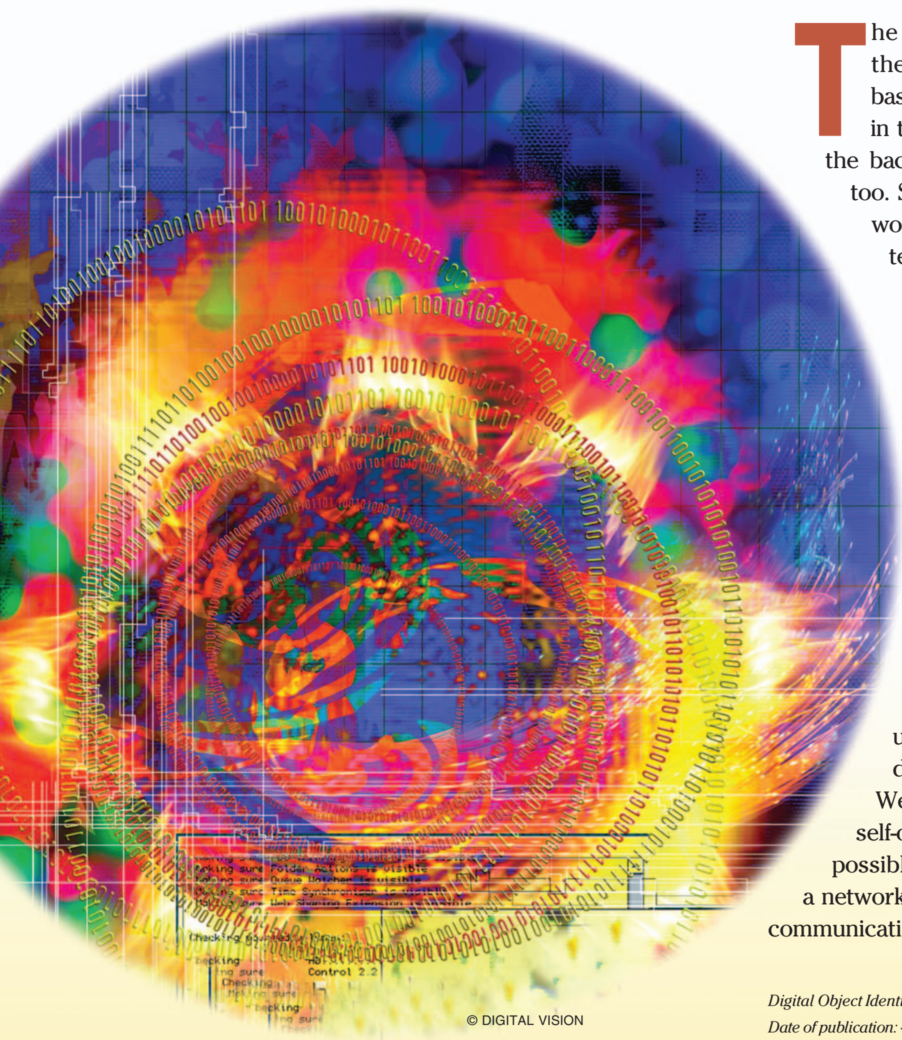
Bilal Zafar, Dominik Schulz,
Soheyl Gherekhloo, and Martin Haardt

© PHOTO F/X2

AD HOC NETWORKING SOLUTIONS

Cooperative MIMO Multihop Networks



© DIGITAL VISION

The decentralization of wireless networks to reduce the dependence on fixed infrastructures such as base stations has become the subject of vast interest in the past few years. In the future, this might form the backbone of general commercial mobile networks too. Such a system would be quite attractive for network operators and users alike, as it is inherently better suited to respond robustly to any unforeseen situation. At the same time, it does not require a fixed infrastructure that reduces both the setup and maintenance costs of a network. However, decentralization gives rise to various network setup and management issues on different open systems interconnection (OSI) layers. In this article, we present a few solutions, on different OSI layers, that provide a good end-to-end performance of such a scheme while ensuring that the network is decentralized and operates in a self-organized manner.

