Wireless Computing in all Area: A Mathematical Approach

Arun Kumar Singh[1*], Wera Dawa[1]

[1]School of MCS, PNG University of Technology, Lae, Papua New Guinea

*Corresponding author email: arun.singh@pnguot.ac.pg

Abstract: Use of wireless interface is one of the milestones of new-generation communication systems, which has an extensive sphere of application in such areas as IoT, mobile devices and for sensors. The research paper under consideration is aimed at investigating wireless computing from the perspective of mathematics and some of the aspects to be discussed are signal propagation, wireless channel characterization, system capacity and error control. We study the basic wireless communication and extrapolate mathematical models/theories and equations for analyzing the nature of the wireless systems, use in networking, and optimization. Wireless computing is one of the most valuable components of the communication in the present world where data transmission can be carried out among various networks without any physical linkages. Wireless computing systems are systems, which involve part of signal processing, part of network optimization and part of information theory; they are the systems that are based on the parameters of ideal systems. In this case, we talk on mathematical models used in the wireless communication channels. propagation models, path loss equations and interference management. Additionally, the paper emphasizes some of the main problems with the usage of graph theory, hints to the queuing theory to structure using adequate algorithm with the overall purpose of improving organizing the network assets as a way of scaling up the wireless networks. This theory goes into detail on most of the advanced concepts like the error correcting codes, modulation schemes and cryptography aspects required in secure communication in wireless computing environment. Therefore, this research aims at establishing a mathematical technique in design, analysis and optimization of wireless systems that we hope that it can be useful in shaping next generation wireless technologies such as 5G and IoT.

Keywords: Wireless Computing, Wireless Communication, Signal Propagation, Network Capacity, Error Correction, Mathematical Modeling

1. INTRODUCTION

Wireless computing refers to the ability to connect devices and exchange information over a wireless medium. the objective of wireless computing is based on the principles of electromagnetic waves communication, channel modeling and network optimization. Key aspects of wireless, such as signal transmission and data packet manipulation, require mathematical models for their prediction. Pertaining to the theoretical foundation of wireless communication in the context of wireless computing and including mathematical supporting forms this paper is also focused. Wireless computing has advanced into a standard element of present-day correspondence structures, merging billions of instruments and empowering an extraordinary trade of data without the demand to link physically. From basic communication through the mobile phones, wireless networks cover smart cities to the IoT making distant devices communicate seamlessly thus changing our everyday life. However, beneath this omnipresent technology is a network of mathematical formula that traditional wireless systems, whose deployment is pivotal to control efficiency, reliability and security in wireless communication [1-4].

This paper, Wireless Computing: Wireless Computing: A Mathematical Approach wants to address the issue of identifying the mathematical structures upon which this field is based upon. Wireless communication is fundamentally the exchange of signals across a medium, which in most cases is the air, a medium that poses definite issues such as signal fade, interference and noise. Such challenges are effectively handled through mathematical models, which assists engineers and scientists deploy better wireless performances. Elements of the physical approach to this mathematical problem include signal processing, network optimization, information

theory, and error correction. Wireless technologies are continuously evolving and with the emergence of 5G and continuous development of IoT that requires more complex and efficient solutions to network, it becomes apparent that new and more advanced mathematical tools are needed. The theory that underlies different areas of networks will be analyzed in this paper including propagation models, queuing theory, graph theory, and optimization algorithms to explain how all these theories can be implemented in order to increase the efficiency, security and scalability of a network. In addition, we will talk about how these mathematical concepts are useful in the modern world problem solving including bandwidth assignment, use of resources in wireless networks, as well as guaranteeing secure communication. By providing a mathematical perspective on wireless computing, this research aims to contribute to the continued advancement of wireless technology, helping pave the way for future innovations in areas like smart grids, autonomous vehicles, and pervasive computing [5].

2. SIGNAL PROPAGATION AND PATH LOSS MODELS

Signal propagation in wireless computing is described by electromagnetic wave propagation over free space. Received signal power is a function of the transmitted power, the distance between the transmitter and receiver, and the physical environment. Most commonly used model to describe signal attenuation is the **Free Space Path Loss (FSPL)** model [6] with program sample shown in Fig. 1 below.

2.1 Free Space Path Loss (FSPL)

The FSPL formula is given by:

FSPL(d,f)=(4πdfc)2\text{FSPL}(d, f) = \left(\frac{4\pi d f}{c}\right)^2FSPL(d,f)=(c4πdf)2

where:

- ddd = distance between transmitter and receiver (in meters),
- fff = signal frequency (in Hz),
- ccc = speed of light (approximately 3×1083 \times 10^83×108 m/s).

This formula represents the ratio of transmitted to received signal power in a free space environment. To express the FSPL in decibels (dB), the equation becomes:

FSPLdB(d,f)=20log⁡10(d)+20log⁡10(f)−147.55\text{FSPL}_{dB}(d, f) = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.55FSPLdB(d,f)=20log10(d)+20log10(f)−147.55

This relationship is essential for determining how much power will be lost as a wireless signal propagates over distance.

