Torrente, a micropayment based Bittorrent extension to mitigate free riding

Felipe K. Shiraishi[1], Vitor H. Perles[1], Hector K. Yassuda[1], Leonardo T. Kimura[1], Ewerton R. Andrade[1,2], Marcos A. Simplicio Jr.[1]

[1]Escola Politécnica, Universidade de São Paulo (USP), Brazil

[2]Universidade Federal de Rondônia (UNIR), Brazil

{fkspoli,vitorhugoperles,hector.kobayashi,leonardo.kimura}@usp.br

ewerton.andrade@unir.br, msimplicio@larc.usp.br

Abstract. *We propose Torrente, a distributed file-sharing solution with economic incentives. Its implementation is built as an extension of BitTorrent protocol, in such a manner that user access to file-sharing swarms is controlled by peers that verify micropayments receipts in a blockchain-based ledger. In addition, by using payment commitments, Torrente facilitates off-chain transactions for faster content sharing. The solution is created as a tool to enhance Amazon Biobank application security, but can be used isolated in cases such as file sharing with monetary incentives.*

1. Introduction

Peer-to-peer (P2P) distributed systems usually deploy mechanisms for avoiding the occurrence of selfish behaviour in their networks [Labs 2017, Cohen 2003, Pant and Kumar 2018, Nair et al. 2008]. A common example is "free riding", which refers to a nodes behavior of avoiding contributing bandwidth with the network, and disconnecting from it as soon as the desired resources and services are obtained. When analyzed from a game theory perspective, this undesirable behaviour is actually an optimal individual strategy to adopt in networks that do not punish or reward collaborative behaviour [Zhang et al. 2009].

Proposals of reward-punishment systems suitable for such distributed systems pertain to three major classes: based on reciprocity, on fixed contribution and monetary incentives [Zhang et al. 2009]. The goal of Torrente is to extend the set of rewards and punishments of Bittorrent, adding monetary incentives to the combination of tit-for-tat and optimistic unchoke mechanisms already existing in the protocol [Cohen 2003]. This extension is particularly aimed at scenarios where: (1) the availability and throughput of big sized files through the network challenges a resources-bounded environment; (2) increasing the availability of data storage and bandwidth by adding server nodes into the network is not an option; and/or (3) the data content is not necessarily of interest for the peers, meaning that they would not store or distribute the data unless some clear incentive is available. In such scenarios, monetary incentives can be used to encourage users to enter the network and to contribute with resources.

This application is conceived and designed to be a security enhancement for the Amazon Biobank application [Kimura et al. 2021]. One of the Amazon Biobank ob-

jectives is to make available DNA files to desiring users. Since DNA files are storage-intensive resources, providing it in a centralized server would make the node availability fragile. In order to enhance its availability protection, this project presents a descentralized storage and distribution solution based in BitTorrent. However, by adopting that solution, other security issues emerges.

The content of the data being stored and distributed should only be accessible by paying users. So, a confidenciality requirement is necessary. The application provides the torrent content encryption option in order to permit its sharing over the network without compromising its access limitation rules. The keys can be bought on the federated centralized peers from the Amazon Biobank. Since key files aren't as big as DNA files, centralized nodes can handle its download requests.

Lastly, since that data will be encrypted, there may be no incentives for peers to colaborate with the network if they aren't interested for its content. That's why the payment component was introduced. To implement such a file sharing system while preserving the distributed nature of Bittorrent, we propose the adoption of a blockchain-based ledger for storing monetary transactions among peers exchanging data pieces. That process requires data artifacts available in a distributed manner in order to achieve authenticity and irreversibility in its operations.

The solution is expected to be built upon highly-scalable micropayment schemes [Perez et al. 2020]. Those schemes are meant for use in scenarios where very frequent financial transactions are expected to occur, such as paying for every second of a video streaming service. For this purpose, they adopt fast and reliable consensus protocols such as Consensus [Chase and Macbrough 2018] or Raft [Androulaki et al. 2018], limiting the latency overhead in a communication protocol. As an additional tweak, though, we use a mechanism similar to PayWord [Rivest and Shamir 1997], enabling some level of off-chain transactions to be executed via hash chaining. The result is that one transaction registered in the blockchain dynamically enables the exchange of an arbitrarily large number of blocks. The benefits of such an off-chain approach are: (1) it limits the number of blocks appended into the blockchain when performing payments; and (2) it reduces even more the underlying latency and processing overheads. Finally, by adopting a blockchain that supports smart contracts (e.g., Hyperledger [Androulaki et al. 2018]), Torrente creates a flexible environment where fine-grained payment rules and restrictions can be defined by the systems peers.

