Thing-to-Thing Electricity Micro Payments Using Blockchain Technology

Thomas Lundqvist, Andreas de Blanche, H. Robert H. Andersson
Department of Engineering Science
University West, Trollhättan, Sweden
{thomas.lundqvist, andreas.de-blanche, robert.andersson}@hv.se

***Abstract*—Thing-to-thing payments are a key enabler in the Internet of Things (IoT) era, to ubiquitously allow for devices to pay each other for services without any human interaction. Traditional credit card-based systems are not able to handle this new paradigm, however blockchain technology is a promising payment candidate in this context. The prominent example of blockchain technology is Bitcoin, with its decentralized structure and ease of account creation. This paper presents a proof-of-concept implementation of a smart cable that connects to a smart socket and without any human interaction pays for electricity. In this paper, we identify several obstacles for the widespread use of bitcoins in thing-to-thing payments. A critical problem is the high transaction fees in the Bitcoin network when doing micro transactions. To reduce this impact, we present a single-fee micro-payment protocol that aggregates multiple smaller payments incrementally into one larger transaction needing only one transaction fee. The proof-of concept shows that trustless, autonomous, and ubiquitous thing-to-thing micro-payments is no longer a future technology.**

***Index Terms*—IoT, Internet of things, Bitcoin, Digital payments, Crypto currency, Smart grid**

I. Introduction

An important enabler technology for the Internet of Things (IoT) era is to make “things” able to automatically and ubiquitously make payments to other “things” without any human interaction. This would, for example, open up the possibility for an IoT device to, on its own, make the decision to rent a fog or cloud server to get extra computational power when needed, to directly pay other devices for internet access, or for an item to pay for its own electricity when plugged into a wall socket – directly to the owner of the socket and with no human interaction.

It is today unclear how to enable this kind of thing-to-thing payments. The payments would typically be small and numerous, and should be autonomous. Current popular payment solutions are not well suited to handle these massive amounts of micro transactions due to limited capacity and high transaction costs. Furthermore, current credit-card based solutions face the problem that you have to share your credit card information with your device and that it then shares it with other devices when performing the payments.

According to a 2015 IBM whitepaper [7] it is not possible to handle payments in the IoT era using the traditional centralized payment approach. They also describe five vectors of disruption for IoT, where the second one is that it will create liquid, transparent marketplaces for real-time matching of supply and demand for physical goods and services. Other work also suggests the creation of an economy plane for the internet [11]. At the same time, electronic payments are becoming ubiquitous and there is a growing demand for cash-less and digital payments.

Digital currencies, like Bitcoin [3], which are based on a decentralized block chain ledger, are promising alternatives to traditional solutions for thing-to-thing payments. Previous research [1], [2], [8], and [9] suggests that blockchain technology could be used to handle thing-to-thing payments. Still, it is so far unclear how this can be done with current state of the art blockchain technology and what limitations and possible pitfalls there might be.

Up until today the original, and most prominent, blockchain implementation is bitcoin [10]. Bitcoin is both a currency and a decentralized payment infrastructure based on a peer-to-peer network. Although there is no central trust authority in the bitcoin network it handled transactions of, on average, 277 000 bitcoins per day during December 2016. On the 1st of January 2017 one Bitcoin can be exchanged for just over 1000 USD. A Bitcoin can be divided into one hundred million Satoshies. Hence, as of January 1st 2017 one Satoshi is worth 0.001 cent.

Bitcoin and blockchains have several promising properties that make them good candidate technologies for handling thing-to-thing payments. First, they are based on a decentralized peer-to-peer network for doing bookkeeping and relaying of transactions. The decentralized nature of bitcoins underlying blockchain allows it to support autonomous and numerous transactions. Another advantage is the very lightweight and cheap way to create new accounts – each device can easily have its own account; a new account can be created in seconds and internet access is not required. Thus, there is no central authority in control of the accounts and the accounts are not directly linked to any individual, only to the IoT device itself.

