Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed

Manjunath Doddavenkatappa, Mun Choon Chan, and A.L. Ananda

School of Computing, National University of Singapore
{doddaven,chanmc,ananda}@comp.nus.edu.sg

Abstract. This paper presents Indriya, a large-scale, low-cost wireless sensor network testbed deployed at the National University of Singapore. Indriya uses TelosB devices and it is built on an active-USB infrastructure. The infrastructure acts as a remote programming back-channel and it also supplies electric power to sensor devices. Indriya is designed to reduce the costs of both deployment and maintenance of a large-scale testbed. Indriya has been in use by over 100 users with its maintenance incurring less than US$500 for almost 2 years of its usage.

In the second part of this paper, we provide an extensive study of all 16 channels of IEEE 802.15.4 supported by CC2420 devices constituting Indriya. The objective is to understand performance differences and correlations that may exist among different channels. The measurement results are of interest to the WSN community as they may be useful in designing WSN routing and MAC protocols that exploit channel-diversity.

1 Introduction

In this paper, we first present Indriya [1], a wireless sensor network testbed deployed at the School of Computing, National University of Singapore (NUS). Indriya has been available for internal use since April 2009 and is publicly available since December 2009. The testbed is used by users from more than 35 universities for research. We are also using it for teaching within NUS.

Indriya is installed with 127 TelosB motes. More than 50% of the motes are equipped with different sensor modules, including Passive Infrared (PIR), magnetometer, accelerometer, etc. Indriya is built on a reliable active-USB infrastructure that employs special USB cables called active cables. The infrastructure provides a remote programming back-channel and it also supplies electric power to the sensor devices.

Indriya's design has the following three advantages:

- Indriya is designed to reduce the costs of both deployment and maintenance of a large-scale testbed. The average installation cost per node in Indriya is substantially less compared to the costs in Motelab [10] and Kansei [7] testbeds. When compared to the Twist testbed [9], which is also centered on an active-USB infrastructure, Indriya avoids the costs and difficulties involved in setting-up and maintaining a large number of single-board computers like NSLU2 [17]. Indriya has been in use for coming to 2 years by

T. Korakis et al. (Eds.): TridentCom 2011, LNICST 90, pp. 302–316, 2012.

over 100 users. The total maintenance cost so far is less than US$500 plus a recurring cost of 1-2 hours per week of time spent by one PhD student. Most of the cost is spent on replacing failed AC-to-DC adapters as these devices are not designed for long-term usage.
- As deployment of Indriya is over three floors, wireless connectivity among nodes is three-dimensional. This allows experimentation of protocols that are sensitive to placement and connectivity, such as geographical routing protocols.
- Unlike most of the existing testbeds which provide only a wireless infrastructure, Indriya is equipped with different types of sensor boards, thus allowing evaluation of WSN applications.

The second contribution of this paper is that we provide an extensive study of all 16 channels of IEEE 802.15.4. In order to leverage available channel-diversity, it is important to understand performance differences and correlations that exist among different channels. However, only limited analysis of these channels exist in the literature [6,3,5] and extensive measurements are lacking. In this paper, we aim to fill this gap by carrying measurements on Indriya.

The rest of the paper is organized as follows. In Section 2, we present a detailed description of Indriya. Section 3 compares Indriya against three WSN testbeds: Motelab [10], Kansei [7], and Twist [9]. We study non-overlapping channels of IEEE 802.15.4 in Section 4. Finally, we conclude in Section 5.

2 Indriya: A Large-Scale Sensor Network Testbed

Indriya is deployed across three different floors of our main School of Computing building (COM1). Figure 1 shows the deployment over 3 floors, covering spaces used for different purposes, including laboratories, tutorial rooms, seminar rooms, study areas, and walkways.[1] Figure 1 also shows the network connectivity at the default maximum transmission power of $0dBm$. The network has several inter-floor links providing 3D connectivity. Interestingly, most of these inter-floor links have packet reception ratio (PRR) close to 1.0 at $0dBm$ transmission power. The 3D connectivity remains even at lower transmit power levels. These observations are big improvement over link quality we observed when MICAz motes are used in our previous measurements. Figures 2(a) and (b) show photographs of Indriya as deployed in our building.

2.1 Motes

It is not feasible to use batteries to power motes for sustained and long-term experimentation, in particular for a large-scale testbed. On the other hand, wall-powering individual nodes incurs significant equipment and labour cost for installing power points and electric cables. In order to avoid these costs, we select USB-based motes so that they can be powered by our remote programming

[1] A real-time map can be found at `http://indriya.comp.nus.edu.sg/motelab/html/motes-info.php`