The vision for future communication systems stems from the need to render more robust networks in the wake of unforeseen situations. Therefore, the need to reduce the dependence on fixed infrastructure is paramount. We envision future communication networks to be self-organizing and cognitive. In this direction, it is possible to view mobile handsets as radio resources in a network and to utilize them as intelligent relays to form communication links in the absence of base stations. Figure 1

Digital Object Identifier 10.1109/MVT.2010.939907
Date of publication: 4 March 2011

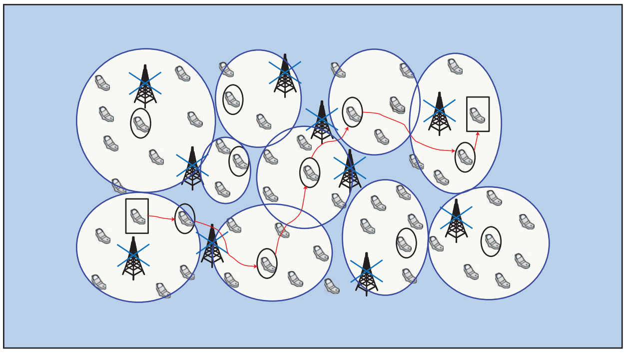


FIGURE 1 Cluster-based multihop system.

illustrates the communication between two nodes in a cluster-based multihop network.

For such a system to become operational to the stage illustrated in Figure 1, we need to go through various steps. The fact that the network has no centralized control means that we have to deal with complex issues of scheduling, synchronization, and interference cancellation in an ad hoc manner that is not easy. Since the exact number of nodes in a particular area is not known and neither are their clocks synchronized, it is difficult to apply channel-access schemes like time division multiple access (TDMA). One option is to perform extensive piloting and gossiping in the network to synchronize clocks of all the nodes. Applying multiple access schemes becomes easier once the network is clustered. However, in an unclustered network, it is very hard to avoid collisions. The clustering process is performed by extensive piloting between the nodes to determine the cluster heads and cluster members. It involves several piloting phases in which each node transmits one short pilot. With each passing phase, the information in the pilot reflects the status of the node. It is reasonable to assume that the collisions are less likely if the piloting messages are short and intermittent and that the time for each clustering phase is quite long in comparison.

We propose to further decrease the probability of interference by employing the frequency division multiple access (FDMA) as well. The entire bandwidth is divided into a certain number of bands, and every node can send signals at one of these frequency ranges. The frequency bands at which the signal can be transmitted are known to the receiver. The received signal will be shifted in the base band by every frequency and then passed through the low-pass filter. If the power of the output signal of the filter exceeds a threshold, the frequency recovery is accomplished and demodulation can be performed, but if it is not the case, it can be concluded that either no signal is sent in this time slot or the sender node is so far away that the sent signal is completely attenuated, and the received SNR is so low that the decoding of the received packet is very defective. In this way, the packets can be decoded by nearby nodes based on the received signal strength indicator (RSSI).

Even though ad hoc mobile networks that support multihopping have been around for a while, we are still some way off the practical deployment of a completely self-organized version of a cluster-based multihop network that can deal with the various issues relating to network management, routing, localization, interference, and suitable physical layer strategies, etc. These issues are not trivial and have been, mostly individually, the subject of a lot of interest over the years. Schemes mentioned in [1]–[3] try to solve the various issues relating to a self-organized network with multihop capabilities. However, there is a lack of architecture that brings together the various methods developed over the years to implement, view, and possibly optimize a complete system based on them. For this purpose, we have built the software Self-Organizing Network with Intelligent Relaying (SONIR) [4], in MATLAB, which implements an end-to-end multihop, virtual multiple-input multiple-output (MIMO) system, capable of dealing with the mobility of nodes in a Rayleigh fading environment. Methods for clustering, mobility management, routing, and virtual MIMO have been implemented. These methods work on different OSI layers. The system shown in Figure 1 is the result of several steps: first, clustering is performed; then, the nodes do extensive piloting to gather as much local information as needed, which helps in identifying the optimal gateways to the neighboring clusters essential for routing. Finally, we are able to establish a route from source to destination. These steps are illustrated in Figure 2.

