Optimizing Packet Size In Radio Transmission
CS450 Project Report

Chuan Liu

December 13, 2008

Abstract

In this project, we investigate the impact of packet size on packet delivery performance of wireless sensor platform. We first design experiments to collect data and measure packet reception ratios (PRR) at different signal to noise ratios (SNR) for different packet sizes. We then derive size-dependent SNR/PRR curves based on the experimental measures. We augment TOSSIM with the size-dependent curves and evaluate the difference between result from simulation and actual testbed. Last we design a protocol for dynamic adjusting packet size to achieve higher PRR in changing environment.

1 Introduction

Accurate modelling radio transmission can help up simulate, analysis and understand the wireless network behavior. The insight into network behavior enables us to build protocols and systems that achieve higher efficiency. Often the models are based on simplified assumptions of variables in real world. We can improve the model by introducing new variable based on observation and experiments.

[Srinivasan *et al.*, 2008] shows there is good correlation between PRR and SNR. The observation makes SNR an ideal predictor of packet reception. Knowing the relationship between SNR and PRR can thus help us simulate radio transmission and design wireless protocol.

Based on the assumption of close relation between PRR and SNR, TOSSIM [Lee *et al.*, 2007] uses SNR/PRR curve with noise models to simulate packet delivery in radio transmission. When a simulated packet delivery event happens at a mote, the simulator calculates the delivery probability based on SNR and use the probability to determine whether the mote receives the packet.

However the work by [Lee *et al.*, 2007] uses a fixed packet size. In this project, we investigate how packet size can affect the PRR. By introducing this new variable, we make the model more accurately reflecting real world. We then improve the current simulation system with the new variable and design a new protocol that make optimal use of packet size to achieve better efficiency. In Section 2, we first describe the experiments and observations. At the end of the section, we introduce the new protocol based on the observation. In Section 3, we evaluate the performance of the protocol in the augmented simulation system. Section 4 concludes our work of the project.

2 Experiments

This section describes data collection, PRR estimation, the derived SNR/PRR curves and the newly proposed protocol.

2.1 Data Collection

We used two TelosB motes to collect data in our experiments. The radio chip attached to this platform is CC2420. CC2420 provides received signal strength indicator (RSSI) for each received packet. We can also

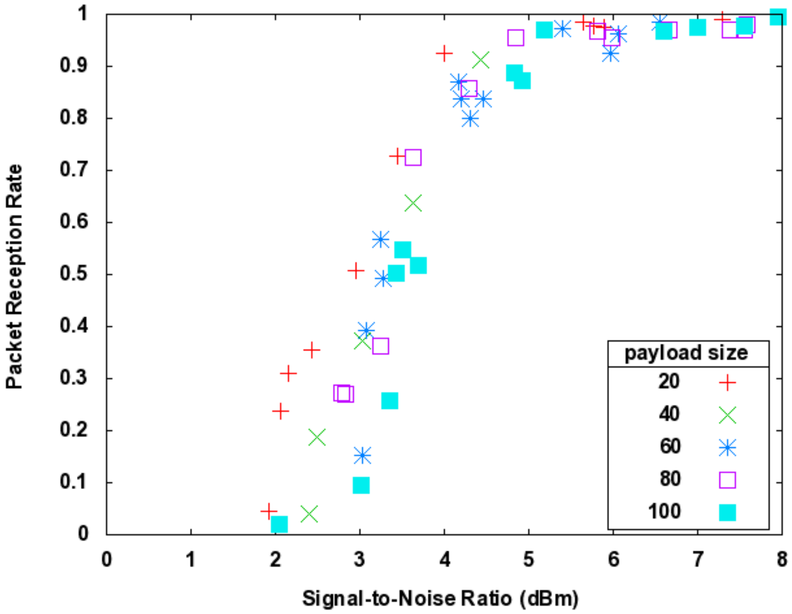

Figure 1: The SNR/PRR estimations for different packet sizes.

read the RSSI value anytime from the chip register. RSSI gives us a good estimation of signal strength. For received packet, RSSI is the sum of noise level and packet signal strength. When there is no traffic, RSSI measures the ambient RF energy.

The testbed is WiFi enabled indoor environment. The environment is under controlled. There is no moving object or other interference during experiments. Experiment shows noise is stable over time. The mean of the noise level is -95.31 dBm and the standard deviation is 0.882.

Among the two motes, one is sender and the other is receiver. Sender sends packet at a fixed interval and receiver samples noise after a shorter interval for every successfully received packet. In this way, we have the same number of noise samples and packet samples. The sending interval is 500ms and the interval between receiving packet and sampling noise is 250ms.

