

# Simulation of Transitional and Bursty Wireless Visual Sensor Networks

Mark Caswell  
Dept. of Computer and Information Science  
Indiana University Purdue University  
Indianapolis, USA  
caswell.mark@gmail.com

Yao Liang  
Dept. of Computer and Information Science  
Indiana University Purdue University  
Indianapolis, USA  
yaoliang@iupui.edu

**Abstract**—Communication links in low-power wireless visual sensor networks (WVSNs) are subject to short-term and long-term noise variations. These variations can cause a WVSN to exhibit prolonged or periodic transitional or bursty transmission performance. In this paper, we present our work on how to generate noise traces that simulate real-world transitional and bursty network behavior in TOSSIM. We develop a toolset called BurstyGen for TOSSIM which can facilitate WVSN protocol designers and application developers to better understand WVSN performance under these conditions. BurstyGen allows users to model both short time-scale and long time-scale variations in WVSN noise environments for the simulation and testing of WVSN system algorithms and protocols.

**Keywords**—wireless networks, link transitional, burstiness, WVSN simulation, TOSSIM, noise model.

## I. INTRODUCTION

Wireless visual sensor networks (WVSNs) is an emerging field combined multimedia communication with wireless sensor networks (WSNs), which plays an important role in visual surveillance applications including traffic monitoring, vehicle parking control, intrusion detection, and so on. In recently reported WVSN systems (e.g., MeshEye [1], FireFly [2], Cyclops [3], CITRIC [4], and WLSN [5]), they all adopt IEEE 802.15.4 protocol for energy-efficient low-power wireless communications, due to the power constraints in wireless motes. However, low-power wireless links in WVSNs and WSNs are noisy and unreliable. To measure the quality of a wireless link, Packet Reception Ratio (PRR) is often adopted. In general, link qualities based on PRR can be described and classified into three categories [3]: (1) good links (with  $PRR > 90\%$ ); (2) poor links (with  $PRR < 10\%$ ); and (3) intermediate, or transitional, links (with  $90\% \geq PRR \geq 10\%$ ). Wireless links can also have a wide range of time-varying PRR values [6, 7]. Packet loss on an individual link can be temporally independent or correlated, due to whether short time-scale noise variation or long time-scale noise variation (i.e. burstiness) exists in the communication environment respectively. Links attempting transmission in transition zone and bursty noise environments are of particular interest.

The quality of a transitional link is very dynamic due to quickly changing environmental noise and interference. Different percentages of transitional links were reported in previous work, ranging from 58% for Roofnet [6] to 14% for a lake testbed [8]. Even though transitional links may not dominate the WSN, they can have an important impact on the

overall performance of the WSN. The network must be able to quickly and efficiently adapt to the changing communication environment. Therefore, protocols and algorithms must be designed with these challenges in mind.

Bursty networks also present a unique challenge to the protocol designer. Small or large portions of a network may exhibit bursty behavior, with significant shifts in overall network performance over time. These shifts in performance can be especially taxing to network protocols and algorithms that are not designed for and tested against these possibilities. In order to maximize network performance, the impact of bursty network behavior should be carefully considered in WSN protocols and algorithms.

Simulation is an important tool widely used today to rigorously study and evaluate WSN and WVSN algorithms and protocols before they can be deployed (e.g., [9, 10, 11]). TOSSIM is a common simulator for TinyOS based WSNs [10, 11]. One interesting feature of the current TOSSIM 2.x is its environmental noise modeling capability [11], which simulates real-world noise patterns using user-provided noise traces and closest-fit pattern matching [12]. This feature enables simulation of the causes of packet loss rather than causing packet loss directly, making it possible to more accurately model real-world communication environments. On the other hand, this data-driven approach relies completely on the measured noise trace provided by the user and therefore may suffer from insufficient collection of the real environmental noise trace in which some important and characteristic real-world link burstiness is not captured. As a result, WVSN/WSN developers are likely not able to test and evaluate their protocols and systems against a wide variety of transitional and bursty network scenarios with TOSSIM 2.x.