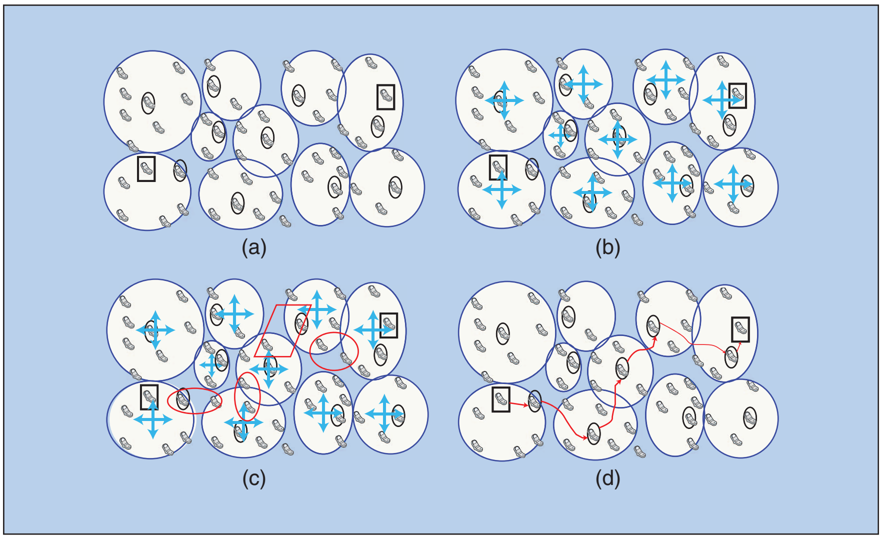


FIGURE 2 Steps for multihop routing in ad hoc networks. (a) Clustering. (b) Local information gathering. (c) Identification of gateways. (d) Multihopping.

Clustering

The most promising option to deal with such cluster-based ad-hoc networks is clustering. It involves grouping together of neighboring nodes to form groups or clusters with one node, designated to act as the local coordinator, known as the *cluster head*. We have implemented two clustering schemes: the RSSI-based method [5] and a global positioning system (GPS)-based scheme to tackle mobility similar to [6]. We are also working on modifying these schemes to better deal with a network changing rapidly due to the movement or failure of nodes. Reference [5] is one of the newest proposed clustering schemes and a good option to simplify the process of clustering through piloting and local information gathering but cannot deal well with the mobility of nodes. A very good scheme to maintain clustering in the face of the mobility of nodes in the system is presented in [6]. We propose to combine these schemes to exploit the common benefits and be able to deal with time-varying real-life networks.

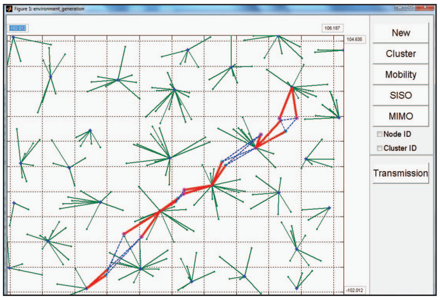


FIGURE 3 Routing path in case of 2×2 cooperative multihop MIMO.

Mobility

Most of the techniques developed for the system in question arise from sensor networks, where nodes are stationary or quasi-stationary. But, if we are to employ such a system for general-purpose communication, we will have to deal with a network in which nodes are moving around at will. The goal of this part of the system is to keep the network operational when faced with highly dynamic mobile nodes. The primary method is to periodically recluster the system to cater to mobility. In SONIR, we provide the option to determine the optimal clustering time based on the quality-of-service (QoS) demand that is of immense interest to network planners. Thus, we can determine the optimal network reclustering time as a function of QoS [bit error rate (BER)] and the speed of the mobiles. Based on simulation results, we can eventually form a mathematical model to determine the reclustering time in relation to these quantities.

