|  |  |  |
| --- | --- | --- |
| **A discussion of Dead Connection Detection, Resource Limits, V$SESSION, V$PROCESS and OS processes (Doc ID 601605.1)** | [IMG_256](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=181165835411733%26id=601605.1%26_adf.ctrl-state=1a25pamq5g_498%20/o%20To%20Bottom)  [To Bottom](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=181165835411733&id=601605.1&_adf.ctrl-state=1a25pamq5g_498 \\o To Bottom) | IMG_257 |

IMG_258

IMG_259

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **In this Document**   |  |  | | --- | --- | |  | [Goal](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=181165835411733&id=601605.1&_adf.ctrl-state=1a25pamq5g_498 \\l GOAL) |  |  |  | | --- | --- | |  | [Solution](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=181165835411733&id=601605.1&_adf.ctrl-state=1a25pamq5g_498 \\l FIX) |  |  |  | | --- | --- | |  | [References](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=181165835411733&id=601605.1&_adf.ctrl-state=1a25pamq5g_498 \\l REF) |   IMG_260  **APPLIES TO:**  Oracle Database - Enterprise Edition - Version 8.0.3.0 and later  Oracle Net Services - Version 9.2.0.1 and later  Information in this document applies to any platform.  \*\*\*Checked for relevance on 02-Sep-2016\*\*\*  **GOAL**  Discuss what can be expected when implementing Dead Connection Detection (DCD) along with Database Resource Limits with user Profiles, inactive sessions and the effects on V$SESSION, V$PROCESS and OS processes  **SOLUTION**  **Dead Connection Detection (DCD)**  These are previously valid connections with the database but the connection between the client and server processes has terminated abnormally.  Examples of a dead connection:  - A user reboots/turns-off their machine without logging off or disconnecting from the database.  - A network problem prevents communication between the client and the server.  In these cases, the shadow process running on the server and the session in the database may not terminate.  Implemented by        \* adding SQLNET.EXPIRE\_TIME = <MINUTES> to the sqlnet.ora file  With DCD is enabled, the Server-side process sends a small 10-byte packet to the client process after the duration of the time interval specified in minutes by the SQLNET.EXPIRE\_TIME parameter.  If the client side connection is still connected and responsive, the client discards the probe packet, ..and another packet will be sent when next interval expires (assuming no other activity on the connection).  If the client fails and client OS sends back an error in responding to the DCD probe packet       \* the Server side process is marked as a dead connection and       \* PMON performs the clean up of the database processes / resources       \* The client OS processes are terminated  **NOTE**: SQLNET.RECV\_TIMEOUT can be set on the SERVER side sqlnet.ora file. This will set a timeout for the server process                   to wait for data from the client process.  **Inactive Sessions:**  These are sessions that remain connected to the database with a status in v$session of INACTIVE.  Example of an INACTIVE session:  - A user starts a program/session, then leaves it running and idle for an extended period of time.    **Database Resource Limits (using user profiles)**  Implemented by       \* Setting RESOURCE\_LIMIT = TRUE in the database startup parameter file (spfile or pfile)       \* Creating or modifying existing user profiles (DBA\_PROFILES) to have one or more resource limit       \* Assigning a profile to a user whose resources are wished to be limited  It could happen that if the idle\_time has been set on the DEFAULT profile, this can lead to an MTS dispatchers being set to 'sniped' and then getting 'cleaned up' via the shell script. The removal of the dispatcher will result in other sessions 'dying' .In that case, If you are to implement resource limits, may be advisable to create new profiles  that be assigned to users and not to change the characteristics of DEFAULT.  Alternatively, if you do change DEFAULT, ensure that all the properties that you  have affected have been fully tested in a development environment.  When a resource limit is exceeded (for example IDLE\_TIME) ... PMON does the following       \* Mark the V$SESSION as SNIPED       \* Clean up the database resources for the session       \* Remove the V$SESSION entry  When a resource limit is exceeded (for example IDLE\_TIME) ... PMON marks the session as SNIPED in V$SESSION.  Then, AFTER the SNIPED session tries to execute any SQL statement, its database resources are cleaned up and its V$SESSION entry is removed.  **Resource Manager**  Resource manager plans can be created to kill inactive sessions with high idle time. Refer to How To Automatic Kill Inactive Sessions using Resource Manager (Doc ID 1935739.1). This document contains the details with an example. You can customize the plan directives as per your requirement.  