This work aims to extend and complement TOSSIM 2.x environmental noise modeling [12] by providing a built-in ability to generate noise traces that simulate real-world noise variation including both transitional and bursty noise periods. We develop a toolset, called BurstyGen, that allows for this simulation. The toolset is relatively simple, being written in Python, and is flexible enough that it can be integrated and adapted in a number of ways. The toolset utilizes the built-in TOSSIM statistical model for radio noise and can be used with a single link or an entire wireless network simulation, therefore complementing the noise simulation functionality of current TOSSIM 2.x. It also does not require any changes to the TOSSIM source code, meaning that it is instantly usable by any

TOSSIM user. Through utilization of this tool, it is hoped that WSN protocol and system designers can advance their understanding and treatment of noise variations, especially transitional and bursty network behaviors more effectively and efficiently.

The remainder of this paper is organized as follows. Section II describes the concept of temporal noise variations. Section III presents our approach to temporal noise variation. Section IV presents our implementation of the TOSSIM BurstyGen toolset. Section V provides validation of our toolset. Finally, conclusions are given in Section VI.

## II. TEMPORAL VARIATIONS IN WSNs

When considering temporal noise variations in WSN communication environments, there are essentially two categories. The first, referred to as short time-scale variation, indicates differences in link quality over a short period of time. The second, referred to as long time-scale variation, indicates changes in link quality over relatively long periods of time.

### A. Short Time-Scale Variation – The Transition Zone

Short time-scale variations in link quality occur in all wireless networks. These variations are visible in both direct measurement of environmental noise and measurement of RSSI at a particular node. As shown by Zamolloa and Krishnamachari [13], these variations occur around a baseline noise floor value and are distributed normally about the baseline. This noise distribution can be represented as follows:

$$\eta_s(t) = \eta_{s0} + \gamma(0, \sigma_s), \quad (1)$$

where  $\eta_{s0}$  is average noise level and  $\gamma(0, \sigma_s)$  is a normally distributed random variable with mean 0 and standard deviation  $\sigma_s$ .

For links with large signal-to-noise ratios and noise floor variations that are small, the short time-scale variations in environmental noise do not impact the network performance. These links will always exhibit very good or perfect link connectivity. For links with signal strengths that are well below the baseline noise level, these short time-scale variations in noise levels will also not change the network performance. These links will always exhibit very poor or no connectivity.

However, for links where the signal-to-noise ratio is very near the minimum required for transmission, the short time-scale variations in environmental noise will cause changes in the ability of a particular link to transmit. These links will exhibit good link quality during some transmissions and poor link quality during other transmissions. These opposing transmission situations occur in close temporal proximity to each other, indicating that the quality of the link and the ability to successfully transmit across it cannot be determined from previous transmission successes or failures. Therefore, links in this range can be considered to exhibit sporadic behavior and are referred to as transitional links.

In practice, although a relatively low percentage of links are found to be transitional [1], the impact of the transition zone can be great if the links in the zone are key to the performance of the total network topology. The design and performance

measurement of network protocols and communication strategies must therefore consider the transition zone. Further, simulation tools must be able to properly simulate transitional links in order to provide a complete and accurate picture of WSN performance.

### B. Long Time-Scale Variation – Bursty Networks

Long time-scale noise variations in WSN communication environments are also of significant importance. A bursty network experiences periods of high noise or interference that may inhibit link quality for that particular time period. During a bursty period, a particular link may not be able to transmit at all. However, once the bursty period ends, the link may become fully operational again. That is, transmission successes and failures are strongly correlated. A transmission failure is often followed by another transmission failure, or a success by a success.

A bursty network can present a unique challenge to the WSN designer, as the protocols and transmission strategies utilized must be able to deal with links that drop out for extended periods of time. To facilitate this, the authors of [8] introduced the concept of link burstiness metric  $\beta$ , a scalar value which is calculated using conditional probability delivery functions (CPDF) [12] generated from empirical data. A link with independent deliveries has a  $\beta=0$ , whereas a link with perfect burstiness has a  $\beta=1$  (see [8] for more details).

## III. NOISE MODEL FOR NOISE VARIATION GENERATION

We present a toolset, called BurstyGen, which is able to generate and simulate both short time-scale and long time-scale network noise variations based on the following proposed noise model. The toolset allows the user to simulate transitional and bursty network links, thereby allowing for detailed testing of WSN protocols and algorithms in these environments.