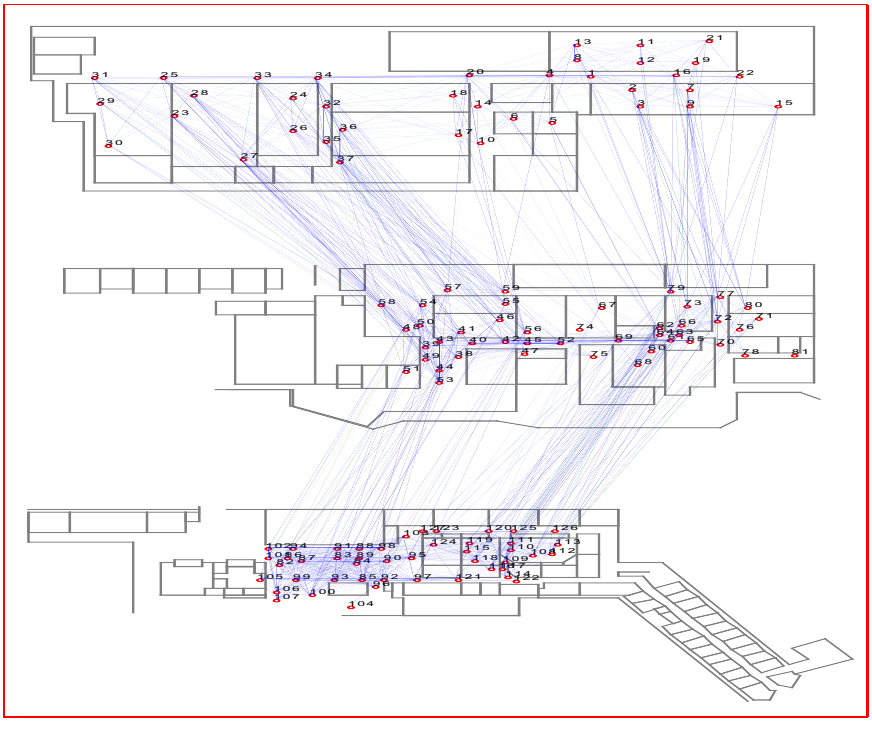
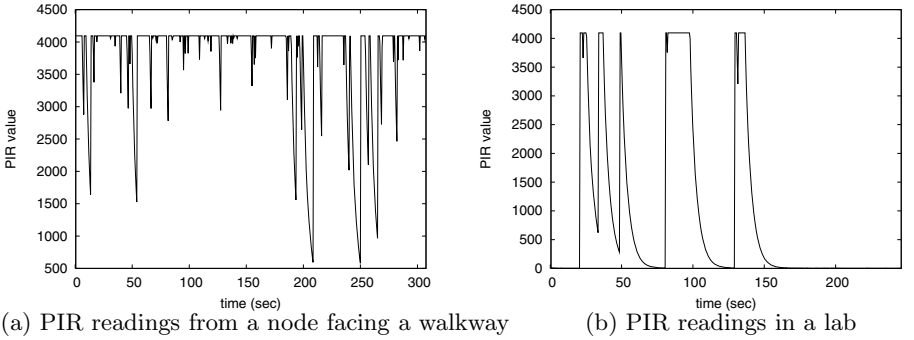


Fig. 1. Deployment of Indriya over three floors that provides 3D wireless connectivity

(a) (b)

Fig. 2. Couple of pictures of Indriya as deployed in our building

back-channel which is built over USB active cables. We choose TelosB devices, the most popularly used USB motes in the WSN community. TelosB has a TI-MSP430 microcontroller with 10KB of RAM used for storing program data only. The program itself (code) is stored in an internal flash of size 48KB. TelosB has Chipcon CC2420 radio transceiver operating at 2.4GHz with the indoor range of approximately 20 to 30 meters.



(a) PIR readings from a node facing a walkway (b) PIR readings in a lab

Fig. 3. Raw readings from two of the PIR sensors on Indriya

2.2 Sensors

More than 50% of the motes in Indriya are installed with different types of sensors, thus allowing experimentation of WSN applications. The main types of sensors deployed are WiEye, SBT80, and SBT30 sensor boards [13]. The WiEye board is commonly used to detect the presence of objects that emit invisible infrared rays, particularly, human beings. The SBT30 board includes visual light, acoustic and infrared sensors. In addition to these sensors, SBT80 also contains temperature, 2-axis acceleration and 2-axis magnetic sensors.

As an illustration, Figure 3(a) plots a WiEye's Passive Infrared (PIR) readings detecting the presence of human beings. The data was collected from a node in Indriya facing a busy walkway in front of a seminar room (SR1). The plot captures the fact that the walkway is busy with the curve staying at higher values during most of the time. Figure 3(b) plots PIR readings taken in a late evening from a node installed in one of our programming labs with a few students moving in-and-out of the lab.

2.3 USB Active Cables

While using a normal USB cable, the maximum distance between a host and a TelosB mote is limited to 5 meters. We overcome this limitation by employing USB active cables. An active cable is a special USB cable that incorporates electronics to sustain data signal so that five of them can be daisy-chained to cover a maximum distance of 25 meters. Our experience suggests that it is extremely important to use high-quality active cables. Otherwise, a host computer often loses USB connection with the sensor devices.

2.4 Design of a Back-Channel for Remote Programming

Existing testbed deployments either individually attach testbed nodes to single-board computers such as Stargate NetBridge [12] or use such computers as *super*

devices with each controlling a group of nodes. In both cases, the single-board computers are accessed over Ethernet. Although the later design is relatively cost-effective, the required number of *super devices* is still quite large. For example, Twist is based on such a design and it uses 46 NSLU2 [17] single-board computers to control 204 sensor nodes.

Unlike these deployments, Indriya does not employ single-board computers. Instead, it is based on an efficient cluster-based design with each cluster consisting of a single cluster-head that can accommodate up to 127 sensor devices. Moreover, an individual cluster can geographically span a circle of diameter of up to $50m$. We are able to cover 3 floors of our large building with only 6 clusters. Compared to existing testbeds, Indriya is the largest in terms of geographical size measuring $23500m^3$. Dimensions of other testbeds can be found in [2].

Figure 4 depicts the design of a cluster of Indriya. A Mac Mini PC [15] constitutes the cluster-head. This PC is a very small footprint computer (19.7cm by 19.7cm by 3.6cm) but as resourceful as a desktop PC. The motes are connected to the cluster-head using USB hubs and active cables. We employ Belkin's 7-port USB hubs and a mix of local-supplied and Aten's USB active cables [14]. Figure 4 also depicts a server to which cluster-heads are connected via Ethernet. The server manages the testbed and provides an user interface.

