# Appendix E

# Configurable Options: Update for BSD/OS 2.1 and Solaris 2.5

This document is an update to Appendix E of *TCP/IP Illustrated, Volume 1: The Protocols* by W. Richard Stevens (Addison-Wesley, 1994), which we refer to in this update as just *the book*. Appendix E of the book describes the TCP/IP configurable options for BSD/386 Version 1.0 and Solaris 2.2 (among others). This update describes the differences from that appendix in the configurable options for BSD/OS 2.1 and Solaris 2.5. This update is not "stand-alone" but is meant to be read in conjunction with the book.

Also be aware that the current errata for the book can be obtained from the author's home page: `http://www.noao.edu/~rstevens`.

Many thanks to Mukesh Kacker of SunSoft for his help with the changes in Solaris 2.5.

## E.1 BSD/OS Version 2.1

This system has changed from the classical BSD configuration to the 4.4BSD approach, using the `sysctl` command (described in Section E.6 of the book).

Most of the parameters affect a single kernel variable, which we show first in our discussion in parentheses. Readers interested in additional details concerning the implementation can look up these variable names in the index of *TCP/IP Illustrated, Volume 2: The Implementation* by Gary R. Wright and W. Richard Stevens (Addison-Wesley, 1995).

 

## IP

`net.inet.ip.forwarding`
(`ipforwarding`) If 0, IP datagrams are not forwarded. If nonzero, IP datagrams are forwarded (that is, the system will act as a router). The initial value is taken from the constant `IPFORWARDING`, if defined when the kernel is compiled, otherwise the initial value is 1 if the constant `GATEWAY` is defined.

`net.inet.ip.forwsrcrt`
(`ipforward_srcrt`) If 0, source routed IP datagrams are not forwarded. If nonzero, source routed IP datagrams are forwarded. The initial value is taken from the constant `IPFORWARDSRCRT`, if defined when the kernel is compiled, otherwise the initial value is 1 if the constant `IPFORWARDING` is defined and if the constant `GWSCREEN` is not defined.

`net.inet.ip.redirect`
(`ipsendredirects`) If 0, ICMP redirects are not sent when forwarding IP datagrams. If nonzero, ICMP redirects will be sent. The initial value is taken from the constant `IPSENDREDIRECTS`, if defined when the kernel is compiled, otherwise the initial value is 1.

`net.inet.ip.ttl`
(`ip_defttl`) The default TTL for both TCP and UDP. The default is 64.

## ICMP

`net.inet.icmp.maskrepl`
(`icmpmaskrepl`) If 0, replies are not sent for ICMP address mask requests. If nonzero, replies are sent for ICMP address mask requests. The initial value is 1 if the constant `GATEWAY` is defined when the kernel is compiled.

`net.inet.icmp.rediraccept`
(`icmp_rediraccept`) If 0, received ICMP redirects are ignored. If nonzero, received ICMP redirects are processed. The default value is 1.

`net.inet.icmp.redirtimeout`
(`icmp_redirtimeout`) This variable is not currently used.

## UDP

`net.inet.udp.checksum`
(`udpcksum`) If nonzero, UDP checksums are calculated for outgoing UDP datagrams. If 0, outgoing UDP datagrams do not contain a checksum. Unlike earlier BSD releases, BSD/OS 2.1 always verifies the checksum of a received UDP datagram (if the sender calculated the checksum). Default is 1.

`net.inet.udp.recvspace`
(`udp_recvspace`) The default size in bytes of the UDP receive buffer. Default is 41600, allowing for 40 1024-byte datagrams.

net.inet.udp.sendmax
(udp_sendspace) The default size in bytes of the UDP send buffer. The size of this buffer is the largest UDP datagram that an application can send. Default is 9216.

### TCP

The four timer variables, tcp_conntimeo, tcp_keepintvl, tcp_keepintvl, and tcp_maxpersistidle are maintained by the kernel in units of 500-ms clock ticks, but are input and output by sysctl in seconds.

net.inet.tcp.conntimeo
(tcp_conntimeo) The amount of time TCP waits for an acknowledgment of a SYN, for either an active open or a passive open. Default is 75 seconds.

net.inet.tcp.do_rfc1323
(tcp_do_rfc1323) If nonzero, TCP will send the window scale and timestamp options (defined in RFC 1323) with the SYN corresponding to an active open, or with the SYN in response to a peer's SYN if the peer sends the options. Default is 1.

