Using GNU Radio for Experiments on Data Distribution in Wireless Ad-hoc Networks

Tatsunori Kimpara† and Susumu Ishihara‡

†Graduate School of Engineering, Shizuoka University, para@ishilab.net
‡Graduate School of Science and Technology, Shizuoka University, ishihara@ishilab.net
3-5-1, Johoku, Hamamatsu-Shi, Shizuoka, 432-8011, Japan

ABSTRACT

Both simulation and experimentation in a real environment are important in the evaluation of protocol performance in wireless ad-hoc networks. Simulation is useful for evaluating systems that are difficult to implement inexpensively. However, it is not easy to simulate the behavior of radio signals in the real world on a large scale. Experimentation is required to evaluate implemented software/hardware systems and to understand their behavior in the real world. However, evaluating wireless ad-hoc networks under various conditions requires many devices, a wide experimentation field, and a huge amount of human resources. In this paper, we discuss using GNU Radio, which can control various wireless signal processing tasks more flexibly than off-the-shelf Wi-Fi and ZigBee devices, to build a flexible, narrow-space wireless network testbed. We evaluated a protocol that disseminates a data item in an opportunistic way and uses a random network coding technique in the real world and through simulation. We then present guidelines for building an experimental environment for a wireless ad-hoc network using GNU Radio in a small space.

1 Introduction

High performance devices and network technology are evolving on a day-by-day basis, and new services using ad-hoc networks, wireless sensor networks (WSNs), and vehicular ad-hoc networks (VANETs) are constantly being proposed. WSNs are expected to be used in environmental research to simulate weather conditions in fire-prone areas such as mountains and to track the positions of moving objects such as animals. VANETs are expected to be used to assist drivers in choosing the best route for avoiding traffic jams, preventing accidents, etc. However, there are several problems with using ad-hoc networks, and many researchers have suggested various techniques and protocols to address them.

Simulation has long been a major research methodology in the field of wireless networks, especially with large-scale ad hoc networks. Simulation enables researchers to evaluate protocols under various conditions by altering the parameters used, e.g., node density, communication range, mobility, and traffic patterns. Network simulation software provides models of popular protocols, and all users have to do is implement the model of the protocol to be evaluated. However, simulations often fail to accurately reproduce the effect of physical behaviors or radio signals such as path loss, interference, multi-path fading, etc. due to the difficulty of modeling such behaviors as well as the high cost of computation. Hence, real-world experimentation is important in terms of both designing and improving the protocols and applications for wireless networks.

The communication range of wireless devices is a significant factor in the development of a wireless network experimental environment. For example, when we construct an ad-hoc network with a 3-hop path using devices with a 100 m communication range, we have to use a large experimental field 300 m in diameter. If a field of such size cannot be used, devices with a shorter communication range have to be chosen. It is difficult to fully control the parameters of off-the-shelf Wi-Fi and ZigBee devices, so these devices are not ideal for constructing the experimental field. Moreover, the more devices that are used, the higher the cost of maintaining the devices. In addition, real-world experimentation with mobile devices is more difficult. Therefore, if we want to use a small experimentation field, we need devices with a short transmission range and flexible control capability of radio signals in order to reduce costs.

We constructed an experiment environment using open source software radio toolkit GNU Radio [1] which can control various wireless signal processing functions.

So far, implementations of software radio have been based on FPGA, and the signal processing functions of each wireless transceiver have to be written in hardware description language to change the FPGA's logic. This means it is not easy to implement a new transceiver on conventional software radio hardware. Adding to this, FPGA hardware is quite expensive, which in turn makes the total cost of software radio systems more expensive. In contrast, GNU Radio enables users to implement software radio using PCs and cheap devices that have A/D conversion and Up/Down conversion capabilities, high-speed SRAM, etc., and the signal processing module blocks are written in C++. Wireless systems are designed by connecting blocks with Python, an object-oriented script language. Up/Down, A/D, and D/A conversion are processed by Universal Software Radio Peripheral (USRP) products that can access various frequency radio bands by using the appropriate daughter board. Moreover, processing blocks are provided by an open-source library, thus making it is easy to develop wireless systems.

One of the most well-known software radio development toolkits is WARP, which was developed by Rice University [2]. WARP is an FPGA-based software development toolkit that has a higher performance than GNU Radio and that can be used to develop broadband wireless communication. However, the FPGA of WARP is more expensive than that of USRP, and therefore, experiments using many nodes will be very expensive. We therefore decided to use GNU Radio and USRP2, a second-generation USRP product.

In this paper, we present guidelines for constructing a real-world experimentation field for basic experiments on data dissemination protocols in wireless ad hoc networks that require various network parameters to evaluate a protocol that is based on R2D2V [3], a data dissemination scheme designed for VANETs. The rest of this paper is organized as follows. We present related work about real-world experiments on wireless ad-hoc networks in section 2. In section 3, we describe basic experiments for measuring the radio transmission range of USRP2. Next, in section 4, we describe indoor and outdoor experimentation environments for evaluating the performance of a data dissemination protocol using random network coding (RNC) in wireless ad-hoc networks. We then discuss the guidelines for experiments on wireless ad hoc networks using GNU Radio. Finally, we conclude the paper in section 5.