Localization and Routing

Localization and routing for self-organized networks is a nontrivial task and has many implementational issues. Having GPS-equipped nodes helps with localization, but it is still quite difficult for the nodes to get an accurate information about their surroundings and neighboring nodes. In our system [4], the cluster heads deal with most of the routing functions. They maintain tables of the IDs of nodes in their cluster and use the nodes to gather information about the presence of neighboring clusters. Moreover, they have tables indicating the most suitable node (or nodes, in case of cooperative MIMO) to use to communicate with a particular neighboring cluster. Ad hoc on-demand distance vector routing (AODV) is a promising option to deal with the routing issue. In our system, we have improved upon AODV in that we update the route when we perform reclustering even during the same session of communication between two nodes to cater for the movement of nodes in the network. In turn, AODV maintains the same route during one session of communication. The cluster maintenance tables and routing tables are also updated periodically, and the optimal update interval can be determined as described in the last section. An interesting and significant future work would be to establish a routing scheme that performs routing based on the (cooperative) MIMO channel between clusters from the start, rather than extending AODV for this purpose.

Cooperative MIMO Schemes

Recently, the idea of cooperative/virtual MIMO has been integrated into such networks as well, for the case of nodes with single antennas. Cooperative or virtual MIMO is a good concept for mobile nodes with a single antenna to be able to achieve the benefits of MIMO communications. It is ideally suited for the distributed clustered system under investigation, where we can employ the nearby nodes to cooperate and form a virtual antenna array. It is also imperative to modify the routing scheme to incorporate the need for multiple paths between clusters. A routing example with 2×2 (virtual) MIMO channels between clusters is shown in Figure 3. In Figure 4, we show the comparison of 2×2 virtual MIMO to the single-input single-output (SISO) case for the example multihop communication scenario in Figure 3. The MIMO gain is clearly visible, thus justifying the use of a virtual MIMO scheme wherever we have idle nodes available. If, however, we do not have any extra nodes available, then SISO multihopping provides a good performance too, thus eradicating the need for fixed infrastructures such as base stations.

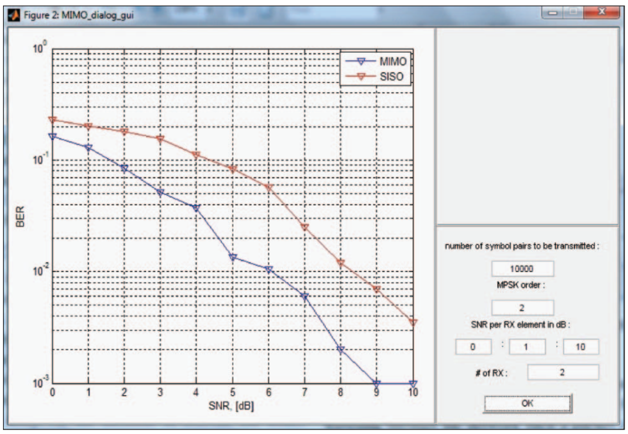


FIGURE 4 BER comparison of 2×2 virtual MIMO multihop with SISO multihop.

We are also investigating other options for the physical layer, such as randomized space-time block codes (R-STBCs), to improve the performance.

Decentralized Resource Mapping

The absence of a global/central control over the network means that we have access to just one node in most cases. It is an important problem to get the resource information about the whole network by just querying one node, because it will help us in managing the network and providing service recovery in bad service areas. We will perform this by a combination of gossiping and flooding of resource-request packets through the network, so that each node can append its local information or QoS information to the packet. After traveling through the network, such a packet can help us to form various resource maps of our decentralized network, such as location mapping and QoS mapping.

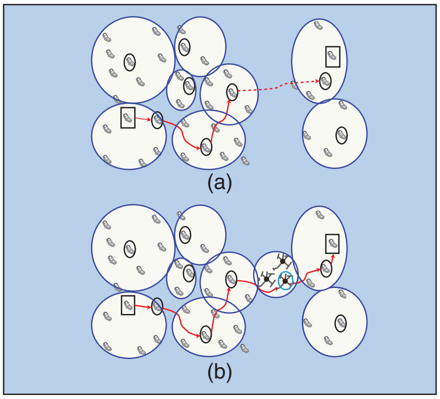


FIGURE 5 Service-recovery mechanism. (a) Detection of areas with bad service. (b) Service recovery.