FSPL Formula:

FSPL (dB)=20⋅log⁡10(d)+20⋅log⁡10(f)+20⋅log⁡10(4πc)\text{FSPL (dB)} = 20 \cdot \log_{10}(d) + 20 \cdot \log_{10}(f) + 20 \cdot \log_{10}\left(\frac{4\pi}{c}\right)FSPL (dB)=20⋅log10(d)+20⋅log10(f)+20⋅log10(c4π)

Where:

- ddd is the distance (in meters).
- fff is the frequency (in Hz).
- ccc is the speed of light (3×108 m/s3 \times 10^8 \, \text{m/s}3×108m/s).

Python Code:

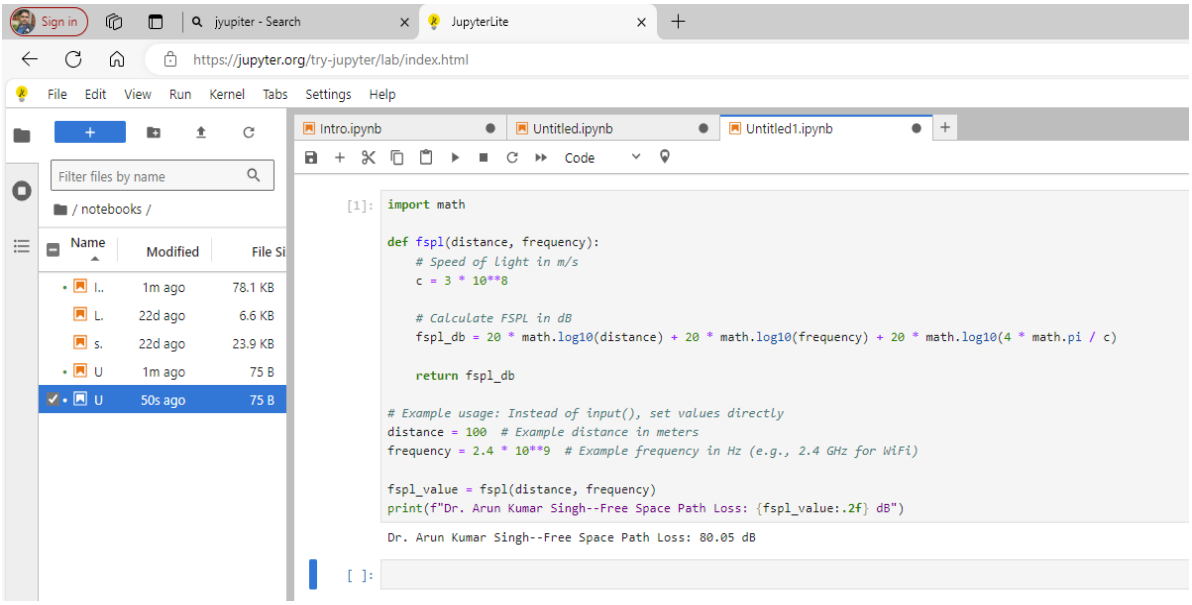

Fig. 1 Python Code for Free Space Path Loss (FSPL)

Explanation:

- **Inputs**: The user enters the distance (in meters) and frequency (in Hz).
- **FSPL Calculation**: The program computes FSPL using the formula.
- **Output**: The result is shown in decibels (dB).

This program helps determine the signal loss in wireless communication, which is crucial for network design and optimization [7].

3. CHANNEL MODELING

Communication through wireless is done on a physical medium, known as channel. Some of the factors which have an impact on the behavior of this channel are some noises, interference and fading. One of those commonly used models for wireless channels is Additive White Gaussian Noise (AWGN) channel with sample program shown in Fig. 2 below.

3.1 AWGN Channel

In an AWGN channel, the received signal y(t)y(t)y(t) is modeled as:

y(t)=x(t)+n(t)y(t) = x(t) + n(t)y(t)=x(t)+n(t)

where:

- x(t)x(t)x(t) = transmitted signal,
- n(t)n(t)n(t) = noise, which is modeled as Gaussian noise with a mean of zero and variance N0N_0N0.

The power of the transmitted signal is often compared to the noise power to determine the signal-to-noise ratio (SNR), which is defined as:

SNR=PsN0\text{SNR} = \frac{P_s}{N_0}SNR=N0Ps

where PsP_sPs represents the signal power and N0N_0N0 is the noise power spectral density. This relationship governs how much interference or noise affects the quality of wireless communication.