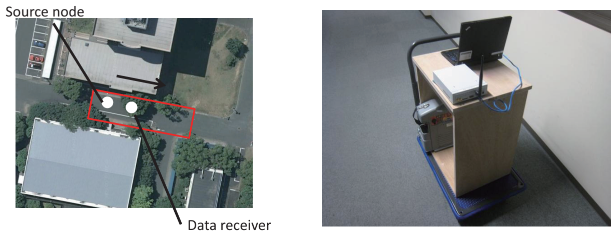
2 Related Work

There has already been much research devoted to ad-hoc network experimentation.

ORBIT is a testbed for wireless networks designed by researchers at Rutgers University [4]. This testbed includes 400 programmable radio nodes for at-scale emulation and has a function for reproducible emulation of wireless network protocols and applications. The radio nodes can communicate with each other by using IEEE 802.11a/b/g, ZigBee, and Bluetooth. Some of the nodes use GNU Radio. All of the nodes are placed in one large room. ORBIT adjusts the transmission range of the nodes in accordance with settings determined by the user, and it also uses an application that controls the links of nodes to construct whatever spuriously multi-hop networks the user wants. However, the main disadvantage of ORBIT is that maintenance of the many devices and the large experimentation field is very expensive.

KanseiGenie is a testbed for WSNs designed by researchers at Ohio State University [5]. This testbed provides 96 “Kansei Nodes” stored in one room. Each Kansei Node is comprised of one XSM, four TelosBs, and one Imote2, each of which can communicate with each other by using IEEE 802.11, 802.15.4, and 900 MHz Chipcon CC1000 radios. KanseiGenie is an interesting case study in the field of testbed development, but like ORBIT, it is very expensive due to the necessity of maintaining the many devices and the large experimentation field.

Network coding (NC) [6] is an effective technique for making optimal use of available network resources by encoding several packets received by intermediate nodes. NC has been shown to be useful for improving throughput and robustness in wireless networks. However, pure NC is useful only when the network topology is fixed. Therefore, random network coding (RNC) [7] has been proposed to enable the use of NC in wireless networks in which nodes move autonomously. Ho et al.[7] showed that randomly selecting coefficients for linear codes over a Galois field (GF) can be used to improve the capacity of networks. NC techniques have been used to develop the protocols of wireless networks [8][9][10].



(a) Outdoor experiment (b) Sending node

Figure 1: Photograph of basic experiment.

Katti et al. showed that NC may improve the throughput of networks when the paths of multiple uni-cast flows intersect in a multi-hop wireless network [8], while Akubczak et al. showed that RNC offers substantially more flexibility, allowing coding over symbols and the use of multiple paths [9]. These protocols were developed to increase wireless network throughput and have been evaluated in real-world environments using PCs equipped with IEEE 802.11. Katti et al. also proposed symbol-level network coding [10] to perform channel access decisions based on the quality of various links in the presence of concurrent transmissions. Symbol-level network coding has been evaluated in real-world environments using GNU Radio and USRP.

3 Basic Experiment for Measuring Transmission Range of USRP2

It is important to evaluate various protocols for increasing throughput and/or robustness on wireless networks in real-world experiments. However, to determine the position of wireless terminals in the experimentation field, we first need to know the communication range of the wireless devices. This is particularly important when developing a narrow experimentation field: we need to know the minimum communication range when various control parameters and modulation functions are used. In this section, we describe the GNU Radio and Universal Software Radio Peripheral (USRP) that we used, explain how we measured the communication range of the USRP2, and discuss the result of the measurement.

3.1 GNU Radio and USRP2

GNU Radio is an open-source software development toolkit that performs various types of signal processing and that includes many of the elements found in radio systems, filters, decoders, demodulators, etc. (called “blocks” in GNU Radio jargon). Users can develop a software radio system using hardware in which GNU Radio connects the processing blocks to make a flow chart. We chose to use a USRP2 due to the high compatibility between GNU Radio and USRP.

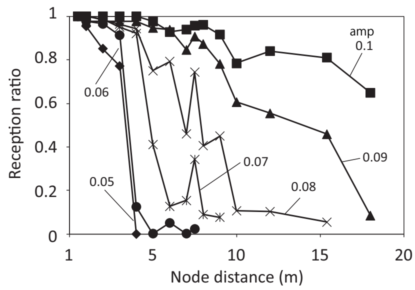
The USRP2 and GNU Radio are responsible for different types of signal processing. GNU Radio mainly deals with complex signal processing, the generating of wireless waveforms, digital modulation, etc., and provides various digital modulation schemes (GMSK, DBPSK, DQPSK, etc.), while USRP2 mainly deals with high-speed signal processing, Up/Down, A/D, and D/A conversion, Up/Down sampling, etc. In a typical sequence, first, GNU Radio pushes data into digital streams when the user sends a data item through GNU Radio and USRP2. Next, GNU Radio sends the digital streams as IQ samples to USRP2 through the Gigabit Ethernet. USRP2 interpolates IQ samples that are sent to GNU Radio to 100 M samples/sec, and then USRP2 Up-converts, D/A-converts, and sends data streams as real-waves. When USRP2 receives a data item, it sends IQ samples to the PC after A/D-converting, Down-converting, and downsampling. The Gigabit Ethernet connects the USRP2 with its host computer.

