附录E 配置选项

我们已经看到了许多冠以“依赖于具体配置”的TCP/IP特征。典型的例子包括是否使能UDP的检验和（11.3节），具有同样的网络号但不同的子网号的目的IP地址是本地的还是非本地的（18.4节）以及是否转发直接的广播（12.3节）。实际上，一个特定的TCP/IP实现的许多操作特征都可以被系统管理员修改。

这个附录列举了本书中用到的一些不同的TCP/IP实现可以配置的选项。就像你可能想到的，每个厂商都提供了与其他实现不同的方案。不过，这个附录给出的是不同的实现可以修改的参数类型。一些与实现联系紧密的选项，如内存缓存池的低水平线，没有描述。

> 这些描述的变量只用于报告的目的。在不同的实现版本中，它们的名字、默认值、或含义都可以改变。所以你必须检查你的厂商的文档（或向他们要更充分的文档）来了解这些变量实际使用的单词。

这个附录没有覆盖每次系统引导时发生的初始化工作：对每个网络接口使用`ifconfig`进行初始化（设置IP地址、子网掩码等等）、往路由表中输入静态路由等等。这个附录集中描述了影响TCP/IP操作的那些配置选项。

E.1 BSD/386版本1.0

这个系统是自从4.2BSD以来使用的“经典”BSD配置的一个例子。因为源代码是和系统一起发布的，所以管理员可以指明配置选项，内核也可重编译。存在两种类型的选项：在内核配置文件中定义的常量（参见`config`(8)手册）和在不同的C源文件中的变量初始化。大胆而又经验丰富的管理员也可以使用排错工具修改正在运行的内核或者内核的磁盘映像中这些变量的值，以避免重新构造内核。

下面列出的是在内核配置文件中可以修改的常量。

IPFORWARDING

这个常量的值初始化内核变量`ipforwarding`。如果值为0（默认），就不转发IP数据报。如果是1，就总是使能转发功能。

GATEWAY

如果定义了这个常量，就使得IPFORWARDING的值被置为1。另外，定义这个常量还使得特定的系统表格（ARP快速缓存表和路由表）更大。

SUBNETSARELOCAL

这个常量的值初始化内核变量`subnetsarelocal`。如果值为1（默认），一个和发送主机具有同样网络号、但不同子网号的目的IP地址被认为是本地的。如果是0，只有在同一个子

网的目的IP地址才认为是本地的。图E-1总结了上述规律。

网络标识符	子网标识符	subnetsarelocal		注释
		1	0	
相同	相同	本地	本地	总是本地的
相同	不同	本地	非本地	依赖于配置
不同		非本地	非本地	总是非本地的

图E-1 对subnetarelocal内核变量的理解

这个变量的值影响了TCP选择的MSS。当给一个本地的目的地址发送报文时，TCP选择的是基于输出接口的MTU的MSS。而发送给一个非本地的地址时，TCP使用变量 `tcp_mssdflt` 作为MSS。

IPSENDREDIRECTS

这个常量的值初始化内核变量 `ipsendredirects`。如果值为1（默认），主机在转发IP数据报时，将发送ICMP重定向。如果是0，不发送ICMP重定向。

DIRECTED_BROADCAST

如果值为1（默认），如果收到的数据报的目的地址是主机的一个接口的直接广播地址，就将它作为一个链路层的广播来转发。如果是0，这些数据报就会被丢弃。

下面的变量也可以改变，它们在目录 /usr/src/sys/netinet中的不同文件中定义。

`tcprexmtthresh`

引起快速重传和快速恢复算法的连续ACK的数目。默认值是3。

`tcp_ttl`

TCP段的TTL字段的默认值。默认值是60。

`tcp_mssdflt`

用于非本地目的地址的默认的TCP MSS。默认值是512。

tcp_keepidle

在发送一个`keepalive`探测报文之前必须等待的500 ms时钟间隔的次数。默认值是14 400（2个小时）。

`tcp_keepintvl`

如果没有收到响应，在两个连续的`keepalive`探测报文之间等待的500 ms时钟间隔的次数。默认值是150（75秒）。

`tcp_sendspace`

TCP发送缓存的默认大小。默认值是4096。

`tcp_recvspace`

TCP接收缓存的默认大小。这个值影响了提供的窗口大小。默认值是4096。

`udpcksum`

如果非0，对输出的UDP数据报计算UDP检验和，并且对于包含了非0检验和的输入UDP数据报要验证它们的检验和。如果值为0，不计算输出的UDP数据报的检验和，也不验证输入UDP数据报的检验和，即使发送者计算了一个检验和。默认值是1。

