**Interprocess Communication in Distributed Wireless Sensor Network**

Luhan Cheng

Faculty of Information Technology

Monash University

lche0021@student.monash.edu

*Abstract*—Wireless Sensor Network (WSN) has been deployed in a wide range of mission critical task from bushfire detection to water quality monitoring. Such system can be abstracted by representing each sensor with one process in computer system. In this report, I conducted an experiment which utilize MPI to simulate the communication pattern in a 4 by 5 grid sensor network whereas all messages passing in network are encrypted. Threads level parallelism is therefore deployed by applying OpenMP to cryptographic operations. The experiment shows that (1) minimized communication overhead can be achieved by MPI (2) OpenMP has the potentiality to speedup encryption/decryption algorithm.

Keywords-component; Inter-process Communication, openmp, Message Passing Interface, Wireless Sensor Network

# Introduction

Inter-process Communication (IPC) defines a set of mechanism that support data sharing and communication among processes[1]. It is commonly used in wireless sensor network (WSN) as communication management method. There are varies methods that have been developed to satisfy the requirements for different applications. For example, TCP/UDP protocol is developed to facilitate web browsing service, message passing in concurrency model and shared memory scheme for all POSIX systems.

This report aims to simulate the communication pattern in a wireless sensor network and discover most efficient communication structure. In addition to the simulation in distributed environment, shared memory parallelization methods are to be discussed to explore the potentiality of accelerating cryptographic operation.

Target network is assumed to be a 2-dimensional cartesian grid, where each coordinate represents a sensor (process). An extra process is introduced to simulate base station in network. The simulation consists of multiple iteration, and for each iteration each sensor sends an encrypted message of random number to its neighbor. An encrypted event is to be reported to base station if the sensor receives at least 3 identical numbers from its neighbor.

There are three objectives being identified in this scenario. (1) Minimize the frequency of message passing between sensor and base station. (2) Minimize the communication among sensors. (3) Speedup the encryption of message with parallelization.

In term of event number being generated. Random seed is feed as the addition of current timestamp and rank number, which keep our simulation close to real world. Due to the fact that we only have very limited amount of resources, high probability of event occurring is necessary to generate meaningful outcomes of simulation, which is achieved by taking first few bits of random number, which limit the generated number falls in range with possible outcomes. The seed feed to random function is chosen as the current timestamp plus rank. I hypothesis that the probability of one particular event being generated is uniformly distributed across all events.

# Design Scheme for IPC

## Justify Chosen IPC

Message Passing Interface (MPI) is a library specification for message-passing in distributed system [2], which is also one of the most used message passing standard in both industry and academy [3]. MPI provides rich features of both point-to-point and collective communication. Compare to other standards, MPI provides several advantages. (1) MPI provides more portable libraries compare to older message passing standards (e.g. parallel virtual machine) [4]. (2) MPI is capable of delivering high performance on HPC system, and it is optimized on the hardware [4][5]. The implementation of MPI standard is vendor specific. One of the most used open source implementations is OpenMPI [6], which is applied in experiment to simulate sensor-sensor and sensor-base communication.

OpenMP stands for Open Multi-Processing. It is an application programming interface that support shared memory parallelization. Its behaviors are defined by a set compiler directives and runtime environment variables [7]. OpenMP follows fork-join model whereas at the start of program, there is only one master thread. A set of slave threads can be dynamically forked, and workload is distributed across slave processes. In the experiment, OpenMP is applied to speedup encryption and decryption process.