2.5 User Interface

Indriya uses Motelab's user interface software which provides web-based access to the testbed nodes. As design of Indriya is cluster-based that differs from that of Motelab, changes are required. Particularly, in the parts of the code that is responsible for communicating with testbed nodes. However, from the perspective of users, clusters in Indriya are transparent and the testbed is simply a wireless network of 127 nodes. The interface allows users to evaluate WSN systems implemented over TinyOS [11], a *de facto* standard operating system for WSNs. Users can upload, monitor, and control their jobs remotely and in real-time.

3 Comparison

In this section, we compare Indriya against three existing testbed deployments: Motelab, Kansei and Twist. Our comparison is mainly from the perspective of deployment cost and difficulties involved in setting-up and maintaining a large-scale testbed.

Motelab is deployed at the Harvard University. It is composed of 190 Tmote Sky sensor nodes with currently 85 of them being active (as of February 2011). Back-channel support is Ethernet-based with individual motes attached to separate Stargate NetBridge single-board computers. Similar to Motelab, devices in Kansei are also individually coupled to devices called Extreme Scale Stargate (XSS) [16] which are similar to NetBridge in configuration. Kansei is deployed at the Ohio State University (OSU) with 210 Extreme Scale Motes (XSMs) [16].

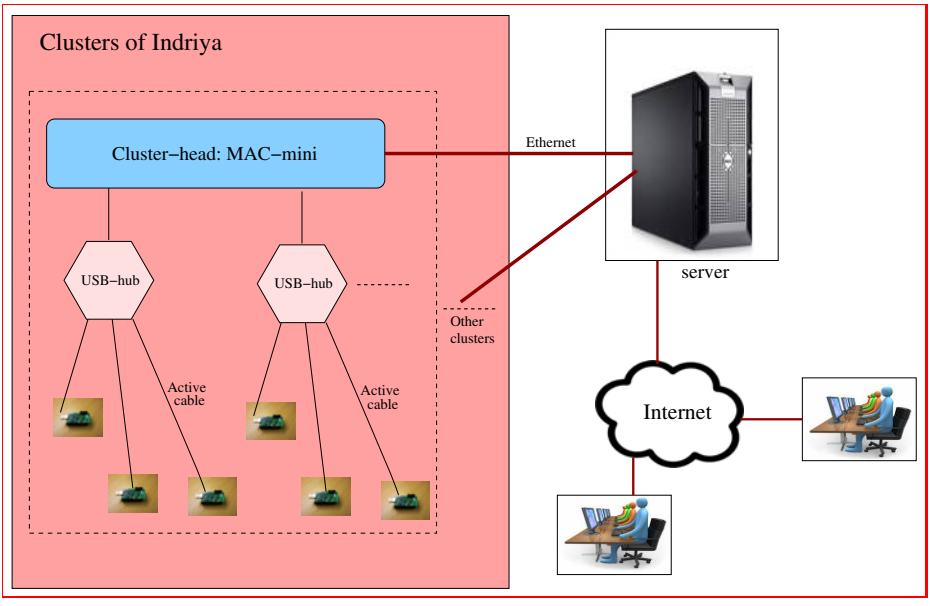


Fig. 4. The cluster-based design of Indriya

On contrary to Motelab and Kansei, Indriya incorporates an efficient cluster-based design eliminating the need for coupling individual motes to separate single-board computers. Instead, we use a MAC Mini capable of controlling 127 sensor devices. This significantly reduces the cost per node in addition to avoiding painstaking difficulties involved in setting-up and maintaining single-board computers. Moreover, single-board devices are typically wall-powered, thus incurring both labour and equipment costs for installing power points and electric cables. Whereas nodes in Indriya are powered over USB.

The average cost per node in Indriya is US$158, which is considerably less compared to the costs in Motelab and Kansei. The cost per node in Motelab and Kansei is almost the same as they incorporate similar designs and devices. The cost is approximately US$548 (NetBridge/XSS: US$449 + Tmote/XSM: US$99) plus the cost of providing wall-power and Ethernet connectivity. Furthermore, extending Indriya with additional motes is also comparatively costs less as each MAC Mini can accommodate up to 127 USB devices and currently we have an average of only 22 nodes per cluster.

We now compare Indriya against Twist which is also centered on an USB infrastructure. Twist uses 46 single-board and wall-powered NSLU2 computers to manage 204 testbed nodes. Whereas in Indriya, we use only 6 Mac Mini PCs to manage 127 nodes with each Mac Mini capable of accommodating more than 100 USB devices. Both testbeds span three floors of a building but with Indriya covering almost 3 times larger geographical volume (as per the dimensions provided in [2]).

In Twist, NSLU2 devices use the OpenSlug distribution of Linux *customized* specifically for the testbed usage [9]. On the other hand, Mac Mini devices in Indriya can use any desktop OS without any specific changes. Currently, we are using Ubuntu Linux running the 2.6.12 kernel without any modification. This also allows us to employ standard tools available for programming and managing sensor motes while NSLU2 like single-board devices demand significant changes or a new set of tools. Another important issue with NSLU2 devices is that they are extremely resource-constrained, particularly, limited flash memory. This demand requirements such as file system over network using NFS like protocols. In summary, all these issues significantly adds to the difficulties involved in setting-up and maintaining a large-scale testbed.

Since both Twist and Indriya incorporate similar USB backbones, the average cost per node in these deployments is more or less same. But, this is true only to the existing deployment instance of Twist, adopting its design is currently expensive. This is because of the fact that NSLU2 is a discontinued product (since 2008). A natural replacement to NSLU2 is either Stargate NetBridge or WRT600N. Stargate NetBridge is a Crossbow [18] modified and expensive version of the original NSLU2. On using either of these latest devices, cost per node in replicating Twist will exceed by at least US$70 per node compared to Indriya.