net.inet.tcp.keepcount
(tcp_keepcnt) The number of consecutive keepalive probes that TCP will send, minus one, before considering the peer as dead. Default is 8 (e.g., TCP will send nine keepalive probes). In Figure 23.1, segments 12 through 20 correspond to nine keepalive probes followed by an RST (segment 21). This is also shown in Figure 25.17 of Volume 2 of this series.

net.inet.tcp.keepidle
(tcp_keepidle) The amount of time a TCP connection must be idle before the first keepalive probe is sent (e.g., the time between segments 11 and 12 in Figure 23.1). Default is 7200 seconds (2 hours).

net.inet.tcp.keepinterval
(tcp_keepintvl) The amount of time between consecutive keepalive probes. Default is 75 seconds.

net.inet.tcp.maxpersistidle
(tcp_maxpersistidle) The maximum amount of time that TCP will send persist probes without receiving a response from the peer before dropping the connection. Section 14.9 of Volume 3 in this series examines this feature in detail, especially with respect to Web servers. Default is 7200 seconds (2 hours).

net.inet.tcp.mssdflt
(tcp_mssdflt) The default TCP MSS for nonlocal destinations. Be aware that this value is normally rounded down to the next multiple of 256 (e.g., 536 becomes 512). Default is 536.

net.inet.tcp.recvspace
(tcp_recvspace) The default size of the TCP receive buffer. Default is 8192.

net.inet.tcp.sendspace
(tcp_sendspace) The default size of the TCP send buffer. Default is 8192.

## E.4 Solaris 2.5

This section notes the parameters that have been *deleted*, describes the *new* parameters that have been added, and describes any parameters whose default value or description has *changed*.

### /dev/ip

Notice that the permissions on `/dev/ip` have changed from earlier Solaris releases from read–write for everyone to read–write only for owner (`root`) and group (`sys`). Therefore you may not be able to query the following values if you do not have adequate permission.

`ip_broadcast_ttl` (new)
: Default TTL for outgoing IP datagrams destined for a broadcast address (overriding the value specified by `ip_def_ttl`). Default is 1 to prevent any router from forwarding the datagram. This value is also used as the outgoing TTL for directed broadcasts that are forwarded (see the `ip_forward_directed_broadcasts` flag).

`ip_cksum_choice` (deleted)

`ip_icmp_err_interval` (new)
: The minimum amount of time between consecutive ICMP error messages (redirects, destination unreachable, time exceeded, parameter problem, and source quench) generated by this host. Default is 500 ms. Notice that this default causes Traceroute from some other host to or through a Solaris 2.5 host to time out its second probe if the round trip time to the Solaris 2.5 host is less than 500 ms.

`ip_ignore_redirect` (new)
: If 0 (default), received ICMP redirect messages are processed. If nonzero, received ICMP redirect messages are ignored.

`ip_ire_pathmtu_interval` (changed)
: The default value of this parameter changed with Solaris 2.3 from 30000 ms (30 sec.) to 600000 ms (10 minutes). This corresponds to the comment at the bottom of p. 156.

`ip_local_cksum` (deleted)

`ip_output_queue` (new)
: (Debug) Determines what happens for non-TCP data when the driver or module below IP is flow controlled. If nonzero (default) the data is queued by IP. If 0, the data is discarded and a source quench is passed up the stream.

`ip_reass_queue_bytes` (new)
: The limit for the IP reassembly queue per physical interface. Default is 1,000,000 bytes. If this limit is exceeded, IP packets awaiting reassembly are deleted, oldest first, until the limit is not exceeded.

`ip_respond_to_address_mask` (deleted)

`ip_respond_to_address_mask_broadcast` (new)
If 0 (default), the host does not respond to ICMP address mask requests that are sent to a broadcast address. If 1, the host does respond. Address mask requests destined to a unicast address are always responded to.

`ip_send_source_quench` (changed)
The default value of this parameter has changed from 1 (send source quenches) to 0 (do not send source quenches). This corresponds to current Internet thinking about ICMP source quench errors (p. 161 of the book and RFC 1812).

`ip_strict_dst_multihoming` (new)
If 0 (default) and if the host is multihomed, the host will accept IP datagrams destined for any of its IP addresses, regardless of the interface on which the datagram arrives. This is called the *weak end system* model in RFC 1122. If 1, the host will only accept IP datagrams that arrive on the interface specified by the destination IP address. This is called the *strong end system* model in RFC 1122.

### /dev/icmp

Notice that the permissions on `/dev/icmp` have changed from earlier Solaris releases from read–write for everyone to read–write only for owner (`root`). Therefore you will not be able to query the following values unless you have root permission.

`icmp_max_buf` (new)
The maximum size of an ICMP send or receive buffer. Default is 262144 bytes.