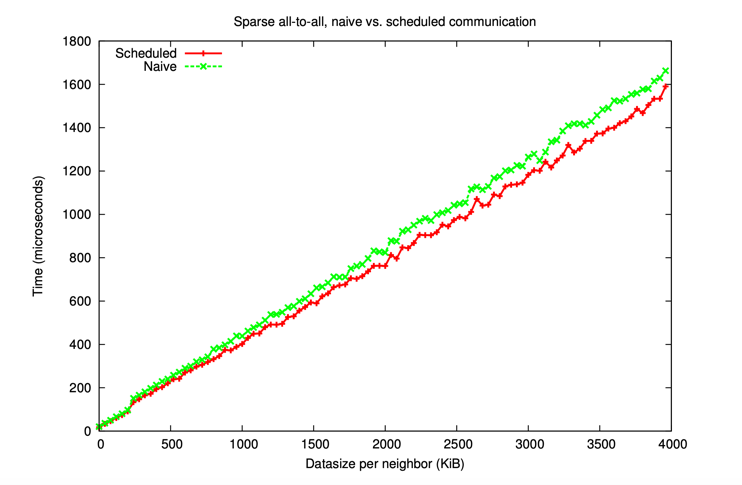
## Overall Topology

The network can be partitioned into two sets {S, B}. Set S consists of 20 sensors which is repartitioned into 4 by 5 cartesian grid. Set B only consist of base station. A random number on all sensors . The event detection criteria specify that an event is to be reported to base station if any sensor has at least 3 of its neighbors generate the same random number. Such criteria indicate strong local connectivity as for each sensor, it need to receive a message from all its neighbors and distribute a message to all neighbors. For the grid with size (|X|, |Y|), the total number of message passing is (.

MPI provides build-in topology constructor for cartesian structure in any dimension. The communicator among sensors is constructed from excluding the rank of base station from MPI\_COMM\_WORLD.

## local message exchange

Random number is then generated, and padding is added (details in section three) to construct a single message, which is to be exchanged by each pair of neighbors. Nearest neighbor communication has been discussed in multiple literatures. In particular, Torsten and Jesper suggested that the scheduled sparse all-to-all can outperform naïve all-to-all operation by using blocking MPI\_Sendrecv[8]. It has been shown that 10% speedup can be observed by applying scheduled all-to-all operation[8], which reduce communication contention on cartesian mesh grid and utilize bidirectional communication link [8]. The application of such operation is limited to case where global knowledge of problem is known, otherwise deadlock could be generated. In the case of WSN, scheduled all-to-all operation is the more optimized.



## global event report

After local message passing stage, each node will need to iterate over all possible event values to determine if an event is triggered. If event is activated, the node will report such event to base station by blocking send the MPI\_COMM\_WORLD. Besides random number generated, other event related information is also reported along, such as iteration number, timestamp and encryption/decryption time. As soon as simulation is completed, each sensor will send it's a summary which includes its basic information to base station. This message signals the completion of simulation.

At base station side, at the start of simulation, assume there are N-1 sensors. Base station will immediately spawn MPI\_Irecv along with an array *(RequestArray[0:2N])* filled with corresponding request. The first half of the array (event buffer) is used for receiving event message, the second half of the array (request array) is used for receiving the message that contains both completion signal and nodes summary of each nodes. The position in request array corresponds to the rank of base station is filled with MPI\_REQUEST\_NULL as null handler. During the simuation, if any request in region *RequestArray[0:N]* is received, the message is decrypted and stored as event, then the same request and MPI\_Irecv will be respawned. If any request in region *RequestArray[N:2N]* is received, it indicates the completion of corresponding rank. Both events receive request and completion signal receive request will be set to MPI\_REQUEST\_NULL. Those two types of message are distinguished by using different tags.

Local message exchange and global event report together forms an iteration. A single simulation consists of multiple such iterations. All events are stored in local memory of base station until being written to storage at the end of simulation. This implementation could potential cause high memory consumption on base rank in large scale simulation but comes with the advantage of more available analysis approaches on event reports.

# encryption

Advanced Encryption Standard (AES) is one of the most widely adopted symmetric encryption algorithms. It offers great security with little computational resource required [9]. In order to parallelize both encryption and decryption operation, AES in counter mode is chosen as the mode of operation. Note that the most computational costly operation in counter mode is the cipher block initialization instead of the XOR operation between plaintext and block [9]. Therefore, the speedup of parallelization may not be observed when message length is insufficient.

In realistic, the keys need to be distributed for security. IV need to be randomly generated in order to preserve the security of message. In the experiment, a few assumptions are made for simplicity, I assume that all the parties involved in communication share the same key and the same initialization vector (IV, or nonce in the context of this report). The implementation of algorithm is taken from WjCryptLib [10]. This implementation offers build-in OpenMP implementation for both encryption and decryption.