### A. Simulation of Noise Floor Variation in TOSSIM

TOSSIM does not currently contain a built-in tool for directly simulating a noise level with short time-scale variations. The BurstyGen toolset fills this gap by introducing this capability. The toolset asks the user for a baseline noise level and a standard deviation for the normal distribution of the noise around the baseline. Using this information and Equation 1, the toolset generates a baseline noise trace with these characteristics. An example of the resulting noise distribution is shown in Figure 1.

The baseline noise level and standard deviation chosen, when compared to a particular link gain value, will determine the resulting PRR of that particular link. For example, if the standard deviation is chosen such that the noise levels exceed the level required for transmission at a particular link's assigned radio gain approximately 20% of the time, the likely resulting PRR is 0.8. The user can set these values to achieve the desired transmission success rate, thereby also influencing the number of transitional links.

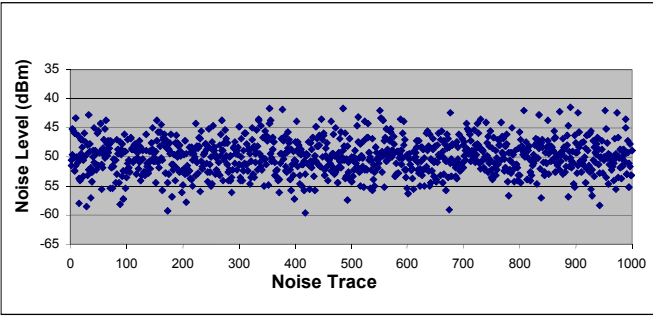


Fig. 1. Baseline noise floor variation generated by BurstyGen with an input average noise level of -50 dBm and a standard deviation of 3 dB.

### B. Simulation of Bursty Networks in TOSSIM

TOSSIM also does not currently contain a built-in tool for directly simulating a noise level with long time-scale variations. The BurstyGen toolset fills this gap by allowing the user to inputs values for a number of bursty network noise trace parameters.

A network burst is a period of high noise that lasts for a specified duration of time. This noise burst is characterized by the length of the burst, the average increase in noise over the noise floor during the burst, and the variation in noise level during the burst around the average.

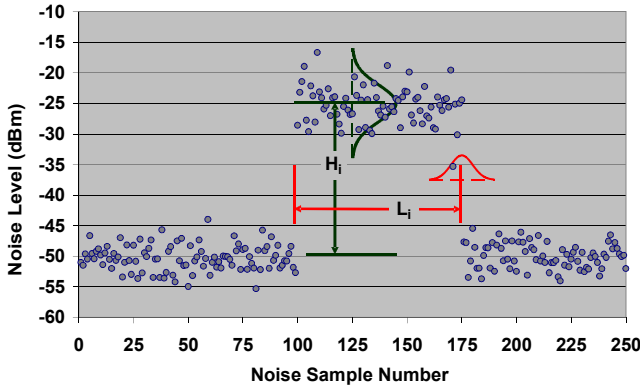


Fig. 2. Example distribution of a noise burst, with relevant dimensions labeled.

An example of a noise burst with normal distribution of the noise levels is shown in Figure 2. Here,  $H_i$  is the difference between the average level of noise during the noise burst and the average level of noise during the non-burst period.  $L_i$  is the duration of the particular noise burst.

Equation 2 shows the noise level during a particular burst period  $i$ . It is similar to Equation 1, but with a different average noise level,  $\eta_{b0,i}$ , and a different standard deviation,  $\sigma_b$ .

$$\eta_{b,i}(t) = \eta_{b0,i} + \gamma(0, \sigma_b) \quad (2)$$

The equation can be related back to Equation 1 through the use of  $H_i$ , as shown in Equation 3.

$$\eta_{b,i}(t) = \eta_{s0} + H_i + \gamma(0, \sigma_b) \quad (3)$$

While Equation 3 fully defines a single noise burst as related to the baseline noise level, there is additional variation that can be introduced from one noise burst to another. Both the average level of the increase in noise and the duration of the noise burst can also vary. The equations below define this variation in terms of a normal distribution about a mean. Note that  $\gamma_H$  stays fixed throughout a particular burst  $i$ .