Torrente is a work in progress implementation that can be installed from the following github repository: `https://github.com/amazon-biobank/Torrente/releases`. Installation, build from source instructions and general documentation can be accessed from its wiki: `https://github.com/amazon-biobank/Torrente/wiki`.

The rest of this article is organized as follows. Section 2 details the Torrente architecture and design decisions. Section 3 describes the required setup for its demonstration. Section 4 discusses the related works. Section 5 explains some relevant use cases and limitations of the proposed solution. Section 6 presents our final considerations.

2. Solution architecture

Torrente solution consists in a client-side application comprising two main components [F. Shiraishi 2021]: a BitTorrent client which is a fork from the open source qBittorrent

solution [qBittorrent 2021]; and Payfluxo, a connection bridge with a Hyperledger backend, which is implemented in node.js. The overall architecture is depicted in Fig. 1.

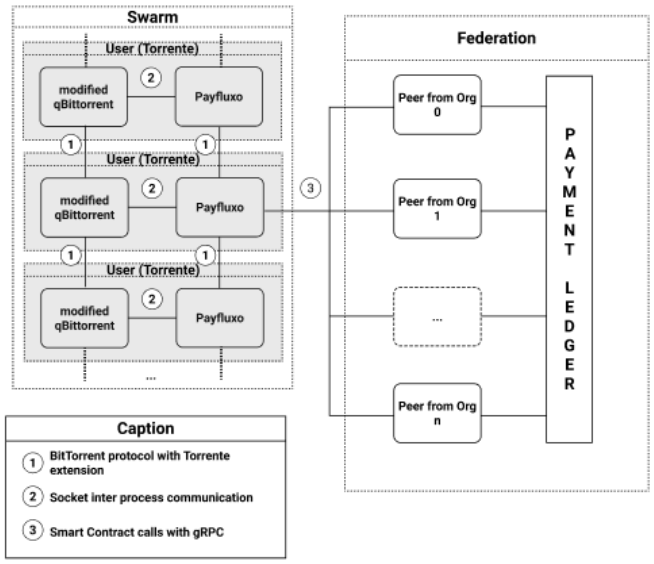

Figure 1. Overall architecture of Torrente.

Essentially, the qBittorrent client was modified to include, besides BitTorrent's download/upload mechanisms, the required apparatus for interworking with a Hyperledger Blockchain. In particular, before enabling the upload of data pieces to a requesting peer, the uploader verifies that: payments commitments are correctly registered in the Hyperledger, so enough of the payer's funds can be used for purchasing data pieces; and, when off-chain PayWord-like payments are made for each data piece, the hash chain links provided by the downloader as payment confirmation are valid. The payment mechanisms employed are further discussed in Sec. 2.3.

As for the Payfluxo component, it was developed to ensure an appropriate communication with the Hyperledger, whose SDK was only available in node.js and Java environments at the time Torrente was being developed. To make the solution behave like a cohesive application, a socket was opened between the Qbittorrent client, using the Qt framework [Qt project 2020], and Payfluxo. The main responsibility of Payfluxo module is, thus, to execute the micropayment protocol between peers and commit the result of these operations, as a request, to peers maintaining the Hyperledger. By notifying the modified BitTorrent client, it can perform its protocol extension over BitTorrent itself.

2.1. Authentication toward Federation

Albeit Torrente may be used with an open (also called non-permissioned) Blockchain as micropayment backend, it is also suited for operating with a Federated (sometimes also named Permissioned) Blockchain (a common use case in Hyperledger-based solutions). For this purpose, Torrente's BitTorrent client provides support for user authentication towards the Federated Blockchain system. The authentication method is based on digital certificates. Specifically, users are expected to create a public and private key pair, and register the public key via an authorized portal provided by the federation (e.g., Fig. 2 shows the portal for the Amazon Biobank project described in [Kimura et al. 2021]).