We use distance as attenuator to adjust RF transmission energy. Since the noise is stable in the testbed, we get different SNR by collecting data at difference distance. For a fixed distance, sender sends out more than 2000 packets (about 20 minutes). The receiver logs RSSI and sequence numbers of all successfully received packets as well as its own noise samples. We estimate the noise floor as the mean of all noise samples and signal strength as the mean of all packet samples. We calculate SNR from signal and noise estimation. By inferring number of total packets from logged (received) sequence numbers, we can calculate PRR. The pair of SNR/PRR values constitutes an estimation point. As with previous study, the SNR estimation is biased upwards, since packet with high RSSI is more likely to be received.

The maximum packet payload size is 114 bytes for TelosB. We carry out experiments for packet payload size (in bytes) of 20, 40, 60, 80 and 100. All data points are shown in Figure 1. For each packet size, we collect PRR/SNR estimations at about 20 different distances. We intentionally choose the distance to have enough points with PRR value between 0 and 1. As previous study shows, only a small change in SNR may result PRR dropping from 1 to 0. If we do not carefully choose the distances, the result may contains PRR only of 1s and 0s, in which situation it will be hard to find the curves.

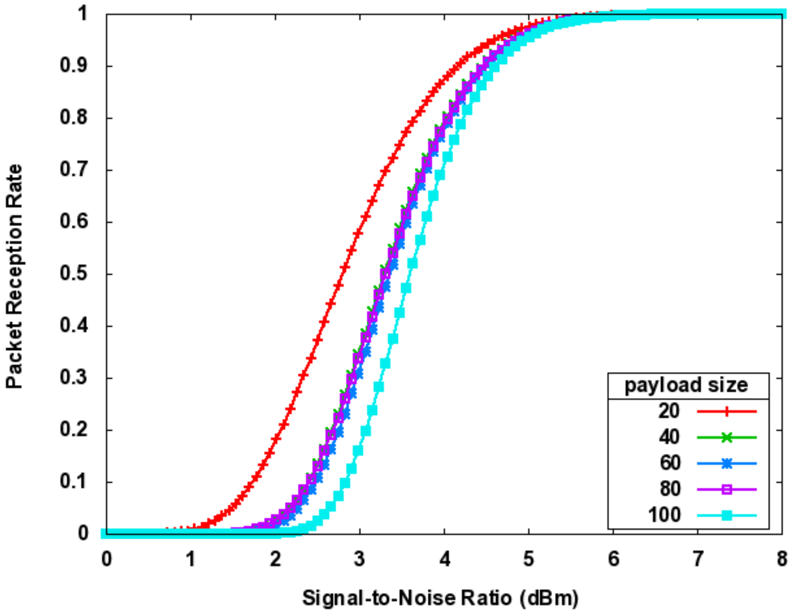

Figure 2: The SNR/PRR curves for different packet sizes.

2.2 SNR/PRR Curves

With reference to TOSSIM source code, we use the following nonlinear function to fit the data points in Figure 1.

$$\left(1-\frac{1}{2}\operatorname{erfc}\left(\beta_1 \times \sqrt{\frac{|10^{(r/10)}-\beta_2|}{2}}\right)\right)^{2f} \qquad \operatorname{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}dt$$

where r is SNR, f is packet size, erfc is complementary error function defined as on the above right, and β_1, β_2 are two parameters adjusting according to data. We find proper values of β_1 and β_2 with method of least squares. The curves that fit previous points are shown in Figure 2. The curves for packet sizes 40, 60 and 80 almost overlap and do not distinguish with each other clearly.

Since larger packet sizes reduce the packet header overhead, we also need to compare the efficiency. We define packet transmission efficiency as follows. The plot of efficiency over SNR for different packet sizes is shown in Figure 3. Still, parts of the curves overlap.

$$\frac{\text{Useful bits received}}{\text{Total bits transmitted}} = \frac{\text{Payload size}}{\text{Total packet size}} \times \text{PRR}$$

2.3 Observations

From Figure 2, we have the following observation on relation between packet size and SNR/PRR.

- The difference in PRR happens in the SNR range from 1 dB to 6 dB. For SNR ≤ 1 dB, PRR is 0 for all packet size; for SNR ≥ 6 dB, PRR is almost 1 for all packet size. The following observations are based on the assumption within this SNR range.
- For the same SNR, lower packet size has higher PRR.
- The difference in PRR increases as the SNR decreases. For example, PRR at SNR 5 dB is almost the same (about 0.98) for both 20 and 100 bytes payload size. On the other hand, at SNR 2.5 dB, PRR