$$\eta_{b,i}(t) = \eta_{s0} + H_0 + \gamma_H(0, \sigma_H) + \gamma(0, \sigma_b) \quad (4)$$

$$L(i) = L_0 + \gamma_L(0, \sigma_L) \quad (5)$$

Finally, the frequency of occurrence must be defined. Equation 6 defines the frequency with which a burst period occurs. It also occurs with variation about a mean value.

$$F(i) = F_0 + \gamma_F(0, \sigma_F) \quad (6)$$

These equations can be used to generate a full set of noise bursts within a particular TOSSIM noise trace.

## IV. BURSTYGEN IMPLEMENTATION

BurstyGen has been implemented utilizing a simple python script. In the following,  $t$  and  $r$  refer to the TOSSIM instance and TOSSIM radio instance, respectively. The nodes are contained in the *nodes* array. All nodes are assumed to have bi-directional links established between all other nodes. It is left to the user to adjust the BurstyGen toolset for other network topologies. The following sections detail the methodology for establishing the BurstyGen noise model.

### A. Establishing a Baseline Noise Trace

The user is given the option of inputting a baseline noise trace from an existing file, or having BurstyGen generate a noise trace from user provided parameter information. In the latter case, the user is asked for the average baseline noise level and the standard deviation of that noise level about the mean.

```
BaselineNoise = input ('Please input the baseline
                        noise level (average): ')

BaselineSigma = input ('Please input the standard
                        deviation of the baseline noise level: ')

noise[0] = BaselineNoise
for i in range(1,10000):
    noise.append(BaselineNoise+
                BaselineSigma*random_sigma())
```

### B. Establishing the Burst Period Metrics

Next, BurstyGen requests the metrics required to build the burst periods into the noise trace. The following metrics are requested from the user.

- Burst Period Offset – The change in noise level from the average baseline noise level to the average noise level during a burst.
- Burst Period Noise Standard Deviation – The standard deviation of the normal distribution of the noise level during a burst period.

- Burst Period Offset Standard Deviation – The standard deviation of the normal distribution of the average burst period offset from one burst period to the next.
- Burst Duration – The length, in noise trace inputs, of the average burst period.
- Burst Duration Standard Deviation – The standard deviation of the normal distribution of the burst duration. The value is stated in number of noise trace inputs.
- Burst Frequency – The frequency, in occurrences per 1000 noise trace inputs, of burst periods.
- Burst Frequency Standard Deviation – The standard deviation of the normal distribution of the burst frequency. The value is stated in occurrences per 1000 noise trace inputs.

By providing the information above, the user can fully define the characteristics of the burst periods experienced by the simulated network.

### C. Adding Burst Periods to the Noise Trace

Once the required information is obtained from the user, BurstyGen adds the burst periods onto the baseline noise trace.

```
i = 0
l = len(noise)
while ( i < l ):
    period = (int)(1000.0/(BurstFrequency +
        BurstFrequencySigma*random_sigma()))
    height = BaselineNoise+BurstOffset+
        BurstOffsetSigma*random_sigma()
    duration = (int)(BurstDuration+
        BurstDurationSigma*random_sigma())
    i = i + period - duration
    for j in range(0,duration):
        if(i+j<len(noise)):
            noise[i+j]=height+BurstSigma*random_sigma()
    i = i + duration
```

### D. Creating the TOSSIM Noise Model

Once the input noise trace model is created, the standard TOSSIM methodology for creating the statistical noise model can be used. The Python script is shown below.

```
for i in range(0,len(nodes)):
    nodes[i].createNoiseModel()
```

This model will then be used by TOSSIM during the execution of the application to determine whether transmissions are successful or not.

## V. BURSTYGEN DEMONSTRATION

We provide examples of short-term and long-term noise variations, resulting in both transitional and bursty link behavior, created using the BurstyGen toolset.

### A. Independent Network, Variations in PRR

In typical environments, the variation in the noise floor occurs as a normal distribution about a fixed mean. This is accomplished in BurstyGen by the specification of an average noise floor value and a standard deviation.