Service Recovery

Our network is inherently robust compared to networks with critical nodes like base stations, because all we need to establish communication is the presence of some nodes to help with relaying. It is, however, important that such a network is also robust in the face of service outages due to nonavailability/failure of nodes by an appropriate placement of new nodes. The ability to form resource maps also helps us to recover our systems in case of failures. This is crucial to maintain the robustness of a self-organized system. Using QoS maps in conjunction with location maps, we can find out the areas of bad coverage and alleviate the situation. At the moment, we are focusing on introducing mobile relays [unmanned aerial vehicles (UAVs), robots, etc.] into such areas to assist in forming virtual MIMO relays. Figure 5 illustrates how we can provide service recovery in our system by employing a swarm of quadrocopter-based relays to form a virtual MIMO cluster.

Physical Layer and Transmission Issues

When considering cooperative MIMO, it is crucial to also look at the impairments. One of these impairments is the synchronization problem. Since the nodes suffer from unequal hardware, self-heating, and environmental changes (e.g., temperature change), the clocks will not run in perfect synchronization. Most of this problem is solved via suitable synchronization protocols on higher and physical layers. On the physical layer, one can make use of multiple synchronization techniques to achieve a certain level of phase synchronization. These techniques include, e.g., reference tone synchronization, where (as proposed in [7]) the receiver emits a signal and all relay nodes change their phase until their respective SNRs are maximized. Another well-known technique is the firefly synchronization shown in [8], which basically makes use of coupled oscillator theory. There are also blind techniques that estimate the phase information from the signal subspace (e.g., [9]). SONIR provides a good platform for comparison of these schemes.

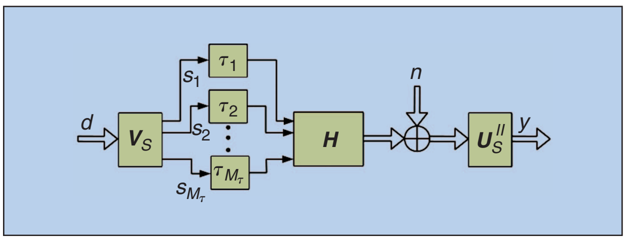


FIGURE 6 Block diagram showing the used synchronization-error model.

While in a simulator environment we are able to simulate under ideal conditions, for the real-world implementation, we have to consider the practical issues shown in [10] and [12]. As seen earlier, these effects can be mitigated via suitable protocols or even via suitable channel estimation [11]. However, some delays are randomly distributed so that their effect is not reflected by the measured channel. These kind of synchronization problems are assumed to cause positive and negative delays within a symbol pulse. This problem has been investigated in [14] for the case of space-time coding. We have adopted the used synchronization-error model to delay errors in eigenmode transmission.

The assumed synchronization-error model is shown in Figure 6. We assume eigenmode transmission without water filling for simplicity. In this figure, $\boldsymbol{V}_{\mathrm{S}}$ is the precoding matrix defined as the matrix of right singular vectors from the economy size singular value decomposition (SVD) of the channel matrix $\boldsymbol{H}$. If $\boldsymbol{s}$ is the vector of symbols to be sent then $x = V_S s$ is the precoded transmit vector. The decoding is done via the matrix $\boldsymbol{U}_{\mathrm{S}}$ containing the left singular vectors accordingly. $\boldsymbol{\Sigma}_{\mathrm{S}}$ is the diagonal matrix of the singular values of $\boldsymbol{H}$. The model assumes a virtual transmit array of M_{T} transmitters and M_{R} collocated antennas at the receiver side. The delays τ_i are assumed to be random variables. The idea is not to directly investigate the influence of delay but to take the pulse-shaping function into account. If $g_{\mathrm{p}}(t)$ is the pulse-shaping function, then $g_{\mathrm{p}}(\tau_i)$ is a random variable as well. For our investigations, we have assumed that all transmitters use the same kind of hardware so that τ_i has the same distribution for every transmitter. Note that this is not an unreasonable simplification, since we can easily extend it to the case of different statistics. As a result we get the following input-output relation:

$$\hat{\boldsymbol{s}} = \sqrt{\frac{P}{M_T}}\boldsymbol{U}_{\mathrm{S}}^{\mathrm{H}}\boldsymbol{H}\boldsymbol{x} + n = \sqrt{\frac{P}{M_T}}\Sigma_S \boldsymbol{V}_{\mathrm{S}}^{\mathrm{H}}\boldsymbol{G}\boldsymbol{V}_{\mathrm{S}}\boldsymbol{s} + n. \qquad (1)$$