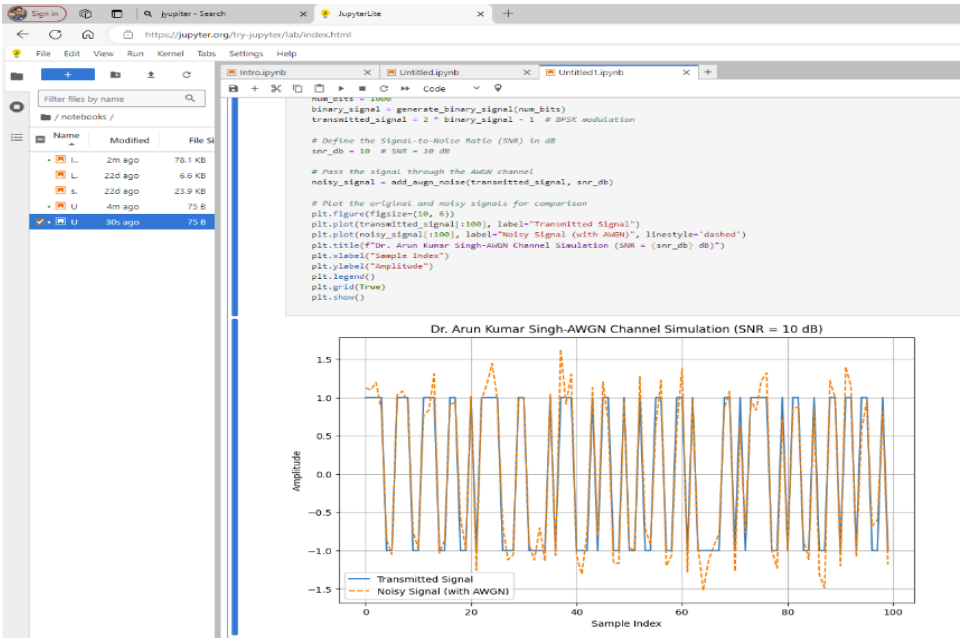

Fig. 2 AWGN Channel

Explanation:

1. **Signal Generation**: We generate a random binary signal of 0s and 1s. This signal is then modulated using Binary Phase Shift Keying (BPSK), where 1 maps to +1 and 0 maps to -1.
2. **AWGN Noise Addition**:
   - The signal is passed through the add_awgn_noise() function, which adds Gaussian noise based on the given Signal-to-Noise Ratio (SNR).
   - The SNR is defined in dB, and we calculate the noise power relative to the signal power using the given SNR.
3. **Plotting**: We plot the first 100 samples of both the transmitted and noisy signals to visualize the effect of the AWGN.

Parameters:

- **SNR (dB)**: Determines the amount of noise added to the signal. Higher SNR means less noise.
- **Signal Length**: You can adjust num_bits to simulate longer signals.

This program helps simulate and visualize the effect of an AWGN channel on a simple communication signal. You can experiment by changing the snr_db value to see how different noise levels affect the signal quality [8].

3.2 Rayleigh Fading

Rayleigh fading occurs when the wireless signal undergoes multiple reflections before reaching the receiver. The signal is modeled as a random variable with a Rayleigh distribution:

p(r)=rσ2e−r22σ2,r≥0p(r) = \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, \quad r \geq 0p(r)=σ2re−2σ2r2,r≥0

where:

- rrr = magnitude of the received signal,
- σ2\sigma^2σ2 = variance.

Rayleigh fading is particularly important in urban environments, where signals encounter many obstacles.

Python Program to Simulate Rayleigh Fading

Below in Fig. 3 is a Python program that simulates Rayleigh fading and visualizes the results using Matplotlib. This program generates a random Rayleigh fading channel and displays the fading envelope over time

Python Code:

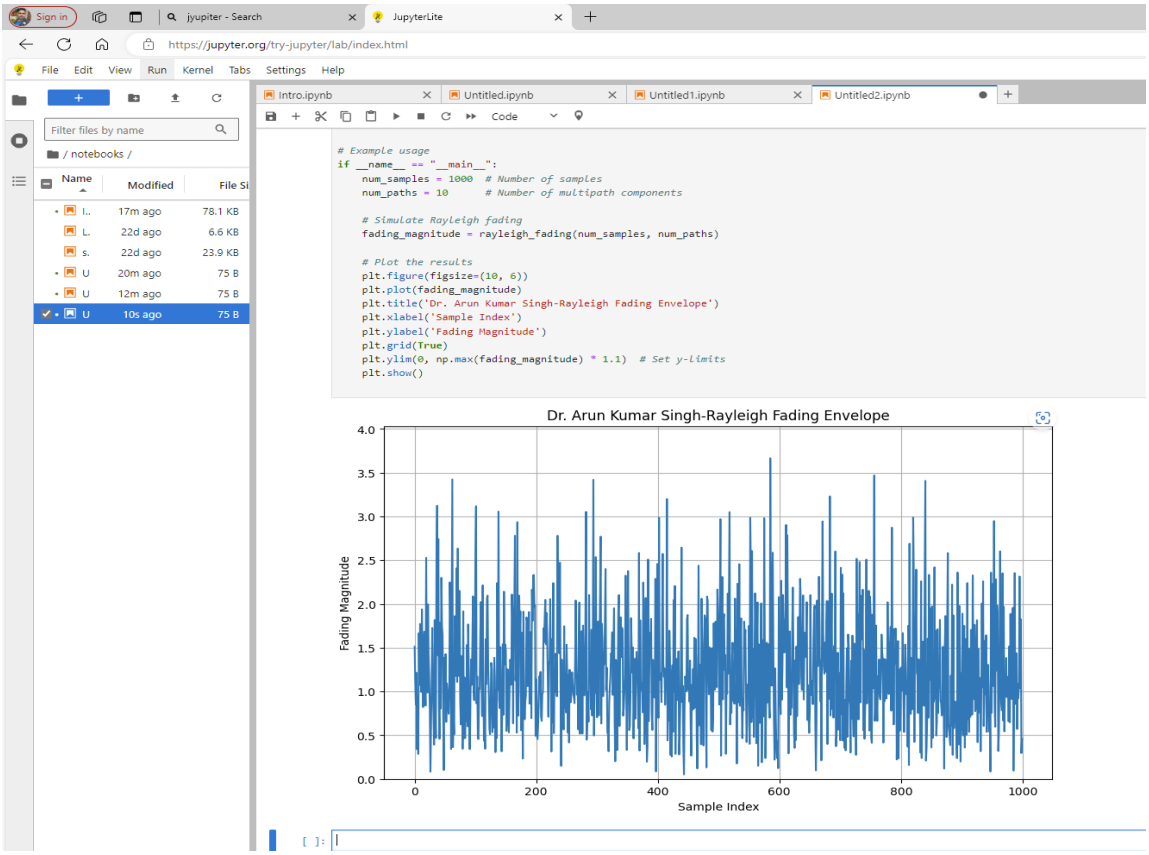

Fig. 3 Python Program to Simulate Rayleigh Fading

Explanation:

1. **Function rayleigh_fading(num_samples, num_paths)**:
   - **Input Parameters**:
     - num_samples: Number of samples to generate.
     - num_paths: Number of multipath components contributing to the fading.
   - **Process**:
     - For each multipath component, the program generates random Gaussian samples representing the contribution of each path.
     - A random phase for each path is generated to simulate the time-varying nature of the fading.
     - The contributions from all paths are summed to obtain the total fading sample.
     - The magnitude (envelope) of the resulting complex signal is calculated and normalized by the square root of the number of paths.
2. **Visualization**:
   - The resulting fading magnitude is plotted using Matplotlib, showing how the signal strength varies over time due to Rayleigh fading.

4. NETWORK CAPACITY AND SHANNON'S THEOREM

The upper bound to the rate at which information can be reliably carried by a wireless channel is then described by Shannon's Capacity Theorem which describes the capacity of the communication channel in consideration of bandwidth and SNR [9].

4.1 Shannon's Capacity Theorem

The capacity CCC of a wireless channel is defined as:

C=Blog⁡2(1+PN0B)C = B \log_2 \left(1 + \frac{P}{N_0 B} \right)C=Blog2(1+N0BP)

where:

- CCC = channel capacity (in bits per second),
- BBB = bandwidth (in Hz),
- PPP = signal power (in watts),
- N0N_0N0 = noise power spectral density (in watts/Hz).

This equation indicates that the capacity increases with bandwidth and signal power, but is limited by noise and interference. Shannon's theorem is fundamental in wireless computing, as it sets a theoretical limit on data transmission rates. The sample Pyton Code program for Network Capacity and Shannon's Theorem is shown in Fig. 4 below.