3.2 Measuring parameters indoors and outdoors using USRP2

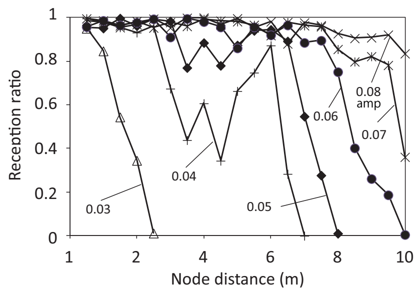
In order to develop an experimental wireless network with multiple hops in a narrow field, the communication range of wireless devices has to be small. We therefore tried to configure various parameters to adjust the communication range of USRP2. Transmission gain, receive gain, transmission bit rate, packet size, modulation scheme, and the amplitude of the carrier signal all affect this communication range. The results of some preliminary experiments showed that adjusting the amplitude of the carrier signal is an effective way to adjust the communication range of USRP2 among these parameters for measuring communication in the short range.

We used two USRP2s outdoors (Fig. 1(b)) to investigate the relationship between communication distance and amplitude of the carrier signal. One USRP2 was configured as a sending node and the other as a receiving node. We used an XCVR2450 daughter board that could support the 2.4 GHz–2.5 GHz and 4.9 GHz–5.85 GHz bands. The maximum transmission power of the board was 100 mW in the former band and 50 mW in the latter. The transmission gain could be controlled in the range of 0–30 dB and the receive gain could be controlled between 0–91 dB. The amplitude of the carrier signal could be adjusted to a value between 0 and 1. When the amplitude of the carrier signal was 1, the full D/A converter scale was used. We used rubber duck omni-directional antennas that covered the wireless radio frequency the daughter board supported. The sending gain of the antennas was 3 dBi.

We measured the success ratio of packet reception to investigate the communication range of USRP2. We fixed a receiving node and moved a sending node using a hand cart (Fig. 1(b)). We used two python programs, benchmark_tx and benchmark_rx, which were bundled with the GNU Radio package to measure the distance at which packets could be successfully decoded. Both programs were relatively simple: benchmark_tx only broadcast packets and benchmark_rx only received packets broadcast by benchmark_tx. These programs detected receiving errors by CRC32. The sending node transmitted 3000 packets. Each packet was 1500 bytes long and the bit rate was 500 kbps. The modulation scheme was DBPSK. We measured the success ratio of packet reception from the number of successfully decoded packets at the receiving node. The center of the wireless radio frequency was set to 5.11 GHz, the transmission gain was set to 0.01 dB, and the receiving gain was set to 75 dB.



(a) Outdoor



(b) Indoor

Figure 2: Packet reception success ratio.

When the USRP2 was used for an extended period of time, the center frequency of the carrier signal was not stable: it shifted a number of kHz. This unstable carrier frequency resulted in a low success ratio of the packet reception. To avoid this effect, we had to adjust the frequency periodically. The experiment was conducted in August on the Shizuoka University campus. The air temperature was 20 ℃ and the humidity was 55%.

People and cars occasionally passed near the experiment field, which affected the success ratio of the packet reception. We therefore conducted the experiments at midnight to avoid these effects. Moreover, if someone or something approached the experiment field, we stopped the measurement immediately and discarded the results obtained within the last few seconds.

The results of the outdoor measurement are shown in Fig. 2(a). The amplitude ranged from 0.05 to 0.1. The packet reception ratio became worsened as the amplitude of the carrier signal became low and the distance became large. The effect of the distance and the amplitude was stable in that changes to their values had a certain regularity. However, when the distance between nodes was between 5 and 10 m, the fluctuation of the packet reception ratio increased. This is probably an effect of multi-path fading. We also performed experiments using an amplitude value ranging from 0 to 1. However, when the amplitude value was large the reception ratio was always 1, and when the value was small the reception ratio was always 0.

We next conducted the indoor experiments. At Shizuoka University, a standard class room is about 13 × 6 m and a big one is about 17 × 8 m. To develop ad hoc networks including a 2- or 3-hop path, it is best if the communication range of USRP2 is smaller than 4 m. Based on the result shown in Fig. 2(a), we set the amplitude of the carrier signal to under 0.07 to keep the communication range of USRP2 under about 4 m. However, in general, the effect of multi-path fading is larger in indoor environments than outdoor ones, so we measured the transmission range in a big classroom (17 × 8 m) to develop the experiment field indoors.

The measurement results of the experiment in the classroom are shown in Fig. 2(b). If we compare Figs. 2(a) and (b), when the communication range of USRP2 in the class room had a distance between nodes of 8–10 m, it is clear that packets could be decoded indoors at an amplitude of 0.07. Packets were rarely successfully decoded outdoors at the same amplitude. When the distance between nodes was around 5 m, the packet reception ratio fluctuated greatly.

