In [5]:

```
import pandas as pd
d = pd.read_csv(r"D:\New folder (4)\log2.csv")
print(d.head())
```

```
   Source Port  Destination Port  NAT Source Port  NAT Destination Port  \
0        57222                53            54587                    53   
1        56258              3389            56258                  3389   
2         6881             50321            43265                 50321   
3        50553              3389            50553                  3389   
4        50002               443            45848                   443   

  Action  Bytes  Bytes Sent  Bytes Received  Packets  Elapsed Time (sec)  \
0  allow    177          94              83        2                  30   
1  allow   4768        1600            3168       19                  17   
2  allow    238         118             120        2                1199   
3  allow   3327        1438            1889       15                  17   
4  allow  25358        6778           18580       31                  16   

   pkts_sent  pkts_received  
0          1              1  
1         10              9  
2          1              1  
3          8              7  
4         13             18  
```

In [6]:

```
d2 = pd.get_dummies(d, columns=['Action'], drop_first=True)
print(d2.head())
```

```
   Source Port  Destination Port  NAT Source Port  NAT Destination Port  \
0        57222                53            54587                    53   
1        56258              3389            56258                  3389   
2         6881             50321            43265                 50321   
3        50553              3389            50553                  3389   
4        50002               443            45848                   443   

   Bytes  Bytes Sent  Bytes Received  Packets  Elapsed Time (sec)  pkts_sent  \
0    177          94              83        2                  30          1   
1   4768        1600            3168       19                  17         10   
2    238         118             120        2                1199          1   
3   3327        1438            1889       15                  17          8   
4  25358        6778           18580       31                  16         13   

   pkts_received  Action_deny  Action_drop  Action_reset-both  
0              1            0            0                  0  
1              9            0            0                  0  
2              1            0            0                  0  
3              7            0            0                  0  
4             18            0            0                  0  
```

In [7]:

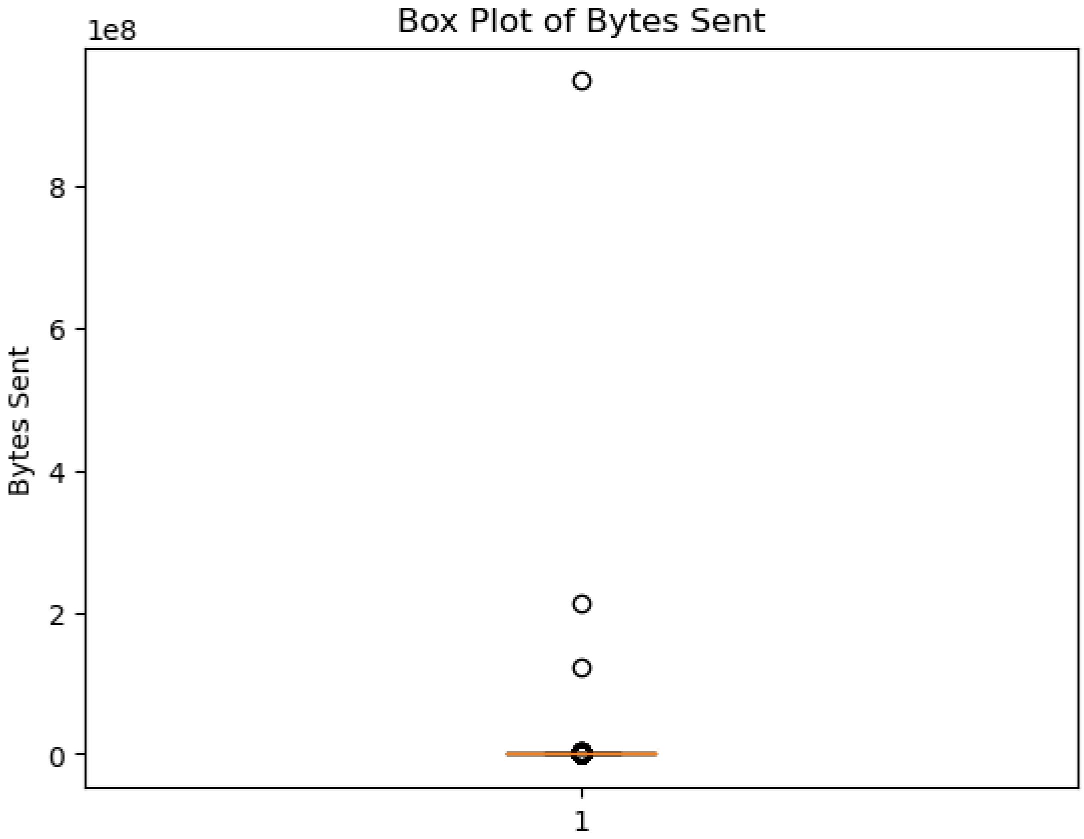
```
m = d['Bytes Sent'].mean()
s = d['Bytes Sent'].std()
l = m - 2 * s
u = m + 2 * s
o = d[(d['Bytes Sent'] < l) | (d['Bytes Sent'] > u)]
print("Outliers in Bytes Sent:\n", o)
```

```
Outliers in Bytes Sent:
        Source Port  Destination Port  NAT Source Port  NAT Destination Port  \
10220         57235             15187            23276                 15187   
33967         15503             62336            46736                 62336   
61429         15792              3478            30536                  3478   

      Action       Bytes  Bytes Sent  Bytes Received  Packets  \
10220  allow  1269359015   948477220       320881795  1036116   
33967  allow   127653507   122661116         4992391   161030   
61429  allow   428935914   213443641       215492273   635946   

       Elapsed Time (sec)  pkts_sent  pkts_received  
10220                9283     747520         288596  
33967                2162      82907          78123  
61429                2242     308738         327208  
```