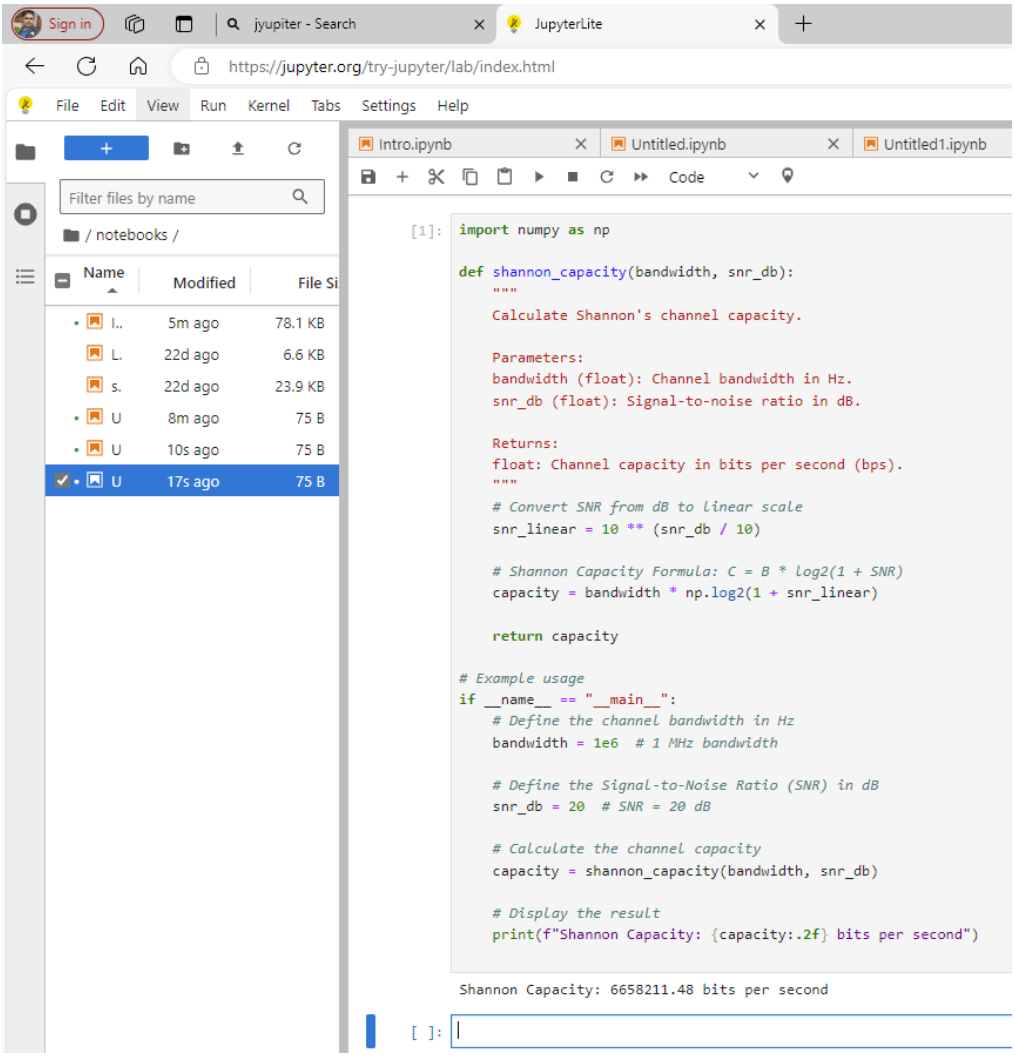

Fig. 4 Network Capacity and Shannon's Theorem-Python Code

Explanation:

1. **SNR in dB to Linear Conversion**: Since the SNR is usually given in decibels (dB), the program first converts it to a linear scale using the formula:

SNRlinear=10(SNRdB10)\text{SNR}_{\text{linear}} = 10^{\left(\frac{\text{SNR}_{\text{dB}}}{10}\right)}SNRlinear=10(10SNRdB)

2. **Shannon's Capacity Formula**: The capacity is then computed using the formula C=B·log⁡2(1+SNRlinear)C = B \cdot \log_2(1 + \text{SNR}_{\text{linear}})C=B·log2(1+SNRlinear), where BBB is the channel bandwidth in Hz, and SNRlinear\text{SNR}_{\text{linear}}SNRlinear is the linear signal-to-noise ratio.

5. ERROR CORRECTION AND CHANNEL CODING

Wireless communication is susceptible to errors due to noise, interference, and fading. Error correction techniques are essential to ensure reliable data transmission. One of the most commonly used error correction schemes is the **Hamming Code**, which detects and corrects bit errors in data transmission [10].

5.1 Hamming Code

The Hamming Code is a type of linear block code used for error detection and correction. The minimum Hamming distance between valid codewords is 3, which allows for the correction of single-bit errors. The codeword length nnn and the message length kkk are related by the equation:

n=2r−1n = 2^r - 1n=2r−1

where rrr is the number of redundant bits required for error detection and correction.
The Hamming distance between two binary strings aaa and bbb of equal length is defined as:

d(a,b)=∑i=1n(ai⊕bi)d(a, b) = \sum_{i=1}^{n} \left(a_i \oplus b_i \right)d(a,b)=i=1∑n(ai⊕bi)

where ⊕\oplus⊕ represents the XOR operation.

Hamming Code Generation (7,4) Example:

For this example, we'll generate the Hamming code (7,4), where:

- 4 bits of data are encoded into 7 bits by adding 3 parity bits.
- The additional parity bits help in error detection and correction.

Steps:

1. Take 4 data bits as input.
2. Calculate the positions of the 3 parity bits.
3. Encode the data with the calculated parity bits.
4. Simulate an error by flipping one bit and detect and correct the error.

