

BIP: 31
Layer: Peer Services
Title: Pong message
Author: Mike Hearn <hearn@google.com>
Comments-Summary: No comments yet.
Comments-URI: <https://github.com/bitcoin/bips/wiki/Comments:BIP-0031>
Status: Final
Type: Standards Track
Created: 2012-04-11

Abstract

This document describes a trivial protocol extension that makes it easier for clients to detect dead peer connections.

Motivation

Today there are a few network related problems that can degrade the Bitcoin user experience:

- 1) Some Bitcoin clients run on platforms that can go to sleep and essentially stop running at any time without warning. Notably, this is very common on both mobiles and laptops (shut the lid). When the system comes back, TCP connections that existed before the sleep still exist but may no longer function correctly, eg, because the IP address has changed, or because the remote peer went away or the connection was timed out by some other system. Currently it can often take a while to notice this has happened.
- 2) The reference Satoshi client is largely single threaded and when placed under heavy load (e.g., because it is downloading the block chain) becomes very slow to respond to network messages. There's no easy way to detect this has occurred, especially if you are just passively waiting for broadcasts from that peer. A way to detect overloaded remote peers and avoid them would both help balance load and provide a better, more responsive system.
- 3) When downloading large data structures like the block chain it is efficient to choose a peer that is near to you network-wise, in order to reduce load on often congested trans-national links and ensure lower latency. Currently it is difficult to measure the latency to a remote peer so clients don't bother, and instead just select a random peer to download from.

All of these can be solved by a backwards compatible protocol modification.

Specification

When the protocol version as negotiated in the "ver" message is greater than 60000, the "ping" message must contain a uint64 field called "nonce". A peer sending "ping" should set the nonce to a random value, and it is then echoed

back by the recipient in a new "pong" message that also contains a single uint64 field.

In this way, the client can send a ping and measure the time taken to receive the corresponding pong. If a client sends two pings before hearing back the first pong, the responses can be distinguished using the nonce. If the client chooses to never overlap pings in this way it should simply set the nonce value to zero.

Backward compatibility

Clients must opt-in to the new feature by advertising a protocol version > 60000 . Clients with older protocol versions are not expected to provide a nonce in the ping message and will not be sent a pong.

Implementation

<https://github.com/bitcoin/bitcoin/pull/932/files>