As postulated previously, it is precisely this short-term variation that leads to some links falling within the transition

zone, where the link quality is neither very good nor very bad, but rather intermittent (i.e., intermediate links). To illustrate that BurstyGen properly simulates this behavior, Figure 3 shows variations in PRR against decreasing radio gain values.

As the radio gain decreases toward the average noise floor, in this case fixed at -95 dBm, the PRR starts to degrade. Over the radio gain range of approximately -85 dBm to -95 dBm, the PRR decreases from 1.00 to 0.00. This 10 dB range represents the transition zone for this particular noise floor distribution. Links with radio gains in the range of -85 dBm to -95 dBm will have intermittent transmission successes.

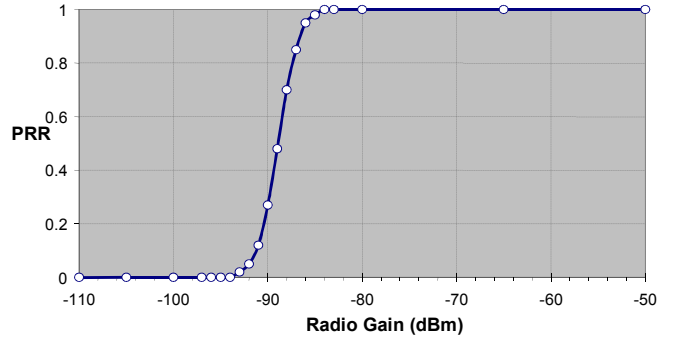


Fig. 3. Variation in PRR as the radio gain approaches the average noise floor (-95 dBm).

(Radio Gain = variable, Average Noise Floor = -95 dBm, Noise Floor Standard Deviation = 2 dB, No Burstiness Used)

### B. Independent Network, Variations in Noise Floor Scatter

Figure 4 shows how the transition zone varies for different values of the standard deviation of the noise floor values, in which the transition zone widens as the variation in the noise floor increases. For a standard deviation of 2 dB, the transition zone is approximately 10 dB wide. For a standard deviation of 5 dBm, the transition increases to approximately 20 dB wide. Finally, when the normal distribution is removed by setting the standard deviation to zero, the transition zone essentially disappears.

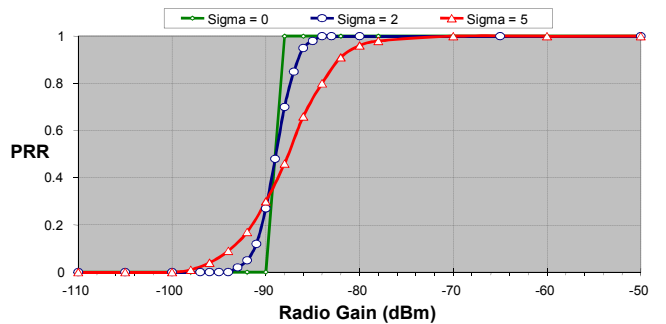


Fig. 4. Variation in PRR as the radio gain approaches the average noise floor (-95 dBm).

(Radio Gain = variable, Average Noise Floor = -95 dBm, Noise Floor Standard Deviation = variable, No Burstiness Used)

These results clearly show that BurstyGen can effectively model short time-scale variations in the noise floor. It makes

this capability available for users to test their improved WSN protocols and transmission strategies using TOSSIM.

### C. Bursty Networks, Example Transmission Stream

Figure 5 shows an example of transmission failures and successes generated through use of the BurstyGen toolset, with 0 and 1 representing a failed and successful transmission respectively. The figure shows periods where transmissions are usually successful and other periods where transmissions are usually unsuccessful. This is the behavior expected in a link with a high burstiness. Also, approximately half of the transmissions appear to be successful. This should result in a PRR of approximately 0.5, which matches the measured PRR on this link of 0.47.

