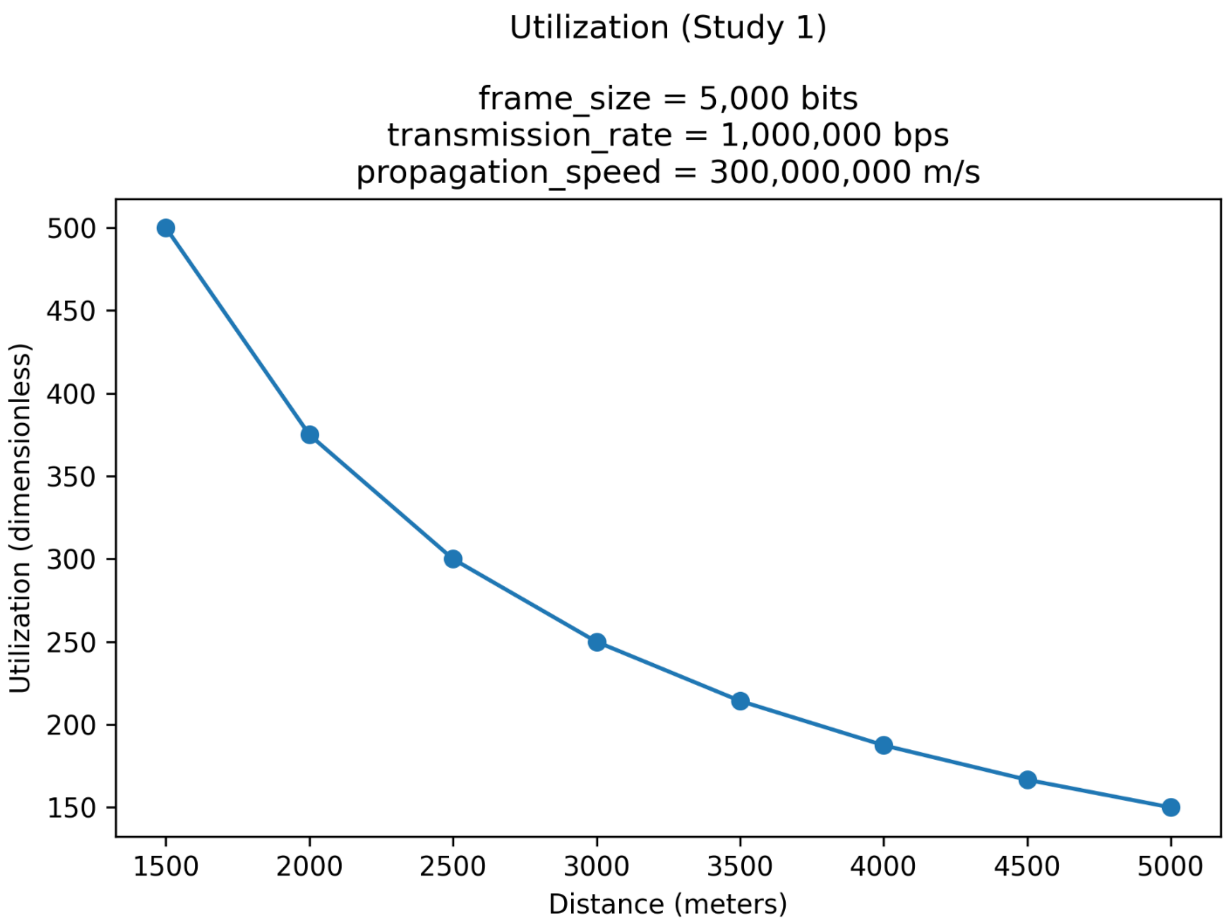
David Mayo
CSE 516-50
Programming Assignment Two

## 1. Plot for Study 1




## 2. Answer to Question 1

It's possible to combine all the given expressions.