In this case, once the inactive time goes beyond the specified MAX\_IDLE\_TIME, PMON marks the session as KILLED. When this KILLED later tries to execute any SQL statement, its database resources are cleaned up and its V$SESSION entry is removed.  **It is strongly recommended that both DCD and Resource Limits with Profiles be implemented in order to clean up resources at both the database and OS level**  This combination **will not** clean up **IDLE / ABANDONED** **/ INACTIVE** connections (OS processes) as these sessions still have active clients  For this case we will see that :       \* PMON has cleaned up the V$SESSION entries .. but both the OS processes and the V$PROCESS entries will still exist       \* SQLNET will continue to be able to send the 10 byte packet successfully until the session is logged off  This condition can be a major problem as       \* The database exhausts PROCESSES and gives ORA-20 maximum number of processes <num> exceeded       \* The OS can become exhausted due to the un needed resources consumed by the abandoned processes  The SYMPTOMS of this condition are       \* The database view V$PROCESS will have no corresponding V$SESSION entry       \* An OS process / thread still exists for the SNIPED session  The solutions to this scenario can are to cleanup the OS processes ... after which the V$PROCESS entries should be removed automatically  **Methods to cleanup OS processes:**       \* UNIX : kill -x ... the OS process at the OS level (typically kill -9)       \* UNIX:  if using a dedicated server, use the following shell script to kill the shadow process (script has been tested on Solaris, AIX, Tru64 and  HPUX):  #!/bin/sh  tmpfile=/tmp/tmp.$$  sqlplus system/manager <<EOF  spool $tmpfile  select p.spid from v\$process p,v\$session s  where s.paddr=p.addr  and s.status in ('INACTIVE,'SNIPED');  spool off  EOF  for x in `cat $tmpfile | grep "^[0123456789]"`  do  kill -9 $x  done  rm $tmpfile  NOTE: If you are running in a shared server environment, you need to be careful not to accidentally kill your dispatchers and/or shared servers. In Oracle 10.2 (or higher) a dedicated connections V$SESSION + V$PROCESS + OS Process can be cleaned up with  ALTER SYSTEM DISCONNECT SESSION '<SID>,<SERIAL>' IMMEDIATE  At this point in versions prior to 10.2 and for shared server connections the only solution is to kill the session at the OS level (see Kill and ORAKILL above)       \* Windows : use the orakill command .... orakill <ORACLE SID> <Thread ID> (see [Note 69882.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=601605.1&id=69882.1) for details)  On occasions we see conditions where a database session has a V$SESSION.STATUS = SNIPED ... and the entry never goes away . This condition can be achieved by implementing Database Resource Limits + Profiles without DCD and allow the database session to exceed the limit in the profile    **Summary of what was discussed here:**  1) DCD initiates clean up of OS and database processes  that have disconnected / terminated abnormally  2) DCD **will not** initiate clean up sessions that are still connected ... but are idle / abandoned / inactive  3) Database Resource Limits + user Profiles clean up database resources for user sessions that exceed resource limits  4) Database Resource Limits + user Profiles **will not** clean up OS processes  5) If DCD and Database Resource Limits + user Profiles are used in combination .. **Dead** Connections OS and Database Resources will be cleaned up  6) **IDLE / ABANDONED / INACTIVE** sessions OS processes **will not** be cleaned up even if DCD and Database Resource Limits + user Profiles are used in combination ... these must be cleaned up manually  7) Resource Manager can be used to kill inactive session, which exceeds the idle time mentioned in the plan directives.  **REFERENCES**  [NOTE:805586.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=601605.1&id=805586.1) - Troubleshooting Session Administration  [NOTE:151972.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=601605.1&id=151972.1) - Dead Connection Detection (DCD) Explained  [NOTE:438923.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=601605.1&id=438923.1) - How To Track Dead Connection Detection(DCD) Mechanism Without Enabling Any Client/Server Network Tracing  [NOTE:1287854.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=601605.1&id=1287854.1) - Troubleshooting Guide - ORA-20: Maximum Number Of Processes (%S) Exceeded  NOTE:1935739.1 - How To Automatic Kill Inactive Sessions using Resource Manager  [NOTE:395505.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=601605.1&id=395505.1) - How to Check if Dead Connection Detection (DCD) is Enabled in 9i ,10g and 11g  [NOTE:1682295.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=601605.1&id=1682295.1) - How to calculate the proper value from processes, sessions, and transactions |