The matrix G is a diagonal matrix with $\boldsymbol{G} = \mathrm{diag}([\mathrm{gp}(\tau_n) \ldots \mathrm{gp}({}^{\tau}M_T)])$. Thereby, we can provide an upper bound of the achievable mutual information when only ones are transmitted.

$$I^{(1)} = \sum_{i=1}^{M_\mathrm{T}} \log_2 \left(1 + \mathrm{SINR}_i^{(1)}\right). \quad (2)$$

The term $\mathrm{SINR}_i^{(1)}$ is the signal-to-interference-and-noise-ratio (SINR) of the ith substream, assuming that only ones are transmitted.

$$\mathrm{SINR}_i^{(1)} = \frac{\mathbb{E}\left\{|v_i^{\mathrm{H}} \boldsymbol{G} v_i|^2\right\}}{\mathbb{E}\left\{\left|v_i^{\mathrm{H}} \boldsymbol{G} \sum_{\substack{l=1\\ l \neq i}}^{M_\mathrm{T}} v_l\right|^2\right\} + \frac{\mathbf{M}_\mathrm{T}}{\lambda_i \rho}}. \quad (3)$$

Here, $\boldsymbol{V_i}$ is the ith right singular vector and $\rho = P/N_0$ with P being the overall power sent by all the relays and N_0 being the power of the uncorrelated, additive, white, Gaussian noise.

Figure 7 clearly shows the loss of mutual information compared with the perfectly synchronized case. For these simulations, we have assumed a root-raised cosine function as the pulse shaper with a rolloff factor of 0.5. The delays τ_i are drawn from a uniform distribution between $-0.8\,T_\mathrm{S}$ and $+0.8\,T_\mathrm{S}$, where T_S is the BPSK symbol interval.

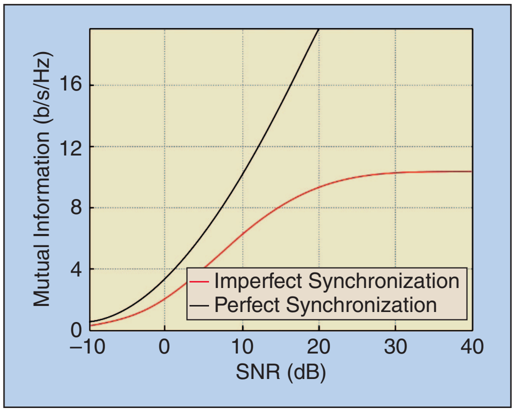


Figure 7 Mutual information for the case of perfect and imperfect synchronization.

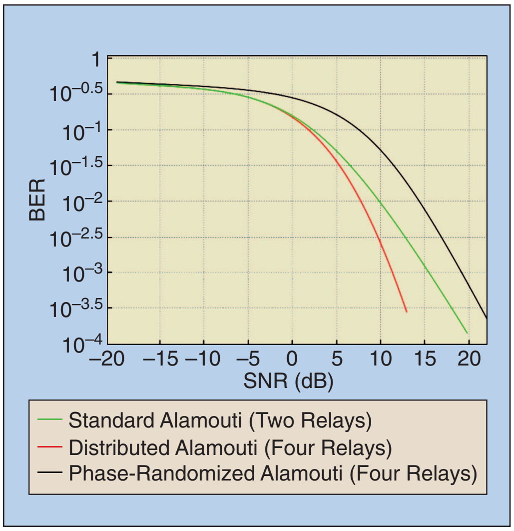


Figure 8 Alamouti in different MISO scenarios.

As seen from the simulation, the gain which can be achieved from virtual MIMO is reduced in all SNR ranges. In the very high SNR range, one can additionally observe a saturation. This is due to the fact that imperfect synchronization results in an intersymbol interference, which itself results in limited SINR. Therefore, a good method of synchronization is very important for such systems.