$$U = \frac{T_F}{2 * T_P} \qquad given$$

$$T_F = \frac{F}{T_R} \qquad given$$

$$T_P = \frac{D}{C} \qquad given$$

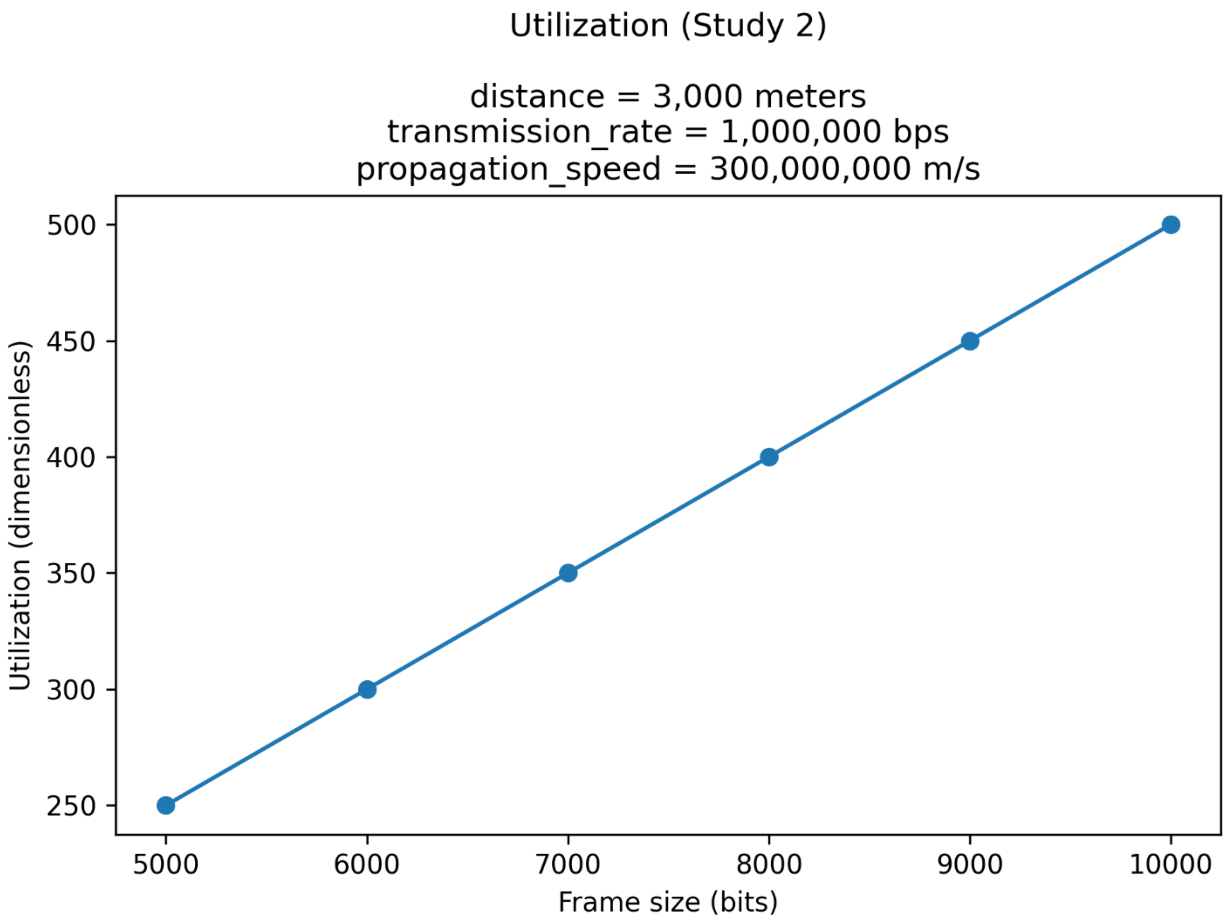
$$U = \frac{F/T_r}{2 * D/C} \qquad combined$$

$$U = \frac{F * C}{2 * D * T_R} \qquad simplified$$

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.