To examine the feasibility of using bitcoin in autonomous thing-to-thing payments, we implemented a proof-of-concept smart cable that connects to a smart wall socket. This cable can pay the socket for the electricity it provides to the device (thing) connected to the end of the cable. Thus, the electricity for anything from a phone to a car can be paid for in bitcoins by the smart cable.

In this paper, we present a proof-of-concept system and our resulting experiences in terms of limitations and possible pitfalls

978-1-5090-5873-0/17/$31.00 ©2017 IEEE

978-1-5090-5873-0/17/$31.00 ©2017 IEEE

when using bitcoin for doing IoT thing-to-thing payments. We find that bitcoin is a feasible payment solution for thing-to-thing payments. However, many low-value payment applications will suffer from the high transaction fee currently present in the bitcoin network. Therefore, a new single-fee micro-payment protocol is presented that can reduce the impact of the high fees.

In Section II we first give some background on Bitcoin before presenting our proof-of-concept system in sections III and IV. We describe the payment protocol and the improved version in Sections V and VI. Section VII presents results and evaluation and Section VIII highlight obstacles and limitations for widespread use of thing-to-thing payments using bitcoins. Section IX discuss related work before concluding the paper in Section X.

II. Bitcoin

Bitcoin is a digital crypto currency that uses a decentralized ledger, the blockchain database, to keep track of all transactions ever performed and all funds held by each account. It was introduced in 2008 by the pseudonym Satoshi Nakamoto [10], which solved the double-spend problem by inventing a proof-of-work algorithm. A good recent overview of Bitcoin and blockchain technology is given by Zohar [12] and the bitcoin project web site [3].

To keep track of all transactions, the bitcoin network uses the blockchain ledger which is replicated among all peer nodes in the bitcoin network. The blockchain is a list of blocks, each one containing multiple transactions. Each block has a pointer to the previous block and the ordering and content of blocks are protected by hash signatures. Bitcoin mining nodes construct new blocks from arriving transactions. This construction is by purpose made difficult and requires considerable mining calculations, the proof-of-work. The effort spent makes it equally difficult to change already included blocks in the blockchain, especially since changing a block in the middle of the chain would require the recreation of all the following blocks. Thus, the blockchain ledger is well protected from modifications and can be regarded as a permanent record of transactions. As an incentive to spend effort mining, the miners are rewarded by newly created bitcoins when a block is created. They also get all transaction fees from the transactions included in the new block.

A single transaction can transfer funds from multiple bitcoin accounts (inputs) to multiple accounts (outputs). The set of all unspent transaction outputs in the blockchain defines the current value held for each bitcoin account. To safeguard the accounts, public-key encryption is used. A bitcoin account consists of one public and one private key as well as an identifier, the bitcoin address, derived from the public key. When a new transaction spends the funds of a certain bitcoin address, a digital signature is needed. This signature must be made using the owner's private key which proves that the owner of the bitcoin address has agreed to spend the outputs.

III. The Smart Cable and Socket System

The aim of our proof-of-concept was to allow a "thing" to pay another "thing" for the electricity it consumes. The scenario we used in this study is based on a smart cable which when plugged into a smart socket, will pay the socket for the electricity it transmits. In the other end of the cable, anything can be connected. In our evaluation, we used a regular toaster to validate the functionality, but it could in principle be a car, a phone or anything else that is powered by electricity. An illustration of the system can be seen in Fig 1.

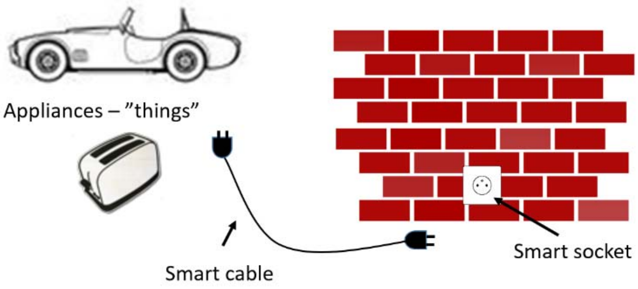


Fig. 1. The proof-of-concept smart cable and socket.