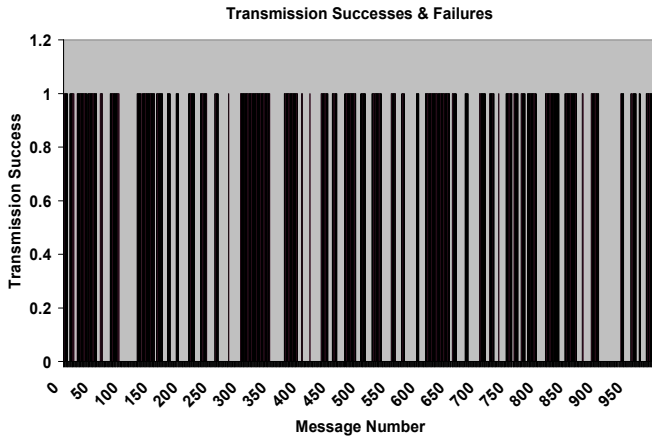


Fig. 5. Transmission attempt successes and failures.

(Radio Gain = -55 dBm, Average Noise Floor = -75 dBm, Burst Offset = 50 dBm, Burst Duration = 100 traces, Burst Frequency = 5 per 1000 traces, no normal variations used)

In order to measure the burstiness metric for this link, it is necessary to first plot the resulting message transmission data on a CPDF chart. The CPDF for this example, shown in Figure 6, clearly shows a strong correlation between successes and failures. Using this CPDF, the burstiness metric  $\beta$  is calculated to be 0.83 according to [8]. The high burstiness metric matches the expectations from the previous figure and thus proves the ability of BurstyGen to simulate bursty network links.

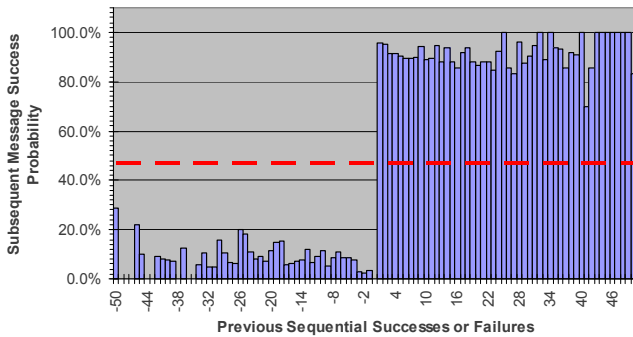


Fig. 6. Resulting CPDF for Example Transmission Scenario.

### D. Bursty Network, Variations in Burst Strength

For a bursty link, the noise burst is characterized by the duration, frequency, and strength of the burst. BurstyGen allows the user to specify these values and creates a noise trace based on these inputs.

Figure 5 shows variations in both PRR and the burstiness metric for a range of burst strength (offset) values. As can be seen, PRR is 1.0 for low burst offsets. This is expected, as the noise burst does not reach levels high enough to interfere with the transmission. However, as the burst offset approaches 35 dBm, the PRR starts to drop off. By 45 dBm, the PRR has reached a steady value of 0.50, since the burst periods only take up 50% of the noise trace (100 duration  $\times$  5 per 1000 = 500 burst traces per 1000 noise traces). Also expected is the width of the transition zone, 10 dB. Since the standard deviation used is the same as that used in Section V.A, the transition zone width is expected to be the same.

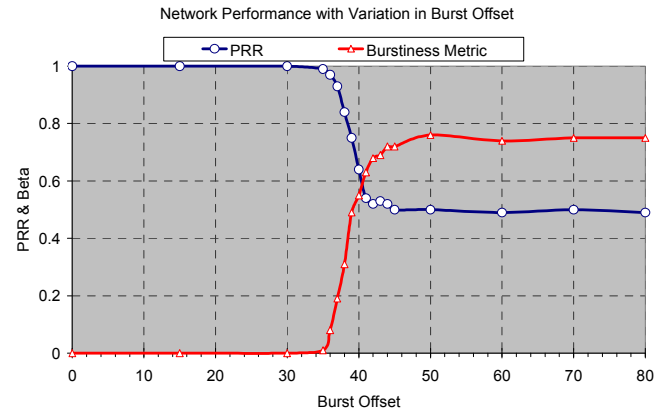


Fig. 7. Variation in PRR and Beta with changes in burst strength.