In cases where it is not feasible to feedback information to the transmitter, space-time codes provide a convenient way to exploit virtual MIMO gains. As mentioned earlier, one possibility is to use a distributed version of Alamouti's code. In this case, half of the transmitting nodes transmit one row of the space-time block code and the other half transmits the remaining row. If the amount of transmitters is odd, one of them transmits the sum of both STBC rows.

The challenge of that scheme is that the receiver has to know which transmitter is going to transmit which STBC row. Thus, there has to be some kind of allocation information distributed in advance. In a real scenario, this may hardly be feasible. This is due to the fact that the network power consumption should be as uniform as possible throughout the whole network. Hence, possible transmitters should choose on their own whether to take part in a transmission or not. Especially in a large network, a hard assignment of STBC rows would result in more overhead and reduced scalability.

As the goal is to have a self-organized network, nodes have to be flexible and just rely on local information. Several approaches have been proposed for that. The basic idea behind them is that every node keeps on listening to its surroundings. By doing this, it gets an estimate of the number of relays that currently take part in transmission. Every such node may also receive some kind of satisfaction feedback from the receiving node. By having this (local) information, any node can decide whether to support the current transmission or not.

Another approach is to not directly use the STBC itself but to let the nodes transmit a linear combination of the columns of a given STBC. The coefficients may be obtained from a special matrix, where each relay selects a row randomly. The extreme case would be to let the relays choose the coefficients for the linear combinations completely randomly. This type of STBC is called R-STBC and has been extensively studied in [13].

The randomization of STBCs can be done in different ways. It is possible to just draw random components independently from a normal distribution for the real and imaginary parts. It is also possible to randomize the phase and magnitude of the coefficients. One example is shown in Figure 8. Here, we have simulated the Alamouti schemes in a multiple-input single-output (MISO) configuration. We have tested the simple centralized Alamouti code with two antennas, the distributed version with four relays, and a randomized Alamouti code with four relays as well. The randomization was performed by setting the magnitude constantly to one and randomly choosing a phase uniformly distributed between $-\pi$ and $+\pi$. The channel was chosen to be a block flat-fading Rayleigh channel. The relays are assumed to be perfectly synchronized.

For the two distributed STBCs, the array gain is clearly visible, since we are using twice the number of relays. The best performance is shown by the distributed version of Alamouti's code. The performance of the randomized version is degraded because of the induced uncertainty when decoding the STBC. This uncertainty slightly reduces the diversity order to a fractional value. However, it is visible that randomization provides a tradeoff between performance and decentralization.

Conclusions

Virtual MIMO can boost the reliability, throughput, and coverage of networks. However, network designers and protocol designers have to take care about the special needs of virtual antenna arrays such as synchronization and self-organization. More work is needed in this field to actually decide when it is worth switching to a virtual MIMO link to guarantee a certain QoS and extend the network's lifetime.

While the implementation of cluster-based multihop virtual MIMO systems presents several serious challenges, the various benefits of such a scheme are clearly visible and render it one of the most promising options for future wireless communication architectures. However, the scheme needs to be optimized on various OSI layers, and some of the possible solutions for optimization have been discussed in this article.

Acknowledgments

This research has been performed within the Graduate School on Mobile Communications (GS-Mobicom) at Ilmenau University of Technology, Germany, which has been partially funded by the German Research Council [Deutsche Forschungsgemeinschaft (DFG)] and the Fraunhofer IIS. The authors gratefully acknowledge this support.

Author Information

Bilal Zafar (bilal.zafar@tu-ilmenau.de) completed his B.E. degree in electrical engineering from the National University of Technology (NUST), Pakistan, in 2007, and his M.S. degree in wireless systems from KTH, Sweden, in 2009. He is currently pursuing his Ph.D. degree at International GS-Mobicom and the Communications Research Laboratory (CRL) at the Ilmenau University of Technology, Germany. His research interests include decentralized and ad hoc networks, cooperative MIMO schemes, routing, network coding, and wireless communications.

Dominik Schulz (dominik.schulz@tu-ilmenau.de) completed his electrical engineering degree at the Ilmenau University of Technology, Germany, where he obtained his Diplom-Ingenieur degree in 2009. Since then, he has been part of International GS-Mobicom and CRL at the Ilmenau University of Technology. His research interests include wireless communications, array signal processing, virtual antenna arrays, and hypercomplex algebra.

