# BitTorrent Token Whitepaper

# BitTorrent Token

## Abstract

TRON Foundation and BitTorrent Foundation are introducing a new cryptographic token called BTT along with an extended version of the BitTorrent protocol in order to create a token-based economy around the usage of networking, bandwidth and storage on hundreds of millions of computers on the internet. Our initial entry point is to introduce token-based optimizations to the existing BitTorrent protocol providing a way for the value of sharing bandwidth and storage to be captured by network participants beyond the point at which the current protocol no longer values it. Our longer-term vision is to broaden the usage of BitTorrent far beyond current use cases to provide a distributed infrastructure platform to third party app developers and to enable consumers to continuously distill small amounts of value from their devices by allowing others to make use of their spare resources.

The first step in our project will create a market-driven mechanism to enable consumers to collaborate to optimize and prolong the lifespan of existing BitTorrent swarms. Using additional extensions to BitTorrent we will subsequently open up opportunities for distributed app developers to launch new apps making use of infrastructure provided by existing BitTorrent clients which already constitute a distributed networked storage platform of unprecedented scale. These new apps will be able to offer incentives (BTT) to users in return for access to cost effective platform resources comprised of an incredibly broad collection of already-deployed network endpoints. The position of these endpoints at the very edge of the Internet will have the additional appeal to developers of being extremely difficult for net-neutrality adversaries to interdict. Finally, the ability of consumers to capture the value of their contributed computing resources within a cryptographic token will give rise to a completely new transactional mechanism for internet consumers that is distinct from either their attention or their credit card.

With over 100 million monthly active users and millions of additional new installs every week, BitTorrent already manages one of the largest distributed computing ecosystems on the Internet. By integrating BTT tokens and transaction processing we will both address existing limitations of BitTorrent and open up a whole new borderless economy exchanging value for compute resources on a global scale. This will be a realization and extension of the vision expressed by Satoshi Nakamoto in the original Bitcoin white paper on its tenth anniversary.

The TRON Foundation and BitTorrent Foundation are legal entities incorporated in the Republic of Singapore.

# Background

## What is BitTorrent?

BitTorrent is a pioneering distributed communication protocol invented by Bram Cohen in 2001. It is a peer-to-peer protocol which uses economic incentives to facilitate the delivery of large and highly demanded files around the internet, eliminating the need for a trusted central server. It is an open protocol that has been independently implemented dozens of times and included in software that has been downloaded and installed billions of times in the past 16 years. Today the protocol is in regular use on well over one hundred million internet-connected computing devices each month. The protocol is continuously updated through an open process of BitTorrent Enhancement Proposals (BEPs) moderated on the developer website[1].

## BitTorrent the protocol

The BitTorrent protocol enables client software endpoints ("clients") to collaborate with each other to enable incredibly efficient and reliable distribution of large files to multiple clients. It does this by attempting to make simultaneous efficient use of every client's upload and download bandwidth to balance peer-to-peer content delivery across a `swarm' of cooperating clients and reduce the reliance on any single weak point (like the connection to a server). The key to understanding how the protocol works is to understand how the underlying economic incentives are implemented.

The protocol is based on a system where files are cut into pieces and pieces are traded between multiple devices which are all trying to get the file at the same time. Cryptographic hashes ("infohashes") of the pieces are used to verify that the pieces being shared are indeed the pieces requested. The system essentially implements a barter economy where individual clients collaborate based on trading pieces of a file they each are trying to download, with delivery bandwidth used as the deciding factor for whom to keep bartering with. Various mechanisms reward the most productive barter exchanges with further barter and punish the least productive ones by disconnecting or even banning the counterparty. Once a client has completed downloading a file, if it continues to upload pieces despite no longer requiring any upload in return, it is considered a "seed." The default in most clients is to "seed" to other downloaders, but this activity is entirely altruistic and there is no economic penalty for end users switching off their BitTorrent client and stopping seeding.

[1] http://www.bittorrent.org/

## BitTorrent the ecosystem

The BitTorrent protocol has been implemented many times and there is healthy competition between companies maintaining different implementations as well as several very popular volunteer-maintained versions. As well as client software that implements the BitTorrent protocol, there are infrastructure providers who independently offer additional useful services (e.g. trackers which introduce peers, torrent sites which index metadata about files being shared and provide access to their associated torrents). This demonstrates how an array of cooperating distributed elements (clients) and semi-distributed elements (trackers, torrent sites) have been successful in maintaining a long-lived and highly robust ecosystem even in the presence of highly motivated attackers. We relied on many lessons learned in the BitTorrent ecosystem as we put together this project plan.

## BitTorrent the company

BitTorrent Protocol is the world's largest decentralized protocol with over 1 billion users, far surpassing the second-ranked decentralized application of Bitcoin (as of October 21, 2018, Bitcoin has a total number of 29.44 million addresses[2]). BitTorrent Inc. invented and maintains the BitTorrent protocol. While there are many implementations of BitTorrent software[3], BitTorrent and µTorrent (often called "utorrent") remain two of the most popular. In 2018, BitTorrent Protocol reached a strategic partnership with TRON Blockchain Protocol. The collaboration of the two makes TRON Protocol the world's largest decentralized ecosystem; It also makes BitTorrent Protocol the biggest decentralized application in the world.

The active usage of software managed by BitTorrent Inc. is incredibly widespread around the world. Software clients maintained by BitTorrent Inc. are in use today by approximately 100 million monthly active users in almost every country in the world with around one million new software installs every day. Over 160 countries have more than 10,000 users, and 23 countries have more than 1 million users. 19 countries have more than 5% of their internet users using our software (comprising almost 60 million users). Furthermore, while there are other providers of BitTorrent software, BitTorrent Inc. clients currently account for an estimated 40% of current BitTorrent protocol activity on the public internet[4].