`icmp_recv_hiwat` (new)
(Debug) The receive high-water mark. Default is 8192 bytes.

`icmp_xmit_hiwat` (new)
(Debug) The transmit high-water mark. Default is 8192 bytes.

`icmp_xmit_lowat` (new)
(Debug) The transmit low-water mark. Default is 1024 bytes.

### /dev/arp

No changes to the three parameters in this module.

### /dev/udp

`udp_max_buf` (new)
The maximum size of a UDP send or receive buffer. Default is 262144 bytes.

`udp_recv_hiwat` (new)
(Debug) The receive high-water mark. Default is 8192 bytes.

`udp_xmit_hiwat` (new)
: (Debug) The transmit high-water mark. Default is 8192 bytes.

`udp_xmit_lowat` (new)
: (Debug) The transmit low-water mark. Default is 1024 bytes.

## /dev/tcp

`tcp_bind_hash` (new)
: (Read-only) Prints information about all TCP endpoints with a state that is greater than or equal to the bound state.

`tcp_co_min` (new)
: (Debug) One of the parameters that determines when the intergrated checksum-and-copy is used for received segments. This parameter is the segment length of the received segment. Default is 64 bytes.

`tcp_conn_hash` (new)
: (Read-only) Prints information about all TCP endpoints with a state that is greater than or equal to the connected state.

`tcp_conn_req_max` (changed)
: The default for this parameter has changed from 5 to 32. Additionally, this parameter can now be set as high as 1024. Section 14.5 of Volume 3 in this series examines this parameter in detail, especially with respect to Web servers. Solaris 2.5.1 has a new counter, `tcpListenDrop`, that can be printed with `netstat -s`, that prints the number of incoming SYNs that are dropped (ignored) becuase there is no room on the listening endpoint's queue.

`tcp_cwnd_max` (changed)
: The default for this parameter has changed from 32768 to 65535.

`tcp_eager_listeners` (deleted)

`tcp_fin_wait_2_flush_interval` (new)
: The length of time an endpoint remains in the FIN_WAIT_2 state after the application does a close (not a half-close). See the end of Section 18.6 for a discussion of this feature. Default is 675000 ms (11 min. and 15 sec.).

`tcp_ip_abort_cinterval` (changed)
: The default for this parameter has changed from 240000 ms (4 min.) to 180000 ms (3 min.).

`tcp_ip_abort_interval` (changed)
: The default for this parameter has changed from 120000 ms (2 min.) to 480000 ms (8 min.).

`tcp_listen_hash` (new)
: (Read-only) Prints information about all TCP endpoints in the listen state.

`tcp_max_buf` (new)
: The maximum size of the TCP send or receive buffer. Default is 262144.

`tcp_queue_hash` (new)
(Read-only) Prints information about all TCP endpoints.

`tcp_naglim_def` (changed)
The default for this parameter has changed from 65535 to 4095.

`tcp_recv_hiwat` (new)
(Debug) This is one of the values that affects the heuristic used to select the default receive buffer size. Default is 8192 bytes. The actual receive buffer size is rounded up to be a multiple of the MSS, so that on an Ethernet with an MSS of 1460, for example, the default receive buffer size (and hence the advertised window) will be 8760.

`tcp_recv_hiwat_minmss` (new)
(Debug) This parameter guarantees that the default receive buffer size will always be at least `tcp_recv_hiwat_minmss` (default of 4) multiplied by the MSS. The actual value of `tcp_recv_hiwat` for a given connection is the maximum of `tcp_recv_hiwat` and `tcp_recv_hiwat_minmss` multiplied by the MSS, with the final result rounded up to a multiple of the MSS. For example, on an Ethernet with an MSS of 1460, the default `tcp_recv_hiwat` of 8192 is greater than 5840 ($4 \times 1460$), so 8192 is rounded up to 8760. But if we set `tcp_recv_hiwat_minmss` to 16, then the default receive buffer size will be 23360 ($16 \times 1460$).

`tcp_rtt_updates` (new)
The number of RTT updates applied to a given TCP connection before the RTT estimator is cached as part of the route. A value of 0 (the default) means do not cache the RTT estimator.

`tcp_strong_iss` (new)
If nonzero (the default), the initial send sequence number for a connection is randomized.

`tcp_xmit_hiwat` (new)
(Debug) The initial send buffer size, which will be rounded up to a multiple of the MSS. Default is 8192.

`tcp_xmit_lowat` (new)
(Debug) The initial low-water mark for the send buffer. Default is 2048.

`tcp_zero_win_probesize` (new)
The number of bytes of data to send when probing a zero window (Chapter 22). Default is 1.