udp_ttl
UDP数据报TTL字段的默认值。默认值是30。
udp_sendspace
UDP发送缓存的默认大小。定义了可以发送最大的UDP数据报。默认值是9126。
udp_recvspace
UDP接收缓存的默认大小。默认值是41 600，允许40个1024字节的数据报。

E.2 SunOS 4.1.3

SunOS 4.1.3使用的方法类似于我们在BSD/386中看到的。因为大部分的内核源代码都没有发布，所以所有的C变量初始化都包含在一个提供的C源文件中。

管理员的内核配置文件（参见config(8)手册）可以定义下面的变量。修改了配置文件之后，需要构造一个新的内核，然后重启动。

IPFORWARDING

这个常量的值初始化内核变量ip_forwarding。如果值为-1，就不转发IP数据报，而且变量的值不能再改变。如果是0（默认），不转发IP数据报，但是如果多个接口都工作，变量的值可以修改为1。如果是1，就总是能转发IP数据报。

SUBNETSARELOCAL

这个常量的值初始化内核变量ip_subnetsarelocal。如果值为1（默认），一个和发送主机具有同样网络号，但不同子网号的目的IP地址被认为是本地的。如果是0，只有在同一个子网的目的IP地址才认为是本地的。图E-1总结了上述规律。当给一个本地的目的地址发送报文时，TCP选择的是基于输出接口的MTU的MSS，而发送给一个非本地的地址时，TCP使用变量tcp_default_mss作为MSS。

IPSENDREDIRECTS

这个常量的值初始化内核变量ip_sendredirects。如果值为1（默认），主机在转发IP数据报时，将发送ICMP重定向。如果是0，不发送ICMP重定向。

DIRECTED_BROADCAST

这个常量的值初始化内核变量ip_dirbroadcast。如果值为1（默认），如果收到的数据报的目的地址是主机的一个接口的直接广播地址，就将它作为一个链路层的广播来转发。如果是0，这些数据报就会被丢弃。

文件/usr/kvm/sys/netinet/in_proto.c定义了下面一些可以修改的变量。一旦修改了这些变量，必须构造一个新的内核，然后重启动。

tcp_default_mss
用于非本地地址的默认TCP MSS。默认值是512。
tcp_sendspace
TCP发送缓存的默认大小。默认值是4096。

`tcp_recvspace`

TCP接收缓存的默认大小。这个值影响了提供的窗口大小。默认值是 4096。

`tcp_keeplen`

一个发往4.2BSD主机的keepalive探测报文必须包含一个字节的数据来得到一个响应。把这个变量的值设置为1是为了兼容于以前的实现。默认值是 1。

`tcp_ttl`

TCP段的TTL字段的默认值。默认值是60。

`tcp_nodelack`

如果非0，对ACK不做延迟。默认值是0。

`tcp_keepidle`

在发送一个keepalive探测报文之前必须等待的 500 ms时钟间隔的次数。默认值是 14 400（2个小时）。

`tcp_keepintvl`

如果没有收到响应，在两个连续的 keepalive探测报文之间等待的500 ms时钟间隔的次数。默认值是150（75秒）。

`udp_cksum`

如果非0，对输出的UDP数据报计算UDP检验和，并且对于包含了非 0检验和的输入UDP数据报要验证它们的检验和。如果值为 0，不计算输出UDP数据报的检验和，也不验证输入UDP数据报的检验和，即使发送者计算了一个检验和。默认值是 0。

`udp_ttl`

UDP数据报TTL字段的默认值。默认值是60。

`udp_sendspace`

UDP发送缓存的默认大小。定义了可以发送最大的 UDP数据报。默认值是9000。

`udp_recvspace`

UDP接收缓存的默认大小。默认值是 18 000，允许两个9000字节的数据报。

E.3 SRV4

SVR4的TCP/IP配置类似于前两个系统，但可用的选项更少。在文件`/etc/conf/pack.d/ip/space.c`5可以定义两个常量，然后必须重新构造内核并且重启动。