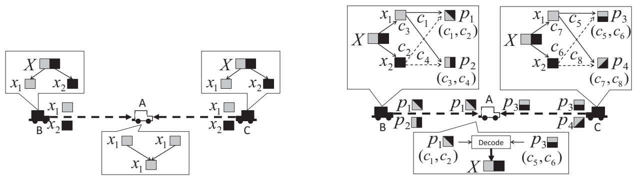
4 Real-world Experiment of Data Dissemination Scheme using RNC

In this section, we describe the ad-hoc network testbed we designed for an experiment on an opportunistic data dissemination scheme using a random network coding (RNC) technique on the basis of the measurement results presented in the previous section. First, we give an overview of RNC and the opportunistic data dissemination scheme that uses this technique, and next, we describe the testbed used for the scheme. Finally, we present the experimental results and compare them with the simulation results.

4.1 Delivery system using random network coding (RNC)

The main advantage of using RNC in data dissemination in wireless ad hoc networks is that it can deliver information with a small number of packets.

An example of RNC in action is shown in Fig. 3. Figures 3(a) and (b) show cases in which car A has broadcast a beacon packet. Let us assume that the beacon includes the position of car A as well as cars that contain data related to this position that they must send back to cars sending the beacon. In the figures, cars B and C are going to send packets of a data item, X, which consists of two parts (x_1 and x_2). Figure 3(a) shows a case in which car A replies to the beacon when RNC is not used. In this case, both B and C broadcast x_1 and x_2. If any packets are lost, A can receive X only if it has received both x_1 and x_2. In other words, if two x_1 packets from A and B are lost, A cannot restore X. Figure 3(b) shows a case in which RNC is used. After receiving the beacon from car A, car B uses RNC to generate two packets ($p_1 = (ax_1 + bx_2)$



(a) No RNC (b) With RNC

Figure 3: Opportunistic data dissemination scheme using random network coding (RNC).

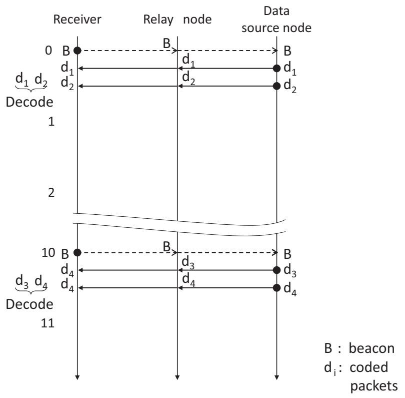
and $p_2 = (cx_1 + dx_2)$) with randomly numbered coefficients (a, b, c, d). It then broadcasts the encoded packets. Car C then generates $p_3 = (ex_1 + fx_2)$ and $p_4 = (gx_1 + hx_2)$ with randomly numbered coefficients (e, f, g, h) and broadcasts them. If the coefficients are linearly independent, car A can restore the original data X when it receives at least two packets from among p_1, p_2, p_3, and p_4. This example demonstrates how RNC can improve reliability in broadcast-based data dissemination in environments that tend to experience a high packet loss rate.

4.2 Experimental method

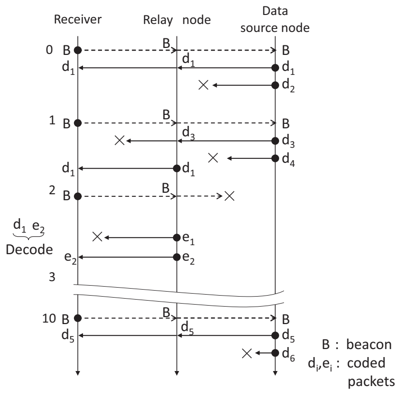
In the previous subsection, we described our evaluation of a scheme that disseminates data items in an opportunistic way by using the RNC technique (Fig. 3(b)). Each node broadcasts a data item when it receives a beacon that requests that data item. The nodes periodically broadcast beacons that include their current position as well as the requested data items. If a node sends a beacon that includes a coordinate of position P, it means that it is requesting data items related to position P. Packets containing a data item sent by nodes that have received a beacon are encoded using RNC.

In our experiment, we used three types of node: source node, relay node, and receiving node. The source node generates one 2000-byte data item every 10 seconds. The receiving node broadcasts one beacon every second until it receives the data item generated by the source node or the number of the beacons reaches 10. Ten seconds after it has sent its first beacon, the receiving node restarts and begins sending a new series of beacons for receiving the new data item generated by the source node. We measured the number of beacons sent by the receiving node before it received each data item generated by the source node. The source node broadcasts encoded packets of the requested data item. The size of each packet is 1000 bytes. When the source node broadcasts the data item, relay nodes can also receive the encoded packets. They hold the packets until the source node generates a new data item. If a relay node receives a beacon from the receiving node, it broadcasts packets encoded by RNC using the encoded packets that it has received from the source node.

The time chart of the experiments is shown in Fig. 4. When the experimentation began, the source node generated a data item and the receiving node broadcast beacons. Figure 4(a) shows an example of a case when there was no communication error. In this case, the source node broadcast encoded packets $d1$ and $d2$ when it received a beacon sent from the re-



(a) Ideal pattern



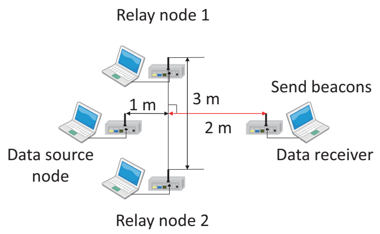
(b) Errors occurred

Figure 4: Time chart of the experiments.