Then, after receiving a digital certificate signed by the Federation, the user should register its private key and digital certificate in the modified qBittorrent client. Torrente will then store the private key encrypted with AES-256-CFB [Dworkin 2001]. The corresponding secret key is derived from the user-provided password using the Lyra2 password-hashing scheme [Andrade et al. 2016], aiming to thwart brute force attempts of decryption via password guessing in case the device where Torrente is installed gets invaded. As a result, after the user provides its password, its authentication toward the federation is handled by Torrente itself, enabling the required payment features.

We note that our prototype can also be executed in an unauthenticated state, in which case the Torrente client acts like a regular qBittorrent client, without the proposed extension over the protocol.

2.2. Blockchain

The Torrente application makes use of a blockchain based ledger to track users funds and register declarations of download intents, which act as payment commitments whose practical effect is freezing funds from a downloader. The peers maintaining the blockchain can then act as gateways to the blockchain services, such as providing requested blockchain states and smart contract methods invocations for the Torrente users.

The motivation for adopting Hyperledger Fabric in our prototype comes from the fact that it supports very efficient consensus mechanisms when operating with a federated blockchain. This facilitates the construction of micropayment solutions, where efficient block validation is a common and important requirement. At the same time, in such a federated environment one can more easily handle misbehavior by users, e.g., via revocation of rights and by applying real-world legal sanctions whenever applicable.

2.3. Download intention declaration

To enable monetary incentives in a peer-to-peer data sharing environment, Torrente operates as follows. As soon as a user adds a torrent to its download queue, it emits a message to Payfluxo to register a download declaration in the back-end Blockchain. This declaration consists in a smart contract method call to freeze funds from the downloader equivalent to the file price. This operation is important for uploaders to decide if they will respond to a request from a peer. After all, uploaders are only expected to engage in sessions for which they can assert the existence of valid reserved funds for the corresponding data to be uploaded.

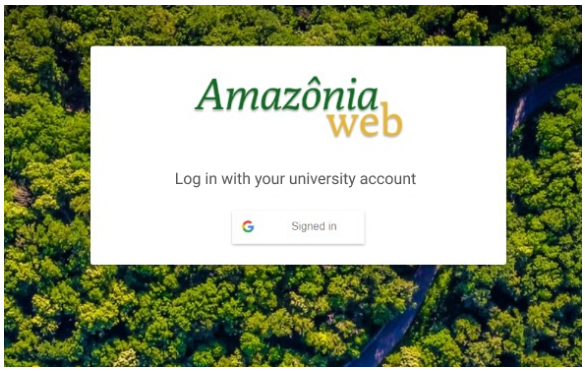

Figure 2. Amazonia Web, the USP portal for students registering.

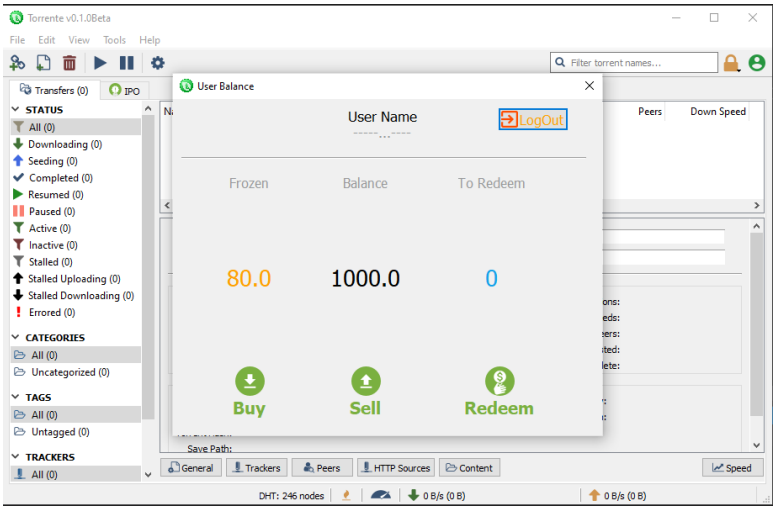

Figure 3. Interface to manually redeem values.

The method adopted to define the price of a complete download operation consists of multiplying the target file size by the piece price defined in the blockchain. The manner by which the price of each piece is defined is outside the scope of this document. For example, the price could be registered in the blockchain as a mutable state that can only be modified by a majority agreement criteria of the federation peers.

