BIP 14 - LINK

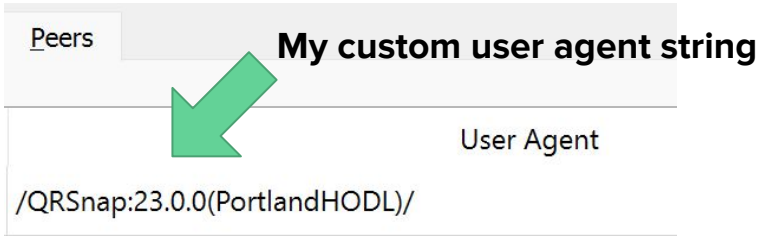
Protocol Version and User Agent

The Goal - Increase Decentralization

- BIP 14 suggests splitting the protocol version away from the client version.
- This would be similar the changing the version of your web browser but not the version of TCP/IP or HTTP protocols that that browser is communicating through.
- This is because there is a definitive difference between the protocol and the client. (*e.x. lets say Bitcoin core adds a new RPC for users to access. This would have no effect on the protocol implementation. As such the client version number should increment but the protocol version should not.*)

```
"version": 249900,
"subversion": "/Satoshi:24.99.0/",
"protocolversion": 70016,
```

or



user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36

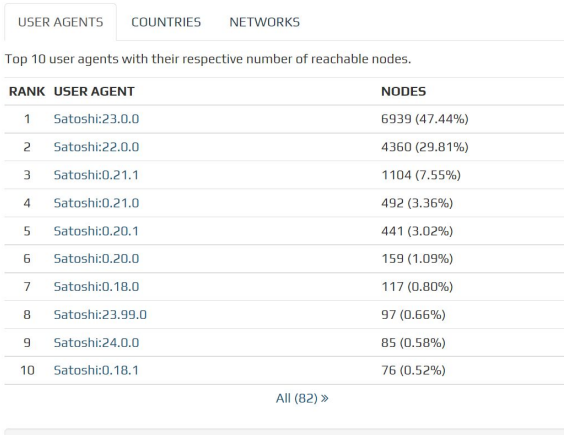
Some details about this BIP to note

```
BIP: 14
Layer: Peer Services
Title: Protocol Version and User Agent
Author: Amir Taaki <genjix@riseup.net>
        Patrick Strateman <bitcoin-bips@covertinferno.org>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0014
Status: Final
Type: Standards Track
Created: 2011-11-10
Post-History: 2011-11-02
```

This BIP was create very early on in Bitcoin History

Though this BIP is old, it's still is relevant and maintains the 'Final' status

Where can you see this in BIP in practice?



USER AGENTS | COUNTRIES | NETWORKS

Top 10 user agents with their respective number of reachable nodes.

RANK	USER AGENT	NODES
1	Satoshi:23.0.0	6939 (47.44%)
2	Satoshi:22.0.0	4360 (29.81%)
3	Satoshi:0.21.1	1104 (7.55%)
4	Satoshi:0.21.0	492 (3.36%)
5	Satoshi:0.20.1	441 (3.02%)
6	Satoshi:0.20.0	159 (1.09%)
7	Satoshi:0.18.0	117 (0.80%)
8	Satoshi:23.99.0	97 (0.66%)
9	Satoshi:24.0.0	85 (0.58%)
10	Satoshi:0.18.1	76 (0.52%)

All (82) »

Bitnodes.io tracks user agent strings - NOT protocol versions

- Bitnodes.io is an amazing use of aggregating the statistics of user agent strings.
- As you can see because of this bip it's easy to aggregate statistics of the various clients operating on the network. These clients remain interoperable because of the fact that though their individual version numbers are increasing over time the protocol version seldomly changes.

A ‘BIP’ of History!

- The original Bitcoin client was named the Satoshi client.
- As time went on other versions and compatible clients were being deployed in a variety of languages (Python, Java, C++, etc ...) and were **developing their own feature sets.** *E.g. BitcoinJ, BitDroid, libbitcoin*
- Because the protocol and version numbers up until this point were linked to each other any increment in the client version incremented the protocol version.
- The protocol version is emitted during the following protocol calls ‘version’, ‘getblocks’, and ‘block’ messages to indicate the software used to create the block.

Rationale

- With non separated version numbers the Satoshi Client versions also became the new protocol version as well. As such the Bitcoin ‘Satoshi’ Client developers held the rest of the implementations on the network hostage to a degree and with each update of the Saoshi client other non satoshi client could be forced into also updating their version numbers as well.
- The protocol should be a shared common collaborative separate from client implementations and allow the Bitcoin network to grow with no or menial fracturing.
- So as such the protocol number would not always increment even if there would be an update to the Satoshi client. This is because of the proposed and implemented differetantion between a **Client** and **Protocol** implementation.

The 2 Line Summary

- Protocol versions represent discrete sets of rules for network communication (protocol) and as such a variation in protocol versions indicates differences such as methods of communication and or features.
- The User agent string is no more or less than an informational tool. A vanity plate for your client implementation per say.