## BitTorrent and Distributed Apps

BitTorrent Foundation has been exploring distributed applications for over a decade. We have investigated adaptations of the BitTorrent protocol as well as entirely novel protocols with the aim of providing services ranging from distributed messaging, BitTorrent-based CDN, peer-to-peer live video streaming, file synchronization and distributed websites.

The incredibly exciting emergence of blockchain technologies has brought a paradigm shift in what is achievable, enabling collaboration between untrusted parties to be persisted over much longer periods

[2] https://www.blockchain.com/charts/my-wallet-n-users
[3] https://en.wikipedia.org/wiki/Comparison_of_BitTorrent_clients
[4] BitTorrent, Inc. internal market share research

than is practical in the BitTorrent ecosystem. But while many new decentralized protocol proposals are suggesting ambitious technical paths forward, almost all are silent on how to address the enormous marketing challenge of building critical mass which is a crucial technical necessity for all distributed systems. Even the few projects which are introducing a cryptographic token to existing user bases have no experience in the suBTTle art of designing a protocol which effectively balances many economic interests at scale. We seek to combine the critical mass of the existing BitTorrent ecosystem, protocol engineering expertise from BitTorrent Foundation and new capabilities introduced by blockchains as pioneered by platforms like TRON.

By integrating blockchain technologies to provide a reliable and scalable mechanism to store and exchange value, we can enable new decentralized applications to be bootstrapped on top of our existing ecosystem. We believe that expanding the BitTorrent ecosystem in combination with blockchain technologies will enable developers of decentralized apps to build on top of it, and help BitTorrent address a universe of exciting new opportunities. We believe that BitTorrent is by far the closest to being able to introduce the infrastructure to support the coming Decentralized Web and the economy to support it.

## Project origins

This project evolved in the light of three basic insights:

1. There is a huge and entirely unrealized opportunity for the application of decentralized BitTorrent technology to many new use cases, and the market is more ready for it now than it has ever been before.
2. Operation of the BitTorrent protocol today suffers from structural inefficiencies which limit the lifespan of BitTorrent swarms and so limit its overall effectiveness as a protocol.
3. Most consumers (including BitTorrent users) are reluctant to pay with fiat currency for things online. The corollary is that people pay with their 'attention' which leads directly to a web dominated by privacy-destroying information monopolies.

We are setting out to improve and extend BitTorrent to address these insights with a project that marries the best of BitTorrent and blockchain technologies.

We will transform BitTorrent to create an infrastructure platform for building elements of the Decentralized Web enabling app developers to directly reward consumers who provide the underlying resources, and enabling consumers to use this 'found value' to transact with publishers and app developers directly without involving fiat currency.

To accelerate introduction, we will start by addressing inefficiencies within how BitTorrent works today. This will stimulate a strong pull for the foundational technology and broad familiarity among consumers with the existence of the token and the user experience and economics around its use.

In parallel we will work with third party developers to develop and promote APIs and a marketplace for distributed infrastructure services based broadly on networking and storage primitives which are staples within the existing BitTorrent technology.

We will also work with third party publishers and app developers beyond the existing BitTorrent ecosystem on services which consumers may spend their tokens on.

In the fullness of time, hundreds of millions of end users around the world will be equipped with a robust new way to distill small amounts of value out of their own technical resources and have many opportunities to spend that value on services of their choosing.

# High Level Project Description

We will extend the BitTorrent protocol and introduce a new TRON TRC-10 cryptographic token called BitTorrent (BTT) in order to implement a distributed infrastructure services economy. Within this economy, end users may offer infrastructure services in small increments in return for tokens, with a blockchain solution for a store of value and medium of exchange that will scale to meet the expected demand.

To speed adoption we will launch a feature called BitTorrent Speed to optimize the operation of BitTorrent within the existing ecosystem. The introduction of this feature will solve problems within BitTorrent as well as prove the effectiveness of using blockchain-based rewards for provision of infrastructure services in small increments across a very large installed base. We will address challenges around rolling out distributed ledger-based transaction processing with low latency on a very large scale. Finally, we will generalize the services offered by BitTorrent clients and make them available to external application developers as a platform on top of which future decentralized apps can be launched.

In the following subsections we will first outline the BitTorrent (BTT) cryptographic tokens around which we plan to build a new economy. Second we present the blockchain technology on top of which transaction processing will operate. Third we outline the proposed approach to optimize the existing BitTorrent protocol called BitTorrent Speed. Fourth, we will describe how BitTorrent Speed will be operationalized with BitTorrent (BTT) tokens. Fifth, we will discuss the generalization of BitTorrent (BTT) Services and describe some of the first decentralized applications being built on it.

## BTT Cryptographic Tokens and Blockchain

BitTorrent is introducing a TRON TRC-10 cryptographic token called BitTorrent (BTT) to act as a general purpose mechanism for transacting in computing resources shared between BitTorrent clients and any other participating service requesters and service providers. BTT will be the unit which denominates transactions for the provision of different services in the BTT-enabled BitTorrent ecosystem.

BTT will be made available as a divisible token allowing for very fine-grained pricing for an evolving set of services within a liquid market of service requesters and service providers.

BitTorrent Inc. will be deploy an “on-chain/off-chain exchange” which exists to facilitate the transfer of tokens between a high-performance private ledger and the public TRON blockchain.

## BitTorrent Speed$^{TM}$ - Incentives to Boost Swarm Lifespan

As observed previously, BitTorrent swarms suffer from structural inefficiencies which lead to frequent premature deterioration or even the death of swarms. Due to bandwidth asymmetry, files frequently complete downloading long before a peer has been able to upload an equivalent amount of bytes. Once the downloading peer has the entire file there is insufficient economic incentive remaining to continue to make the file available to other downloaders ("seeding"). As a result of people leaving swarms without contributing as much bandwidth as they have consumed, many BitTorrent swarms do not last as long as they otherwise could.