In [8]:

```python
import matplotlib.pyplot as plt
plt.boxplot(d['Bytes Sent'].dropna())
plt.title('Box Plot of Bytes Sent')
plt.ylabel('Bytes Sent')
plt.show()
```

```
Matplotlib is building the font cache; this may take a moment.
```



In [9]:

```python
mv = d.isnull().sum()
print("Missing Values:\n", mv)
```

```
Missing Values:
 Source Port             0
Destination Port        0
NAT Source Port         0
NAT Destination Port    0
Action                  0
Bytes                   0
Bytes Sent              0
Bytes Received          0
Packets                 0
Elapsed Time (sec)      0
pkts_sent               0
pkts_received           0
dtype: int64
```

In [10]:

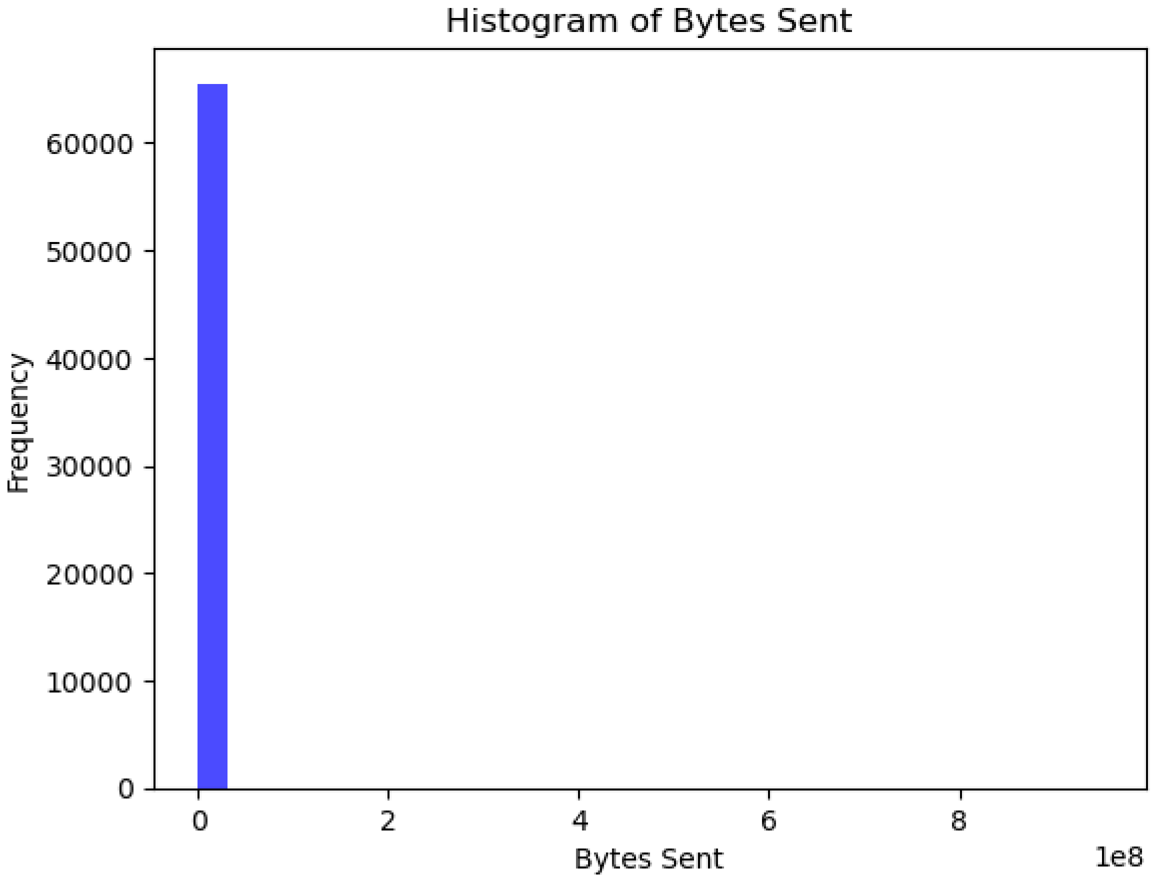
```
d.fillna(d.mean(), inplace=True)
print("Dataset after filling missing values:\n", d.head())
```

```
Dataset after filling missing values:
     Source Port  Destination Port  NAT Source Port  NAT Destination Port  \
0         57222                53            54587                    53   
1         56258              3389            56258                  3389   
2          6881             50321            43265                 50321   