The system works in the following way:

1. The socket is normally in **standby mode**. This means that the socket is providing a limited amount of power, just enough to power the smart cable for a limited time.
2. When the smart cable is inserted, the cable turns on and starts **communicating** with the socket.
3. The cable then pays for a certain amount of electric energy using bitcoin and the socket goes into **payed mode** and starts to keep track of power use.
   - If the cable fails to pay, the socket goes into a **lockout mode**, turning off power for a certain period, for example, 1 min, before returning to **standby mode**.
4. The socket notifies the cable when the energy quota is about to run out. The cable can then choose to make a new payment to keep the electricity flowing.
5. When the cable is detached, the socket returns to **standby mode**.

For this to work, the cable needs to be set up with a bitcoin account containing bitcoins (a bitcoin address and the public/private key pair). The user should not need to be aware of the payments and only plug the cable and the toaster in as usual. In this proof-of concept the intelligence is implemented in the cable, but it could just as well have been included in the appliance. Furthermore, if the cable runs out of funds it will not be able to make any new payments until the user has transferred more bitcoins to the cable's account.

IV. Implementation

In our prototype thing-to-thing payment system, we focused mainly on the verification of basic functionality. Thus, we chose to build the smart cable and the smart socket proof-of-concept using inexpensive commodity parts and using the bitcoin blockchain technology. Figure 2 shows a block diagram of our prototype system.

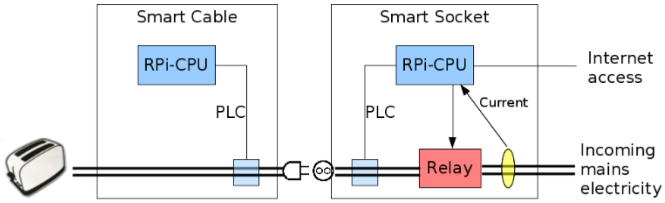


Fig. 2. The proof-of-concept system. The smart socket can control electricity supply via a relay and measure current to calculate transferred energy.

For the proof-of-concept system we chose to use Raspberry Pi single-board computers as embedded controllers both in the cable and in the socket. The smart socket controls the electricity supply via relays and uses a current sensor to calculate transferred energy. Furthermore, the smart socket needs internet access in order to forward the payment transactions to the bitcoin network. In our proof-of-concept implementation the socket is connected to the internet using a Fast Ethernet cable. However, it could just as well have used Wi-Fi, Power-Line Communication (PLC), or any other available technology.

For communication between the socket and the cable, commodity PLC devices are employed to enable TCP/IP communication. By utilizing PLC communication there is a one-hop, dedicated, connection between the socket and cable without the need for additional cables or wireless communication. This connection is used by the payment protocol for negotiating the power payment and handling the bitcoin transaction process.

V. The Basic Payment Protocol

For doing payments and communicating between the cable and socket, we first created a basic protocol where each request for energy was paid for by using a separate bitcoin transaction. This proved to be quite expensive in terms of bitcoin transaction fees. We later added support for doing incremental micro-payments in order to aggregate a larger amount of transactions for a one-time fee. The protocol is written in Python and uses the built-in Simple XML-RPC libraries. Our basic protocol is illustrated in Fig. 3.

A description of the calls made by the cable to the socket can be seen in Table I. The cable initiates a payment using ‘power_request’ sending along its bitcoin address and getting in return a prepared transaction skeleton ready to sign using its private key. In the basic proof-of-concept implementation the cable always pays a fixed amount of 10000 satoshi as the value of the transaction (approx. $0.1). The ‘pay_transaction’ call then sends the signed data to be broadcast on the bitcoin network. If the transaction broadcasts successfully, the socket decides that it has been payed and enters the payed mode. The cable regularly uses the ‘get_status’ call to find out if it needs to pay or not (state is standby or payed, respectively) and to see how much more remaining energy it can use before a new payment is needed. We use a third-party API[1] to simplify the bitcoin transaction handling instead of contacting nodes in the bitcoin network directly.