In some cases it is possible for a swarm to allow the completion of a download even in the absence of a seed. This possibility is computed and displayed in some clients as an "availability" metric[5], typically expressed as the number of distributed copies available.  If there is at least one active non-seed peer holding each of the pieces, then the file is said to be "available."

The BitTorrent protocol includes a design decision as "rarest first" which dictates that when a client faces a choice of which of its remaining undownloaded pieces to request, it should prefer to request the pieces it knows are held by the fewest peers in the set to which it has connected. This mechanism is intended to flatten the distribution of pieces to decrease the likelihood of a swarm losing a key peer or peers who are the sole providers of a required piece.

These two considerations mean that seeds are not strictly necessary to complete a download. But research has shown that in the majority (approximately 86%) of seedless cases, this sort of collective reconstruction is not feasible and seeders do have a significant impact on file availability.[6]

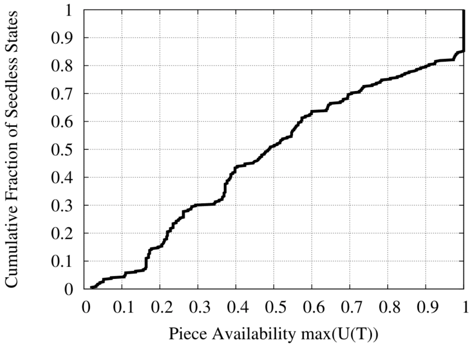


Figure 1, source: *Unraveling BitTorrent's File Unavailability: Measurements and Analysis*

To be clear, BitTorrent overall functions quite well already, and nothing we propose in this optimization will reverse the way the current protocol works. Nor are we suggesting an optimization that is expected to increase the aggregate sharing behavior to new participants who previously were not adding torrents. The

[5] https://wiki.vuze.com/w/Availability

[6] *Unraveling BitTorrent's File Unavailability: Measurements and Analysis* https://ieeexplore.ieee.org/document/5569991

addition we have in mind is simply an overlay on top of the current protocol which will allow existing participants in BitTorrent swarms to allocate resources to each other more efficiently. To this end, we are developing a new BitTorrent feature called BitTorrent Speed, designed to enable peers to offer each other cryptographic-token-based incentives to continue to seed files after the initial download has completed.

The existing BitTorrent barter market will continue intact, but participating BitTorrent clients will implement a new set of extensions to the BitTorrent protocol allowing end users to engage in a market where both BTT tokens and upload speed are used as market inputs.

BitTorrent Speed is a feature that will be integrated into future BitTorrent and µTorrent clients and will enable users to advertise their bids within a swarm and trade BTT in exchange for continued prioritized access to seeds. The intended result is that peers will choose to seed for longer, leading to better swarm longevity and faster download speeds for all swarm participants.

Our approach to implementing BitTorrent Speed starts with how BitTorrent currently allocates resources. BitTorrent uses a sharing algorithm called “tit-for-tat” which is implemented using a mechanism called “choking”. BitTorrent clients classify peers as either choked or unchoked. Only unchoked peers are eligible to receive data from the client. The choke state of all peers is re-calculated periodically (typically every 15 seconds). An example choking algorithm might sort peers by how much data the client has received from each one since the choking algorithm was last executed. The first n peers are then unchoked and the rest choked, where n is the number of unchoke slots, a fixed value chosen by each client implementation. Seeds do not receive any data from peers so they use the amount of data sent to each peer instead. This means seeds optimize for maximum throughput with no regard for fairness, or anything else for that matter.

There are also a number of unchoke slots, typically one, which are reserved for a separate choking algorithm called “optimistic unchoking”. Optimistic unchoking selects peers to unchoke in a random or round-robin fashion. This allows new peers an opportunity to receive some data so that they can start reciprocating with other peers.

Choking is the primary means of allocating resources within a BitTorrent swarm. It is this mechanism that we will adapt to allow for clients to offer rewards to others for continued seeding of content they want access to. Allowing clients to bid BTT for preferential treatment by the choking algorithm gives them a powerful tool to offer incentives to seeds to remain in a swarm.

### Operational Description of BitTorrent Speed

Peers will act as both Service Requesters and Service Providers. A peer offering BTT in exchange for other users’ local resources is a service requester and a peer offering such services in exchange for BTT is a service provider.

## Service Discovery

The BitTorrent Speed application lifecycle begins when peers discover each other via existing BitTorrent protocol mechanisms: they announce to a tracker using an infohash or find peers for a given hash in the DHT[7]. In this way, infohashes naturally segment the space of all peers into swarms of users with a common interest in exchanging pieces of a set of files.

Potential service providers in a swarm are either seeds (peers having a complete locally downloaded copy of a torrent) or peers with partial copies. These service providers advertise what pieces they have available via the existing protocol `have` message.

### Agreement

#### Initial Balance

Prior to making the first bid in their client's lifetime, a service requester must establish a BTT balance. It does this by placing some BTT into a payment channel between the service requester and service provider.

#### Initial Bidding Round

The initial bid is sent via a new bid BitTorrent protocol extension message, sent to each peer having some pieces required by the service requester. The message contains the number of BTT the service requester is willing to bid per piece.

#### Announcing to Trackers

An extension to the BitTorrent tracker protocol, the `bid-announce` key, allows clients to include their current bid when they announce themselves to trackers. The extension adds two new request parameters which allow clients to request peers with the highest bids. Due to the long intervals between announces (30 minutes or more), clients must not trust bids returned by trackers. If a peer's bid turns out to be radically lower than what the tracker claimed then clients should disconnect that peer.

