Data analysis of internet usage of graduate student in Indian University

In [11]:
```
import pandas as pd
import matplotlib.pyplot as mp
import numpy as np
import seaborn as sns

df = pd.read_csv('F:\\JNTUH  data science\\Assignments\\ML_DS_Assignments\\ML DL Assignment 1\\internet_session.csv')
```

In [12]:
```
df
```

Out[12]:

	name	start_time	usage_time	IP	MAC	upload	download	total_transfer	seession_break_reason
0	user1	2022-05-10 02:59:32	00:00:36:28	10.55.14.222	48:E7:DA:58:22:E9	15861.76	333168.64	349030.40	Idle-Timeout
1	user1	2022-05-10 18:53:27	00:01:49:56	10.55.2.253	48:E7:DA:58:22:E9	16957.44	212152.32	229109.76	Idle-Timeout
2	user1	2022-05-10 21:20:44	00:01:35:00	10.55.2.253	48:E7:DA:58:22:E9	14080.0	195153.92	209233.92	Idle-Timeout
3	user1	2022-05-11 00:37:42	00:00:26:00	10.55.2.253	48:E7:DA:58:22:E9	5242.88	40806.4	46049.28	Idle-Timeout
4	user1	2022-05-11 02:59:38	00:00:11:52	10.55.2.253	48:E7:DA:58:22:E9	22067.2	10772.48	32839.68	Idle-Timeout
...	...	...	...	...	...	...	...	...	...
4707	user9	2022-11-04 01:11:34	00:06:54:32	10.55.4.189	DA:2F:97:0E:B7:D0	107960.32	2390753.28	2495610.88	Idle-Timeout
4708	user9	2022-11-04 10:26:09	00:00:23:49	10.55.4.59	DA:2F:97:0E:B7:D0	11407.36	209674.24	221081.60	Idle-Timeout
4709	user9	2022-11-04 20:41:42	00:01:24:13	10.55.15.186	DA:2F:97:0E:B7:D0	18995.2	373657.6	392652.80	Idle-Timeout
4710	user9	2022-11-05 00:21:06	00:08:49:43	10.55.4.159	DA:2F:97:0E:B7:D0	46602.24	593766.4	640368.64	Idle-Timeout
4711	user9	2022-11-05 20:55:37	00:01:06:20	10.55.2.33	DA:2F:97:0E:B7:D0	21237.76	298536.96	319774.72	NaN

4712 rows × 9 columns

In [13]:
```
# Finding null values
df.isna().sum()
```

Out[13]:
```
name                    0
start_time              0
usage_time              0
IP                      0
MAC                     0
upload                  0
download                0
total_transfer          0
seession_break_reason   9
dtype: int64
```

In [14]:
```
# Dropping null values as they won't affect the analysis
df = df.dropna().copy()
```

In [15]:
```
df.isna().sum()
```

Out[15]:
```
name                    0
start_time              0
usage_time              0
IP                      0
MAC                     0
upload                  0
download                0
total_transfer          0
seession_break_reason   0
dtype: int64
```

In [16]:
```
# Searching for duplicated values
df.duplicated().sum()
```

Out[16]:
```
0
```

In [17]:
```
df.dtypes
```

Out[17]:
```
name                    object
start_time              object
usage_time              object
IP                      object
MAC                     object
upload                  object
download                object
total_transfer         float64
seession_break_reason   object
dtype: object
```

In [18]:
```
df.upload[1643]
```

Out[18]:
```
'240B'
```

In [19]:
```
# value of upload is '240B' at 1643.
# Extract the numeric values from upload columns and convert them to float type
df['upload'] = df['upload'].str.extract('(\d+)', expand=False)
df.upload = df.upload.astype(float)

df['download'] = df['download'].str.extract('(\d+)', expand=False)
df.download = df.download.astype(float)
```

In [20]:
```
# converting 'usage_time datatype from object to datetime'
df['usage_time'] = df['usage_time'].str.replace('00:', '', 1)
df['usage_time'] = pd.to_datetime(df['usage_time'])
```

