In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import pandahouse
from scipy import stats
from tqdm import tqdm_notebook
import ipywidgets
import swifter
```

Применение алгоритма линеаризации целевой метрики для повышения чувствительности статистических тестов

Экспериментальные группы:

exp_group = 1 – всё по-старому

exp_group = 2 – рекомендации «похожих на лайкнутые постов»

exp_group = 0 – всё по-старому

exp_group = 3 – рекомендации «постов, которые лайкали похожие на вас люди»

Загружаем данные

In [5]:
```
connection = {
    'host': 'https://clickhouse.lab.karpov.courses',
    'password': 'dpo_python_2020',
    'user': 'student',
    'database': 'simulator_20231220'
}
```

In [6]:
```
query03 = '''SELECT
    exp_group,
    user_id,
    sum(action = 'like') AS likes,
    sum(action = 'view') AS views,
    likes/views AS ctr
FROM simulator_20231220.feed_actions
WHERE
    toDate(time) BETWEEN '2023-11-18' and '2023-11-24'
AND
    exp_group IN (0,3)
GROUP BY
    exp_group,
    user_id'''

df03 = pandahouse.read_clickhouse(query03, connection = connection)
```

In [7]:
```
df03.head()
```

Out[7]:

	exp_group	user_id	likes	views	ctr
0	3	115383	12	44	0.272727
1	3	123580	2	11	0.181818
2	0	4944	8	41	0.195122
3	0	4504	5	15	0.333333
4	0	121508	6	25	0.240000

In [8]:
```
query12 = '''SELECT
    exp_group,
    user_id,
    sum(action = 'like') AS likes,
    sum(action = 'view') AS views,
    likes/views AS ctr
FROM simulator_20231220.feed_actions
WHERE
    toDate(time) BETWEEN '2023-11-18' and '2023-11-24'
AND
    exp_group IN (1,2)
GROUP BY
    exp_group,
    user_id'''

df12 = pandahouse.read_clickhouse(query12, connection = connection)
```

In [9]:
```
df12.head()
```

Out[9]:

	exp_group	user_id	likes	views	ctr
0	1	109963	3	15	0.200000
1	1	26117	32	141	0.226950
2	1	138232	18	73	0.246575
3	1	26295	39	141	0.276596
4	1	18392	7	32	0.218750

Линеаризируем лайки

In [10]:
```
# Функция линеаризации CTR: linearized_likes = likes - CTRcontrol * views
def linerized_likes(likes, views, CTRcontrol):
    linerized_likes = likes - CTRcontrol * views
    return linerized_likes
```

In [11]:
```
CTRcontrol_03 = df03.loc[df03['exp_group'] == 0]['likes'].sum() / df03.loc[df03['ex
CTRcontrol_12 = df12.loc[df12['exp_group'] == 1]['likes'].sum() / df12.loc[df12['ex

CTRcontrol_03, CTRcontrol_12
```

Out[11]:
```
(0.20983799195924746, 0.2096041628394293)
```

In [12]:
```
df03['linearized_likes'] = df03.apply(
    lambda x: linerized_likes(x['likes'], x['views'], CTRcontrol_03), axis = 1
)
```

```
df12['linearized_likes'] = df12.apply(
    lambda x: linerized_likes(x['likes'], x['views'], CTRcontrol_12), axis = 1
)
```

Сравниваем группы 0 и 3

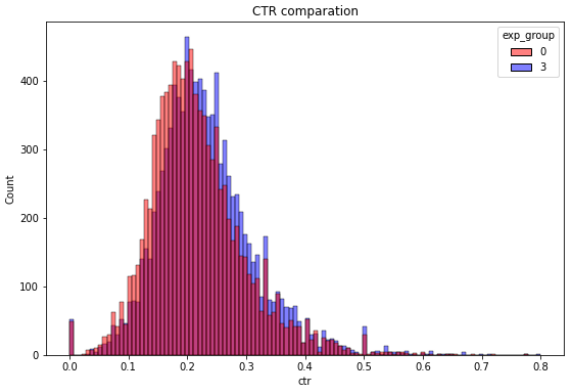
Расчёт

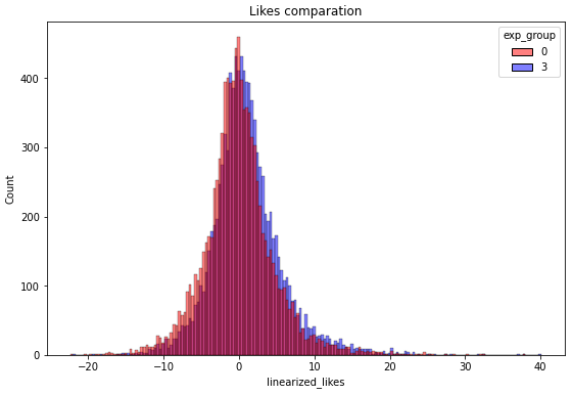
In [13]:
```
print(f'''
    Linearized likes test: {stats.ttest_ind(
        df03.loc[df03['exp_group'] == 0]['linearized_likes'],
        df03.loc[df03['exp_group'] == 3]['linearized_likes'],
        equal_var = False
    )}, \n
    CTR test: {stats.ttest_ind(
        df03.loc[df03['exp_group'] == 0]['ctr'],
        df03.loc[df03['exp_group'] == 3]['ctr'],
        equal_var = False
    )}'''
)
```