A second tracker protocol extension, `bid-scrape`, allows service providers to retrieve lists of infohashes and recent bids for service against those infohashes. This allows service providers to find torrents in need of supplemental bandwidth in a very efficient and decentralized way.

[7] For detailed description of BitTorrent protocol operations see https://en.wikipedia.org/wiki/BitTorrent

## Reserve prices

Each torrent a client is seeding has a reserve price associated with it. We plan to implement a user-configurable mechanism with defaults designed to enable rewards sought by seeders to grow over time. The default reserve price starts at zero when the torrent is completed and increases as a function of the time since the torrent was last known to be possessed by a peer. Possession by another peer can be proven by a peer submitting a proof of possession of a selected chunk within the torrent.

When a peer connection is opened the client sends a reserve price message containing the reserve price and the index of a chunk whose hash may be sent to prove another seed possesses the torrent. The reserve price message is also sent on all connections when the reserve price or proof chunk changes.

The proof of possession may be sent to the client in a source proof message which contains the piece index, chunk index, and the hash of the chunk. When a valid proof is received the torrent's reserve price is reset to zero.

The chunk used as the challenge to prove possession of a piece should be one which the client has not uploaded recently. The client maintains a bitmap for each torrent where each bit represents one chunk. When the client uploads a chunk its corresponding bit is set to one. When all bits for a torrent become one they are cleared to zero. The client selects which chunk to require by taking the output of a pseudo-random number generator (PRNG) seeded with a secret value XORed with the torrent's infohash. If the selected chunk's bit is set to one then the PRNG is invoked again until the selected chunk's bit is zero.

### Auto-Bid

For the initial release clients will use a simplified auto-bidding mechanism. In this version, the client simply bids a fraction of the remaining BTT balance in its wallet. The bid is calculated as such:

bid = (spending rate) * remaining balance in BTT / (remaining download in bytes / 1024)

This formula implies that as the download progresses, the bid can change. For the initial release, the client will not re-bid until the bid changes by more than 10% from the previous bid, and the spending rate (a parameter that can vary from 0.0 to 1.0 depending on how aggressive the client should be in bidding) will be defined to be 1.0.

In the future, this simple algorithm will be refined. For example, based on existing bid message traffic and current transfers the client can estimate market rate for unchoke slots. The client also has a picture of piece rarity, through normal bittorrent mechanics. Clients may choose to automatically bid limited amounts, if it seems likely they will be able to send rare pieces to more peers who are bidding for slots at a higher rate. This incentive-based behavior more closely models network bandwidth topology than classic tit-for-tat.

### Bidding User Interface

Bidding will take place by default in an automated fashion. Users' clients will earn and bid to and from their token balance on their behalf. We may expose user interface controls to enable users to turn on or off the feature, turn it on or off for certain torrents, adjust the spending rate parameter, set reserve prices, or exercise even more fine-grained control over the bidding process.

### Bidding Revisions and Frequency

As the client may receive data for less than its maximum bid (and very frequently for free, as is currently the case in BitTorrent), the bid computed by dividing the remaining total spend by the remaining data will creep upwards over time. The client can implement any heuristic it likes to determine when to send bid messages with a new bid value, but it should not send new bids more than once a minute. For example, the client could send new bids when its bid value changes by more than 10 percent. If the user changes the total BTT amount then of course the client should send new bids immediately.

### Match Making

A peer participating in the traditional BitTorrent protocol makes decisions about which other peers to send data to ("unchoke") on a periodic basis, largely on the basis of how quickly it is receiving data from each. We extend this unchoke mechanism so that a service provider will include both bid data and peer upload rate in its decisions about which peer to unchoke. In mixed swarms with BTT-enabled BitTorrent clients and legacy BitTorrent clients, service requesters will offer BTT to seeders but download speeds will be saturated without regard for whether a given seed was offered BTT or not. This will retain BitTorrent's competitive bandwidth market where a seed offering fast speeds will continue to be effective regardless of if another one has set a reserve price in BTT in return for seeding.

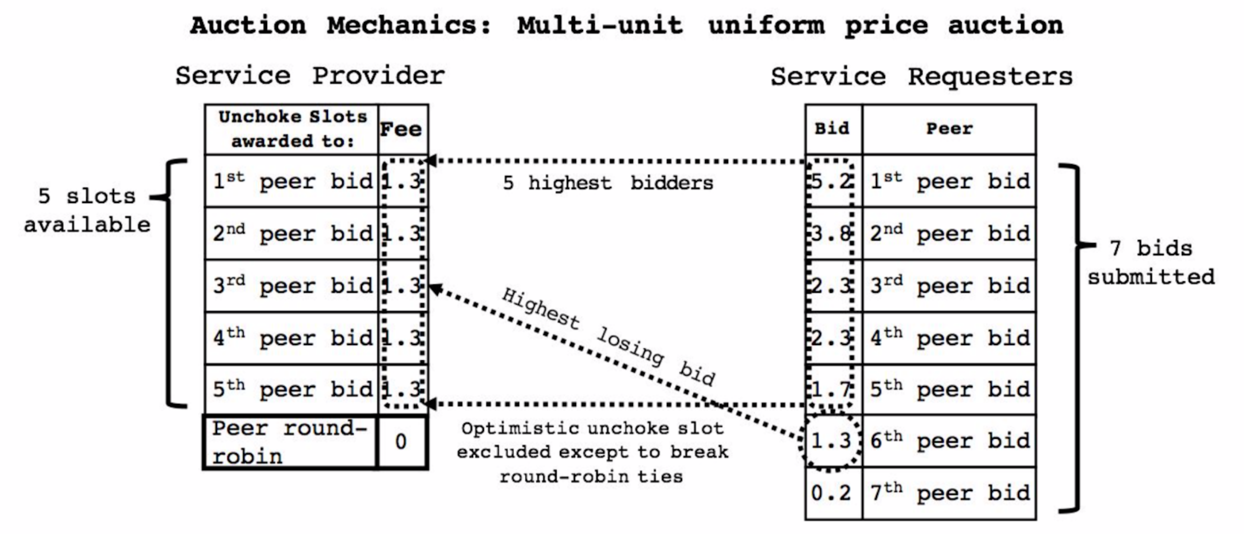
The optimistic unchoke slot should not be subject to the same auction format as the regular unchoke slots. Caution must be exercised when dealing with optimistic unchoking due to its importance in allowing new peers to bootstrap into the swarm. If the client is using a round-robin algorithm for optimistic unchoking, it should only apply an auction to break ties between peers which have gone the same amount of time since being choked. In practice this means auctioning of the optimistic unchoke slot will typically only happen between bidders who have never been unchoked.