4 Non-overlapping Channels of CC2420: An Extensive Measurement

In this section, we present our experimental measurement of all 16 channels of IEEE 802.15.4 supported by CC2420. The channels are defined in the 2.4 GHz ISM band with each channel spreading in 2 MHz bandwidth. The inter-channel separation is 3 MHz. Nevertheless, not all channels are orthogonal to one another with the potential of close channel interference [5]. In this paper, we focus on understanding the performance differences and correlations that may exist among these channels. In particular, we are interested in results that are useful in deciding which channel to switch to when the need arises.

For our experimentation, we choose 44 nodes spanning an entire floor of Indriya. We choose these nodes as they form the largest floor-wise segment of Indriya. In addition, the deployment covers diverse environments ranging from open corridors to obstruction-filled partitions. Similar to the approach used in [8], we choose one node at a time and instruct it on the USB backbone to send 200 unicast packets on each of its neighbor-links. The packets are inter-spaced by 10ms and neighbor-links are selected one after the other. We repeat this procedure w.r.t to each of the 44 nodes. Furthermore, this entire process is repeated on every channel. We carry out our experiments at the default maximum transmission power of $0dBm$ and we consider links A→B and B→A as two different links.

The rest of this section is organized into four subsections. Section 4.1 focuses on both network-wide and link-wise performance differences. In Section 4.2, we

attempt to understand the correlations that may exist among different channels. We study the correlation between performance (PRR) and signal strength (RSSI) on every channel in Section 4.3. In Section 4.4, we analyze whether the short-time stability of RSSI observed in [4] can be generalized across all 16 channels.

4.1 Performance of Different Channels

Figure 5(a) plots the number of communication links (neighbor-links) observed on each of the 16 channels. With 44 nodes, the maximum number of links is 44x43 = 1892.

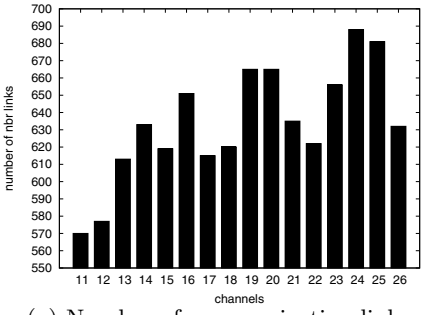
Performance differences are evident with the maximum difference of 118 links between channel-11 and 24. More interestingly, although the average number of links per channel is 633, only 392 links are common to all 16 channels. Figure 5(b) plots the average sum of the number of new links that are added and number of links that are removed when we switch the network from the given channel to another. The error bars represent standard deviation. We can observe that channel-21 has the minimum change of 128 links. The maximum change of 147 links can be seen on switching from channel-11 and this is expected as the least number of links is observed on channel-11. These two figures imply that the underlying communication topology on which routing protocols build their routes can vary significantly from channel to channel.

Figure 5(c) plots the probability of a link having PRR greater than certain value (complementary CDF (CCDF)). We can observe significant differences in the performance among the channels with gaps between individual curves widening as the value of PRR increases. The probability of a link being good (links with PRR > 0.9 [4]) varies across a wide range of 0.39 to 0.86. Channel-26 shows the best performance with 86% of its links performing *good*. This observation justifies the choice of using channel-26 as the default channel in TinyOS. The worst channel is 13, with the probability of having a good link being 0.39.

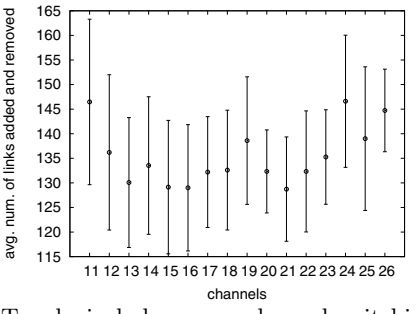
We now turn our attention to the performance differences on individual links. We consider all the 392 links that are common to all 16 channels. Figure 5(d) captures the variations by plotting the standard deviation (SD) of the reception ratios of 16 channels for each of the 392 links. The SD is computed for each link using the 16 samples of PRR, one for each channel. The links in the figure are sorted in increasing order of their mean PRR. The results exhibit substantial variations with the maximum standard deviation being close to 0.4. We argue that this result is positive as the variations indicate that it may be possible to find a good channel when the current channel is being degraded. We will investigate this issue in more detail in the following Section 4.2.

4.2 Correlations among Different Channels

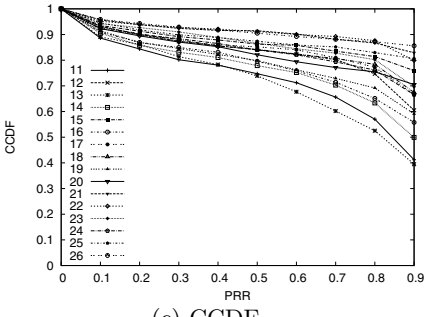
From the perspective of WSN routing and MAC protocols that exploit channel-diversity, it is important to quantify the performance of different channels relative



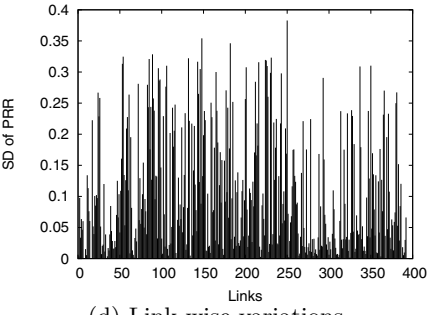
(a) Number of communication links



(b) Topological change on channel-switching



(c) CCDF



(d) Link-wise variations

Fig. 5. Network-wide and link-wise variations in the performance of different channels

to one another so that a WSN routing or MAC protocol will be able to exploit differences in channel performance appropriately. There are two possible reasons why an active sensor node may want to switch from one channel to another. First, the current channel used may be poor. Second, the current channel may be unavailable due to reservation by another node. In order to quantify the potential gain of exploiting channel diversity, we measure the following parameters.