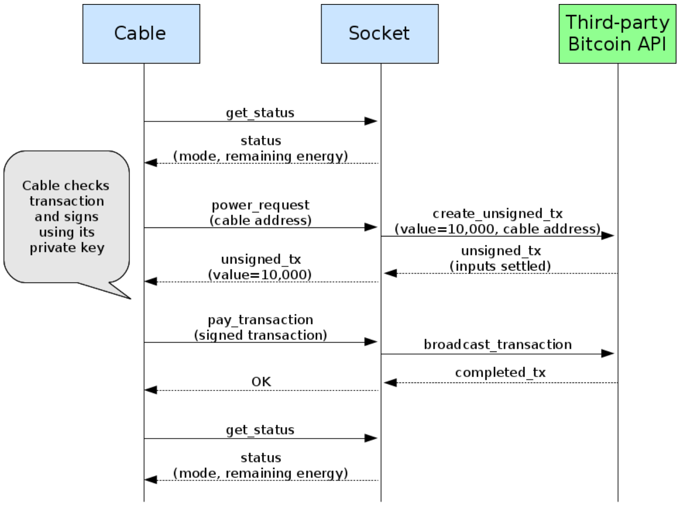

Cable
Socket
Third-party Bitcoin API
get_status
status (mode, remaining energy)
Cable checks transaction and signs using its private key
power_request (cable address)
create_unsigned_tx (value=10,000, cable address)
unsigned_tx (value=10,000)
unsigned_tx (inputs settled)
pay_transaction (signed transaction)
broadcast_transaction
completed_tx
OK
get_status
status (mode, remaining energy)

Fig. 3. The basic payment and management protocol. A third-party bitcoin service is used by the socket to simplify the creation and submission of transactions. The payment transaction is a two-step process. First, an unsigned transaction is prepared (create_unsigned_tx), then the cable must sign the transaction to verify payment and the ownership of the bitcoin funds. Finally, the signed transaction is sent to the bitcoin network (broadcast_transaction). The value 10,000 is an example.

TABLE I. DESCRIPTION OF CALLS FROM THE CABLE TO THE SOCKET

Call	Description
get_status	Returns status: current state (standby or payed), used and remaining energy
power_request	Initiate a request to pay for more energy. The cable’s bitcoin address is sent to the socket. The socket then calls the third-party API (create_unsigned_tx) to create a new unsigned transaction using the cable address as input and the socket’s bitcoin address as output. The third-party API manages the matching of the input address to the existing unspent outputs in the bitcoin blockchain and prepares the binary data that is to be signed by the cable.
pay_transaction	The cable first signs the unsigned transaction using its private (secret) key. Then, this call delivers the signed transaction to the socket. The socket forwards this signed transaction to the third-party API for broadcasting in the bitcoin network. If the transaction is deemed successful, the socket enters the payed mode.

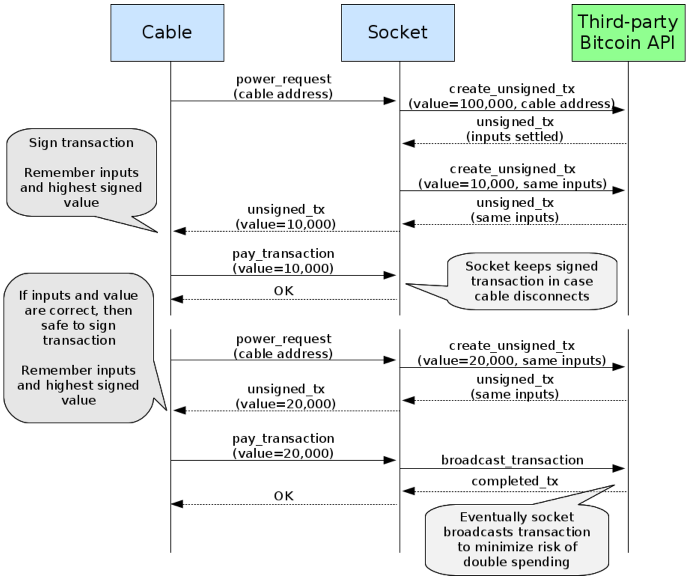
VI. The Single-Fee Micro-Payment Protocol