In [26]:
```
df['start_time'] = pd.to_datetime(df.start_time)
```

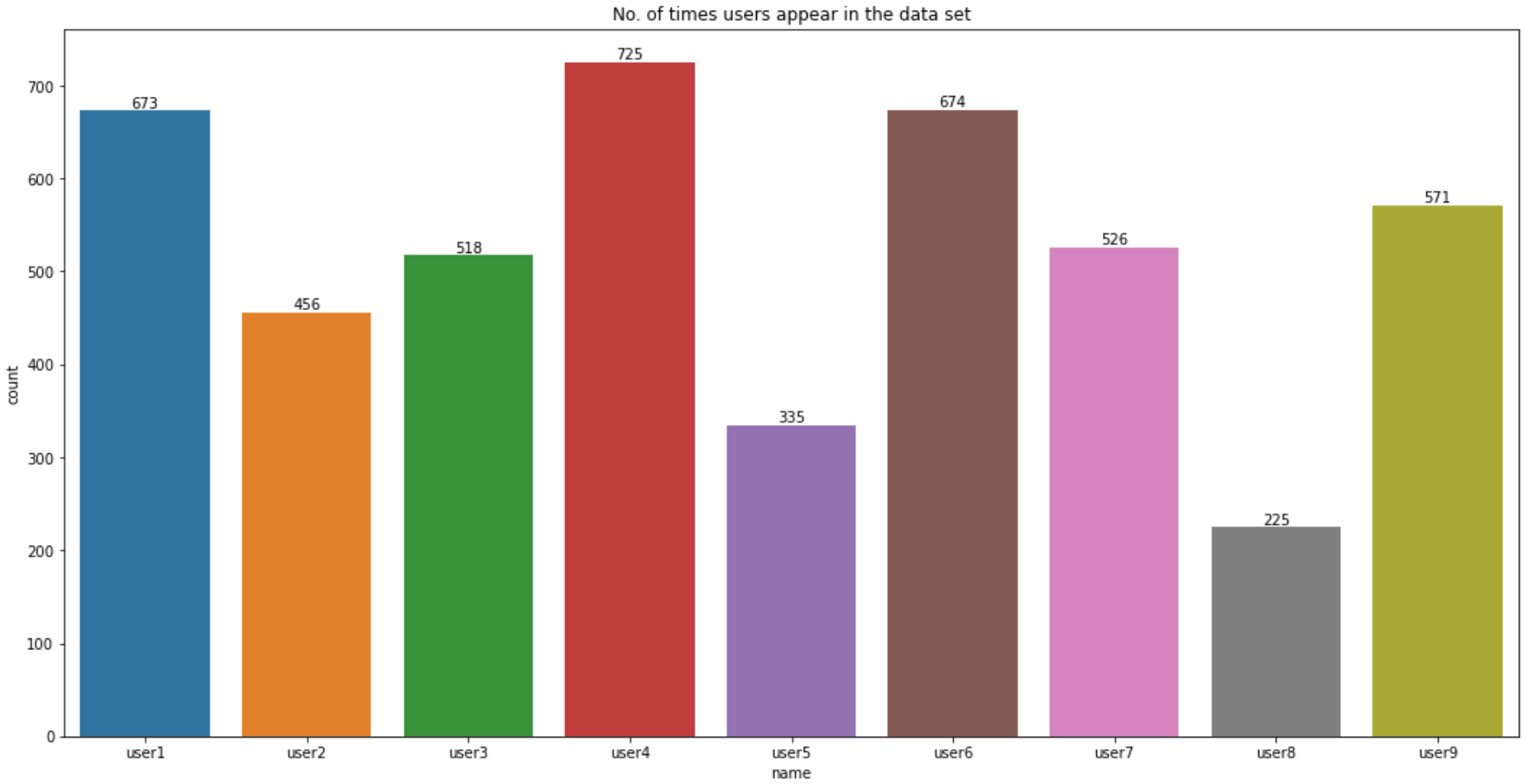
In [27]:
```
df.dtypes
```

Out[27]:
```
name                            object
start_time              datetime64[ns]
usage_time              datetime64[ns]
IP                              object
MAC                             object
upload                         float64
download                       float64
total_transfer                 float64
seession_break_reason           object
dtype: object
```