First, we measure the likelihood that channel x has a higher PRR than channel y. Denote this value as g_{x-y}. Hence, g_{15-13} indicate the likelihood that a higher PRR can be gained by choosing the channel-15 over 13. Second, we measure that given two channels x and y, the likelihood that channel x has similar PRR as channel y. Denote this value as e_{x-y}. Hence, e_{15-13} indicate the likelihood that channels 15 and 13 have similar PRRs. We consider two PRR values as similar if their difference is within 0.001.

In order to measure g_{x-y} and e_{x-y} parameters, we use our 16 packet-traces collected on 44 nodes of Indriya with each trace corresponding to a specific channel. We compute probability values by comparing PRR values observed on different channels on each of the 392 communication links that are common to all channels. For example, if PRR on channel x compared to channel y is greater on each of the 392 links, then we note g_{x-y} as 1.0. We similarly compute e_{x-y} based on whether observed PRR values are similar.

Table 1. The G-matrix

	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
11	0.00	0.22	0.48	0.33	0.10	0.20	0.18	0.16	0.18	0.09	0.07	0.11	0.17	0.12	0.02	0.03
12	0.61	0.00	0.67	0.45	0.16	0.32	0.32	0.26	0.27	0.14	0.14	0.20	0.26	0.20	0.04	0.04
13	0.42	0.23	0.00	0.29	0.08	0.19	0.15	0.13	0.15	0.08	0.06	0.10	0.19	0.12	0.02	0.04
14	0.52	0.39	0.58	0.00	0.10	0.27	0.25	0.23	0.26	0.13	0.13	0.20	0.26	0.21	0.03	0.03
15	0.70	0.58	0.81	0.65	0.00	0.47	0.50	0.44	0.45	0.31	0.28	0.42	0.47	0.36	0.10	0.06
16	0.61	0.44	0.70	0.49	0.16	0.00	0.33	0.30	0.30	0.18	0.17	0.25	0.31	0.23	0.06	0.04
17	0.68	0.49	0.73	0.53	0.18	0.39	0.00	0.35	0.35	0.22	0.21	0.31	0.39	0.28	0.06	0.04
18	0.68	0.50	0.76	0.57	0.20	0.42	0.36	0.00	0.36	0.21	0.19	0.28	0.38	0.25	0.07	0.05
19	0.65	0.49	0.74	0.54	0.19	0.38	0.35	0.33	0.00	0.18	0.19	0.31	0.34	0.24	0.06	0.05
20	0.73	0.60	0.80	0.62	0.23	0.47	0.45	0.41	0.43	0.00	0.28	0.39	0.44	0.34	0.09	0.06
21	0.73	0.58	0.82	0.62	0.24	0.49	0.49	0.43	0.45	0.26	0.00	0.36	0.44	0.31	0.08	0.06
22	0.71	0.54	0.79	0.59	0.22	0.46	0.45	0.41	0.40	0.24	0.20	0.00	0.39	0.29	0.08	0.05
23	0.67	0.50	0.73	0.57	0.19	0.43	0.38	0.34	0.39	0.21	0.19	0.25	0.00	0.26	0.07	0.05
24	0.70	0.52	0.78	0.55	0.20	0.43	0.43	0.38	0.39	0.23	0.22	0.35	0.41	0.00	0.05	0.05
25	0.76	0.64	0.86	0.69	0.30	0.51	0.54	0.49	0.49	0.36	0.33	0.46	0.53	0.39	0.00	0.06
26	0.76	0.64	0.84	0.69	0.32	0.53	0.56	0.49	0.51	0.37	0.32	0.49	0.54	0.39	0.11	0.00

Table 2. The E-matrix

	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
11	1.00	0.17	0.10	0.16	0.20	0.19	0.14	0.17	0.17	0.18	0.20	0.18	0.17	0.18	0.22	0.21
12	0.17	1.00	0.10	0.16	0.26	0.23	0.19	0.24	0.24	0.26	0.28	0.26	0.24	0.28	0.32	0.32
13	0.10	0.10	1.00	0.14	0.11	0.10	0.11	0.11	0.11	0.12	0.12	0.11	0.08	0.10	0.12	0.12
14	0.16	0.16	0.14	1.00	0.25	0.24	0.22	0.20	0.20	0.25	0.25	0.20	0.17	0.24	0.29	0.29
15	0.20	0.26	0.11	0.25	1.00	0.36	0.32	0.37	0.36	0.46	0.47	0.36	0.34	0.43	0.61	0.61
16	0.19	0.23	0.10	0.24	0.36	1.00	0.28	0.28	0.32	0.35	0.34	0.29	0.27	0.34	0.42	0.43
17	0.14	0.19	0.11	0.22	0.32	0.28	1.00	0.29	0.30	0.33	0.30	0.24	0.23	0.30	0.40	0.40
18	0.17	0.24	0.11	0.20	0.37	0.28	0.29	1.00	0.31	0.38	0.38	0.32	0.27	0.37	0.44	0.45
19	0.17	0.24	0.11	0.20	0.36	0.32	0.30	0.31	1.00	0.39	0.36	0.30	0.27	0.37	0.45	0.44
20	0.18	0.26	0.12	0.25	0.46	0.35	0.33	0.38	0.39	1.00	0.46	0.36	0.35	0.43	0.55	0.57
21	0.20	0.28	0.12	0.25	0.47	0.34	0.30	0.38	0.36	0.46	1.00	0.44	0.37	0.47	0.59	0.61
22	0.18	0.26	0.11	0.20	0.36	0.29	0.24	0.32	0.30	0.36	0.44	1.00	0.36	0.36	0.45	0.46
23	0.17	0.24	0.08	0.17	0.34	0.27	0.23	0.27	0.27	0.35	0.37	0.36	1.00	0.34	0.39	0.41
24	0.18	0.28	0.10	0.24	0.43	0.34	0.30	0.37	0.37	0.43	0.47	0.36	0.34	1.00	0.56	0.56
25	0.22	0.32	0.12	0.29	0.61	0.42	0.40	0.44	0.45	0.55	0.59	0.45	0.39	0.56	1.00	0.83
26	0.21	0.32	0.12	0.29	0.61	0.43	0.40	0.45	0.44	0.57	0.61	0.46	0.41	0.56	0.83	1.00