(Radio Gain = -50 dBm, Average Base Noise Floor = -95m dB, Noise Std. Dev. = 2 dB, Burst Duration = 100 traces, Burst Frequency = 5 per 1000 traces)

The burstiness metric curve also matches expectations. Burstiness starts at 0.0 and increases over the transition zone, eventually reaching a steady-state value of approximately 0.78. The link exhibits very strong burstiness as the burst offset reaches 45 dBm. The final value of the burstiness metric varies with the duration of the bursts. This is explored more fully in the next section.

### E. Bursty Network, Variations in Burst Duration

Longer bursts should result in highly correlated transmissions and high burstiness metrics. Shorter, more frequent bursts should result in less correlated transmissions and lower burstiness metrics. Figure 8 shows the variation in PRR and the burstiness metric over a range of burst durations.

As expected, the PRR linearly declines from 1.0 at a burst duration of 0 (no burst) to 0.0 at a burst duration of 200 (all burst), because the burst frequency is fixed. Therefore, the total time the network spends in a burst increases as the duration of each burst increases.

The burstiness metric curve also exhibits the expected behavior, up to a burst duration of 125 traces. The burstiness increases steadily as the burst duration increases. This is



because the periods of low and high performance become increasingly correlated. However, after burst duration of 125 traces, the burstiness metric begins to decrease. This decrease is due to the fact that very few transmissions are succeeding, leading to limited data available to calculate a valid burstiness metric. In other words, the burstiness metric can no longer be considered a useful measure in this range.

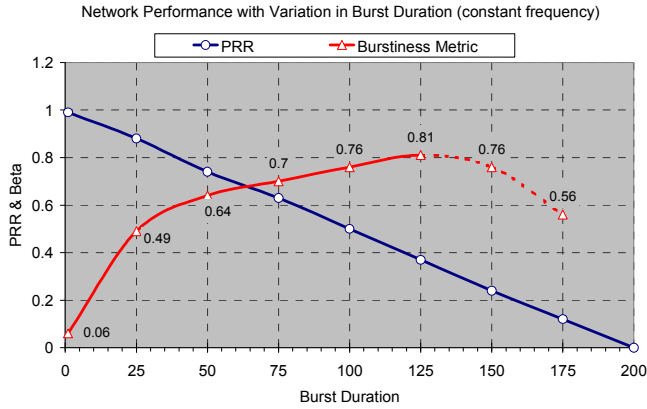


Fig. 8. Variation in PRR and Beta with changes in burst duration.

(Radio Gain = -50 dBm, Average Base Noise Floor = -95 dBm, Burst Offset = 50 dBm, Noise Std. Dev. = 2 dB, Burst Frequency = 5 per 1000 traces)

#### F. Bursty Network, Variations in Burst Frequency

Finally, we show how the burstiness metric for a network varies as the burst frequency changes. In this analysis, the total time spent by the network inside burst periods is kept constant. Therefore, as the burst frequency goes up, the burst duration is decreased to maintain a constant number of burst noise traces.

Figure 9 shows the variation in PRR and the burstiness metric as the burst frequency increases. Also shown is the required decrease in burst duration (yellow line, right y-axis). For this analysis, a radio gain of -50 dBm, a mean noise level of -95 dBm, and a burst offset of 50 dBm was used.

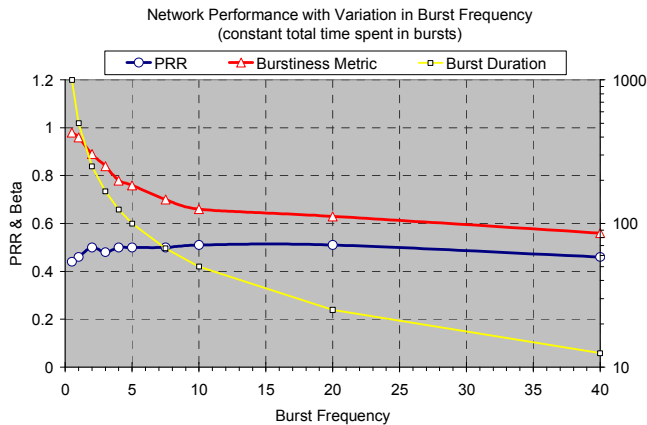


Fig. 9. Variation in PRR and Beta with changes in burst frequency.