IPFORWARDING

这个常量的值初始化内核变量 `ipforwarding`。如果是0（默认），不转发IP数据报。如果是1，就总是能转发IP数据报。

IPSENDREDIRECTS

这个常量的值初始化内核变量 `ipsendredirects`。如果值为1（默认），主机在转发IP数据报时，将发送ICMP重定向。如果是0，不发送ICMP重定向。

前两节中，我们描述的许多变量在内核中都有定义，但必须修补内核来改变它们。例如，存在一个名为`tcp_keepidle`的变量，它的值是14 400。

E.4 Solaris 2.2

Solaris 2.2是较新的Unix系统的典型代表，它为管理员提供了一个可以改变 TCP/IP系统配置选项的程序。这样可以不必通过修改源文件和重新构造内核来进行配置。

配置程序是ndd(1)。我们可以运行程序，看看在UDP模块中可以检验和修改的参数：

```
solaris % ndd /dev/udp \?
udp_wroff_extra                 读、写
udp_def_ttl                     读、写
udp_first_anon_port             读、写
udp_trust_optlen                读、写
udp_do_checksum                 读、写
udp_status                      只读
```

我们可以指明 5个模块：/dev/ip、/dev/icmp、/dev/arp、/dev/udp和/dev/tcp。问号参数（为了防止外壳程序解释问号，我们在它前面加了一个反斜线）告诉ndd程序列出那个模块的所有参数。查询一个变量的值的例子是：

```
solaris %ndd /dev/tcp tcp_mss_def
536
```

为了修改一个变量的值，我们需要有超级用户的权限，输入：

```
solaris #ndd -set /dev/ip ip_forwarding 0
```

这些变量可以划分为三种类型：

1) 系统管理员可以修改的配置变量（如， ip_forwarding）。

2) 只能显示的状态变量（如， ARP快速缓存）。这个信息一般通过命令 ifconfig，netstat和arp以一种更好理解的格式提供。

3) 用于内核源代码的排错变量。使能一些这种变量可以在运行时产生内核的排错输出，当然这会降低系统的性能。

现在我们可以描述每个模块的参数了。所有的参数如果没有注明“（只读）”，就是可读写的。只读的参数是上面第 2种情况的状态变量。我们对于第 3种情况的变量注明了“（排错）”。如果不另外说明，所有的计时变量都以毫秒指明，这和其他系统不同，其他系统一般以 500 ms时钟间隔的次数来指明时间。