When it comes time for the client to run its choking algorithm as described above, it first compiles a list of eligible bidders. A bidder is eligible if it is equal-to-or-greater-than the torrent's reserve price, if any is defined.

The integration of an auction into choking will vary depending on how the client's choking algorithm is implemented. The example algorithm described above could be modified to first sort peers by highest eligible bid, then by most data received. When an eligible bidder is unchoked, the service provider first sends a new protocol extension message bid-response to the bidder containing the rate in BTT/byte the bidder is expected to pay and the details of the payment channel to which the BTT must be sent. The indicated rate must be less-than-or-equal-to the rate received from the bidder. This message is followed by a normal unchoke message. If a winning bidder was already unchoked and the required payment rate has changed, the bidder is first choked and all requests rejected before the bid-response and unchoke messages are sent.

Clients may implement any auction format they please, but a variant of the Vickrey-Clarke-Groves auction is expected to produce optimum results. See Figure 2 for outline of mechanics, which turns out to be a multi-unit uniform price auction.

Each service requester bids on only one unchoke slot per service provider. Clients would therefore charge each unchoked bidder the rate of the highest losing bid, or the reserve price of the requested torrent, whichever is higher. If there is only a single bidder it is charged the reserve price. This means that there is always a base level incentive for bidders to ask for service as in the absence of any other bids they will get the service for the reserve price regardless of what they bid.


Auction Mechanics: Multi-unit uniform price auction
Service Provider
Service Requesters
5 slots available
5 highest bidders
Highest losing bid
Optimistic unchoke slot excluded except to break round-robin ties
7 bids submitted

| Unchoke Slots awarded to: | Fee |
|---|---|
| 1st peer bid | 1.3 |
| 2nd peer bid | 1.3 |
| 3rd peer bid | 1.3 |
| 4th peer bid | 1.3 |
| 5th peer bid | 1.3 |
| Peer round-robin | 0 |

| Bid | Peer |
|---|---|
| 5.2 | 1st peer bid |
| 3.8 | 2nd peer bid |
| 2.3 | 3rd peer bid |
| 2.3 | 4th peer bid |
| 1.7 | 5th peer bid |
| 1.3 | 6th peer bid |
| 0.2 | 7th peer bid |

Figure 2.

## Transaction Processing

Once a service requester receives unchoke and bid-response messages, it sends a contract to the private ledger for the amount of a single piece at BTT/byte rate.

We define the Client to be the party sending BTT, and the Seeder to be the party receiving it.

1. Client creates a public key (K1) and requests a public key from the Seeder (K2). 2. Client creates and signs but does not send to a payment channel a transaction (T1) that sets up a payment of BTT/byte rate times the number of bytes needed, to an output requiring both the Seeder's private key and a client key,

using OP_CHECKMULTISIG. 3. Client creates a refund transaction (T2) that is connected to the output of T1 which sends all the money back to the Client. It has a time lock set for some time in the future, several times longer the the expected download time, plus a few hours. The Client doesn't sign it, and provides the unsigned transaction to the Seeder. By convention, the output script is "2 K1 K2 2 CHECKMULTISIG" 4. The Seeder signs T2 using its private key associated with K2 and returns the signature to the client. Note that it has not seen T1 at this point, just the hash (which is in the unsigned T2). 5. The Client verifies the Seeder's signature is correct and aborts if not. 6. The Client signs T1 and passes the signature to the seeder, which now sends the transaction to the payment channel (either party can do this if they both have connectivity). This locks in the BTT. 7. The Client then creates a new transaction, T3, which connects to T1 like the refund transaction does and has two outputs. One goes to K1 and the other goes to K2. It starts out with all value allocated to the first output (K1), ie, it does the same thing as the refund transaction but is not time locked. The Client signs T3 and provides the transaction and signature to the Seeder. 8. The Seeder verifies the output to itself is of the expected size and verifies the

Client's provided signature is correct. 9. When the Client wishes to pay the Seeder, it adjusts its copy of T3 to allocate more value to the Seeder output and less to its own. It then re-signs the new T3 and sends the signature to the Seeder. It does not have to send the whole transaction, just the signature and the amount to increment by is sufficient. The Seeder adjusts its copy of T3 to match the new amounts, verifies the signature and continues.

This continues until the transfer ends, or the timeout from step 3 is getting close to expiry. The Seeder then signs the last transaction it saw and sends it to the payment channel, allocating the final amount to itself. The refund transaction is needed to handle the case where the Seeder disappears or halts at any point, leaving the allocated value in limbo. If this happens then once the time lock has expired the Client can send the refund transaction to the payment channel and get back all the BTT.

The lock time and sequence numbers avoid an attack in which the Seeder provides pieces, and then the Client double-spends the output back to themselves using the first version of TX2, thus preventing the Seeder from claiming the BTT. If the user does try this, the TX won't be included right away, giving the Seeder a window of time in which it can observe the TX from the payment channel, and then send the last version it saw, overriding the Client's attempted double-spend.