3         50553              3389            50553                  3389   
4         50002               443            45848                   443   

  Action  Bytes  Bytes Sent  Bytes Received  Packets  Elapsed Time (sec)  \
0  allow    177          94              83        2                  30   
1  allow   4768        1600            3168       19                  17   
2  allow    238         118             120        2                1199   
3  allow   3327        1438            1889       15                  17   
4  allow  25358        6778           18580       31                  16   

   pkts_sent  pkts_received  
0          1              1  
1         10              9  
2          1              1  
3          8              7  
4         13             18  
```

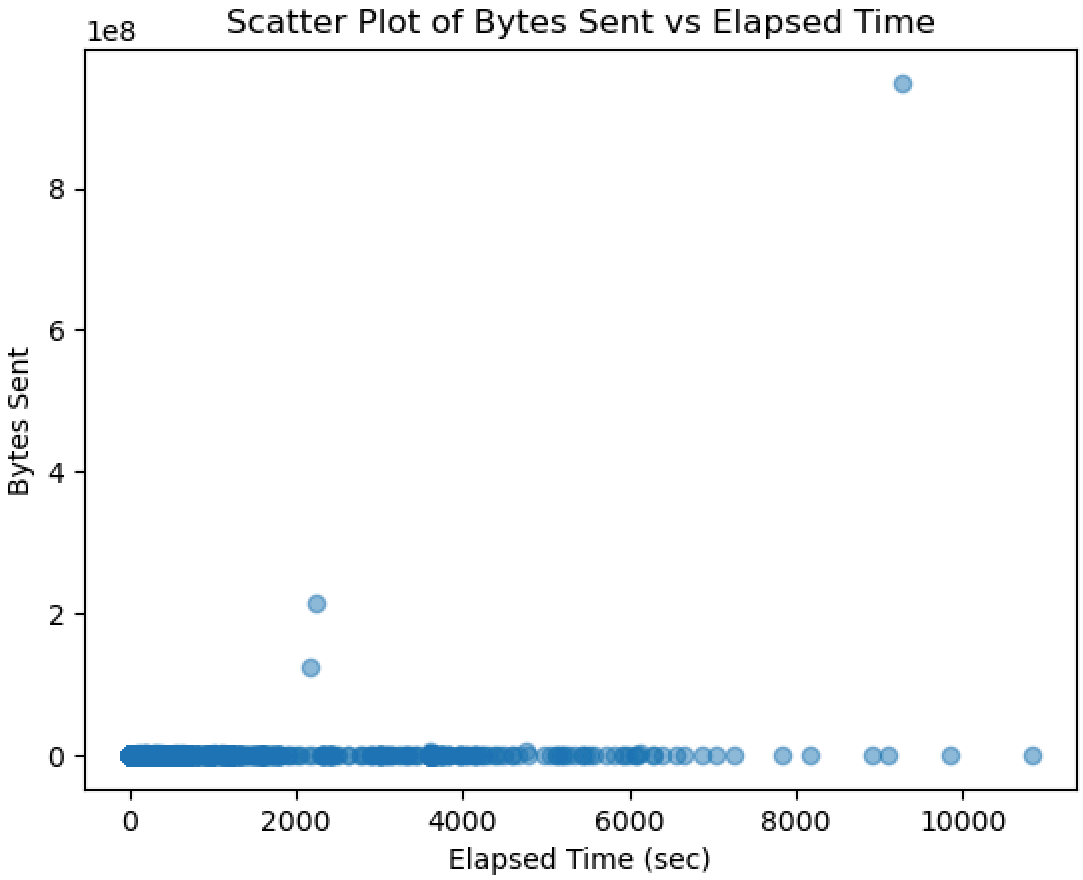
```
C:\Users\22bad059\AppData\Local\Temp\ipykernel_9752\3930674370.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.mean is deprecated. In a future
version, it will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of numeric_only to sile
nce this warning.
  d.fillna(d.mean(), inplace=True)
```