In [28]:
```
# The total number of users
df.name.value_counts()
```

Out[28]:
```
user4    725
user6    674
user1    673
user9    571
user7    526
user3    518
user2    456
user5    335
user8    225
Name: name, dtype: int64
```

In [50]:
```
mp.figure(figsize=(18, 9))
ax = sns.countplot(x='name', data=df)
ax.bar_label(ax.containers[0])
mp.title("No. of times users appear in the data set")
mp.show()
mp.clf()
```



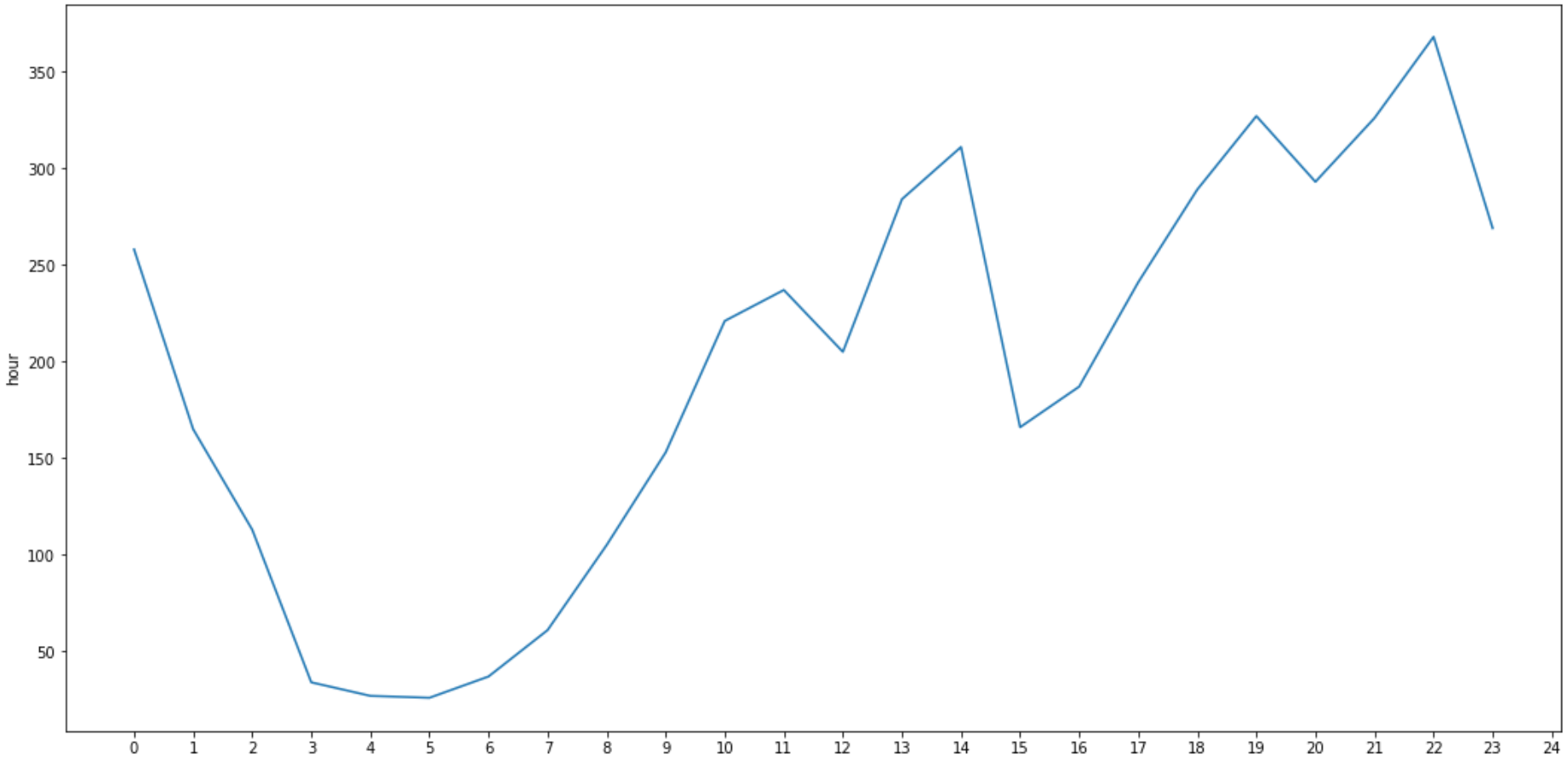

```
<Figure size 432x288 with 0 Axes>
```

Analysis on data

What is the most frequent internet activity time of the day ?