Fig. 5 shown below is a sample Python program for **Hamming Code** (7,4):

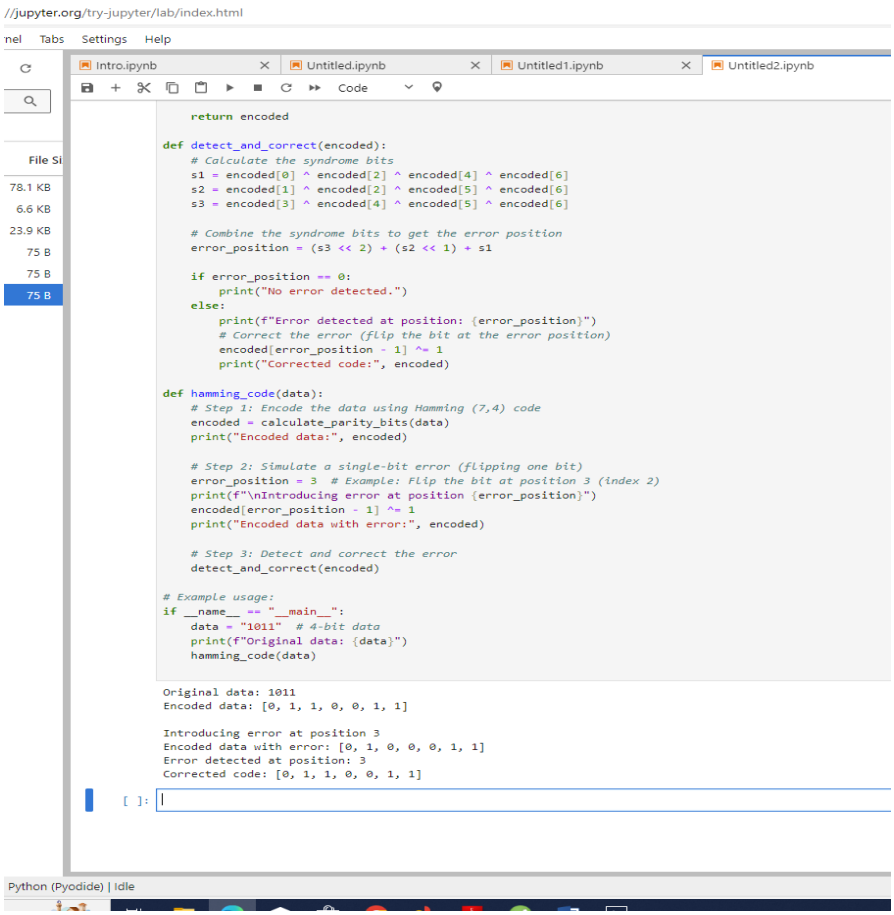

Fig. 5 Python Program for Hamming Code

Explanation:

1. **Encoding Data**:
   - The function calculate_parity_bits() takes 4 bits of data as input and calculates the parity bits.
   - The positions of the parity bits are calculated as follows:
     - p1=d1⊕d2⊕d4p1 = d1 \oplus d2 \oplus d4p1=d1⊕d2⊕d4
     - p2=d1⊕d3⊕d4p2 = d1 \oplus d3 \oplus d4p2=d1⊕d3⊕d4
     - p3=d2⊕d3⊕d4p3 = d2 \oplus d3 \oplus d4p3=d2⊕d3⊕d4
2. **Simulating an Error**:
   - In the hamming_code() function, we simulate a single-bit error by flipping a bit (position 3, in this example).

6. OPTIMIZATION OF WIRELESS NETWORKS

Network optimization is critical in wireless computing for efficient resource allocation and improved quality of service [11-19]. One mathematical technique widely used for optimization is **Linear Programming (LP)**.

6.1 Linear Programming (LP) for Wireless Network Optimization

In wireless networks, resource allocation problems (such as power distribution, bandwidth allocation) can often be modeled as LP problems. A typical LP problem for power allocation can be formulated as follows:

Minimize:f(x)=∑i=1nPi\text{Minimize:} \quad f(x) = \sum_{i=1}^{n} P_iMinimize:f(x)=i=1∑nPi Subject to:Ci=Bilog⁡2(1+PiN0Bi)≥Ri∀i\text{Subject to:} \quad C_i = B_i \log_2 \left(1 + \frac{P_i}{N_0 B_i} \right) \geq R_i \quad \forall iSubject to:Ci=Bilog2(1+N0BiPi)≥Ri∀i

where:

- PiP_iPi = power allocated to user iii,
- BiB_iBi = bandwidth allocated to user iii,
- RiR_iRi = minimum required data rate for user iii,
- N0N_0N0 = noise spectral density.

This formulation ensures that power is allocated in such a way as to minimize total power consumption while meeting the users' data rate requirements.

Fig. 6 shown below is a sample Python program that uses LP for **resource allocation in a wireless network**, where the objective is to maximize the total network throughput.

Problem:

- There are multiple wireless users, each with a required data rate.
- The network has a limited number of resources (e.g., bandwidth or power), and the goal is to allocate resources to maximize the total throughput.
- We want to solve the problem using Linear Programming.

