[HDFS-3436](https://www.google.com/url?q=https://issues.apache.org/jira/browse/HDFS-3436&sa=D&source=editors&ust=1751359166337146&usg=AOvVaw1QYluOg2IR4SNy-1Uw0xbo)

adding new datanode to existing pipeline fails in case of Append/Recovery

**(1)  Log information**

**(1.1) Roles in this case**

 DFSClient (client-side)  asks DN1(or DN2)(server-side) to transfer block

 DN1(or DN2) (client-side)  transfers block to DN4(server-side)

**(1.2) Symptoms**

Senario: There are 4 DNs in the cluster; File is written to 3DNs, DN1->DN2->DN3; Stop DN3; Now append is called.

DFSClient:

**2012-04-24 22:06:09,947** INFO  hdfs.DFSClient (DFSOutputStream.java:createBlockOutputStream(1063)) - Exception in createBlockOutputStream

java.io.IOException: Bad connect ack with firstBadLink as \*\*\*\*\*\*\*:50010

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.createBlockOutputStream(DFSOutputStream.java:1053)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.setupPipelineFor**Append**OrRecovery(DFSOutputStream.java:943)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:461)

After append is called, it finds that DN3 is broken when set up the pipeline for append

**2012-04-24 22:06:09,947** WARN  hdfs.DFSClient (DFSOutputStream.java:setupPipelineForAppendOrRecovery(916)) - Error Recovery for block BP-1023239-10.18.40.233-1335275282109:blk\_296651611851855249\_1253 in pipeline \*\*\*\*\*:50010, \*\*\*\*\*\*:50010, \*\*\*\*\*:50010: bad datanode \*\*\*\*\*\*:50010  (DN3)

Since there is a bad datanode(DN3) in pipeline, it starts error recovery for the block.

**2012-04-24 22:06:10,072** WARN  hdfs.DFSClient (DFSOutputStream.java:run(549)) - DataStreamer Exception

java.io.EOFException: Premature EOF: no length prefix available

                      at org.apache.hadoop.hdfs.protocol.HdfsProtoUtil.vintPrefixed(HdfsProtoUtil.java:162)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.transfer(DFSOutputStream.java:866)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.addDatanode2ExistingPipeline(DFSOutputStream.java:843)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.setupPipelineForAppendOr**Recovery**(DFSOutputStream.java:934)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:461)

When setup the pipeline for recovery, a new datanode DN4 will be added to the pipeline, then the block will be transferred from DN1(or DN2) to DN4.

During this recovery, EOF exception happens. When “Premature EOF: no length prefix available” occurs, it is likely that DataNode has stopped the data transfer, and it does not respond to DFSClient, so DFSClient will not be able to write and read data.

**2012-04-24 22:06:10,072** WARN  hdfs.DFSClient (DFSOutputStream.java:hflush(1515)) - Error while syncing

java.io.EOFException: Premature EOF: no length prefix available

                      at org.apache.hadoop.hdfs.protocol.HdfsProtoUtil.vintPrefixed(HdfsProtoUtil.java:162)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.transfer(DFSOutputStream.java:866)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.addDatanode2ExistingPipeline(DFSOutputStream.java:843)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.setupPipelineForAppendOrRecovery(DFSOutputStream.java:934)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:461)

java.io.EOFException: Premature EOF: no length prefix available

                      at org.apache.hadoop.hdfs.protocol.HdfsProtoUtil.vintPrefixed(HdfsProtoUtil.java:162)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.transfer(DFSOutputStream.java:866)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.addDatanode2ExistingPipeline(DFSOutputStream.java:843)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.setupPipelineForAppendOrRecovery(DFSOutputStream.java:934)

                      at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:461)

Now append to file fails due to addDatanode2ExistingPipeline is failed.

DataNode side:  (the log is from DN1 or DN2, which is assigned to transfer the block to DN4)

**2012-05-17 15:39:12,261** ERROR datanode.DataNode (DataXceiver.java:run(193)) - host0.foo.com:49744:DataXceiver error processing TRANSFER\_BLOCK operation  src: /127.0.0.1:49811 dest: /127.0.0.1:49744

java.io.IOException: BP-2001850558-xx.xx.xx.xx-1337249347060:blk\_-8165642083860293107\_1002 is neither a RBW nor a Finalized, r=ReplicaBeingWritten, blk\_-8165642083860293107\_1003, RBW

  getNumBytes()         = 1024

  getBytesOnDisk()  = 1024

  getVisibleLength()= 1024

  getVolume()           = E:\MyWorkSpace\branch-2\Test\build\test\data\dfs\data\data1\current

  getBlockFile()        = E:\MyWorkSpace\branch-2\Test\build\test\data\dfs\data\data1\current\BP-2001850558-xx.xx.xx.xx-1337249347060\current\rbw\blk\_-8165642083860293107

  bytesAcked=1024

  bytesOnDisk=102

at org.apache.hadoop.hdfs.server.datanode.DataNode.transferReplicaForPipelineRecovery(DataNode.java:2038)

                      at org.apache.hadoop.hdfs.server.datanode.DataXceiver.transferBlock(DataXceiver.java:525)

                      at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.opTransferBlock(Receiver.java:114)

                      at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.processOp(Receiver.java:78)

                      at org.apache.hadoop.hdfs.server.datanode.DataXceiver.run(DataXceiver.java:189)

                      at java.lang.Thread.run(Unknown Source)