In [30]:
```
df['hour'] = pd.to_datetime(df['start_time']).dt.hour
freq_time = df['hour'].value_counts().sort_index()
mp.figure(figsize=(18, 9))
sns.lineplot(data=freq_time)
mp.xticks(np.linspace(start=0, stop=24, num=25))
```

```
mp.show()
mp.clf()
```




```
<Figure size 432x288 with 0 Axes>
```

From the above graph we can conclude that most users were active at 22:00 hour per day.

How often the ip changes ?

In [32]:
```
base_ip = '10.55.2.253'
ip_count = 0
for i in range(1, df.shape[0]):
    if df.iloc[i]['IP'] != base_ip:
        ip_count +=1
        base_ip = df.iloc[i]['IP']
```

```
print('No. of times the Ip address changed: ' + str(ip_count))
```

```
No. of times the Ip address changed: 2302
```

How often the device changed.

In [33]:
```
base_device = '48:E7:DA:58:22:E9'
device_count = 0
for i in range(1, df.shape[0]):
    if df.iloc[i]['MAC'] != base_device:
        device_count +=1
        base_device = df.iloc[i]['MAC']

print('No. of times the device changed: ' + str(device_count))
```

```
No. of times the device changed: 1223
```

What is the average usage per hour , per day and per month ?

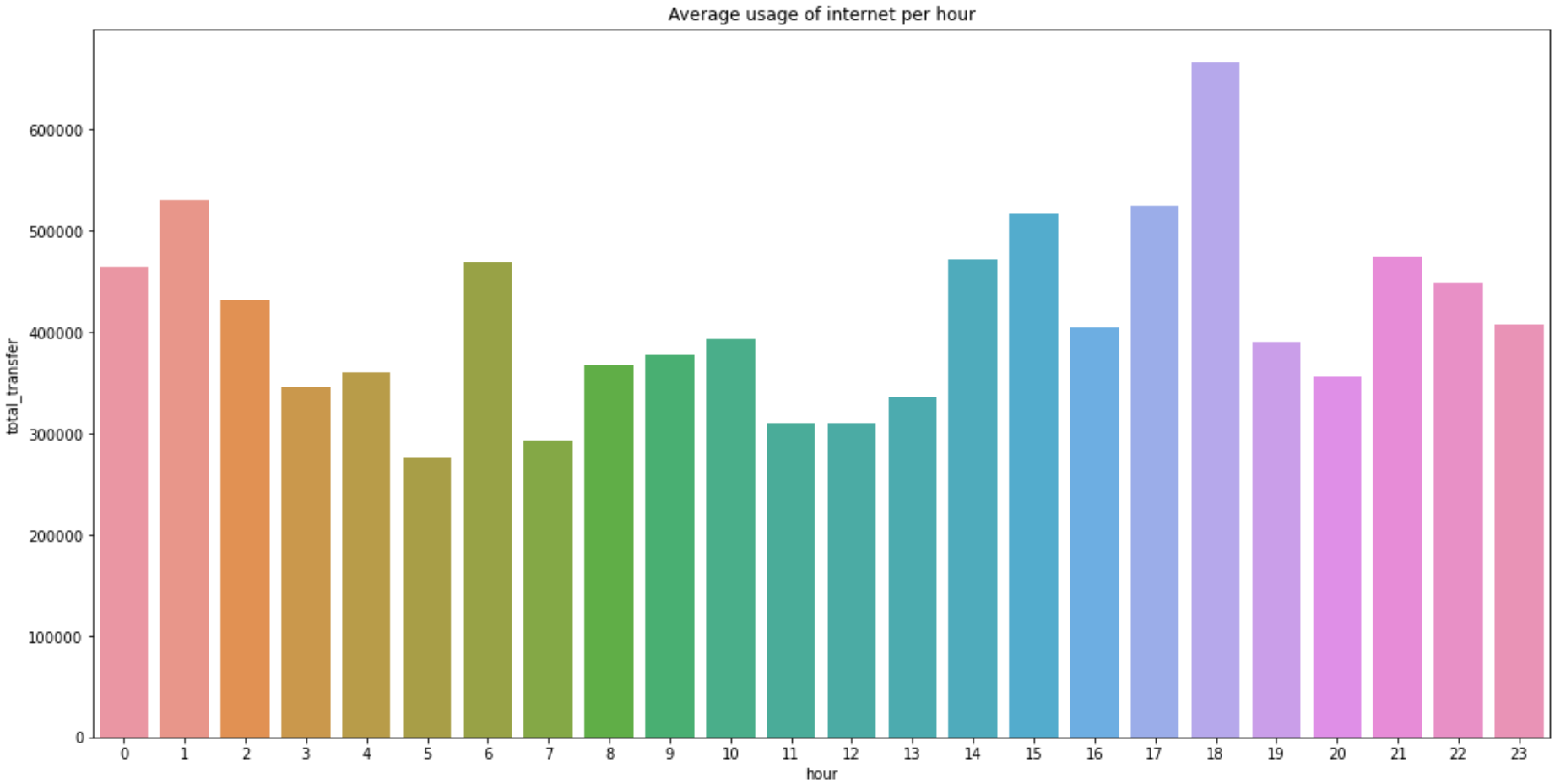
In [38]:
```
df['day'] = df['start_time'].dt.day
df['month'] = df['start_time'].dt.month

average_hr = df.groupby('hour').total_transfer.mean()
print('The average usage per hour:\n ' +  str(round(average_hr, 2)))
```

```
The average usage per hour:
 hour
0     464530.44
1     530880.86
2     431576.11
3     345303.34
4     359809.44
5     275960.91
6     468959.59
7     292886.83
8     366681.92
9     377480.64
10    393259.12
11    309492.45
12    310137.98
13    335270.58
14    472403.71
15    517005.11
16    403919.40
17    525423.69
18    666590.76
19    389841.79
20    355862.80
21    474038.34
22    449600.50
23    407785.08
Name: total_transfer, dtype: float64
```