2.4. Payment procedure

For every new connection made with a peer, a payment session is opened before any data piece is exchanged. When a downloader successfully receives a piece from an uploader, it generates a hash chain, similarly to what is proposed in Payword, and registers it in the Blockchain in the form of a signed commitment with an expiration date. The commitment is then sent to the uploader, who checks its validity and status in the Blockchain. The availability of funds by the download is also verified, by asserting the correspondence of the received commitment with the download declaration. Once the commitment is accepted, the data sharing procedure effectively starts. Specifically, the downloader provides one pre-image hash link payment for each downloaded data piece that passes the BitTorrent integrity check. This allows the uploader to perform an off-chain verification of the payment's validity, thus enabling the delivery of the subsequent data piece. Since this approach involves simply hash computations, it can be done very efficiently, without introducing any noticeable delays in the communication.

2.5. Payment redeem

Every payment commitment includes an expiration date within which payments can be redeemed. The payment redeem by the uploader may be activated in three different ways: manually; automatically, when close to the redeem expiration date; or automatically, when the communication between downloader and uploader is halted for a long enough period.

In our prototype, the manual redeem activation may be performed by the user by accessing the funds balance dialog when authenticated. A button labeled "redeem" initiates that use case as shown in Fig. 3.

A payment redeem operation does not invalidate a payment channel established with the downloader. Instead, as long as there are still enough funds associated with the original commitment, its hash chain can be reused for future redeems, with smaller values. Even though this allows uploaders to perform one redeem operation per provided block, they are expected to refrain from doing so aiming to reduce the number of Blockchain operations performed (and, hence, the eventual payment of associated fees).

3. Proposed demonstration

The proposed demonstration for this application consists of at least two desktop devices, e.g., with a Windows or a Debian distribution operation systems, where Torrente is appropriately installed. They can be inside the same LAN, or in different LANs. If they are positioned in diferent LANs, each one behind a NAT, the NAT devices must have appropriate PCP, NAT-PMP or uPnP protocols embedded. If these devices doesn't have an appropriate implementation for these protocols, the NAT devices must have a previous port forwarding over the port 9003 for the demonstration machines. It is important to emphasize that if simplicity is desired for the demonstration, having both machines in the same LAN is the optimal setup.

Also, a encrypted file with the mentioned methods with the appropriate credentials must be available for both devices. A suggested example file may be provided by the authors.

To perform a file sharing session, a big file torrent generation is recommended. The downloader node may use the torrent file or the magnetic link in order to initiate a download session. In order to begin the download session, users must be in authenticated states. The expected result of the demonstration is to check the redeemable value rendered in the balance dialog after a upload session is done.

4. Related Works

To the best of our knowledge, the main solutions in the literature that handle the problem of free-riding via monetary incentives, complementing possible reciprocity mechanisms, are BitTrusty [Pant and Kumar 2018], Floodgate [Nair et al. 2008], and Filecoin over the InterPlanetary File System (IPFS) [Labs 2017]. Even though those schemes were studied and some of their requirements were incorporated as guidelines in the design of Torrente, there are relevant differences among those solutions, as discussed in what follows.

BitTrusty [Pant and Kumar 2018] proposes solving free-riding by encouraging users by remunerating them with Bitcoin. Only limitation of the proposed solution is that, due to the adoption of Bitcoin as underlying payment solution, the overall performance of the system is likely to suffer from the reasonably long delays required for transaction validation. Torrente avoids such issues by combining two mechanisms. The first is a micropayment system, built upon the Hyperledger infrastructure with the Raft consensus protocol, to ensure that transactions are validated much faster than what is obtained with Bitcoin's proof-of-work. The second is a Payword-like mechanism for creating an association between each micro transaction and uploaded data blocks. This facilitates off-chain payments, which can later be validated in batch in the Blockchain, thus reducing the Blockchain growth and enabling even faster transactions.

Floodgate [Nair et al. 2008] introduces another form of encouraging contribution in a file distribution network using a micropayment system. The protocol presented enables monetized P2P data sharing without the explicit use of cryptocurrencies. Due to this centralized nature, Floodgate create a single point of failure, weakening some of the main advantages of distributed data sharing networks like BitTorrent: its resilience and scalability. Torrente, on the other hand, employs a distributed micropayment scheme that is compatible with the distributed nature of BitTorrent, using smart contracts and cryptocoins to enforce payments according to the amount of data uploaded by the payee.