In [11]:

```
plt.hist(d['Bytes Sent'], bins=30, color='blue', alpha=0.7)
plt.title('Histogram of Bytes Sent')
plt.xlabel('Bytes Sent')
plt.ylabel('Frequency')
plt.show()
```



In [12]:
```
plt.scatter(d['Elapsed Time (sec)'], d['Bytes Sent'], alpha=0.5)
plt.title('Scatter Plot of Bytes Sent vs Elapsed Time')
plt.xlabel('Elapsed Time (sec)')
plt.ylabel('Bytes Sent')
plt.show()
```



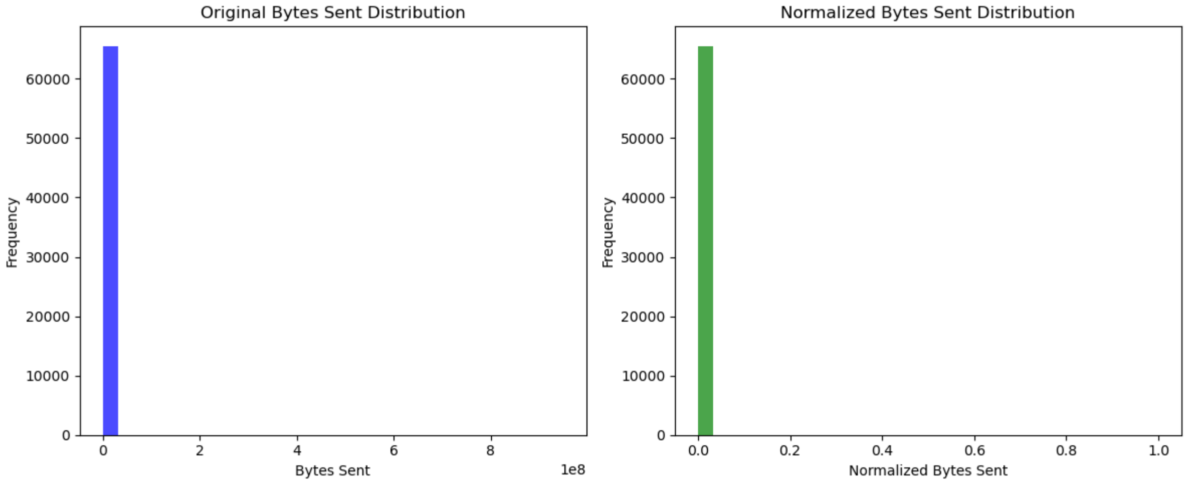
In [19]:

```
from sklearn.preprocessing import MinMaxScaler
s = MinMaxScaler()
d['B_S'] = s.fit_transform(d[['Bytes Sent']])
d['B_R'] = s.fit_transform(d[['Bytes Received']])
print("Dataset with Normalized Bytes:\n", d[['Bytes Sent', 'B_S', 'Bytes Received',
```

```
Dataset with Normalized Bytes:
    Bytes Sent           B_S  Bytes Received           B_R
0          94  3.584694e-08              83  2.586622e-07
1        1600  1.623655e-06            3168  9.872794e-06
2         118  6.115066e-08             120  3.739695e-07
3        1438  1.452855e-06            1889  5.886903e-06
4        6778  7.082933e-06           18580  5.790294e-05
```