In [41]:

```
mp.figure(figsize=(18, 9))
sns.barplot(x='hour', y='total_transfer' , data=df, ci=None, estimator=np.mean)
mp.title("Average usage of internet per hour")
mp.show()
mp.clf()
```



<Figure size 432x288 with 0 Axes>

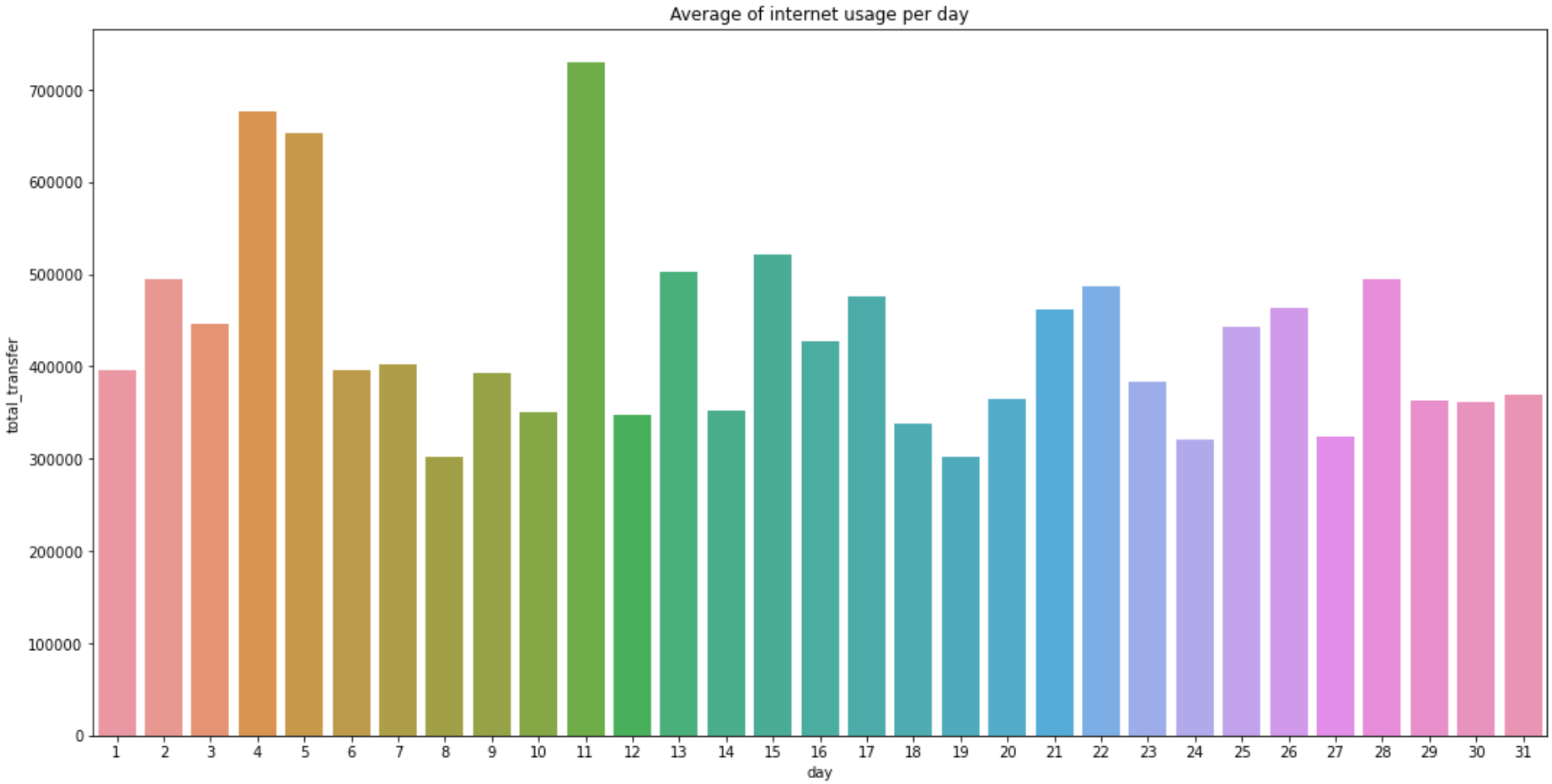
From the above graph we can conclude that the highest of average internet usage is done at 18:00 hrs and lowest at 5:00 hr.

In [45]:
```
# Per day
average_day = df.groupby('day').total_transfer.mean()
print('Average of internet usage per day:\n ' +  str(round(average_day, 2)))
```

```
Average of internet usage per day:
 day
1     396705.04
2     494496.48
3     445865.63
4     676332.03
5     652195.66
6     396261.75
7     402259.89
8     301859.57
9     393521.97
10    350665.02
11    729857.65
12    346695.95
13    501906.70
14    352701.10
15    521520.51
16    426719.39
17    475795.71
18    337490.93
19    301941.32
20    365130.12
21    462211.69
22    486595.37
23    383153.93
24    320598.94
25    443689.47
26    463432.02
27    324318.12
28    494576.34
29    363645.61
30    361418.88
31    369118.01
Name: total_transfer, dtype: float64
```

In [46]:

```
mp.figure(figsize=(18, 9))
sns.barplot(x='day', y='total_transfer' , data=df, ci=None, estimator=np.mean)
mp.title("Average of internet usage per day")
mp.show()
mp.clf()
```