Filecoin [Labs 2017] is a crypto-currency associated with a file storage system protocol running over IPFS. Its rewarding mechanism consists in applying a Proof of Spacetime and Proof of Replication to assert if a file is appropriately stored by its peers. The main difference between Torrente and Filecoin is that the latter focus on rewarding appropriate content storage, while the former's goal is to ensure that peers storing some piece of data actually participate in the data sharing process. In other words, a peer who has some piece of data in storage is only rewarded by Torrente if it actually distributes this data to interested parties. Conversely, in Filecoin peers might be rewarded even if they adopt a selfish behavior of keeping the data for themselves.

5. Discussion

It is acknowledged that the changes in the protocol to support the micropayments will impact the performance of both download and upload speeds in the network. However, determining how much it will degrade and the ways of minimizing it requires series of tests and adjustments that can be developed in future works.

Another point of interest in further investigations is the vulnerabilities that the implementation may introduce to the protocol. One known vulnerability is the possibility of collusion of malicious peers to get a shared file for free. This is possible due to the payment being made after the piece is received by the downloader. This can be mitigated by a large number of pieces, but other strategies and their effectiveness can be studied in the future.

The usages of this implementation can also be explored ahead. In this work, a generic incentivized file-sharing system is presented. It can be extended for many areas of interest like content and media. This implementation will be used at and was motivated by Amazon Biobank as mentioned in introduction [Kimura et al. 2021].

6. Conclusion

During the elaboration of this work, it was possible to implement a successful proof of concept modification of the QBittorrent using the related services Payfluxo and Hyperledger Fabric. The implementation in the current state enabled a set of users to share a file and interact with the blockchain and the hash chain to pay and get paid for shared content.

In conclusion, this work presents a practical proof of concept that implements an economically incentivized distributed file-sharing system using micropayments. The many use cases and implications of this Bittorrent protocol extension can be further be explored in future works.

Acknowledgments: This work was supported by Ripple's University Blockchain Research Initiative, and by CNPq (grant 304643/2020-3).

References

Andrade, E., Simplicio, M., Barreto, P., and Santos, P. (2016). Lyra2: Efficient password hashing with high security against time-memory trade-offs. *IEEE Transactions on Computers*, 65(10):3096–3108.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S. W., and Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proc. of the 13th EuroSys Conference (EuroSys'18)*, New York, NY, USA. Association for Computing Machinery.

Chase, B. and Macbrough, E. (2018). Analysis of the XRP ledger consensus protocol. Technical report, Ripple Labs, Inc.

Cohen, B. (2003). Incentives build robustness in bittorrent. `www.bittorrent.org/bittorrentecon.pdf`.

Dworkin, M. (2001). *(SP 800-38A) Recommendation for Cipher Modes of Operation*. National Institute of Standards and Technology, Gaithersburg, MD, USA.

F. Shiraishi, V. Perles, H. Y. (2021). Torrente github repository. `github.com/amazon-biobank/Torrente`.

Kimura, L., Andrade, E., Carvalho, T., and Simplicio, M. (2021). Amazon Biobank: sustainable development built upon rainforest's biodiversity. In *Planetary Health Annual Meeting and Festival*.

Labs, P. (2017). Filecoin: A decentralized storage network. https://filecoin.io/filecoin.pdf.

Nair, S. K., Zentveld, E., Crispo, B., and Tanenbaum, A. S. (2008). Floodgate: A micropayment incentivized p2p content delivery network. In *Proc. of 17th Int. Conf. on Computer Communications and Networks*, pages 1–7.

Pant, S. and Kumar, V. (2018). Bittrusty: A bitcoin incentivized peer-to-peer file sharing system. In *IEEE 3rd Int. Conf. on Computing, Communication and Security (ICCCS)*, pages 148–155.

Perez, D., Xu, J., and Livshits, B. (2020). Revisiting transactional statistics of high-scalability blockchains. In *Proc. of the ACM Internet Measurement Conference*, page 535–550, New York, NY, USA. Association for Computing Machinery.

qBittorrent (2021). qBittorrent (github). `github.com/qbittorrent/qBittorrent`.

Qt project (2020). Qt documentation. `https://doc.qt.io/`.

Rivest, R. L. and Shamir, A. (1997). Payword and micromint: Two simple micropayment schemes. In *Security Protocols*, pages 69–87, Berlin, Heidelberg. Springer.

Zhang, K., Antonopoulos, N., and Mahmood, Z. (2009). A review of incentive mechanism in peer-to-peer systems. In *1st Int. Conf. on Advances in P2P Systems*, pages 45–50.