

How to Configure TCP Keep-Alive Parameters

Table of contents

1. How to Configure TCP Keep-Alive Parameters	3
2. Procedure	4
2.1 Check Current TCP Keep-Alive Settings	4
2.2 Apply Temporary (Dynamic) Changes	5
2.3 Test the Settings	6
2.4 Make Changes Permanent	7
2.5 Estimate Connection Timeout	8
2.6 Tuning Recommendations	9

1. How to Configure TCP Keep-Alive Parameters

This guide explains how to configure and tune TCP Keep-Alive parameters in OpenVMS TCP/IP Services.

It also explains how to estimate the time required to detect and close a broken TCP connection.

2. Procedure

2.1 Check Current TCP Keep-Alive Settings

Run the following command to display current TCP parameters:

```
$ SYSCONFIG -Q INET
```

Note

If SYSCONFIG is not defined in your current process, run

\$ @SYS\$MANAGER:TCPIP\$DEFINE_COMMANDS or invoke the command as

\$ TCPIP SYSCONFIG -Q INET.

Example output:

```
TCP_KEEPAIVE_DEFAULT = 0
TCP_KEEPIIDLE = 14400
TCP_KEEPCNT = 8
TCP_KEEPIINTVL = 150
```

Parameter descriptions:

Parameter	Description
TCP_KEEPAIVE_DEFAULT	Enables (1) or disables (0) Keep-Alive by default
TCP_KEEPIIDLE	Idle time before sending the first probe (half-second units)
TCP_KEEPCNT	Number of retry probes before giving up
TCP_KEEPIINTVL	Time between retry probes (half-second units)

2.2 Apply Temporary (Dynamic) Changes

You can modify these settings without rebooting:

```
$ SYSCONFIG -R INET TCP_KEEPA_LIVE_DEFAULT=1
$ SYSCONFIG -R INET TCP_KEEPI_DLE=60
$ SYSCONFIG -R INET TCP_KEEPCNT=4
$ SYSCONFIG -R INET TCP_KEEPI_NTVL=30
```

Note

This will only apply to sockets built after the Keep-Alive setup.

2.3 Test the Settings

To test the configuration restart the telnet service using:

```
$ @SYS$MANAGER:TCPIP$TELNET_SHUTDOWN  
$ @SYS$MANAGER:TCPIP$TELNET_STARTUP
```

Important

This disconnects all active Telnet sessions.

Then carry out the following:

- Establish an inbound Telnet connection
- Monitor the process related the TNA device
- Disconnect the network (for example, unplug the cable)
- Observe how long it takes before the connection is terminated

2.4 Make Changes Permanent

To persist settings across reboots, edit:

```
TCPIP$ETC:SYSCONFIGTAB.DAT
```

Important

The parameters shown in the following examples are provided for illustrative purposes only and should not be interpreted as recommended values. Replace them with values appropriate for your environment.

Example configuration:

```
INET:
    TCP_KEEPA_LIVE_DEFAULT=1
    TCP_KEEPI_DLE=60
    TCP_KEEPCNT=4
    TCP_KEEPI_NTVL=30
```

2.5 Estimate Connection Timeout

Note

The values of TCP_KEEPIDLE and TCP_KEEPINTVL are in half seconds. For example, setting TCP_KEEPINTVL to 30 results in an interval of 15 seconds.

Considering the earlier example, you can estimate the failure detection time (D) window using this formula:

$$(\text{INTVL}) \times (\text{CNT} + 1) < D < (\text{IDLE}) + (\text{INTVL}) \times (\text{CNT} + 1)$$

Example:

$$(15) \times (4+1) < D < (30) + (15) \times (4+1)$$

$$75 \text{ seconds} < D < 105 \text{ seconds}$$

The detection time (D) is between 75 and 105 seconds, depending on when the failure occurs. In the worst case, the connection fails immediately after activity on the connection, forcing TCP to wait the full idle period first.

The sequence is:

- Wait "TCP_KEEPIDLE" (30s), send a probe (the disconnection occurs somewhere in that period)
- Wait "TCP_KEEPINTVL" (15s), send a probe
- Repeat this last operation "TCP_KEEPCNT" times (15s x 4 = 60s)
- Wait for an answer "TCP_KEEPINTVL" (15s) (This explains the (keepcnt+1) of the above formula)

2.6 Tuning Recommendations

When a connection is made over an expensive link (for example, ISDN), use a large idle value to reduce network traffic:

```
TCP_KEEPIDLE=14400    ! Corresponds to 7200 seconds (2 hours) between probes
```

If a connection failure must be detected as quickly as possible, use a small value. For example:

```
TCP_KEEPIDLE=150      ! Corresponds to 75 seconds before first probe
```