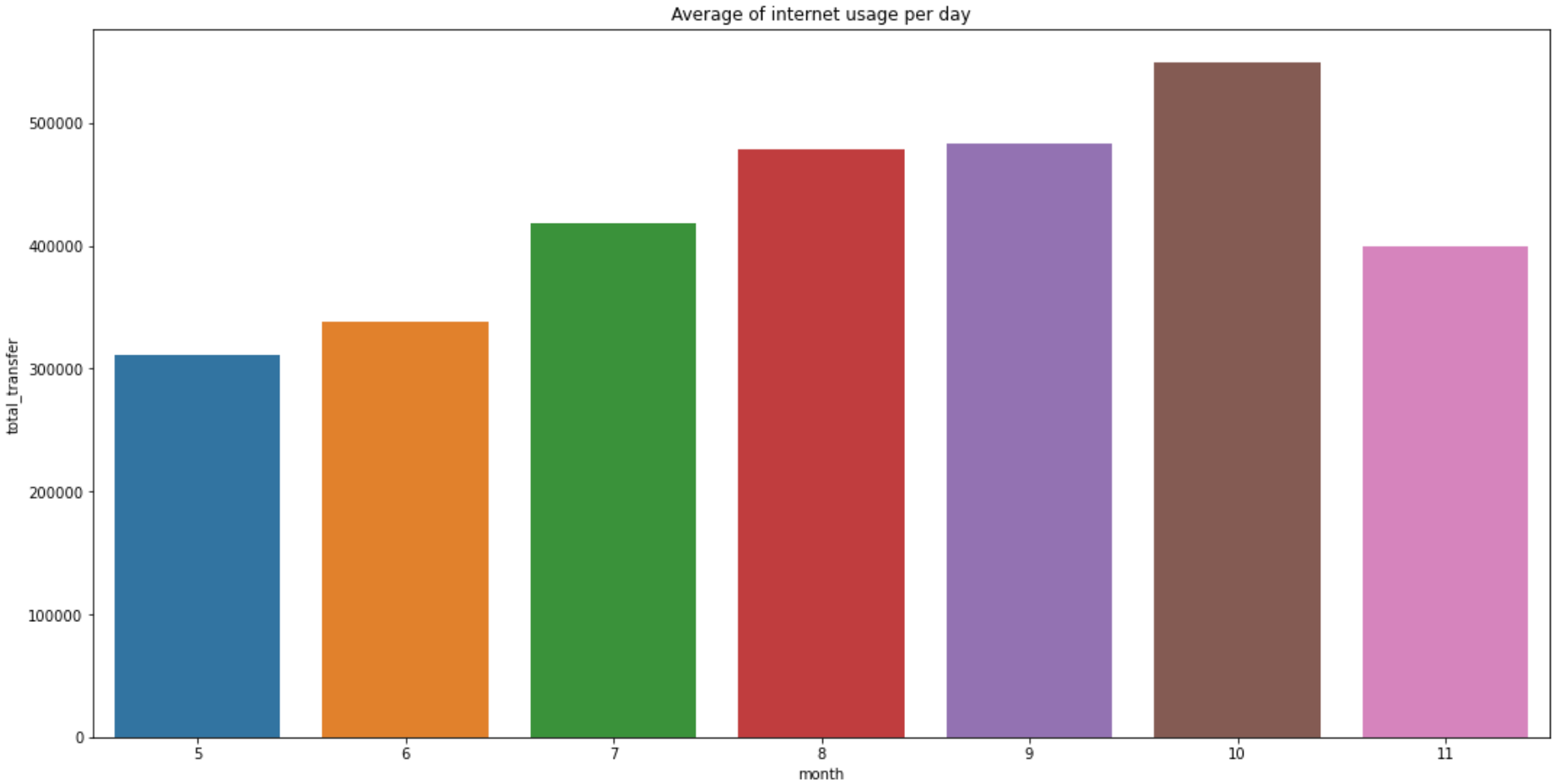
<Figure size 432x288 with 0 Axes>

In [47]:
```
# Per month
average_m = df.groupby('month').total_transfer.mean()
print('Average of internet usage per day:\n ' +  str(round(average_m, 2)))
```

```
Average of internet usage per day:
 month
5     311177.16
6     338418.08
7     418583.99
8     479042.44
9     482955.52
10    549467.63
11    399804.11
Name: total_transfer, dtype: float64
```

In [48]:

```
mp.figure(figsize=(18, 9))
sns.barplot(x='month', y='total_transfer' , data=df, ci=None, estimator=np.mean)
mp.title("Average of internet usage per day")
mp.show()
mp.clf()
```




```
<Figure size 432x288 with 0 Axes>
```

From the above graph we can conclude that users used internet the most in the month of October with the total transfer of 549467.63 and least in the month of May with the total transfer of 311177.16.