The set of g_{x-y} values can be organized in the form of a matrix G, similarly for e_{x-y} using matrix E. For consistency, the rows and columns of these matrices are numbered from 11 to 26 with each row/column corresponding to a specific channel.

The G- and E-matrices are depicted in the Tables 1 and 2 respectively. As an illustration, we can see that by switching from channel 20 to 14, the likelihood of obtaining a better channel is 0.62, while the likelihood of doing worse is 0.13 (1 - 0.62 - 0.25).

We highlight important inferences based on the values in these matrices:

- Channels like 25 and 26 tend to perform better than the other channels in many cases and so one of them may be chosen as the default channel. However, when these channels cannot be utilized either due to interference or unavailability, looking at corresponding columns in E-matrix, it can be seen that many other channels can perform as good as channels 25 and 26 with high probabilities of up to 0.61.
- For most of the channels, there exists significant variations in the performance, such that it is likely to find a better channel if the existing channel cannot be utilized.
- Even channel-13, which tends to perform the worst, can serve as a reasonable substitute if the current channel is unavailable.

In conclusion, while the G- and E-matrices may need further refinement (for example, effects of location and temporal variations must be accounted), we believe the data from these two tables indicate that there is great potential in designing WSN routing and MAC algorithms based on dynamic channel-switching.

4.3 Correlation between Performance (PRR) and RSSI/LQI

In this section, we measure the correlation between performance and RSSI/LQI[2] over all 16 channels. Our aim is to verify whether the observations made in [8] on channel-11 are consistent on all 16 channels.

Figure 6 plots average values of PRR, RSSI and LQI observed on all 16 channels on each of the common 392 links. The links on x-axes are sorted in the increasing order of their mean PRR values. The figure shows a positive correlation of PRR generally being higher when RSSI/LQI are greater. Such a correlation indicates the potential for using RSSI and LQI as estimators of the link quality over any channel. Given such a potential, it would be interesting to analyze the correlation further. Kannan et al., have observed a strong correlation of PRR being at least 0.85 on the links where RSSI is greater than $-87dBm$ in [8]. Such an observation is useful to routing protocols particularly in instantaneous link quality assessment. However, observations in [8] is limited to channel-11. Consequently, we verify whether they can be generalized across different channels.

Figures 7(a) and (b) show the correlation of PRR and RSSI on channels 15 and 11 respectively. Each data point in the plots corresponds to a specific link and it is a pair of RSSI and PRR observed on the link. The error bars represent

[2] The term LQI stands for "Link Quality Indicator", it is a measure provided by the CC2420 transceiver and it indicates the quality of the channel during reception of a packet.