As normal, when the service requester receives an unchoke message, the service provider will begin to send pieces.

If for some reason the transfer does not complete after a timeout, the service requester is choked and receives no further data. Repeated failures to transfer BTT by a service requester can result in the service provider banning the service requester. Banned service requester peers are disconnected and any attempt to reconnect by the service requester is rejected for some period of time. Similarly, failure to verify data from the service provider can result in the service provider being banned.

Each party progressively contributes bandwidth (pieces) or BTT, with a signed transaction produced for each step in the process. The maximum breach exposure of the service provider at any given time is therefore one piece worth of bandwidth, and since service requesters pay only on verified delivery, they have zero breach exposure.

## Generalized BTT Services

Optimizing the existing BitTorrent protocol is an obvious first step in the introduction of a cryptographic token but it barely scratches the surface of what is rapidly becoming possible. The precedent being set is allowing users to store value from sharing small amounts of infrastructure in order to spend it later. We are setting ourselves on a path to dramatically extend both the earning opportunities and the spending opportunities for users of BTT-enabled BitTorrent clients. To address earning opportunities we are developing a range of generalized BTT services and preparing to open up the platform to 3rd party developers who could make use of those services provided in return for paying BTT.

As a result of extensive discussions with partners interested in our platform we have concluded that there will be three BTT Services offered at first:

(1) A decentralized content delivery service to enable service requesters to advertise bids and pay BTT for bandwidth to receive a particular piece of content. This service will be well suited for mass distribution of content, especially in the presence of censors or other types of attackers. Service providers will be incentivized to make available content which they can serve to as many people as possible, thus ensuring robust performance even with very large numbers of service requesters.

(2) A decentralized storage service to enable service requesters to pay for storage over time. Service providers will agree to store some amount of data and provide proofs-of-storage to the service requester on-demand. Service requesters will also be able to download the stored data from the service provider for a pre-arranged fee. Service providers will seek out content which offers the highest payment rate over time. This service will be useful for remote backup and sharing of private data among small groups.

(3) A decentralized proxying service to enable service requesters to pay a client for retrieval of content by URL. This will be useful to highly mobile applications or those which seek to evade IP-level network controls. The protocol will also be designed to allow content to be requested in chunks. This will, for example, allow clients with intermittent connectivity, such as mobile users relying on wifi, to reliably retrieve web resources without needing to maintain an open connection long enough to receive the complete contents.

More BTT Services can be implemented and introduced into the network of service providers as demand emerges from new BTT Applications. BitTorrent Foundation will provide a forum for discussion and standardization of new BTT Services similar to what it provides for the BitTorrent protocol.

As characterized in Figure 3, the various enhancements to the BitTorrent protocol as well as the BTT transaction processing approach outlined will be formally documented. These components will serve as building blocks for distributed applications.

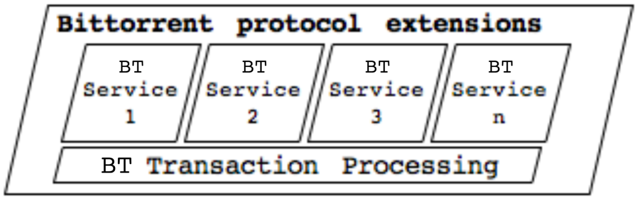


Figure 3.

The BitTorrent protocol extensions will be submitted for comments to the BitTorrent community BEP process — an informal but open standards-setting process[8] and facilitated by BitTorrent Foundation that has guided protocol enhancements for over a decade. Following community feedback we will develop and test our implementation of these extensions via engineering and release management practices which are well established at BitTorrent.

Furthermore, as is our normal practice with highly important updates , we will subsequently release these extensions as an open source library and establish support and incentives for integration into third-party BitTorrent or other clients in order to broaden the pool of clients that can support BTT Applications as much as possible.

## BTT Incentives

The continued evolution of the BitTorrent ecosystem in this type of productive direction will require coordination and incentives provided to a broad range of existing and future participants. Other BitTorrent client implementers, third-party app developers and other online publishers will all be eligible for a system of BTT incentive awards.

The BitTorrent ecosystem has proven over many years that millions of people will enthusiastically share their resources if they can cooperate safely and securely bound by the rules of a protocol they trust. By introducing a mechanism for value storage and exchange we aim to greatly broaden the universe of possible participants - either service requesters, service providers or both. To maximize chances for success it is vital that we ensure that BitTorrent Inc. is not a central monopolist in the BTT-enabled BitTorrent ecosystem, just as it is far from a monopolist in the BitTorrent ecosystem of today. This will require coordination of activities and provision of incentives to a broad range of existing and future participants.

The BTT Project depicted in Figure 4 is one where the success of ecosystem partners will lead to increasing returns for all ecosystem participants.

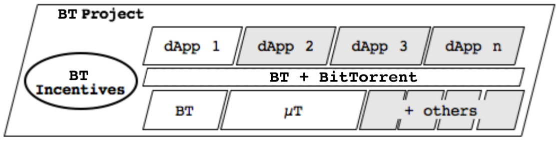


Figure 4

The purpose of the BTT Incentives will be to:

- Promote the BTT project to current and prospective participants, be they service providers, service requesters or both - this means finding and introducing new application developers who are interested in participating in service requests or service provision; - Govern membership and participation rules among participants in the BTT Ecosystem with an overriding objective of establishing a level playing field on

[8] http://www.bittorrent.org/beps/bep_0000.html

which all can participate fairly; - Govern the equitable and transparent distribution of rewards and incentives such that

promising ideas have a fair chance and productive outcomes are fairly rewarded; - Work in conjunction with volunteers on bittorrent.org to facilitate discussion around future

related BitTorrent protocol extensions.