/dev/ip

ip_cksum_choice

（排错）在IP检验和算法的两个独立实现之中选择一个。

ip_debug

（排错）如果大于0，使能内核打印排错信息功能。值越大输出的信息越多。默认为 0。

ip_def_ttl

如果运输层没有指明，指定输出IP数据报默认的TTL。默认值是255。

ip_forward_directed_broadcasts

如果值为1（默认），如果收到的数据报的目的地址是主机的一个接口的直接广播地址，就将它作为一个链路层的广播来转发。如果是 0，这些数据报就会被丢弃。

ip_forward_src_routed

如果为1（默认），就转发包含一个源路由选项的接收数据报。如果为 0，这些数据报将被

丢弃。

ip_forwarding

指明系统是否转发进入的IP数据报：0表示不转发，1表示总是转发，2（默认）表示只有当两个或两个以上接口都工作时才转发。

ip_icmp_return_data_bytes

一个ICMP差错返回的除了IP首部以外的数据字节的数目，默认是64。

ip_ignore_delete_time

（排错）一个IP路由表项（IRE）最小的生命期。默认是30秒（这个参数以秒记，不是毫秒）。

ip_ill_status

（只读）显示每个IP下层数据结构的状态。每个接口存在一个下层数据结构。

ip_ipif_status

（只读）显示每个IP接口数据结构的状态（IP地址、子网掩码等等）。每个接口存在一个这种结构。

ip_ire_cleanup_interval

（排错）扫描IP路由表，删除过时表项的时间间隔。默认是30 000 ms（30秒）。

ip_ire_flush_interval

从IP路由表中无条件地刷新ARP信息的间隔。默认是1200 000 ms（20分钟）。

ip_ire_pathmtu_interval

路径MTU发现算法尝试增加MTU的间隔。默认是30 000 ms（30秒）。

ip_ire_redirect_interval

来自ICMP重定向的IP路由表项被删除的间隔。默认是60 000 ms（60秒）。

ip_ire_status

（只读）显示所有的IP路由表项。

ip_local_cksum

如果为0（默认），IP不为通过环回接口发送和接收的数据报计算IP检验和或者更高层的检验和（即TCP、UDP、ICMP或IGMP）。如果为1，就要计算这些检验和。

ip_mrtdebug

（排错）如果为1，使能内核打印多播路由的排错输出。默认是0。

ip_path_mtu_discovery

如果为1（默认），IP执行路径MTU发现。如果是0，IP不会在输出的数据报中设置“不分片”比特。

ip_respond_to_address_mask

如果为0（默认），主机不响应ICMP的地址掩码请求。如果为1，主机则响应。

ip_respond_to_echo_broadcast

如果为1（默认），主机响应发往一个广播地址的ICMP回显请求。如果为0，则不响应。

ip_respond_to_timestamp

如果为0（默认），主机不响应ICMP的时间戳请求。如果为1，则响应。

ip_respond_to_timestamp_broadcast

如果为0（默认），主机不响应发往一个广播地址的ICMP时间戳请求。如果为1，则响应。

ip_rput_pullups

（排错）来自于网络接口驱动程序的缓存数目的计数，它需要增长以访问整个 IP首部。引导时它被初始化为0，并且可以被复位为0。

ip_send_redirects

如果为1（默认），当主机作为一个路由器时，它发送ICMP重定向。如果为0，则不发送。

ip_send_source_quench

如果为1（默认），当输入的数据报被丢弃时，主机生成 ICMP源抑制差错。如果为 0，则不生成这种差错。

ip_wroff_extra

（排错）在缓存中为IP首部分配的额外空间的字节数。默认是 32。

/dev/icmp

icmp_bsd_compat

（排错）如果为1（默认），收到的数据报的IP首部的长度字段的值被调整为不包括 IP首部的长度。这和伯克利演变的实现是一致的，用于读原始的 IP或原始的ICMP分组的应用程序。如果为0，则不改变长度字段的值。

icmp_def_ttl

输出ICMP报文的默认的TTL。默认值为255。

icmp_wroff_extra

（排错）在缓存中为IP选项和数据链路首部所分配的额外空间的字节数。默认是 32。

/dev/arp

arp_cache_report

（只读）ARP的快速缓存。

arp_cleanup_interval

ARP登记项从ARP快速缓存中被删除的时间间隔。默认是 300 000 ms（5分钟）（IP为完成的ARP传输维护着它自己的快速缓存；参见 ip_ire_flush_interval）。

arp_debug

（排错）如果为1，使能打印ARP驱动程序的排错输出。默认是 0。

/dev/udp

udp_def_ttl

输出UDP数据报的默认的TTL。默认值是255。

udp_do_checksum

如果为1（默认），为输出的UDP数据报计算UDP检验和。如果为0，输出的UDP数据报不包含一个检验和（和其他大多数的实现不一样，这个 UDP检验和标志并不影响进入的数据报。如果一个接收到的数据报有一个非 0的检验和，它总是要被验证）。