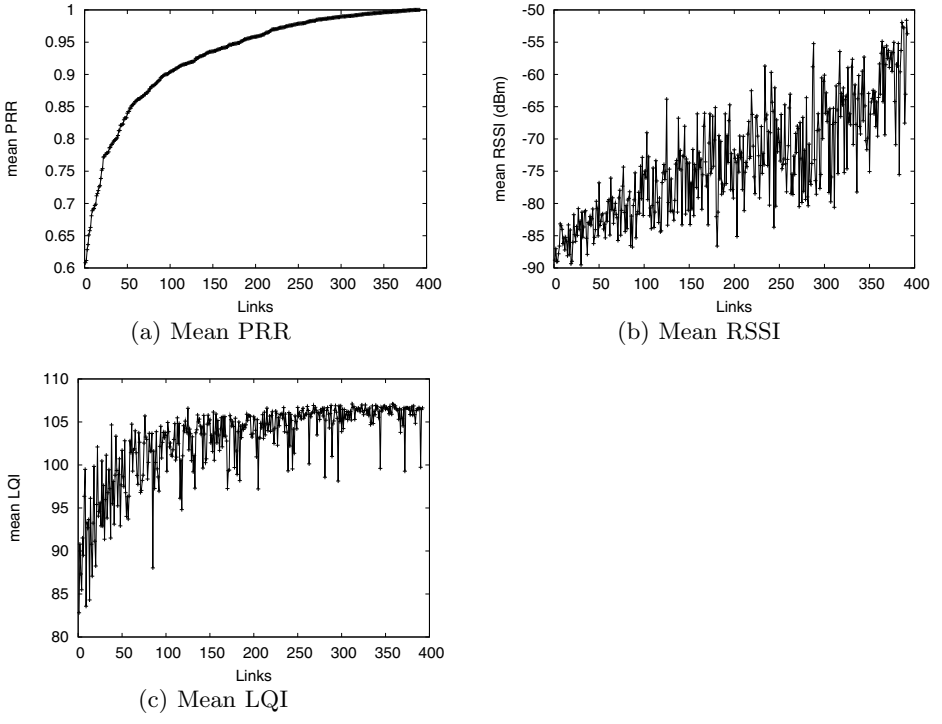


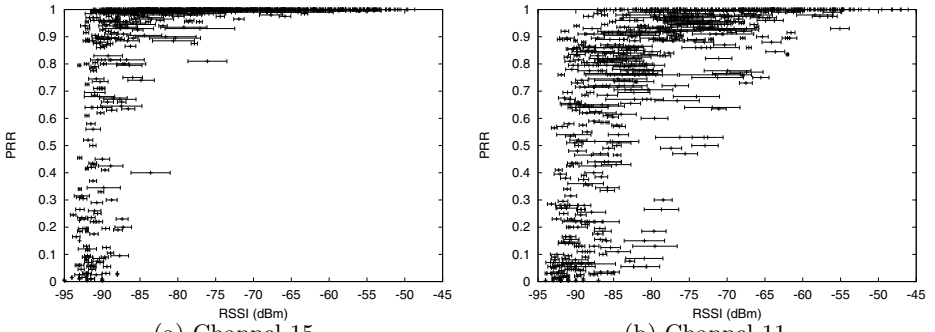
Fig. 6. Mean values of RSSI, LQI and PRR over 16 channels on 392 individual links

standard deviation of the RSSI values. On channel-15, a correlation of PRR being minimum 0.85 for RSSI greater than $-85dBm$ can be observed, whereas on channel-11, we can observe that a large number of links having PRR well below the 0.85 although their RSSI values are larger than $-85dBm$. Nevertheless, for RSSI greater than $-80dBm$, most of the links on channel-11 have PRR of at least 0.85.

We found that PRR and RSSI correlation on channels 12, $17-18$, and $20-26$ is consistent with the observations on channel-15. The correlation on channels 13, 14, 16 and 19 are similar to that on channel-11. Moreover, the correlation on channel-15 is also strongly consistent with the results on channel-11 in [8]. However, on channel-11 itself observations differ slightly with the lower bound being a higher value of $-80dBm$ whereas it is $-87dBm$ in [8]. This may be accounted for spatial differences.

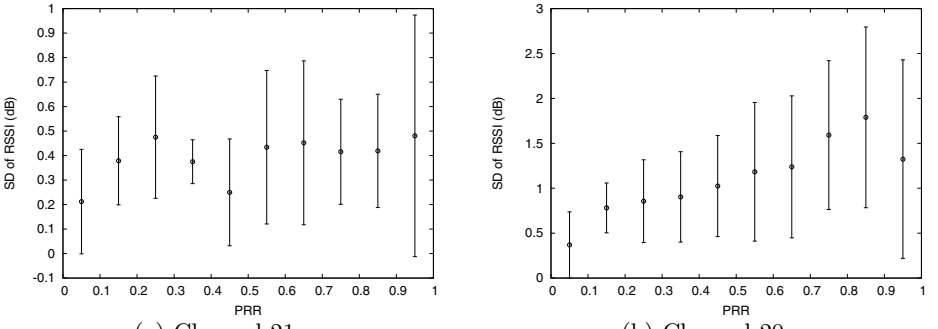
4.4 Short-Time Stability of RSSI

We now try to verify the generality of the results observed in [4], which shows that RSSI on channel-26 remains quite stable over short-time spans. Figures 8(a) and (b) plots the mean of the standard deviations of RSSI against the PRR on



(a) Channel-15 (b) Channel-11

Fig. 7. Correlation between PRR and RSSI on different channels



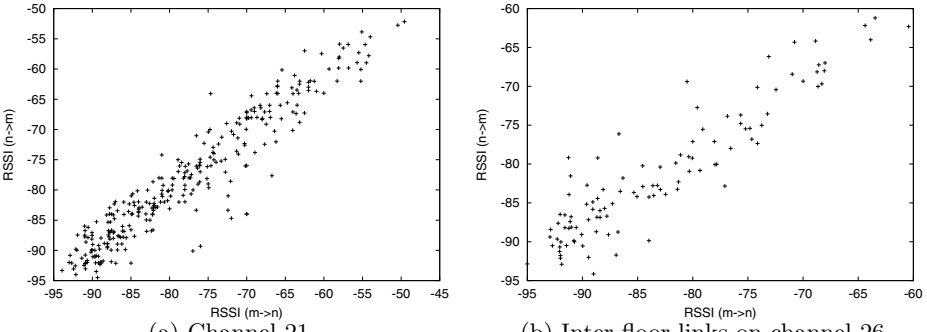
(a) Channel-21 (b) Channel-20

Fig. 8. Short-time stability of RSSI across different channels

channels 21 and 20 respectively. Each data point represents mean of the standard deviations of RSSI samples observed over a group of links. We categorize links into 10 different groups based on PRR. For example, a link with PRR 0.15 belongs to the group $(0.1, 0.2]$ $(10\% - 20\%)$, the PRR 0.35 is assigned to the bin $(0.3, 0.4]$ and so on.

Recall that on every link we send 200 unicast packets with inter-packet interval being $10ms$. This allows us to consider a duration of 2 seconds to constitute the short-time span. RSSI values measured every 10ms over a short duration of 2 seconds vary minimally on all channels. As an illustration, it is clear from Figure 8(a) that RSSI values are quite stable on channel-21 with mean of the standard deviation being well below $1dB$ at all considered PRR bins. The channels $22, 26$, and 25 performed close to the channel-21. However, the mean value ranges up to a maximum of 1.78 dB on other channels such as channel-20 as depicted in Figure 8(b).