(Radio Gain = -50 dBm, Average Base Noise Floor = -95 dBm, Burst Offset = 50 dBm, Noise Std. Dev. = 2 dB)

The PRR stays constant throughout the range of burst frequencies. This is because the total time spent in a noise burst is held constant. Therefore, the same number of transmission successes and failures is expected.

The burstiness metric also behaves as expected, starting at nearly 1.0 for a burst frequency of 1 per 1000 traces (highly correlated noise pattern) and decreasing as the burst frequency increases. Interestingly, the burstiness metric would not be expected to reach 0.00 until a burst frequency of nearly 500 per 1000. At this point, each burst would almost be a single noise trace long and the network would appear to behave independently to the CPDF. In fact, negative burstiness metrics might be expected if the network began to alternate between successive failed and successful transmissions.

## VI. CONCLUSIONS

The BurstyGen approach and toolset presented in this paper allow users of TOSSIM 2.x to accurately and efficiently model a network with both short time-scale and long time-scale variations in the noise environment. It takes advantage of the existing TOSSIM 2.x radio modeling approach and does not require any changes to the TOSSIM source code. The approach can be adapted for small or large networks and can be customized for a variety of situations. In particular, the simulation results have been clearly demonstrated to match what would be expected of these network scenarios. Using our BurstyGen toolset, the generated noise trace can be combined with the collected noise trace of *moderate* size, which can efficiently extend and complement current TOSSIM 2.x for WSN noise simulation to cover a wide variety of network burstiness scenarios. Thus, the presented toolset can be used to better explore the behavior of WWSN protocols and systems in environments that experience temporal variations in noise.

## REFERENCES

- [1] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "MeshEye: A Hybrid-Resolution Smart Camera Mote for Applications in Distributed Intelligent Surveillance," *Int. Symp. Information Processing in Sensor Networks*, pp. 360-369, April 2007.
- [2] A. Rowe, D. Goel, R. Rajkumar, "FireFly Mosaic: A Vision-Enabled Wireless Sensor Networking System," *Proc. 28th IEEE Int. Real-Time Systems Symposium*, pp. 459-468, Dec. 2007.
- [3] M. Rahimi, R. Baer, O. I. Iroez, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks," *Proc. 3rd ACM Int. Conf. Embedded Networked Sensor Systems*, pp. 192-204, Nov. 2005.
- [4] P. Chen, P. Ahammad, C. Boyer, S. Huang, L. Lin, E. Lobaton, et al. "CITRIC: A Low-Bandwidth Wireless Camera Network Platform," *Proc. ACM/IEEE Int. Conf. Distributed Smart Cameras*, pp. 1-10, Sept. 2008.
- [5] M. Chitnis, Y. Liang, J. Y. Zheng, P. Pagano, G. Lipari, "Wireless Line Sensor Network for Distributed Visual Surveillance". The 6th ACM Int. Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, pp. 71 - 78, Spain, Oct. 2009.
- [6] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *SIGCOMM*, pp. 121-132, 2004.
- [7] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," *SIGCOMM*, pp. 51-62, 2006.
- [8] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Lewis, "The  $\beta$ -factor: Measuring Wireless Link Burstiness," *Proc. 6th ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2008.

- [9] L. Girod, T. Stathopoulos, N. Ramanathan, J. Elson, D. Estrin, E. Osterweil, and T. Schoellhammer, "A system for simulation, emulation, and deployment of heterogeneous sensor networks," Proc. 2nd SenSys, pp. 201–213, New York, NY, 2004.
- [10] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Simulating large wireless sensor networks of tinyos motes," Proc. 1<sup>st</sup> SenSys, Los Angeles, CA, pp. 126-137, 2003.
- [11] TOSSIM 2.x. <http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM> (accessed on Sept. 26, 2019)
- [12] H.J. Lee, A. Cerpa, and P Levis, "Improving wireless simulation through noise modeling," Proc. 6<sup>th</sup> IPSN, Massachussets, pp. 21-30, 2007.
- [13] M.Z. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Trans. Sensor Netw.* Vol. 3, Issue 2, pp. 1-34, June 2007.