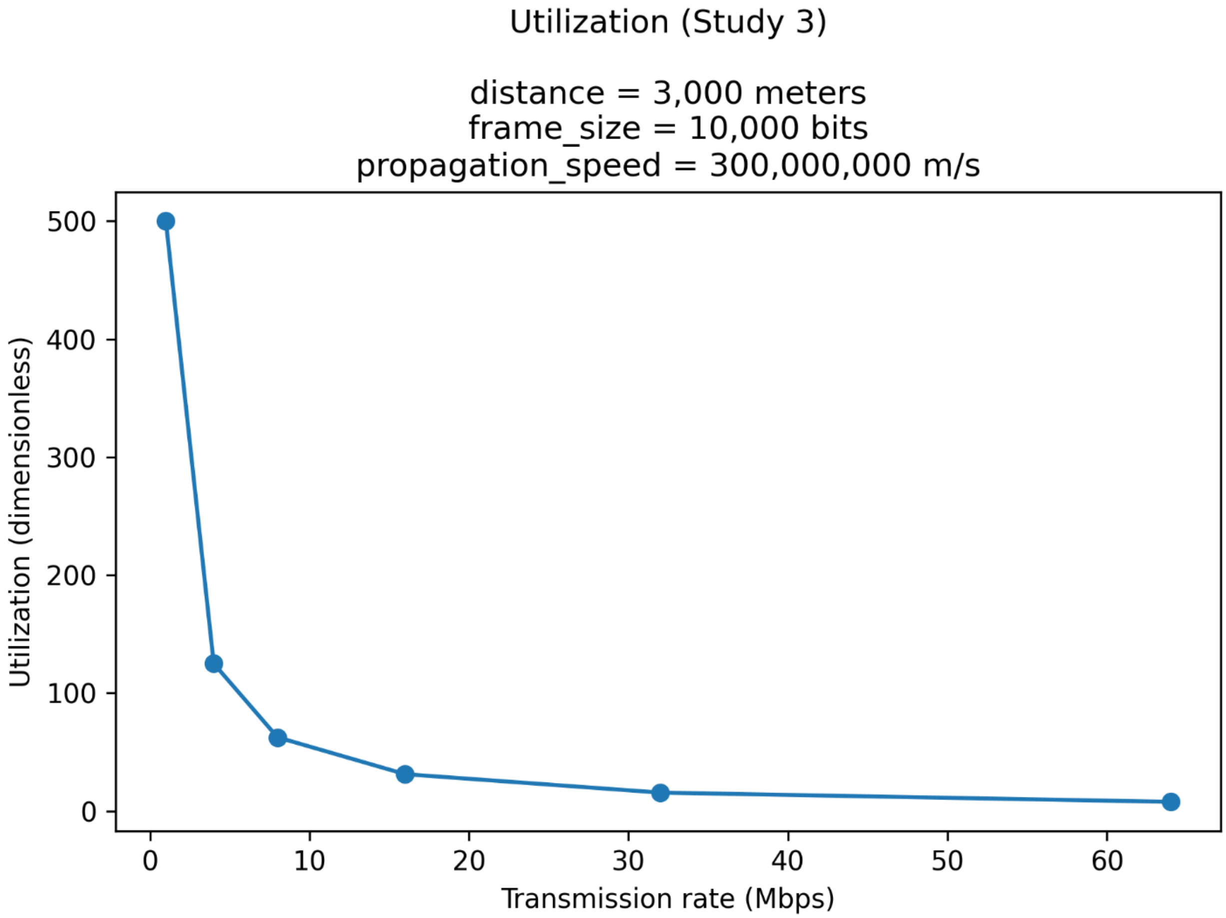
## 3. Plot for Study 2




## 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




## 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"
```

```python
    + 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
```

```python
# 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()
```