udp_largest_anon_port

可以为UDP临时端口分配的最大端口号。默认是 65535。

`udp_smallest_anon_port`
可以为UDP临时端口分配的最小端口号。默认是32768。

`udp_smallest_nonpriv_port`
一个进程需要超级用户的权限才能给自己分配一个小于这个值的端口号。默认是 1024。

`udp_status`
（只读）所有本地的UDP端点的状态：本地IP地址和端口，远端IP地址和端口。

`udp_trust_optlen`
（排错）不再使用。

`udp_wroff_extra`
（排错）在缓存中为IP选项和数据链路首部所分配的额外空间的字节数。默认是 32。

/dev/tcp

`tcp_close_wait_interval`
2MSL的值：在TIME_WAIT状态花费的时间。默认是240 000 ms（4分钟）。

`tcp_conn_grace_period`
（排错）当发送一个SYN时，在定时器间隔上附加的时间。默认是 500 ms。

`tcp_conn_req_max`
在一个监听的端口上挂起的连接请求的最大数目。默认是 5。

`tcp_cwnd_max`
拥塞窗口的最大值。默认是32768。

`tcp_debug`
（排错）如果为1，使能打印TCP的排错输出。默认是0。

`tcp_deferred_ack_interval`
在发送一个延迟的ACK之前等待的时间。默认是50 ms。

`tcp_dupack_fast_retransmit`
引起快速重传、快速恢复算法的连续的重复 ACK的数目。默认是3。

`tcp_eager_listeners`
（排错）如果为1（默认），TCP在将一个新的连接返回给一个挂起的被动打开的应用程序之前需要进行三次握手。这是大多数的 TCP实现采用的方式。如果为 0，TCP将呼入连接请求（收到的SYN）传递给应用程序，并不完成三次握手直到该应用程序接受了这个连接（把这个值置为0可能引起很多已经存在的应用程序不能用）。

`tcp_ignore_path_mtu`
（排错）如果为1，路径MTU发现算法忽略接收到的需要 ICMP分段的报文。如果为0（默认），使能TCP的路径MTU发现。

`tcp_ip_abort_cinterval`
当TCP进行一个主动打开时，整个重传超时的值。默认是 240 000 ms（4分钟）。

`tcp_ip_abort_interval`
一个TCP连接建立以后，整个重传超时的值。默认是 120 000 ms（2分钟）。

`tcp_ip_notify_cinterval`

当TCP正在进行一个主动打开时，TCP通知IP去寻找一条新路由超时的值。默认是10 000 ms（10秒）。

`tcp_ip_notify_interval`

TCP为一个已经建立的连接通知IP去寻找一条新路由超时的值。默认是10 000 ms（10秒）。

`tcp_ip_ttl`

用于输出TCP段的TTL。默认为255。

`tcp_keepalive_interval`

在发出一个keepalive探测报文之前，一个连接保持空闲状态的时间。默认为 7200000 ms（2小时）。

`tcp_largest_anon_port`

为TCP临时端口分配的最大端口号。默认为 65535。

`tcp_maxpsz_multiplier`

（排错）指明了报文流首部将应用程序写的数据分装成几个 MSS。默认是1。

`tcp_mss_def`

非本地的目的地址的默认的MSS。默认是536。

`tcp_mss_max`

最大的MSS。默认为65495。

`tcp_mss_min`

最小的MSS。默认为1。

`tcp_naglim_def`

（排错）每个连接的Nagle算法阈值的最大值。默认是 65535。每个连接的值以MSS的最小值或这个值开始。TCP_NODELAY插口选项将每个连接的值设置为 1，以禁止Nagle算法。

`tcp_old_urp_interpretation`

（排错）如果为1（默认），采用紧急指针的一个以前的（但更常见的） BSD的理解：它指向紧急数据最后一个字节后的一个字节。如果为 0，采用主机需求RFC理解：它指向紧急数据的最后一个字节。

`tcp_rcv_push_wait`

（排错）在把接收数据传递给应用程序之前，可以缓存的没有设置 PUSH标志的数据的最大字节数。默认是16384。

`tcp_rexmit_interval_initial`

（排错）初始的重传超时间隔。默认是 500 ms。

`tcp_rexmit_interval_max`

（排错）最大的重传超时间隔。默认是 60 000 ms（60秒）。

`tcp_rexmit_interval_min`

（排错）最小的重传超时间隔。默认是 200 ms。

`tcp_rwin_credit_pct`

（排错）在对每个接收的段进行流量控制检查之前，必须达到的接收缓存窗口的百分比。默认是50%。

`tcp_smallest_anon_port`

分配给TCP临时端口的开始端口号。默认是32768。

`tcp_smallest_nonpriv_port`

一个进程需要有超级用户的权限才能给自己分配一个小于这个值的端口号。默认是1024。

`tcp_snd_lowat_fraction`

（排错）如果非0，发送缓存的低水平线是发送缓存的大小除以这个值。默认是0（禁止）。

`tcp_status`

（只读）所有TCP连接的信息。

`tcp_sth_rcv_hiwat`

（排错）如果非0，把报文流首部的高水平线设置为这个值。默认为0。

`tcp_sth_rcv_lowat`

（排错）如果非0，把报文流首部的低水平线设置为这个值。默认为0。

`tcp_wroff_xtra`

（排错）在缓存中为IP选项和数据链路首部所分配的额外空间的字节数。默认是32。

E.5 AIX 3.2.2

AIX3.2.2允许在运行时使用no命令设置网络选项。它可以显示一个选项的值，设置一个选项的值，或者将一个选项的值设置为默认。例如，显示一个选项，我们键入：

```
aix % no -o udp_ttl
udp_ttl = 30
```