# Result Discussion

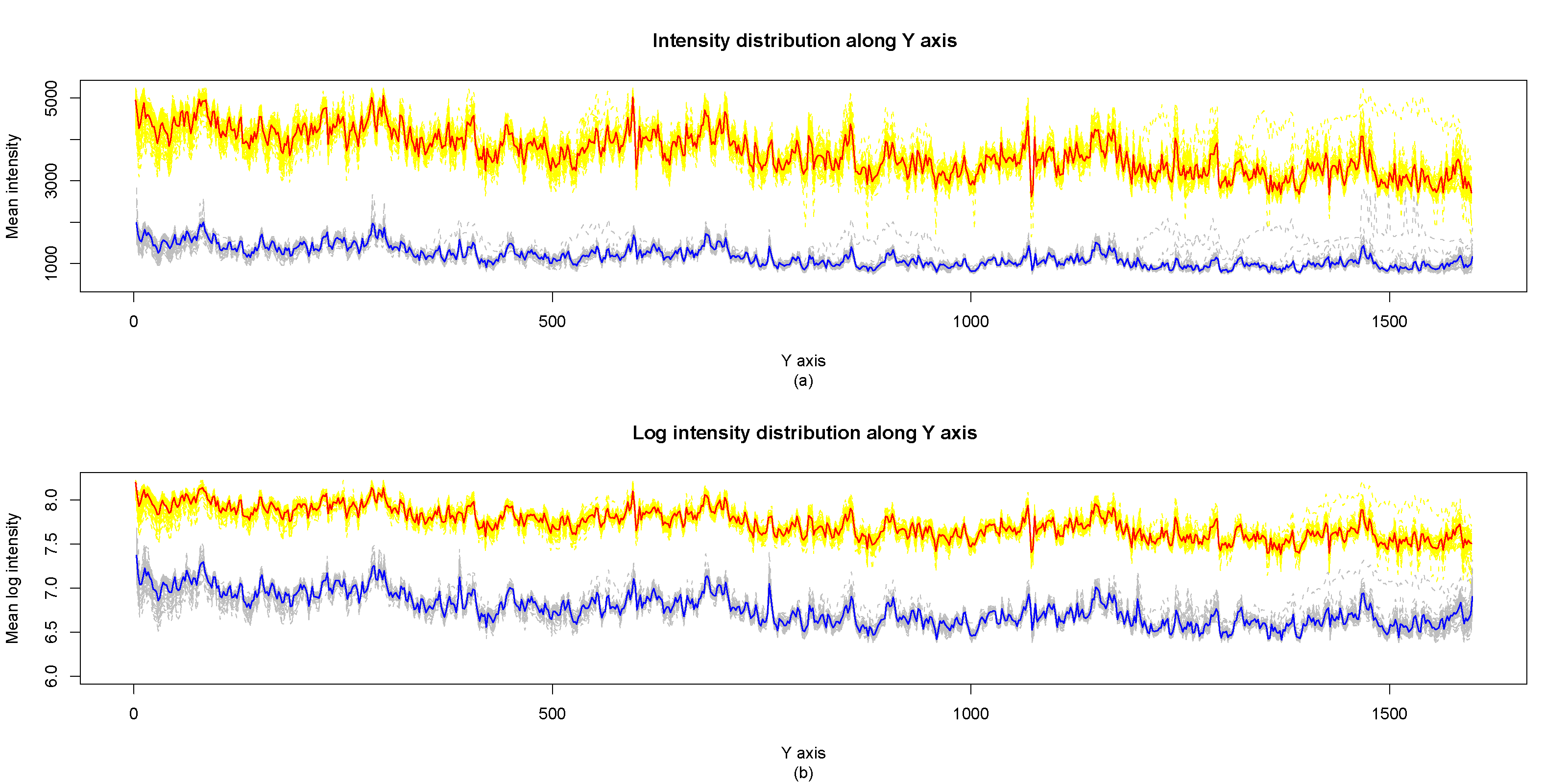
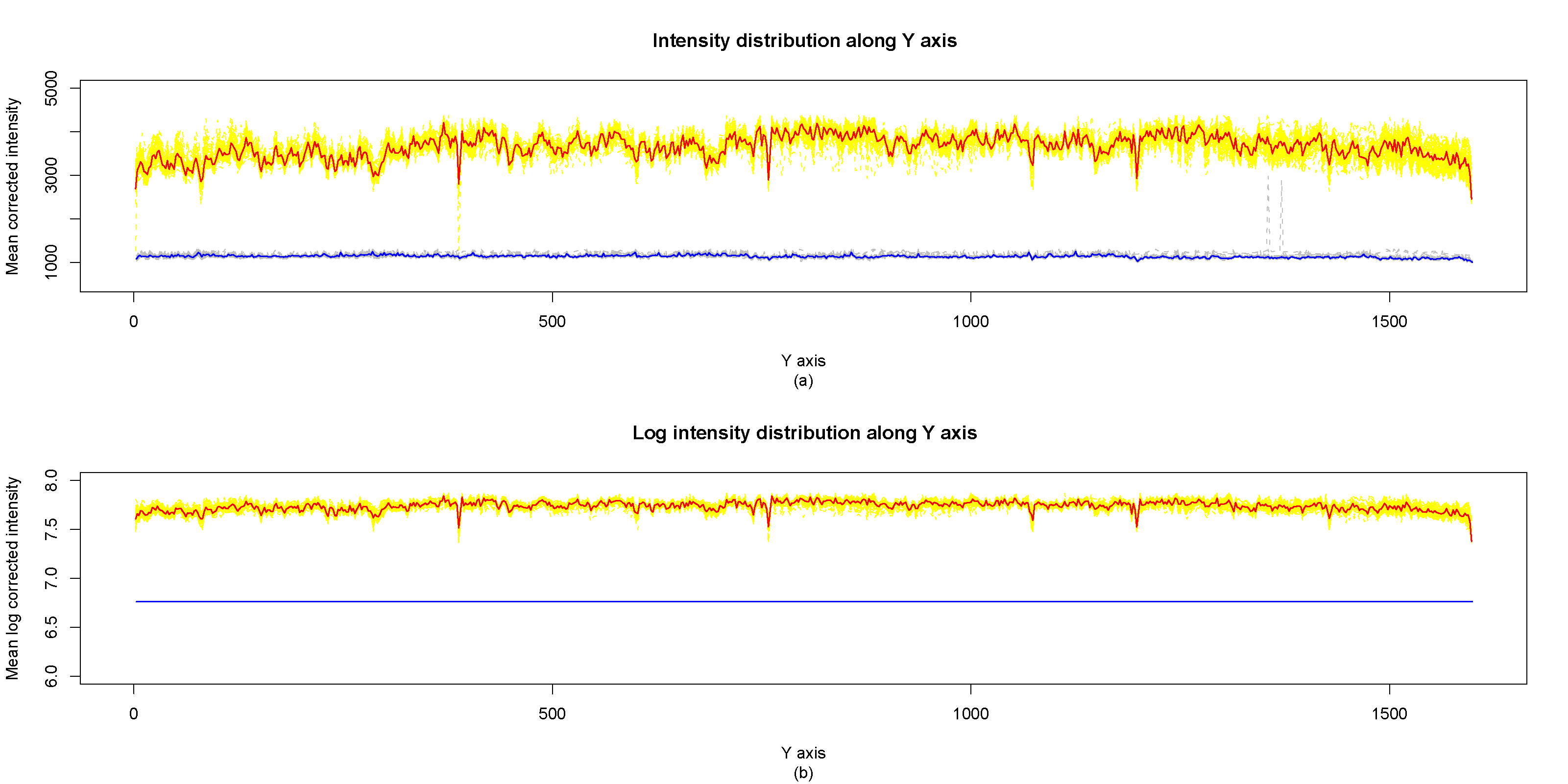
1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
8. Electronic Publication: Digital Object Identifiers (DOIs):

Article in a journal:

1. D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

Article in a conference proceedings:

1. H. Goto, Y. Hasegawa, and M. Tanaka, “Efficient Scheduling Focusing on the Duality of MPL Representatives,” Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

1. Example of a TWO-COLUMN figure caption: (a) this is the format for referencing parts of a figure.