
Subject: [patch 0/3][NETNS45][V2] remove timewait sockets at namespace exit
Posted by [Daniel Lezcano](#) on Thu, 27 Sep 2007 11:04:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Denis Lunev spotted that using a reference to the network namespace with the timewait sockets will be a waste of time because they are pointless while we will remove the network stack at network namespace exit.

The following patches do the following:

- fix missing network namespace reference in timewait socket
- do some changes in timewait socket code to prepare the next patch, especially split code taking a lock
- do the effective timewait socket cleanup at network namespace exit.

The following code is a test program which creates 100 timewait sockets.

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/poll.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <arpa/inet.h>

#include <unistd.h>

#define MAXCONN 100

int client(int *fds)
{
    int i;
    struct sockaddr_in addr;

    close(fds[1]);

    memset(&addr, 0, sizeof(addr));

    addr.sin_family = AF_INET;
    addr.sin_port = htons(10000);
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    if (read(fds[0], &i, sizeof(i)) == -1) {
        perror("read");
    }
}
```

```

        return 1;
    }

    for (i = 0; i < MAXCONN; i++) {
        int fd = socket(PF_INET, SOCK_STREAM, 0);
        if (fd == -1) {
            perror("socket");
            return 1;
        }

        if (connect(fd, (const struct sockaddr *)&addr, sizeof(addr))) {
            perror("connect");
            return 1;
        }
    }

    return 0;
}

int server(int *fds)
{
    int i, fd, fdpoll[MAXCONN];
    struct sockaddr_in addr;
    socklen_t socklen = sizeof(addr);

    close(fds[0]);

    fd = socket(PF_INET, SOCK_STREAM, 0);
    if (fd == -1) {
        perror("socket");
        return 1;
    }

    memset(&addr, 0, sizeof(addr));

    addr.sin_family = AF_INET;
    addr.sin_port = htons(10000);
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    if (bind(fd, (const struct sockaddr *)&addr, sizeof(addr))) {
        perror("bind");
        return 1;
    }

    if (listen(fd, MAXCONN)) {
        perror("listen");
        return 1;
    }
}

```

```

    if (write(fds[1], &i, sizeof(i)) == -1) {
        perror("write");
        return 1;
    }

    for (i = 0; i < MAXCONN; i++) {
        int f = accept(fd, (struct sockaddr *)&addr, &socklen);
        if (f == -1) {
            perror("accept");
            return 1;
        }
        fdpoll[i] = f;
    }

    return 0;
}

int main(int argc, char *argv[])
{
    int fds[2];
    int pid;

    if (pipe(fds)) {
        perror("pipe");
        return 1;
    }

    pid = fork();
    if (pid == -1) {
        perror("fork");
        return 1;
    }

    if (!pid)
        return client(fds);
    else
        return server(fds);
}

--

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
