Ex. No. 6	**Real-Time Data Processing with Spark Streaming**
Youtube Link	**https://youtu.be/JwIXSCBvpws**
Date of Exercise	20.10.25

AIM

To process real-time streaming data using Apache Spark Streaming by counting the occurrence of words from a live data stream (e.g., data received from a TCP socket).

Procedure:

1. **Start Spark environment:**
   Ensure Spark and dependencies are properly installed.
2. **Set up a socket stream using Netcat:**
   Open a terminal and run:

   nc -lk 9999

   This opens a socket on port 9999 to simulate live data input.
3. **Write the Spark Streaming Python code** to receive and process text input from the socket.
4. **Run the Spark Streaming application:**
   Execute the Python code to start processing real-time data from the socket.
5. **Enter sample text into the Netcat terminal** and observe the real-time word count in the Spark terminal.

Program:

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
# Step 1: Initialize SparkContext and StreamingContext with 5-second batch interval
sc = SparkContext("local[2]", "NetworkWordCount")
ssc = StreamingContext(sc, 5)
# Step 2: Create a DStream that connects to localhost:9999
lines = ssc.socketTextStream("localhost", 9999)
# Step 3: Split each line into words
words = lines.flatMap(lambda line: line.split(" "))
# Step 4: Count each word in each batch
pairs = words.map(lambda word: (word, 1))
wordCounts = pairs.reduceByKey(lambda x, y: x + y)
# Step 5: Print the result
wordCounts.pprint()
# Step 6: Start the streaming computation
ssc.start()
# Step 7: Wait for the computation to terminate
ssc.awaitTermination()
```

Output:

Input in Netcat Terminal (localhost:9999):

spark streaming example

spark is powerful

Output in Spark Terminal:

Time: 2025-07-15 11:45:00

('spark', 1)

('streaming', 1)

('example', 1)

Time: 2025-07-15 11:45:05

('spark', 1)

('is', 1)

('powerful', 1)

Result :

The Spark Streaming application successfully received and processed real-time data from a socket stream. Word counts were generated for each 5-second batch of streaming text input, demonstrating the core functionality of real-time data processing using Apache Spark.