MPTCP 接收

Thursday, August 6, 2015 11:41 AM

接收顺序分析http://blog.chinaunix.net/uid-20424888-id-96008.html

TCP消息接收分析

http://blog.csdn.net/russell_tao/article/details/9950615

TCP包的处理流程

接收:

ip_local_deliver

↓

tcp_v4_rcv() (tcp_ipv4.c)→_tcp_v4_lookup()

↓

tcp_v4_do_rcv(tcp_ipv4.c)→tcp_rcv_state_process (OTHER STATES)

↓Established

tcp_rcv_established(tcp_in→tcp_ack_snd_check,(tcp_data_snd_check,tcp_ack (tcp_input.c)

↓ ↓ ↓ ↓

tcp_data tcp_send_ack tcp_write_xmit

↓ (slow) ↓(Fast) ↓

tcp_data_queue tcp_transmit_skb

↓ ↓ ↓

sk->data_ready (应用层) ip_queque_xmit

tcp_ipv4.c

tcp_v4_do_rcv:

这里会分很多不同的队列，为了有效的处理

真正的是receive

>>

tcp_v4_do_rcv 里面有mptcp_v4_do_rcv会判断调用的是否为meta_sk来选择调用哪个

http://blog.sina.com.cn/s/blog_4826456d0100jcws.html

mptcp_v4_do_rcv:/

* We only process join requests here. (either the SYN or the final ACK) */

>>

tcp_input.c

tcp_rcv_established:

调用tcp_rcv_established函数将SKB加入sk-> receive_queue中

```
        //tcp_rcv_established中语句 __skb_queue_tail(&sk->
sk_receive_queue, skb);
        //完成队列的添加
```

tcp_rcv_established函数的工作原理是把数据包的处理分为2类：fast path和slow path，其含义显而易见。这样分类

的目的当然是加快数据包的处理，因为在正常情况下，数据包是按顺序到达的，网络状况也是稳定的，这时可以按照fast path

mptcp slow path 应该就是为了检查一大堆东西

>>

tcp_data_queue 数据排队到receive queue

```
/* MPTCP: we always have to call data_ready, because
             * we may be about to receive a data-fin, which still
             * must get queued.
             */
            sk->sk_data_ready(sk, 0);通知应用进程数据排队了
```

```
    /* As we successfully allocated the mptcp_tcp_sock, we have to
     * change the function-pointers here (for sk_destruct to work correctly)
     */
    sk->sk_error_report = mptcp_sock_def_error_report;
    sk->sk_data_ready = mptcp_data_ready;
    sk->sk_write_space = mptcp_write_space;
    sk->sk_state_change = mptcp_set_state;
    sk->sk_destruct = mptcp_sock_destruct;
```

mptcp_data_ready mptcp_input.c

>>

mptcp_queue_skb 处理mapping full的数据放入meta_sk 的queue里面

一般是ofo_queue 利用原始其对无序的进行排序

mptcp_parse_options mptcp_input.c

接收处理options