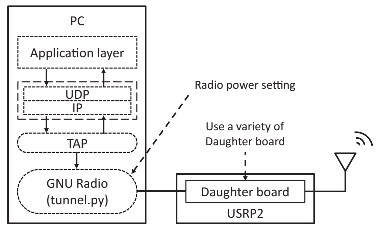
ceiving node just after the experiment started. The receiving node received encoded packets from the source node and then restored the original data item by decoding the encoded packets. The receiving node then stopped sending beacons and did not start sending them again until the source node generated a new data item.

Figure 4(b) shows an example of when some packets were lost. When the experiment began, the source node generated a data item and the receiving node broadcast beacons. When the source node received a beacon from the receiving node, it broadcast encoded packets $d1$ and $d2$. However, the receiving node and the relay node failed to receive $d1$. The receiving node re-broadcast the beacon after one second. After receiving the second beacon from the receiving node, the source node generated two new encoded packets (d_3 and d_4) with new random coefficients for the same data item that had been sent as the reply to the previous beacon. The receiving node received d_1 from the relay node but failed to receive d_3 and d_4 from the source node. Since the receiving node had received only one encoded packet, it could not decode the original data item. The receiving node then sent the third beacon after one second. This beacon was received by the relay node but not by the source node. The relay node generated new encoded packets (e_1 and e_2) using the packets it received from the source node (d_1, d_3, and d_4) and then broadcast the encoded packets. The receiving node received $e2$. It had now received two encoded packets and could therefore restore the original data item by decoding d_1 and e_2. It then stopped broadcasting beacons. In cases like this, when RNC is not used, packets generated by simply dividing the data items (X_1 and X_2) are broadcast without RNC encoding.

We conducted four experiments: (i) RNC and one relay node, (ii) RNC and two realy nodes, (iii) without RNC and one relay node, and (iv) without RNC and two relay nodes.



(a) Layout of nodes



(b) System overview

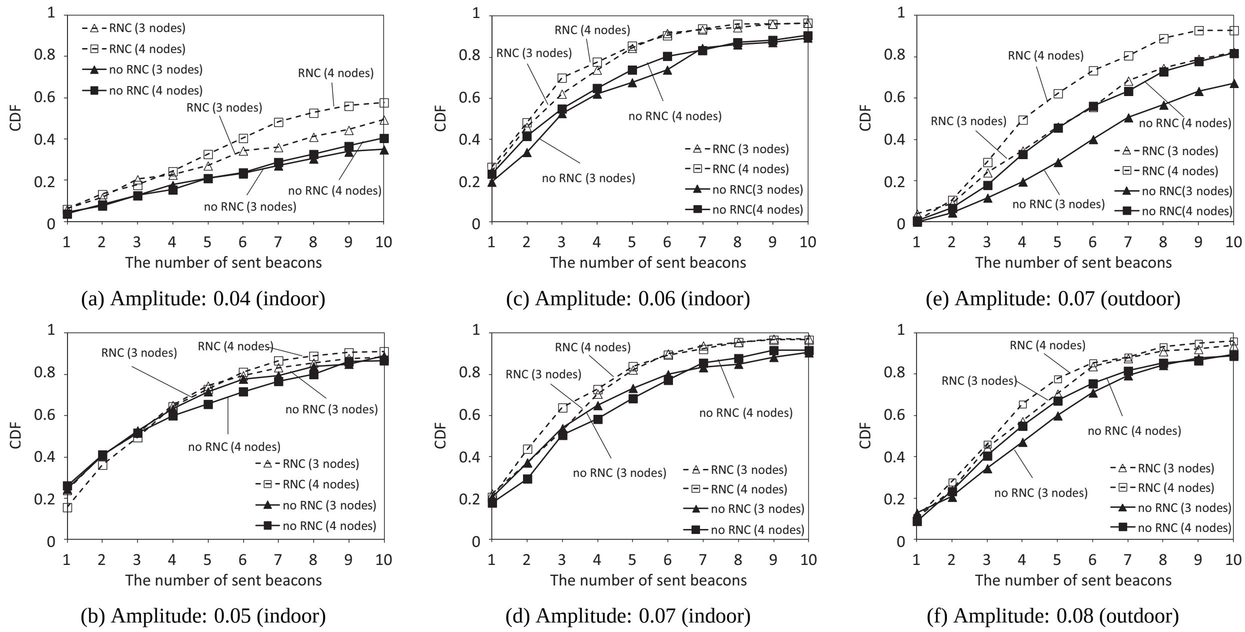
Figure 5: Experimental environment.

4.3 Experimental environment

We developed an environment field on the basis of the measurement results discussed above in order to evaluate the protocol, which is an opportunistic data dissemination scheme using the RNC technique. We used the same configuration as the first measurement. The amplitude of the carrier signal was set to 0.04–0.07 indoors and 0.07–0.08 outdoors.