Python Program for Linear Programming (using SciPy):

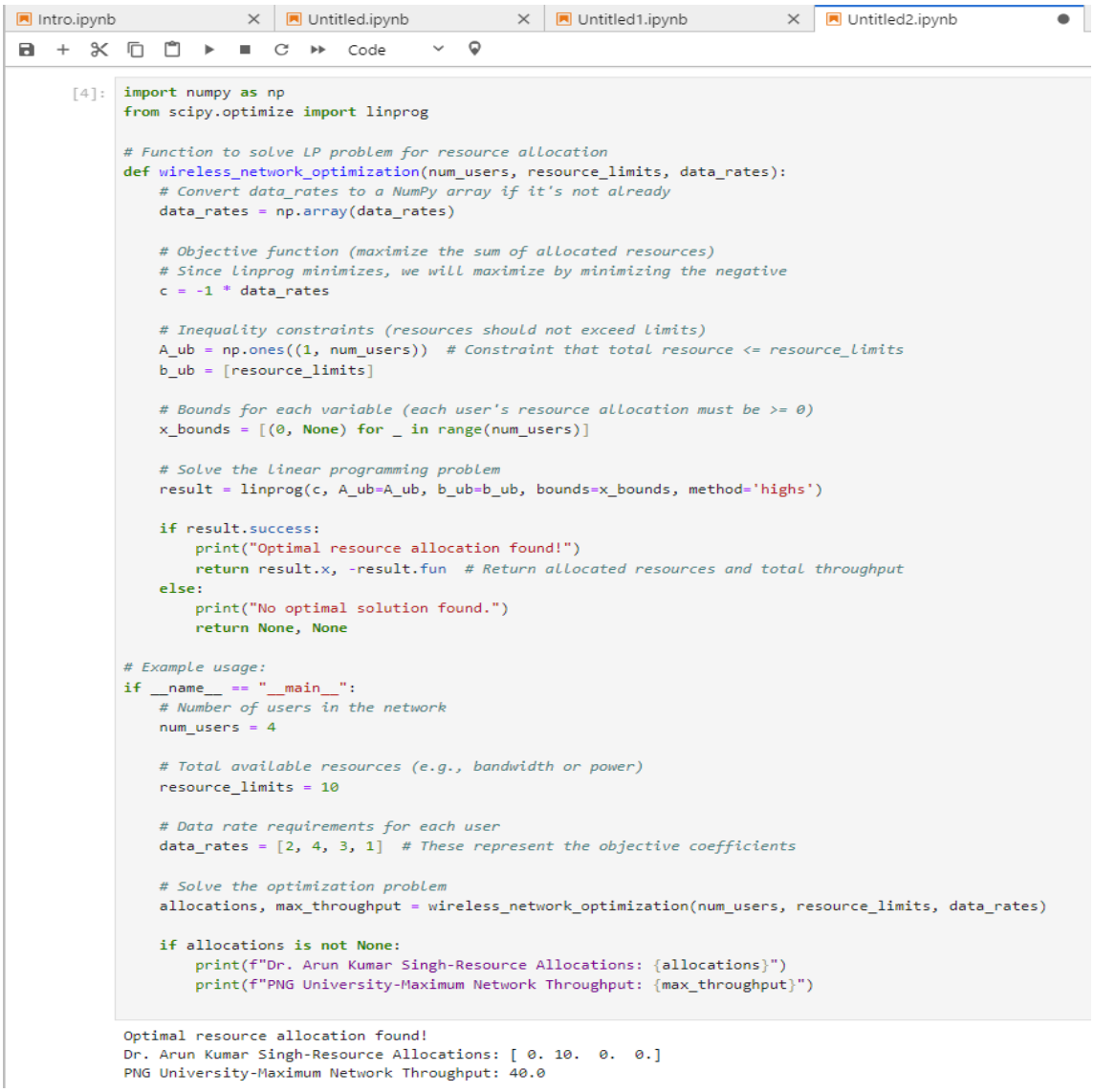

Fig. 6 Linear Programming (LP) for Wireless Network Optimization

Explanation:

1. **Objective Function**: The objective is to maximize the total network throughput. This is represented by the sum of the allocated resources. Since SciPy's linprog minimizes the objective, we negate the data rates to turn it into a maximization problem.
2. **Inequality Constraints**: The total resources allocated to users cannot exceed the available resource limits. This is enforced through the constraint: ∑i=1nxi≤resource_limits\sum_{i=1}^{n} x_i \leq \text{resource_limits}i=1∑nxi≤resource_limits Here, xix_ixi represents the resource allocated to each user.
3. **Bounds**: The allocation for each user must be non-negative (no negative resources), hence the bounds 0≤xi0 \leq x_i0≤xi.
4. **Linear Programming Solver**: The linprog function from SciPy is used to solve the LP problem. The method highs is a reliable solver for large-scale problems.