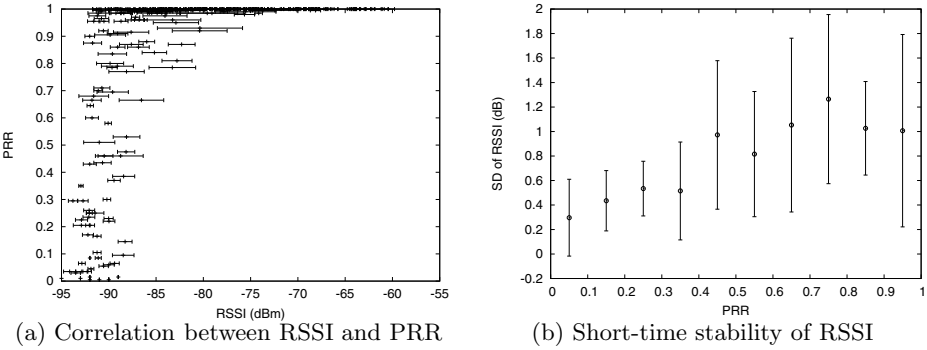
We now verify symmetry of CC2420 links on different channels. The authors in [8] observe that links are symmetric on channel-11. We found that such observation is strongly general across all channels and as an illustration, Figure 9(a) depicts symmetric nature of CC2420 links on channel-21.



(a) Channel-21 (b) Inter-floor links on channel-26

Fig. 9. Symmetric nature of 15.4 links

Finally, we present measurement results for 230 inter-floor links between the first and second floor nodes of Indriya. It would be interesting to see if the observations made so far apply on inter-floor links. For this measurement, we limit to channel-26. As shown in Figure 10(a), a strong correlation between PRR and RSSI exists even on inter-floor links. The short-time stability is depicted in Figure 10(b). Moreover, these links are also reasonably symmetric as depicted in the Figure 9(b).



(a) Correlation between RSSI and PRR (b) Short-time stability of RSSI

Fig. 10. Correlation and short-time stability on inter-floor links

5 Conclusion

The twofold contribution of this paper can be concluded as follow. By presenting Indriya, we demonstrate that large-scale WSN testbeds can be reliably built over USB infrastructures without employing single-board and wall-powered computers. Our infrastructure considerably reduces both deployment and maintenance costs. To date, Indriya is in operation for almost 2 years with minimal maintenance cost of less than US$500.

The second contribution of the paper is that it provides an extensive measurement and analysis of the non-overlapping channels of IEEE 802.15.4 supported by CC2420. We analyzed both network-wide and link-wise performance differences that exist among different channels. We presented data that capture performance of all 16 channels relative to one another. These results are of interest to the WSN community at large as they illustrate significant potential gain for algorithms and protocols based on dynamic channel-switching. Finally, we also demonstrated that the correlation and short-time stability of RSSI respectively observed in [8] and [4] can be generalized to all 16 channels.

Acknowledgement. We would like to thank the anonymous reviewers for their insightful comments and help in improving the quality of this paper. We are also very thankful to Chetan Ganjihal and Technical Services staff of our School of Computing for their active involvement in deploying Indriya. This material is based upon work partially supported under the grant #R-252-000-359-112.

References

1. Doddavenkatappa, M., Chan, M.C., Ananda, A.L.: Indriya: A Wireless Sensor Network Testbed at NUS, `http://indriya.comp.nus.edu.sg`
2. Omprakash, G., Rodrigo, F., Kyle, J., David, M., Philip, L.: Collection Tree Protocol. In: Proceedings of SenSys 2009 (November 2009)
3. Xing, G., et al.: Multi-channel Interference Measurement and Modeling in Low-Power Wireless Networks. In: Proceedings of 30th IEEE Real-Time Systems Symposium (December 2009)
4. Srinivasan, K., Maria, A., Saatvik, A., Philip, L.: The β-factor: Measuring Wireless Link Burstiness. In: Proceedings of ACM SenSys (November 2008)
5. Wu, Y., et al.: Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks. In: Proceedings of INFOCOM 2008 (2008)
6. Doherty, L., Lindsay, W., Simon, J.: Channel-Specific Wireless Sensor Network Path Analysis. In: Proceedings of Computer Communications and Networks (2007)
7. Arora, A., et al.: Kansei: A High-Fidelity Sensing Testbed. In: IEEE Internet Computing, Special Issue on Large-Scale Sensor Networks (March 2006)
8. Srinivasan, K., Levis, P.: RSSI is Under Appreciated. In: Proceedings of the Third Workshop on Embedded Networked Sensors (2006)
9. Handziski, V., et al.: A Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Network. In: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (2006)
10. Werner-Allen, Swieskowski, Welsh: MoteLab: a wireless sensor network testbed. In: Proceedings of Fourth International Symposium on Information Processing in Sensor Networks (2005)
11. TinyOS Project, `http://www.tinyos.net`
12. Memsic Corporation, `http://www.memsic.com/`
13. EasySen Corporation, `http://www.easysen.com/`
14. Aten Corporation, `http://www.aten.com/`
15. Mac Mini PC, Apple Corporation, `http://www.apple.com/macmini/`
16. Exscal Project, `http://cast.cse.ohio-state.edu/exscal/`
17. LinkSys NSLU2, `http://www.cisco.com/`
18. Crossbow Corporation, `http://www.xbow.com/`