David Mayo  
CSE 516-50  
Programming Assignment Two

## 1. Plot for Study 1

A graph of a graph

Description automatically generated

## 2. Answer to Question 1

It's possible to combine all the given expressions.

This simplified version is what is calculated in my code and used for all answers.

As we can see both from the plot and from the formula, U is inversely proportional to distance. This makes sense from a networking standpoint, since as distance increases, the latency increases, and the stop-and-wait protocol is inefficient when latency is high.

## 3. Plot for Study 2

A graph of a graph

Description automatically generated with medium confidence

## 4. Answer to Question 2

As we see from the formula and the plot, utilization is directly proportional to the frame size. This makes sense from a networking standpoint because as the latency remains constant and the amount of data per frame increases, the efficiency increases.

## 5. Plot for Study 3

A graph of a number of miles

Description automatically generated with medium confidence

## 6. Answer to Question 3

As we can see from the formula and the plot, utilization is inversely proportional to data transmission rate.

From a networking standpoint, this makes sense, though it seems counterintuitive at first. The number of frames that can be sent in some time period is dominated by the latency, not the transmission rate, so that increasing the transmission rate has virtually no impact on how much data is actual successfully transmitted in the

## 7. Code/excel sheet for calculating the values.

"""

Python3.11 code for solving CSE 516-50 programming assignment two

Author: David Mayo

Code available at

https://github.com/davidmayo/cse516\_assignment2/blob/main/script.py

A relatively recent version of matplotlib is required to run this code.

"""

import matplotlib.pyplot as plt

# See report for explanation of this function

def utilization(

    frame\_size: float,

    propagation\_speed: float,

    transmission\_rate: float,

    distance: float,

) -> float:

    return frame\_size \* propagation\_speed / (2 \* transmission\_rate \* distance)

#####################

#      Study 1      #

#####################

# Calculations

frame\_size = 5000

transmission\_rate = 1e6

propagation\_speed = 3e8

distances = [1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000]

utilizations = [

    utilization(

        frame\_size=frame\_size,

        propagation\_speed=propagation\_speed,

        transmission\_rate=transmission\_rate,

        distance=d

    )

    for d

    in distances

]

# Plot

fig, ax = plt.subplots(layout="constrained")

ax: plt.Axes

ax.plot(distances, utilizations)

ax.scatter(distances, utilizations)

ax.set\_xlabel(f"Distance (meters)")

ax.set\_ylabel(f"Utilization (dimensionless)")

ax.set\_title(

    f"Utilization (Study 1)\n\n{frame\_size = :,.0f} bits\n"

    + f"{transmission\_rate = :,.0f} bps\n{propagation\_speed = :,.0f} m/s"

)

fig.savefig("./study1.png", dpi=300)

plt.show()

#####################

#      Study 2      #

#####################

# Calculations

frame\_sizes = [5000, 6000, 7000, 8000, 9000, 10000]

transmission\_rate = 1e6

propagation\_speed = 3e8

distance = 3000

utilizations = [

    utilization(

        frame\_size=f,

        propagation\_speed=propagation\_speed,

        transmission\_rate=transmission\_rate,

        distance=distance

    )

    for f

    in frame\_sizes

]

# Plot

fig, ax = plt.subplots(layout="constrained")

ax: plt.Axes

ax.plot(frame\_sizes, utilizations)

ax.scatter(frame\_sizes, utilizations)

ax.set\_xlabel(f"Frame size (bits)")

ax.set\_ylabel(f"Utilization (dimensionless)")

ax.set\_title(

    f"Utilization (Study 2)\n\n{distance = :,.0f} meters\n"

    + f"{transmission\_rate = :,.0f} bps\n{propagation\_speed = :,.0f} m/s"

)

fig.savefig("./study2.png", dpi=300)

plt.show()

#####################

#      Study 3      #

#####################

# Calculations

frame\_size = 10000

# transmission\_rate = 1e6

transmission\_rates = [

    1e6,

    4e6,

    8e6,

    16e6,

    32e6,

    64e6,

]

propagation\_speed = 3e8

distance = 3000

utilizations = [

    utilization(

        frame\_size=frame\_size,

        propagation\_speed=propagation\_speed,

        transmission\_rate=t\_r,

        distance=distance

    )

    for t\_r

    in transmission\_rates

]

# Plot

fig, ax = plt.subplots(layout="constrained")

ax: plt.Axes

transmission\_rates\_in\_mbps = [t\_r / 1e6 for t\_r in transmission\_rates]

ax.plot(transmission\_rates\_in\_mbps, utilizations)

ax.scatter(transmission\_rates\_in\_mbps, utilizations)

ax.set\_xlabel(f"Transmission rate (Mbps)")

ax.set\_ylabel(f"Utilization (dimensionless)")

ax.set\_title(

    f"Utilization (Study 3)\n\n{distance = :,.0f} meters\n"

    + f"{frame\_size = :,.0f} bits\n{propagation\_speed = :,.0f} m/s"

)

fig.savefig("./study3.png", dpi=300)

plt.show()