The node layout in the experiment is shown in Fig. 5(a). There are three data delivery routes from the source node to the receiving node, one a 1-hop path route between the source and the receiving nodes, and the two 2-hop paths via a relay node. In our experiments, the nodes were set in an outdoor field (as shown in Fig 1(a)) and a big classroom. We conducted the experiments on holidays or at midnight to avoid the effect of moving objects or people.

The system configuration of the experiment is shown in Fig. 5(b). We used the TAP system to send and receive IP packets through the GNU Radio-based wireless communication system. TAP is a virtual interface that emulates Ethernet and enables user programs to send and receive communication data that are regularly treated in the data link layer. We used an application layer program for the operations described in this section. Data items that are broadcast by UDP are passed on to tunnel.py, which is one of GNU Radio 's bundled programs that supports the TAP interface.



(a) Amplitude: 0.04 (indoor)
(c) Amplitude: 0.06 (indoor)
(e) Amplitude: 0.07 (outdoor)
(b) Amplitude: 0.05 (indoor)
(d) Amplitude: 0.07 (indoor)
(f) Amplitude: 0.08 (outdoor)

Figure 6: CDF of reception rate for indoor and outdoor environments.

Table 1: Packet loss rate of beacon packets and data packets.

		Beacon			Data		
	Amp.	Rcv.-Src.	Rcv.-Relay 1	Rcv.-Relay 2	Rcv.-Src.	Rcv.-Relay 1	Rcv.-Relay 2
Indoor	0.04	57.9%	93.4%	82.9%	78.3%	91.3%	74.1%
	0.05	33.5%	23.7%	100%	60.6%	83.8%	100%
	0.06	50.2%	36.1%	59.7%	69.0%	68.8%	77.4%
	0.07	31.7%	23.2%	20.6%	64.3%	66.7%	84.9%
Outdoor	0.07	48.5%	14.5%	11.5%	92.3%	73.1%	71.7%
	0.08	45.4%	15.5%	10.1%	71.5%	69.2%	72.2%

4.4 Experimental results

The results of indoor and outdoor experiments using an amplitude of 0.04–0.08 are shown in Fig. 6. The graphs in the figure show the cumulative distribution function (CDF) of the number of beacons sent until the original data item is decoded at the receiving node. The x-axis shows the number of sent beacons. The CDF values when the number of sent beacons is 10 mean the total data reception rates. Table 1 shows the loss rate of beacons and data packets between the receiving node and other nodes. The temperature of the experiment site was 25 ℃ with 27% humidity.

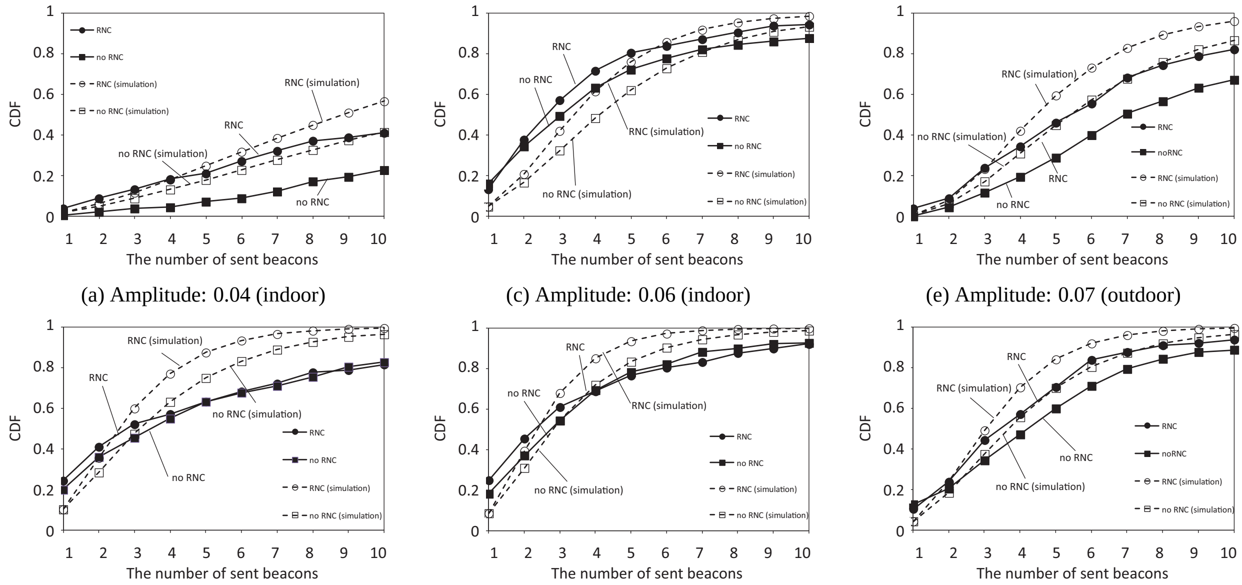
Figure 6(a)–(d) shows the indoor results and (e)–(f) the outdoor. Using RNC improved the data reception ratio compared with not using it. Moreover, there was a significant difference between the data reception ratio with and without RNC when the amplitude of the carrier signal was small. As shown in Table 1, the packet loss rate was high when the amplitude of the carrier signal was low. This demonstrates that the RNC is particularly effective for improving the reliability of data dissemination when the packet loss rate is high.