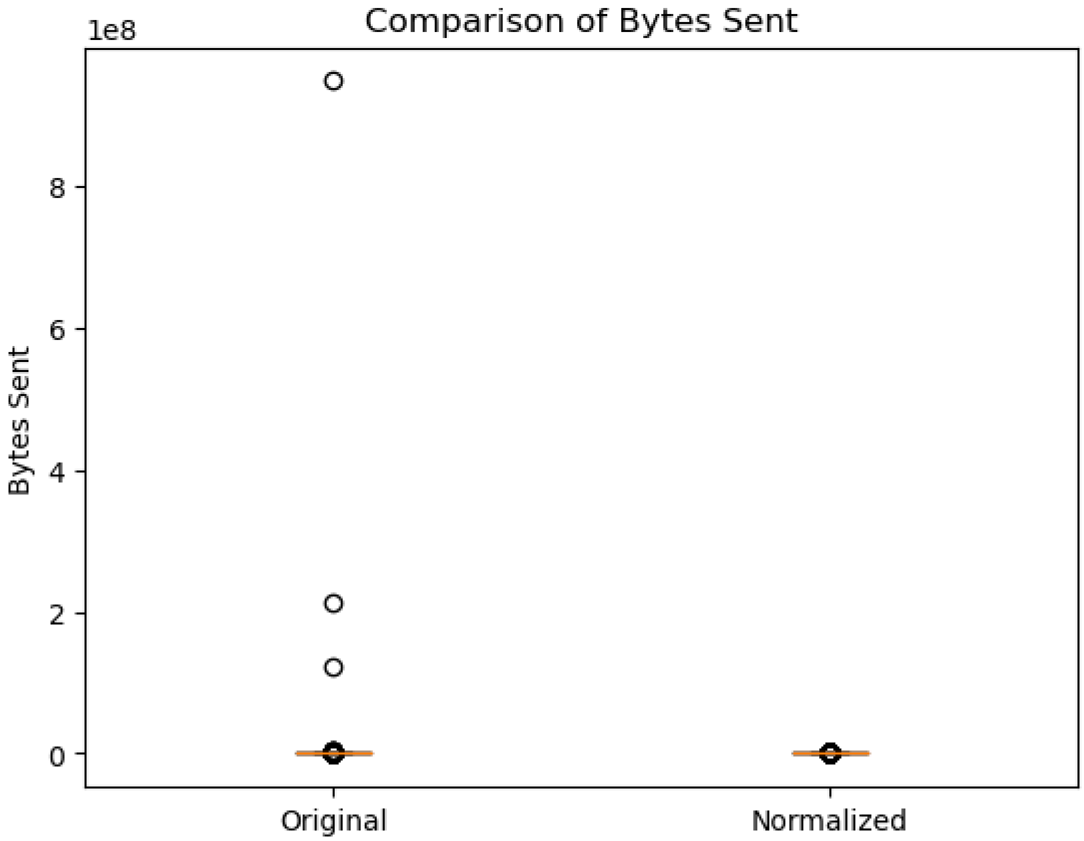
In [20]:

```
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.hist(d['Bytes Sent'], bins=30, color='blue', alpha=0.7)
plt.title('Original Bytes Sent Distribution')
plt.xlabel('Bytes Sent')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
plt.hist(d['B_S'], bins=30, color='green', alpha=0.7)
plt.title('Normalized Bytes Sent Distribution')
plt.xlabel('Normalized Bytes Sent')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



In [21]:

```
plt.boxplot([d['Bytes Sent'], d['B_S']], labels=['Original', 'Normalized'])
plt.title('Comparison of Bytes Sent')
plt.ylabel('Bytes Sent')
plt.show()
```


Comparison of Bytes Sent
1e8
8
6
4
2
0
Bytes Sent
Original
Normalized

```
In [ ]:
```