Messengers - Bug #1803

msgr: behave better when ending TCP connections

12/07/2011 04:56 PM - Greg Farnum

Status:	Won't Fix	**% Done:**	0%
Priority:	Normal		
Assignee:			
Category:			
Target version:			
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:		**ceph-qa-suite:**	
Regression:	No	**Pull request ID:**	
Severity:		**Crash signature:**	

Description

TV is telling me that if we're not confirming that each side of the connection calls ::shutdown() on the socket, we're not ending our TCP connection properly. Obviously it can work out okay even so, but we want to be good citizens and fixing this up will likely reduce the edge cases where we need to call mark_disposable() on pipes.

History

#1 - 12/07/2011 05:20 PM - Josh Durgin

- Priority changed from Normal to High

This actually caused a deadlock with ffsb on the kernel client - ffsb ended up with 1006 connections in the CLOSING state, and the osd had 1006 in FIN_WAIT2. This made the osd hit max open file descriptors at 1024. (The other osd crashed for a different reason).

#2 - 12/09/2011 11:18 AM - Greg Farnum

- Assignee set to Greg Farnum

I'm going to see if I can handle this in userspace today — fixing it in the kernel client will be another ticket.

#3 - 12/12/2011 10:49 AM - Greg Farnum

- Status changed from New to In Progress

From the little I'm reading in Unix Network Programming, it looks like we're just doing this wrong — we call shutdown(RD_WR) and then try to read, which never works. And we don't call close() until we get our successful read (or after timeouts when we mark_disposable).
So presumably just fixing that will deal with it.

#4 - 12/12/2011 04:45 PM - Greg Farnum

And I've flipped back and forth umpteen times today about what's going on. At this point I can conclude that nobody on our end knows, but probably one of close() or shutdown() is actually removing the buffer (probably close()). So the proper fix is going to involve reworking the messenger so that it does separate shutdown calls for SO_WR and then does shutdown() for SO_RD after receiving an EOF from the other side.

#5 - 12/19/2011 03:48 PM - Greg Farnum

- Priority changed from High to Normal

#6 - 12/23/2011 02:48 PM - Greg Farnum

- Status changed from In Progress to New

#7 - 02/22/2013 12:21 PM - Ian Colle

- Assignee deleted (Greg Farnum)

#8 - 09/24/2014 03:12 AM - Loic Dachary

- Status changed from New to Resolved

Not sure at which point this problem was fixed but it is doubtful that it stayed around for the past three years unnoticed.

#9 - 09/24/2014 10:37 AM - Greg Farnum

- Status changed from Resolved to New

This has been greatly improved with the addition of our socket timeouts and things, but I don't think it's properly resolved yet. It will get a great deal easier when the messenger doesn't have a thread<->socket relationship.

#10 - 04/12/2017 04:04 PM - Sage Weil

- Status changed from New to Won't Fix

#11 - 03/12/2019 11:12 PM - Greg Farnum

- Project changed from Ceph to Messengers

- Category deleted (msgr)