There was no difference between the number of beacons for the one-relay node case and the two-relay node case (Fig. 6(b)). The packet loss rate between the receiving node and relay node 2 was 100% (Table 2), so for the indoor experiment with amplitude 0.05 we only evaluated the 3-node case. Please note that the center frequency of relay node 2 shifted after the experiment started, which ultimately affected the measurement result. The presented graph (Fig. 6(b)) includes this effect in order to demonstrate the effect of using GNU Radio and USRP2.

4.5 Comparison of experimental and simulation results

We conducted a packet-level simulation of the data dissemination scheme in the same scenario as the experiments so that we could compare the results. The wireless communication between nodes was simply modeled as the success or failure of packet transmission. The behaviors of the MAC layer protocol and physical propagation of the wireless signals were not considered.

First, we assigned the simulation model the packet loss rate derived from the measurement in real environments, as shown in Tables 1 and 2. These packet loss rates were obtained when there was no background traffic, so the actual packet loss rate during the data dissemination experiments might be slightly different. The simulation results of 1000 trials when using



(a) Amplitude: 0.04 (indoor) (c) Amplitude: 0.06 (indoor) (e) Amplitude: 0.07 (outdoor)

(b) Amplitude: 0.05 (indoor) (d) Amplitude: 0.07 (indoor) (f) Amplitude: 0.08 (outdoor)

Figure 7: Comparison of simulation and measurement results. Packet loss rate results were the same in both cases.

Table 2: Estimated packet loss rate obtained by fitting the simulation results with the measurement results.

		Beacon		Data	
	Amp.	Rcv.-Src.	Rcv.-Relay	Rcv.-Src.	Rcv.-Relay
Indoor	0.04	65%	93.4%	84%	91.3%
	0.05	40%	35%	75%	85%
	0.06	60%	40%	70%	70%
	0.07	40%	40%	71%	75%
Outdoor	0.07	57%	23%	92%	73%
	0.08	53%	25%	76%	73%

only one relay node are shown in Fig. 7(a)–(f).

As the graphs in Fig. 7(a)–(f) show, the difference between the simulation results and the measurement results is large. This may be due to the difference between the actual packet loss rate during the data dissemination operation and the packet loss rate obtained without background traffic.

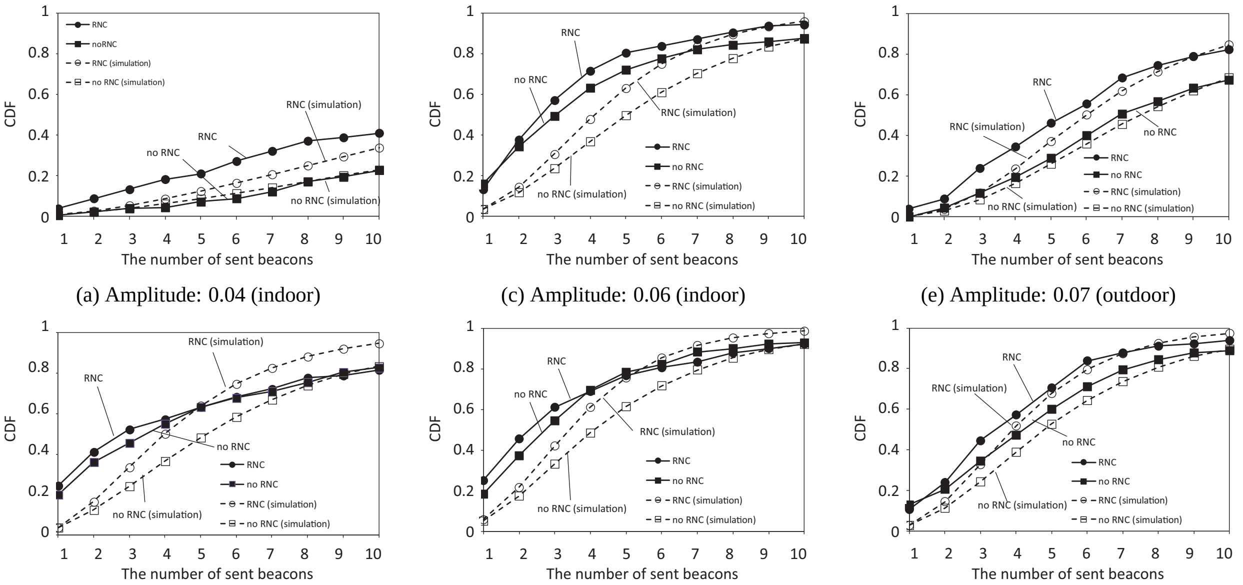
To estimate the actual packet loss rate during the data dissemination operation, we changed the packet loss rate in the simulation so that the simulation results were closer to the measurement results of a case when RNC was not used (summarized in Table 2).

