4.4. **TCP_NODELAY** and Small Buffer Writes

As discussed briefly in Transmission Control Protocol (TCP) (sect-Realtime_Tuning_G
General_System_Tuning-Network_determinism_tips.html#form-Realtime_Tuning_Gu
Network_determinism_tips-Transmission_Control_Protocol_TCP), by default TCP us
collect small outgoing packets to send all at once. This can have a detrimental effect

Using `TCP_NODELAY` and `TCP_CORK` to improve network latency

1. Applications that require lower latency on every packet sent should be run on s
enabled. It can be enabled through the **`setsockopt`** command with the sockets

```
# int one = 1; # setsockopt(descriptor, SOL_TCP, TCP_NODELAY, &
```

2. For this to be used effectively, applications must avoid doing small, logically rela
`TCP_NODELAY` is enabled, these small writes will make TCP send these multiple
packets, which can result in poor overall performance.

   If applications have several buffers that are logically related and that should be
be possible to build a contiguous packet in memory and then send the logical p
configured with **`TCP_NODELAY`**.

   Alternatively, create an I/O vector and pass it to the kernel using **`writev`** on a s
`TCP_NODELAY`.

3. Another option is to use **`TCP_CORK`**, which tells TCP to wait for the application to
sending any packets. This command will cause the buffers it receives to be app
buffers. This allows applications to build a packet in kernel space, which may be
different libraries that provides abstractions for layers. To enable **`TCP_CORK`**, set
`setsockopt` sockets API (this is known as "corking the socket"):

```
# int one = 1; # setsockopt(descriptor, SOL_TCP, TCP_CORK, &one
```

4. When the logical packet has been built in the kernel by the various components
to remove the cork. TCP will send the accumulated logical packet right away, w
packets from the application.

```
# int zero = 0; // setsockopt(descriptor, SOL_TCP, TCP_CORK, &ze
```

Related Manual Pages

For more information, or for further reading, the following man pages are related to th section.

- tcp(7)
- setsockopt(3p)
- setsockopt(2)