```
    Linearized likes test: Ttest_indResult(statistic=-16.186230032932844, pvalue=
1.4918137745326139e-58),

    CTR test: Ttest_indResult(statistic=-13.935320516755773, pvalue=6.216047483062
228e-44)
```

Визуализация

In [17]:
```
plt.figure(figsize=(20,6))
ctr_plot = plt.subplot(1,2,1)
ctr_plot.title.set_text('CTR comparation')
sns.histplot(
    data = df03,
    x = 'ctr',
    hue = 'exp_group',
    palette = ['r', 'b'],
    alpha = 0.5,
    kde = False)

likes_plot = plt.subplot(1,2,2)
likes_plot.title.set_text('Likes comparation')
sns.histplot(
    data = df03,
    x = 'linearized_likes',
    hue = 'exp_group',
    palette = ['r', 'b'],
    alpha = 0.5,
    kde = False)

plt.show()
```





Выводы по группам 0 и 3

После линеаризации лайков p-уровень значимости T-теста кратно уменьшился. На распределении видно снижение количество выбросов, а также стали более явно виден сдвиг на хвостах распределений групп.

Сравниваем группы 1 и 2

Расчёт

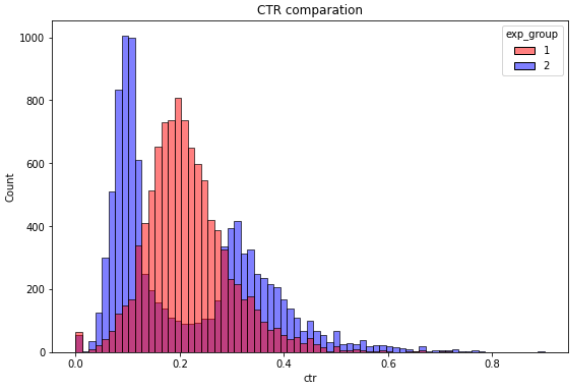
In [15]:
```
print(f'''
    Linearized likes test: {stats.ttest_ind(
        df12.loc[df12['exp_group'] == 1]['linearized_likes'],
        df12.loc[df12['exp_group'] == 2]['linearized_likes'],
        equal_var = False
    )}, \n
    CTR test: {stats.ttest_ind(
        df12.loc[df12['exp_group'] == 1]['ctr'],
        df12.loc[df12['exp_group'] == 2]['ctr'],
        equal_var = False
    )}'''
)
```

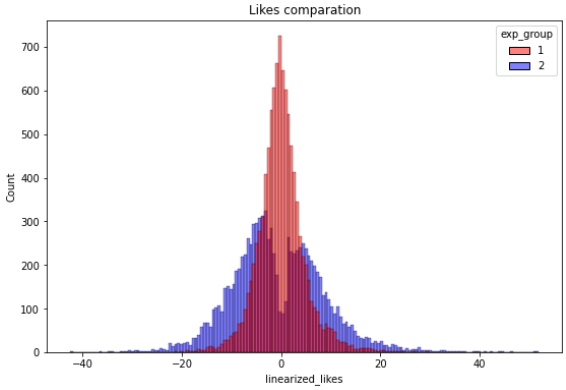
```
    Linearized likes test: Ttest_indResult(statistic=5.936377101934479, pvalue=2.9
805064038668164e-09),

    CTR test: Ttest_indResult(statistic=0.4051491913112757, pvalue=0.6853733311407
51)
```

Визуализация

In [16]:
```
plt.figure(figsize=(20,6))
ctr_plot = plt.subplot(1,2,1)
ctr_plot.title.set_text('CTR comparation')
sns.histplot(
    data = df12,
    x = 'ctr',
    hue = 'exp_group',
    palette = ['r', 'b'],
    alpha = 0.5,
    kde = False)

likes_plot = plt.subplot(1,2,2)
likes_plot.title.set_text('Likes comparation')
sns.histplot(
```

```
    data = df12,
    x = 'linearized_likes',
    hue = 'exp_group',
    palette = ['r', 'b'],
    alpha = 0.5,
    kde = False)

plt.show()
```





Выводы по группам 1 и 2

После линеаризации лайков p-уровень значимости T-теста кратно уменьшился. Более того, теперь T-тест видит различия между средними наших выборок. Также теперь более замены значения между пиковыми значениями распределений.

Однако, полученный результат не оказывает сильного влияния на интерпретацию графика: и в том, и в другом случае видно, что поведение пользователи тестовой группы разделилось на тех, у кого CTR вырос и тех, у кого он упал, причём последних ощутимо больше.