Figures 8(a)–(f) show the simulation result obtained using the estimated packet loss rate (Figs. 8(a)–(d) for the indoor experiment and Figs. 8(e) and (f) for the outdoor). The difference between the simulation and measurement results when using RNC is bigger for the indoor environment than for the outdoor environment. If we compare Tables 1 and 2, the difference of the packet loss rate is less than 10%. We therefore conclude that the communication condition of the indoor environment was unstable. In contrast, in the outdoor environment, the difference between the simulation and measurement results of the packet loss rate as well as the packet reception ratio is small. We conclude that the experiment in the outdoor environment was under a stable communication condition and that the results are therefore reliable.

4.6 Discussion

As shown in Fig. 5(a), the distance between the receiving node and the source node was longer than that between the receiving node and the relay node, and therefore the packet loss rate between the receiving node and the relay node was lower than that between the receiving node and the source node (as shown in Table 1). Moreover, the indoor packet loss rate between the receiving node and the relay node was higher than that outdoors when the amplitude was 0.04–0.07. One reason for this is the effect of fading, which was stronger in the indoor environment than in the outdoor environment. The variation of packet loss when the amplitude was 0.04–0.07 was higher indoors than outdoors when the amplitude was 0.07–0.08. Since the variation of the packet loss rate is less sensitive to the amplitude of the carrier signal, it is difficult to configure indoor communication environments, as expected. However, it is still possible to evaluate the behavior of protocols in a less than ideal communication environment.

Because the fading effect in the outdoor environment was relatively weak, the communication environment was less sensitive to the parameters than the indoor environment, mak-



(a) Amplitude: 0.04 (indoor)
(c) Amplitude: 0.06 (indoor)
(e) Amplitude: 0.07 (outdoor)
(b) Amplitude: 0.05 (indoor)
(d) Amplitude: 0.07 (indoor)
(f) Amplitude: 0.08 (outdoor)

Figure 8: Simulation results with modified packet loss rate.

ing it easier to configure the environment. We should therefore use an outdoor environment to develop experimental fields. If we conduct a large scale experiment in which the multi-hop counts are larger than 2, the possibility of decoding the packets will decrease because it is difficult for receiving nodes to receive all the divided packets. However, if we use RNC, the nodes can decode packets even if they only receive a minimum number of packets. We assume that in such cases the use of RNC will significantly improve the reliability of data dissemination.

As shown in Table 1, the relationship between the packet loss rate and the amplitude was not stable. However, the packet loss rate increased as the amplitude decreased. The packet loss rate and the amplitude seemed to change almost regularly. This means we can control the packet loss rate by changing the amplitude value.

5 Conclusion

We have presented guidelines for building an experimental environment for a wireless ad-hoc network using GNU Radio in a small space. In the measurement experiments, we configured the parameters of GNU Radio to measure the communication range of USRP2. Based on the results of preliminary experimentation to obtain the parameters between the amplitude of the carrier signal and the communication distance of the USRP2, we developed a narrow experiment field using GNU Radio and USRP2 and evaluated a protocol that disseminates a data item in an opportunistic manner with a random network coding technique. Experimental and simulation results showed that it is much easier to configure an outdoor environment than an indoor one. We also found that the stability feature of the carrier frequency strategy affects the measurement results. Our future work will focus on the development of an experiment environment that can support mobile nodes.

REFERENCES

[1] GNU Radio: ⟨http://gnuradio.org/redmine/⟩ (accessed 2012-2-8).

[2] WARP:⟨http://warp.rice.edu/⟩ (accessed 2012-2-8).

[3] Kusumine, N. and Ishihara, S.: R2D2V: RNC based Regional Data Distribution on VANETs, *Proc. IEEE Vehicular Network Conference* (*VNC2010*), pp.271–278 (2010).

[4] Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H. and Singh, M.: Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols, *Proc. Wireless Communications and Networking Conference*, vol.3, pp.1664–1669 (2005).

[5] Ertin, E. and Arora, A. and Ramnath, R. and Naik, V. and Bapat, S. and Kulathumani, V. and Sridharan, M. and Zhang, H. and Cao, H. and Nesterenko, M.: Kansei: a testbed for sensing at scale, *Proc. 5th Conf. Information Processing in Sensor Networks*, pp.399–406 (2006).

[6] Ahlswede, R., Cai, N., Li, S. Y. R. and Yeung, R. W.: Network information flow, *IEEE Trans. Information Theory*, Vol.46, No.4, pp.1204–1216 (2000).

[7] Ho, T., Medard, M., Koetter, R., Karger, D. R., Effros, M., Shi, J. and Leong, B.: A random linear network coding approach to multicast, *IEEE Joumal on Selected Areas in Communications*, vol.22, pp.107–120 (2004).

[8] Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M. and Crowcroft, J.: XORs in the air: practical wireless network coding, *Proc. ACM SIGCOM'06*, pp.243–254 (2006).

[9] Chachulski, S., Jennings, M., Katti, S. and Katabi, D.: Trading Structure for Randomness in Wireless Opportunistic Routing, *Proc. ACM SIGCOMM'07*, pp.169-180 (2007).

[10] Katti, S. and Katabi, D. and Balakrishnan, H. and

Medard, M.: Symbol-level network coding for wireless mesh networks, *Proc. ACM SIGCOM'08*, pp.401-412 (2008).