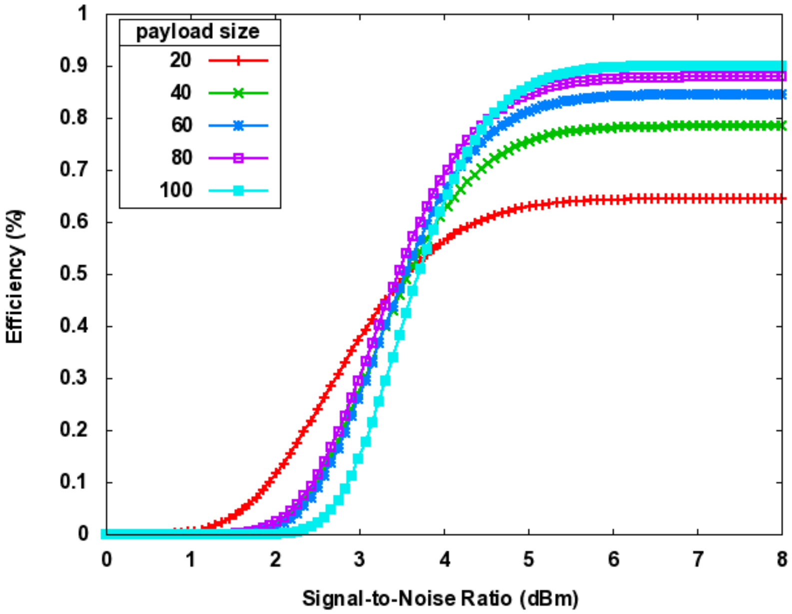

Figure 3: The efficiency over SNR for different packet sizes.

for 100 bytes payload size is almost 0, while the PRR for 20 bytes payload size at the same SNR is about 0.4.

From Figure 3, we observe the relation between efficiency and SNR: payload size 100 bytes has the best efficiency at SNR $\geq$ 4.5 dB; 80 bytes payload size has the best efficiency in SNR range 3.5 db $\sim$ 4.5 dB; and 20 payload size has best efficiency in lower SNR range.

2.4 Protocol Design

Based on the observation of the relation between efficiency and SNR, we propose the following protocol[1]. First the protocol uses packet payload sizes (in bytes) of 3 fixed options, i.e. 20, 80 and 100. The protocol is then divided into sender part and receiver part. The receiver send instruction on optimal packet size to sender based on its estimation of average SNR; the sender adjust its packet sizes based on the instruction from receiver as well as its own guess of environment condition by counting number of sequentially lost packets. We described the two parts separately in below and give the reason behind the protocol afterward.

Receiver

- For each packet received, calculate SNR by measuring its background noise and the message's RSSI. Update its average SNR with the new SNR measure. The average SNR is a exponential moving average calculated with the following formula where α is the smoothing factor.

$$\text{average SNR} = \alpha \cdot \text{SNR} + (1 - \alpha) \cdot \text{average SNR}$$

- Send back acknowledgement containing optimal packet size based on the average SNR.

---

[1] The protocol described here is slightly different with the one presented in class.

Sender

- Start with largest possible packet size.
- Reduce its packet size when a number of packets failed to delivery, i.e. no acknowledgement received. The number differs for different packet size. We empirically set the number as follows.
  - At 100 bytes payload size, if no acknowledgements received for 2 sequential packets, the sender downgrade to use 80 payload length.
  - At 80 bytes payload size, if no acknowledgements received for 4 sequential packets, the sender downgrade to use 20 payload length.
- Upon receiving the acknowledgement, sender increases its packet size to the proper value as indicated by receiver in acknowledgement.

First, we explain the sender's choice of number of failed delivery for adjusting packet size. Assume acknowledgement follows the same SNR/PRR relation. Suppose PRR $= p$ for data packet and PRR $= q$ for acknowledgement. The two are different since acknowledgement packet is smaller. Acknowledgement has a larger chance to fail, i.e. $(1-p)+p(1-q)$, than the data packet, i.e. $1-p$. For payload length 100 and SNR 4.5 dB, $p \approx 0.9$ and $q \approx 0.9$. The probability of not receiving 2 sequential acknowledgement is 0.0361. For payload length 80 and payload length 3.5 dB, $p \approx 0.5$ and $q \approx 0.8$. The probability of not receiving 4 sequential acknowledgements is 0.13. Both probabilities are small enough to make us believe the situation must be worse for that to happen.

Second, the low probability of receiving acknowledgement is also the reason the sender uses two methods to adjust its packet size. Since under low SNR condition, the sender can hardly receive feedback from receiver. In this case, it can still adjust to smaller packet size to let its data packet have better pass through.

Then we explain why the protocol let receiver decide the proper packet size to use. In general the receiver gets more packets than sender and thus it can make better estimation of SNR. Better estimation of SNR leads to better guess of optimal packet size to use. Therefore in the proposed protocol, we let sender set its packet size according to receiver.