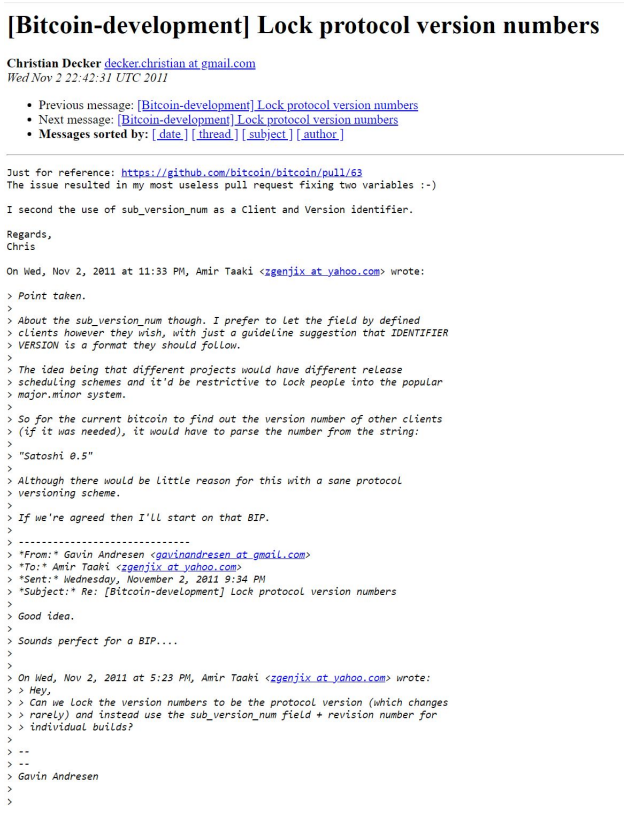
So how does this BIP Split the Protocol and Client Versions?

- Network packets ‘`version`’ and ‘`getblocks`’ both contain a ‘version’ field. This field represents the version of the Network Protocol. Blocks versions represent the client they were created by.
- The ‘`version`’ packets also contain a **`sub_version_num`** this was previously unused and now and now is set to contain the user-agent of the client. These user agents as proposed are a modified implementation of browser user agents.
- The sub_version_num has been renamed to user agent and currently has a maximum length of 255 characters.

Proposed Rules for User Agent Strings

- Based on RFC 1945 browser user agents but with more structure
- Technically no format is defined only guidelines
- Format of … */Name:Version/Name:Version/.../*
  - Version numbers in the form of Major.Minor.Revision (2.6.41)
  - Repository builds using a date in the format of YYYYMMDD (20110128)
- Comments are also allowed as well! - This is done by enclosing the comment into a set of parentheses.
- To add this comment the user may add uacomment='your cool name here' and it will be applied your user agent string within a set of brackets delimiting a comment!

Some cool extras...



[Bitcoin-development] Lock protocol version numbers

Christian Decker decker.christian at gmail.com
Wed Nov 2 22:42:31 UTC 2011

- Previous message: [Bitcoin-development] Lock protocol version numbers
- Next message: [Bitcoin-development] Lock protocol version numbers
- **Messages sorted by:** [date] [thread] [subject] [author]

Just for reference: https://github.com/bitcoin/bitcoin/pull/63
The issue resulted in my most useless pull request fixing two variables :-)

I second the use of sub_version_num as a Client and Version identifier.

Regards,
Chris

On Wed, Nov 2, 2011 at 11:33 PM, Amir Taaki <zgenjix at yahoo.com> wrote:

> *Point taken.*
>
> *About the sub_version_num though. I prefer to let the field by defined*
> *clients however they wish, with just a guideline suggestion that IDENTIFIER*
> *VERSION is a format they should follow.*
>
> *The idea being that different projects would have different release*
> *scheduling schemes and it'd be restrictive to lock people into the popular*
> *major.minor system.*
>
> *So for the current bitcoin to find out the version number of other clients*
> *(if it was needed), it would have to parse the number from the string:*
>
> *"Satoshi 0.5"*
>
> *Although there would be little reason for this with a sane protocol*
> *versioning scheme.*
>
> *If we're agreed then I'll start on that BIP.*
>
> *----------*
> **From:* Gavin Andresen <gavinandresen at gmail.com>*
> **To:* Amir Taaki <zgenjix at yahoo.com>*
> **Sent:* Wednesday, November 2, 2011 9:34 PM*
> **Subject:* Re: [Bitcoin-development] Lock protocol version numbers*
>
> *Good idea.*
>
> *Sounds perfect for a BIP....*
>
>
> *On Wed, Nov 2, 2011 at 5:23 PM, Amir Taaki <zgenjix at yahoo.com> wrote:*
> *> Hey,*
> *> Can we lock the version numbers to be the protocol version (which changes*
> *> rarely) and instead use the sub_version_num field + revision number for*
> *> individual builds?*
>
> *--*
> *--*
> *Gavin Andresen*
>
>

Following the thread of BIP 1. The thread on the linux foundation email lists linked is the thread that I believe to be the original discussion that led to this becoming a BIP. Gavin Andresen ACKs the idea as ‘Sounds perfect for a BIP... November 2 2011 was the original thread, BIP submission was one November 11 2011. Others such as Luke-JR also chimed in as well.

LINK

Sources

1. https://github.com/bitcoin/bips/blob/master/bip-0014.mediawiki
2. https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2011-November/000715.html