7. CONCLUSION

Therefore, wireless computing is a revolution of the kind of communication, computing, and transmission of information in the society. Mathematical model and mathematical approaches are very important for the effective and efficient working of wireless communication system. By applying mathematical models which include linear programming, stochastic process and statistical method, engineers and researchers are in a position to solve a number of problems that arise among wireless networks for example signal processing, resource management, error control and network optimization. The application of the above mathematical models makes it possible to have better insight into some of the issues affective wireless environment including fading, interferences, and capacity limitations. In addition, the proposal helps enforce the design of high-quality algorithms and protocols suitable for dynamic environments to provide quality services to users. As the wireless technologies are being developed further and further, demands for novel mathematical theories and techniques will grow even more, resulting in progress in the 5G network and IoT, to mention just a few domains. Last of all, the combination of mathematics and wireless computing also creates conditions for developing new communication means based on current achievements and stimulating the constant progress of digital integration in the modern world. On extending the notion of wire-less computing, mathematics will continue to form a fundamental base for wireless computing developments, for the solutions to the existing problems as well as the possibilities in future.

REFERENCE

[1] Abdel-Basset, Mohamed, et al. "A novel intelligent medical decision support model based on soft computing and IoT." *IEEE Internet of Things Journal* 7.5 (2019): 4160-4170.

[2] Alsharif, Mohammed H., et al. "Unleashing the potential of sixth generation (6G) wireless networks in smart energy grid management: A comprehensive review." *Energy Reports* 11 (2024): 1376-1398

[3] Arivazhagan, C., and V. Natarajan. "A Survey on Fog computing paradigms, Challenges and Opportunities in IoT." *2020 international conference on communication and signal processing (ICCSP)*. IEEE, 2020.

[4] Bharany, Salil, et al. "A systematic survey on energy-efficient techniques in sustainable cloud computing." *Sustainability* 14.10 (2022): 6256.

[5] Fan, Tiantian, et al. "Energy aware edge computing: a survey." *High-Performance Computing Applications in Numerical Simulation and Edge Computing: ACM ICS 2018 International Workshops, HPCMS and HiDEC, Beijing, China, June 12, 2018, Revised Selected Papers 2*. Springer Singapore, 2019.

[6] Haibeh, Lina A., Mustapha CE Yagoub, and Abdallah Jarray. "A survey on mobile edge computing infrastructure: Design, resource management, and optimization approaches." *IEEE Access* 10 (2022): 27591-27610.

[7] Hazra, Abhishek, et al. "Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges." *Computer Science Review* 48 (2023): 100549.

[8] Kaur, Loveleen, and Rajbir Kaur. "A survey on energy efficient routing techniques in WSNs focusing IoT applications and enhancing fog computing paradigm." *Global transitions proceedings* 2.2 (2021): 520-529.

[9] Mishra, Sambit Kumar, et al. "Energy-efficient service allocation techniques in cloud: A survey." *IETE Technical Review* 37.4 (2020): 339-352.

[10] Medara, Rambabu, and Ravi Shankar Singh. "A review on energy-aware scheduling techniques for workflows in IaaS clouds." *Wireless Personal Communications* 125.2 (2022): 1545-1584.

.[11] Popli, Sakshi, Rakesh Kumar Jha, and Sanjeev Jain. "A survey on energy efficient narrowband internet of things (NBIoT): architecture, application and challenges." *IEEE access* 7 (2018): 16739-16776.

[12] Sadri, Ali Akbar, et al. "Data reduction in fog computing and internet of things: A systematic literature survey." *Internet of Things* 20 (2022): 100629.

[13] Singh, Arun Kumar. "Digital Era in Papua New Guinea (PNG): Novel Strategies of the Telecom Service Provider Companies." *Social Capital in the Age of Online Networking: Genesis, Manifestations, and Implications*. IGI Global, 2023. 230-248.

[14] Singh, Arun Kumar. "Fundamentals of Data Visualization and Its Applications in Business." *Data Visualization Tools for Business Applications*. IGI Global, 2025. 1-28.

[15] Singh, Arun Kumar. "A Basic Process of Python Use for IOTAP, Data Science, and Rapid Machine Learning Model Development." *Fraud Prevention, Confidentiality, and Data Security for Modern Businesses*. IGI Global, 2023. 84-104.

[16] Shivahare, Basu Dev, et al. "Survey Paper: Study of Natural Language Processing and its Recent Applications." *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*. IEEE, 2022.

[17] Songhorabadi, Maryam, et al. "Fog computing approaches in IoT-enabled smart cities." *Journal of Network and Computer Applications* 211 (2023): 103557.

[18] Verma, Richa, and Shalini Chandra. "A systematic survey on fog steered IoT: Architecture, prevalent threats and trust models." *International Journal of Wireless Information Networks* 28 (2021): 116-133.

[19] Yu, Dongmin, Zimeng Ma, and Rijun Wang. "Efficient smart grid load balancing via fog and cloud computing." *Mathematical Problems in Engineering* 2022.1 (2022): 3151249.