3 Evaluation

Section 2 presents experimental derived SNR/PRR curves for different packet sizes. We modify TOSSIM to make it use different curve for different packet size. We also implement the protocol and carry out experiments to evaluate the performance of the protocol by simulating it with the modified version of TOSSIM.

Lots of experiments can be designed to compare various aspects of the protocol. Because of the time limitation of the course project, we only perform a simple experiment to test packet reception. We discuss other possible experiments at end of the section.

To see whether the new protocol can achieve a higher data reception in long run, we compare it with other two baseline applications. One only sends fixed length packets of 20 bytes long and the other only sends data packet of 100 bytes long. We let the three applications (two baseline applications and one using the new protocol) send out 100 Kb data. We record all received data packets at receiver side. We use the noise model of Casino Lab testbed [2] to generate noise in the experiments. We also experiment with different signal strengths by adjusting the gain value of radio links in TOSSIM. The gain values used in the experiment are -90, -91, -92, -93, -94 and -95. In general, the lower gain value leads to a lower SNR when the noise is the same.

The result are shown in Figure 4. It can be observed in the graph the application using new protocol has the best or close to best data reception rate within every signal strength category. Though the new protocol not always has the best data reception rate, there is no significant difference when comparing to

---

[2] The choice is of no particular reason. There are two noise sample files in TOSSIM noise directory. Meyer Library testbed is one collected by [Lee *et al.*, 2007] and used for introduction purpose in TOSSIM tutorial. So we pick another one.

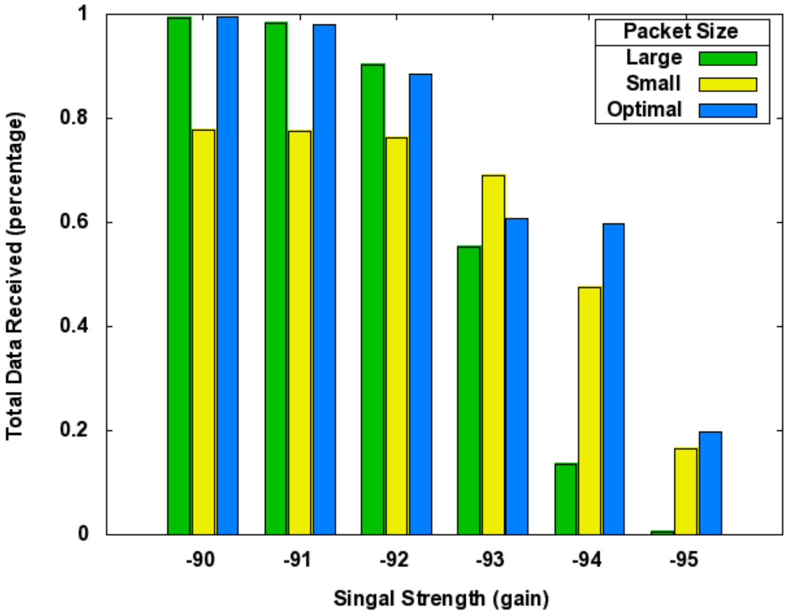

Figure 4: The comparison of data reception rate among the three applications.

best. This makes it beat the two baseline applications since both of them has worst data reception rate for some gain. For example at one end with good signal strength (gain = -90), the new protocol outperforms the baseline application with fixed small packet size by slightly over 20%; on the other end (gain = -95) while the application with large packet size almost completely fails to delivery any data, the new protocol can still achieve 20% data reception rate.

Though this experiment shows some merits of the new protocol, there are still other aspects of the protocol not tested. For example, we only tested the protocol with a single noise environment. Experiments with other noise environments may lead to other behaviour of the protocol. Also our experiment is carried out by simulation using TOSSIM. If we can experiment with actual testbed, our result will be more convincing.

4 Conclusion

In this project, we study the impact of packet size on SNR/PRR curve. Our study shows larger packet size has better reception rate with high SNR while smaller packet size performs better in more noisy environment. Based on our experiment result, we design a protocol that can adjust the packet size according to SNR measure. The protocol shows a more robust performance under different conditions in our evaluation.

Future directions of this work include: 1) collecting more SNR/PRR estimations to derive more accurate curves; 2) study the protocol with other noise environment/model and test it in real testbed.

The data and code can be downloaded at http://cs.jhu.edu/ chuan/cs450_project.tar.bz2.

References

[Lee *et al.*, 2007] HyungJune Lee, Alberto Cerpa, and Philip Levis. Improving wireless simulation through noise modeling. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, pages 21–30, 2007.

[Srinivasan *et al.*, 2008] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low power wireless. Technical Report SING-08-03, Stanford University, 2008.