Once the BTT project is running sustainably we may consider transitioning the rules and procedures it has established for administering BTT incentives into a lower overhead instrument such as a decentralized autonomous organization (DAO). To begin with, however, the immediate future will require more human ingenuity and flexibility which is why we believe that we must invest heavily (both people and tokens) in BTT Incentives.

# Implementation Considerations

## Blockchain

BTT Applications will be supported by active BitTorrent users who number in the tens of millions each day. To minimize the opportunities for fraud, BTT Applications will provide service in small increments then wait for payment to be confirmed before more service is provided. This will require transactions to be handled at a granular level and confirmed in a matter of seconds, and ideally in less than a second. Furthermore, even our most conservative estimates of capacity requirements anticipate dozens of transactions per second. With these needs in mind it is clear that existing public blockchains will not be able to support on-chain processing and settlement any time soon.

## User Controls

We plan to introduce features like BitTorrent Speed and BTT transactional support into BitTorrent and µTorrent clients in a phased way to allow us to iterate towards the clearest possible user education journey and thus to optimize end-user participation. Participation in the BTT transactions is required to be both fully disclosed and optional for end users.

## BitTorrent Wallets

As part of the rollout of new BitTorrent and µTorrent software that can participate in the BTT applications we will be distributing integrated cryptocurrency wallets to all users. As we will be distributing these wallets on a large scale to mass market end users, and not necessarily to cryptocurrency enthusiasts, we will need to pay close attention to simplicity and usability.

Bootstrapping: The first available BTT application will be the BitTorrent Speed feature which will be unproven at the outset. Given new service requester services will take time to become prevalent at any scale, we may additionally pursue a strategy to pre-seed the market with promotional quantities of BTT.

Use Case Diversification for BTT: The introduction of BTT wallets on the scale anticipated by this project may create opportunities for new uses for the token that are wholly unrelated to BitTorrent technology. We expect that many millions of users will be able to accumulate small quantities of BTT that may not have material value until they are aggregated by service providers. That is to say that users who may accumulate only a small amount of BTT from providing services will look for ways to spend that token that go beyond their need for incentivizing seeding. In due course we expect to evangelize this new capability and explore partnerships that can accelerate merchant acceptance of this new type of micropayment mechanism. This will be particularly advantageous to merchants who want to aggregate and use tokens to pay for infrastructure services to support their ongoing services.

We expect to be able to establish an economy as characterized in Figure 5 where BTT are introduced into the economy primarily by distributed app developers, are then traded between service requesters and service providers within and beyond the BitTorrent ecosystem, and may ultimately aggregate in commercially significant pools at some service providers who may be part of the BitTorrent ecosystem or may not. At this point of the cycle the BTT will be returned via the open market to new service requesters who would like to exchange them for distributed infrastructure services provided by BitTorrent users.

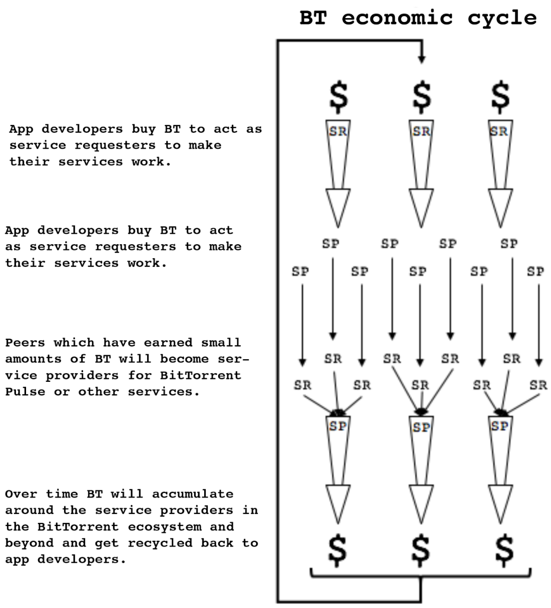

BT economic cycle
App developers buy BT to act as service requesters to make their services work.
App developers buy BT to act as service requesters to make their services work.
Peers which have earned small amounts of BT will become service providers for BitTorrent Pulse or other services.
Over time BT will accumulate around the service providers in the BitTorrent ecosystem and beyond and get recycled back to app developers.
$
SR
SP

Figure 5

Identity: BitTorrent as a protocol has never provided any type of identity service that goes beyond identifying a client on a particular IP + port number. That is to say that BitTorrent essentially identifies instances of software running on machines rather than people. This is analogous to the identity framework behind cryptocurrencies. If you have access to the cryptographic token wallet software that stores a token then it is generally assumed that it is your token. With the implementation of BTT we expect to follow a very similar approach to identity and expect to tie BTT tightly to a participating piece of client software. Beyond the possibility of placing a password on that wallet, we do not currently anticipate that the BTT project will directly give rise to the need for any additional new layer of identity management in BitTorrent.

# BTT Token Issuance

Our objectives for an offering are to:

We will create a total supply of 990,000,000,000 BTT. The total supply with be divided as follows: the allocation to BitTorrent, Inc. torrent client users as part of client install and onboarding, the allocation to existing TRX holders, the initial supporters and investors, the BitTorrent Foundation and team, the TRON Foundation.

Users of BitTorrent, Inc.'s torrent clients (and possibly other torrent clients which choose to implement the required set of protocol extensions) will be able to submit a CAPTCHA or proof of work which will allow them access to an initial balance of BTT.

# Conclusion

We have presented our motivation, qualifications and plans for extending BitTorrent via the BTT project starting with a new core feature designed to improve the BitTorrent ecosystem, a new cryptographic token and a practical implementation of cryptographic token transaction processing at scale.