Soheyl Gherekhloo (soheyl.gherekhloo@tu-ilmenau.de) received his B.E. degree from the Ilmenau University of Technology, Germany, in 2010, and is currently pursuing his master's degree in information and communication technology at the same university and is also working as a research assistant in CRL.

Martin Haardt (martin.haardt@tu-ilmenau.de) studied electrical engineering at the Ruhr-University Bochum, Germany and at Purdue University, USA. He received his Diplom-Ingenieur (M.S.) degree from the Ruhr-University Bochum in 1991 and his Doktor-Ingenieur (Ph.D.) degree from Munich University of Technology in 1996. In 1997, he joined Siemens Mobile Networks in Munich, Germany, where he was responsible for strategic research for third-generation mobile radio systems. From 1998 to 2001, he was the director for international projects and university cooperations in the mobile infrastructure business of Siemens in Munich, where his work focused on mobile communications beyond the third generation. Since 2001, he has been a full professor in the Department of Electrical Engineering and Information Technology and a head of the Communications Research Laboratory at Ilmenau University of Technology, Germany. He has received the 2009 Best Paper Award from the IEEE Signal Processing Society, the Vodafone (formerly Mannesmann Mobilfunk) Innovations Award for outstanding research in mobile communications, the ITG Best Paper Award from the Association of Electrical Engineering, Electronics, and Information Technology (VDE), and the Rohde and Schwarz Outstanding Dissertation Award. His research interests include wireless communications, array signal processing, high-resolution parameter estimation, as well as numerical linear and multilinear algebra.

References

[1] C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, Sept. 1997.

[2] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Commun. Survey Tutorials*, vol. 7, no. 1, pp. 32–48, 2005.

[3] J. N. AL-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[4] B. Zafar, S. Gherekhloo, A. Asgharzadeh, M. T. Garrosi, and M. Haardt, "Self-organizing network with intelligent relaying (SONIR)," in *Proc. IEEE Int. Conf. Mobile Ad-hoc and Sensor Systems (IEEE MASS)*, San Francisco, CA, Nov. 2010, pp. 765–767.

[5] S. Fazackerley, A. Paeth, and R. Lawrence, "Cluster head selection using RF signal strength," in *Proc. Canadian Conf. Electrical and Computer Engineering (CCECE)*, 2009, pp. 334–338.

[6] B. An and S. Papavassiliou, "A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks," in *Proc. CISS*, Johns Hopkins, Baltimore, 2001, pp. 387–395.

[7] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Scalable feedback control for distributed beamforming in sensor networks," in *Proc. IEEE Int. Symp. Information Theory*, Sept. 2005, pp. 137–141.

[8] A. Tyrrell, A. T. G. Auer, and C. Bettstetter, "Firefly synchronization in ad hoc networks," in *Proc. MiNEMA*, Workshop, 2006, p. 4.

[9] T.-R Tsai and D.-F. Tseng, "Subspace algorithm for blind channel identification and synchronization in single-carrier block transmission systems," *IEEE Trans. Signal Processing*, vol. 88, no. 2, pp. 296–306, 2008.

[10] T.-D. Nguyen, O. Berder, and O. Sentieys, "Impact of transmission synchronization error and cooperative reception techniques on the performance of cooperative MIMO systems," in *Proc. IEEE Int. Conf. Communications*, 2008, pp. 4601–4605.

[11] Y. Jing and B. Hassibi, "Distributed space-time coding in wireless relay networks," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 3524–3536, Dec. 2006.

[12] M. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," in *Proc. Wireless Communications and Networking*, 2003, pp. 1266–1273.

[13] B. S. Merken and A. Scaglione, "Randomized space-time coding for distributed cooperative communication," in *Proc. IEEE Int. Conf. Communications*, Istanbul, Turkey, 2006, pp. 5003–5017.

[14] R. C. Palat, A. Annamalai, and J. H. Reed, "Upper bound on bit error rate for time synchronization errors in band-limited distributed MIMO network," in *Proc. IEEE Wireless Communications and Networking Conf.*, 2006, pp. 2058–2063.

VT