**JVM HEAP MEMORY INCREASE DUE TO PROTOCAL VIOLATION ISSUE:**

This is quite different where the heap memory did not reach the peak utilization but the Managed servers reach 75% often and get back to normal. Heap reaching 75% is not an issue in usual JVM behaviour, but in our application we have configured in such a way that the node is put on STANDBY when the hap utilization increases to 75%.

The heap dump will help you to identify the issue only when the memory increases to peak utilization. We had touch time in nailing down the issue.

Initially we analysed the thread dumps taken during the issue reported time period. We could find requests were getting struck during the Weblogic servers trying to connect to database.

The issue can occur when there is change in JDBC driver, Database Upgrade and other firewall or network related changes in the environment.

**From Thread Dump:**

Please find the traces below,

SimpleAsyncTaskExecutor-14" prio=10 tid=0x00002aaabe1f6800 nid=0x1c5e runnable [0x0000000060a81000]

java.lang.Thread.State: RUNNABLE

at java.net.SocketInputStream.socketRead0(Native Method)

at java.net.SocketInputStream.read(SocketInputStream.java:129)

at oracle.net.ns.Packet.receive(Packet.java:300)

at oracle.net.ns.DataPacket.receive(DataPacket.java:106)

at oracle.net.ns.NetInputStream.getNextPacket(NetInputStream.java:315)

at oracle.net.ns.NetInputStream.read(NetInputStream.java:260)

at oracle.jdbc.driver.T4CMAREngine.getNBytes(T4CMAREngine.java:1517)

at oracle.jdbc.driver.T4CMAREngine.unmarshalNBytes(T4CMAREngine.java:1485)

at oracle.jdbc.driver.T4CMAREngine.unmarshalBuffer(T4CMAREngine.java:2039)

at oracle.jdbc.driver.T4CMAREngine.unmarshalCLR(T4CMAREngine.java:1805)

at oracle.jdbc.driver.T4CMAREngine.unmarshalCLR(T4CMAREngine.java:1718)

at oracle.jdbc.driver.T4CMAREngine.unmarshalCLR(T4CMAREngine.java:1936)

at oracle.jdbc.driver.T4CMAREngine.unmarshalDALC(T4CMAREngine.java:2279)

at oracle.jdbc.driver.T4CTTIoer.unmarshal(T4CTTIoer.java:275)

at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:447)

at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:192)

at oracle.jdbc.driver.T4C7Ocommoncall.doOLOGOFF(T4C7Ocommoncall.java:61)

at oracle.jdbc.driver.T4CConnection.logoff(T4CConnection.java:543)

- locked <0x000000070e355868> (a oracle.jdbc.driver.T4CConnection)

at oracle.jdbc.driver.PhysicalConnection.close(PhysicalConnection.java:3984)

**From WEBLOGIC Server logs -** "java.sql.SQLException: Protocol violation: [0]**:**

And also we could find the Protocol Violation error on logs during the issue reported time period as below,

####<Jul 9, 2015 2:15:48 AM EDT> <Error> <JDBC> <HOSTNAME> <pnbprod09-bc> <[ACTIVE] ExecuteThread: '15' for queue: 'weblogic.kernel.Default (self-tuning)'> <weblogic> <> <> <1436422548087> <BEA-001112> <Test "SELECT 1 FROM DUAL" set up for pool "DATASOURCE \_NAME" failed with exception: "java.sql.SQLException: Protocol violation: [0]".>

**COMMANDS FOR TROUBELSHOOTING THE SERVER LIFECYCLE DEBUGS:**

Please find the steps to enable the lifecycle debugs for the server , it will required to restart the server to take this change effect.   
  
Add below mentioned -d flag in the server startup script of JVM(Managed Server) or setdomainenv.sh file.   
  
-Dweblogic.slc=true   
  
-Dweblogic.slcruntime=true   
  
-Dweblogic.debug.DebugServerLifeCycle=true   
  
-Dweblogic.StdoutDebugEnabled=true

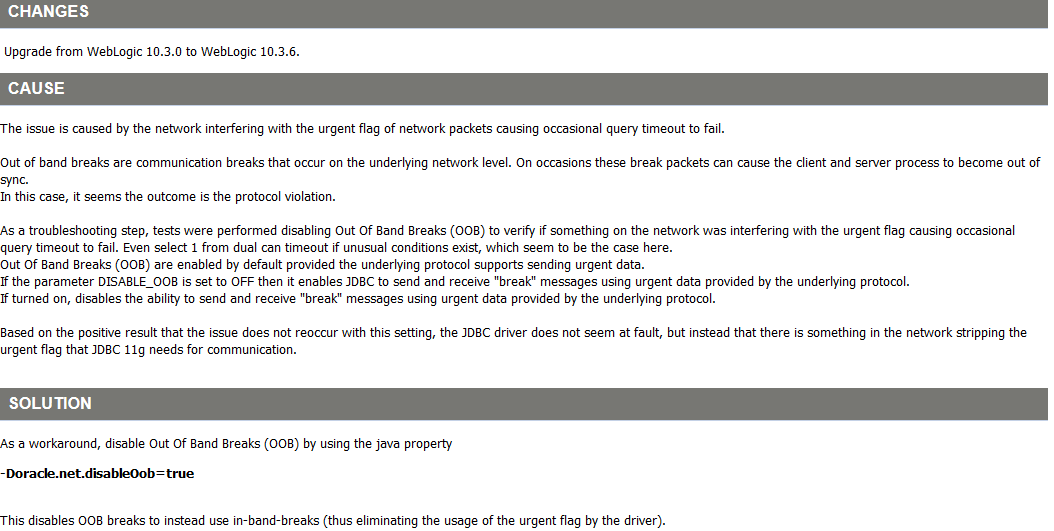
-XX:HeapDumpPath=/home/grd/pnb\_prod/HeapDump/pnbprod01

-XX:+HeapDumpOnOutOfMemoryError

**Oracle Document & suggestions in Weblogic End:**

During the Analysis we found an Oracle document related to the issue that we found in the log traces, and the same document has been provided to Oracle for their verification and suggestions.

From Oracle Doc,



**Oracle Recommendation in Database end:**

As recommended by Oracle, we had removed the parameter “SQLNET.EXPIRE\_TIME=10” in sqlnet.ora file.