下面的选项可以被修改。

`arpt_killc`

在删除一个不活动的、完成的ARP项之前等待的时间（以分钟计）。默认是20。

`ipforwarding`

如果为1（默认），总是转发IP数据报。如果为0，则禁止转发。

`ipfragttl`

等待重新装配的IP数据报片的生存时间（单位为秒）。默认是60。

`ipsendredirects`

如果为1（默认），当转发IP数据报时，主机将发送ICMP重定向。如果为0，则不发送ICMP重定向。

`loop_check_sum`

如果为1（默认），对通过环回接口发送的数据报计算IP检验和。如果为0，则不计算这个检验和。

`nonlocsrcroute`

如果为1（默认），就转发包含一个源路由选项的接收数据报。如果为0，就丢弃这些数据报。

`subnetsarelocal`

如果值为1（默认），一个和发送主机具有同样网络号，但不同子网号的目的IP地址被认为是本地的。如果是0，只有在同一个子网的目的IP地址才认为是本地的。图E-1总结了上述规律。当给一个本地的目的地址发送报文时，TCP选择的是基于输出接口的MTU的MSS。当发送给一个非本地的地址时，TCP使用默认（536）作为MSS。

`tcp_keepidle`

在发送一个keepalive探测报文之前等待的500 ms时间段的倍数。默认值是14 400（2小时）。

`tcp_keepintvl`

如果没有收到响应，在发送下一个 keepalive探测报文之前等待的 500 ms时间段的倍数。默认值是150（75秒）。

`tcp_recvspace`

TCP接收缓存的默认大小。它影响了提供的窗口大小。默认值是 16 384。

`tcp_sendspace`

TCP发送缓存的默认大小。默认是 16 384。

`tcp_ttl`

TCP报文段TTL字段的默认值。默认值是60。

`udp_recvspace`

UDP接收缓存的默认大小。默认值是 41 600，允许40个1024字节数据报。

`udp_sendspace`

UDP发送缓存的默认大小。定义了可以发送的最大的 UDP数据报。默认是9216。

`udp_ttl`

UDP数据报TTL字段的默认值。默认值是30。

E.6 4.4BSD

4.4BSD是第一个为多个内核参数提供动态配置的伯克利版本。使用 sysctl（8）命令。参数的名字看起来就像SNMP的MIB变量的名字。查看一个参数，我们键入：

```
vangogh %sysctl net.inet.ip.forwarding
net.inet.ip.forwarding = 1
```

要修改一个参数的值需要有超级用户的权限，键入：

```
vangogh #sysctl -w net.inet.ip.ttl=128
```

可以修改下面的参数。

`net.inet.ip.forwarding`

如果为0（默认），就不转发IP数据报。如果为1，则使能转发功能。

`net.inet.ip.redirect`

如果为1（默认），当转发 IP数据报时，主机将发送 ICMP重定向。如果为 0，则不发送ICMP重定向。

`net.inet.ip.ttl`

TCP和UDP默认的TTL。默认值是64。

`net.inet.icmp.maskrepl`

如果为0（默认），主机不响应ICMP地址掩码请求。如果为1，则响应。

`net.inet.udp.checksum`

如果为1（默认），对输出的UDP数据报计算UDP检验和，并且对于包含了非0检验和的呼入UDP数据报要校验它们的检验和。如果值为 0，不计算输出UDP数据报的检验和，也不验证输入UDP数据报的检验和，即使发送者计算了一个检验和。

另外，在本附录前面部分描述的许多变量分散在几个不同的源文件中（`tcp_keepidle`、`subnetarelocal`等等），它们也可以被修改。