We have outlined how we are generalizing this approach to enable other distributed app developers to use a distributed infrastructure platform composed of over 100 million consumer BitTorrent clients for the provision of networking and storage resources in return for BTT. In particular we have described three novel distributed apps from independent app developers that plan to make use of the platform and outlined the value they see in it.

We have described the mission and operation of the BTT Incentives Program which will be dedicated to driving the number and success of the BTT distributed apps. It will manage the progressive release and distribution of BTT to ecosystem participants which are successful in driving useful platform adoption.

We have discussed some important implementation considerations and challenges and how we expect to address them.

And we have presented a plan for how tokens will be issued and shared in the pursuit of a stable and thriving economy built around the the sharing of computing resources by millions of ecosystem participants.

The potential of this project is compelling not just because of the disruptive decentralized applications that it will enable, but also because of its open ecosystem approach that will welcome and reward participants at every level and finally because of the enormous head start the BitTorrent ecosystem enjoys in the building and deploying a decentralized computing economy.

# References

BitTorrent https://en.wikipedia.org/wiki/bittorrent_(software) BitTorrent clients https://en.wikipedia.org/wiki/Comparison_of_bittorrent_clients BitTorrent protocol: https://en.wikipedia.org/wiki/bittorrent BitTorrent company: https://en.wikipedia.org/wiki/bittorrent_(company)

[*] BitTorrent.org: http://www.bittorrent.org [*] BitTorrent.org BEP Process: http://www.bittorrent.org/beps/bep_0000.html [*] BitTorrent protocol specification: http://www.bittorrent.org/beps/bep_0003.html

[*] BitTorrent.com: http://www.bittorrent.com [*] µTorrent.com: http://www.utorrent.com

[*] Libutp https://github.com/bittorrent/libutp [*] Newly open source BitTorrent protocol aims to unclog tubes https://arstechnica.com/information-technology/2010/05/BitTorrent-open-sources-new-protocol-implementation/

[*] Unraveling BitTorrent's File Unavailability: Measurements and Analysis http://ieeexplore.ieee.org/document/5569991/

# Appendices

# FAQ

## Why not rewrite the BitTorrent protocol?

The maturation of cryptocurrency projects like Bitcoin is one more proof (just like BitTorrent) that distributed protocols can implement incentives that allow large numbers of untrusted network participants to collaborate productively. Bitcoin’s novelty compared to BitTorrent is that it introduces the blockchain concept enabling collaboration can endure over time, unlike BitTorrent where collaborations are transient and occur in wholly separate and unrelated events called ‘swarms’. We considered a fundamental rewrite of the BitTorrent protocol to allow collaboration to be persisted over time and to ensure ‘the right seeding behavior’ was rewarded so that more long-tail content (valuable content with only occasional demand) would be available for longer. We imagined a protocol which would both download (like BitTorrent) and hand out longer-term incentives (like bitcoin mining rewards). After lengthy consideration we discounted this approach for several reasons:

1. Difficulty of the problem - implementing an incentive system at the protocol level requires precise thinking about objectives. We found it impossible to articulate clearly what the long-tail seeding objectives should be and how to avoid gaming them - there are plenty of BitTorrent swarms that die because literally no-one cares (e.g. some better version of a file becomes available) - the only tractable answer seemed to be to implement a voting system to let consumers judge, but that seemed to call into question the desire to wrap everything into the protocol. In short, trying to programmatically discern

between what should and should not be preserved seemed like a problem we were poorly equipped to answer without asking end users. 2. The need to be strictly better than existing BitTorrent (a.k.a. “soft-fork not hard-fork”) - any protocol rewrite would have to be compatible with the existing BitTorrent ecosystem - this immediately rules out things like penalties for not seeding - consumers would just choose to use clients which implemented the ‘old’ BitTorrent protocol which did not penalize them. The parallel to this issue within the existing Bitcoin space is the growing difficulty of implementing hard forks. The BitTorrent ecosystem is now so big that a hard fork would have an extremely low chance of success. 3. Conviction that we were over-complicating the solution - the likely need for human agency in the system (people voting) convinced us we should focus on simpler extensions to BitTorrent as-is and design a voting system that was based around an existing cryptocurrency. This has the advantage of allowing the market to determine what should be seeded while leaving BitTorrent enhanced but not changed at its core.

## Why did BitTorrent not include incentives when it was invented?

In fact early research into projects that were forerunners to BitTorrent did try to imagine how a system of persistent incentives might be managed. They foundered largely due to the difficulty of finding an effective solution to `keep the score’ while operating at scale. Blockchain and distributed ledger solutions using cryptographic tokens present a powerful new way to keep the score such that transactions can be processed and a ledger may be managed at scale even in the absence of perfect trust between all counterparties.

## How can this solution help me get around net neutrality adversaries?

Some examples: Proxying from IP to IP will enable users to find content that is blocked by an ISP in their geography by connecting to it via an intermediary to which both site and requester can connect.

## How will you protect end users computers from malicious attacks?

The usage of end-users’ technical resources will be strictly limited to the provision of technical services like networking or storage within carefully bounded limits. Network connections will be protected by uTP - a self-adjusting bandwidth mechanism which ensures applications throttle back if there is any indication of other apps (even on other devices) using the network connection. Storage will be encrypted and limited to a user-configurable maximum. And users will be able to configure which applications they accept and which they do not. The provision of BTT services is limited to simple infrastructure operations and in no way will permit untrusted third parties to execute code on a user’s device.

## Can users opt out? What if they don’t want to provide their resources or earn tokens?

Yes, users will always be able to configure the parameters of their sharing or switch it off entirely if they choose. There should be nothing mandatory about this ecosystem and users will retain the right to opt out for any reason.