One drawback with the basic micro-payment protocol introduced in the previous section is the relatively large transaction fee. The bitcoin transaction fee is a payment to the miners, currently (Jan 2017) the typical fee ranges from 10,000 to 25,000 Satoshis (approx. $0.1 to $0.25). If the electricity payments are of similar sizes, the added fee becomes 100% – quite excessive.

To reduce the impact of the mining fee, we devised an improved protocol supporting micro transactions. The protocol uses the fact that multiple transactions using the exact same inputs can be created and signed by the cable as long as only (the last) one is broadcast to the bitcoin network. Otherwise it would

[1] We used blockcypher.com as the third-party API, but similar services by other companies would work equally well.

be an attempt of double-spending the inputs which is not possible, since only one transaction would succeed. The idea is that the cable can sign progressively higher valued transactions that the socket holds on to until it eventually chooses to broadcast the highest one based on a tradeoff between the risk of not getting payed and paying the transaction fee. A larger aggregated sum means lower relative fee. But the longer the socket holds on to the transactions the higher the risk of losing the payment if the cable, or someone in possession of its keys, spends the inputs before the socket does. An overview of the improved micro-transaction protocol can be seen in Fig 4 and Table II contains descriptions of the changed behaviors of calls and actions for the improved protocol.


Cable
Socket
Third-party Bitcoin API
power_request (cable address)
create_unsigned_tx (value=100,000, cable address)
unsigned_tx (inputs settled)
Sign transaction

Remember inputs and highest signed value
create_unsigned_tx (value=10,000, same inputs)
unsigned_tx (same inputs)
unsigned_tx (value=10,000)
pay_transaction (value=10,000)
Socket keeps signed transaction in case cable disconnects
OK
If inputs and value are correct, then safe to sign transaction

Remember inputs and highest signed value
power_request (cable address)
create_unsigned_tx (value=20,000, same inputs)
unsigned_tx (same inputs)
unsigned_tx (value=20,000)
pay_transaction (value=20,000)
broadcast_transaction
completed_tx
OK
Eventually socket broadcasts transaction to minimize risk of double spending

Fig. 4. Improved protocol for micro-payments. In this scenario, the socket chooses to broadcast already at value=20,000 Satoshis. It could continue to ask for higher and higher values and finally broadcast when the value reaches, for example, 100,000. The first 'create_unsigned_tx' sets up 100,000 as the maximum value. The chosen inputs can then be used for all lower-valued incremental transactions up to the maximum 100,000. This also makes sure that the cable has enough funds to eventually pay the maximum value. The values 10,000 and 100,000 are just examples.

TABLE II. BEHAVIOR OF THE MICRO-PAYMENT PROTOCOL

Call / Action	Description
power_request	The first time, the socket calls the third-party API using 'create_unsigned_tx' twice. First, to establish suitable transaction inputs using the maximum value of 100,000 Satoshis. Then, to create the first new unsigned transaction using the exact same inputs and the value 10,000 for the first payment. For subsequent calls, a new unsigned transaction with an incrementally higher value is created. This works as long as the original inputs have not been spent.
Cable signs	When the value is 10,000, the cable can sign the unsigned transaction using its private (secret) key as before. If a higher value is to be signed, the cable must first check if the inputs match the previously used inputs, then it knows that the previous transaction has been discarded. The value should then also be one payment amount more than the previously signed transaction. The cable must remember the inputs used and the highest signed value.
pay_transaction	If the value is less than the maximum, the socket stores the transaction for later use. Otherwise, it broadcast the transaction. If the cable disconnects at any time, it should broadcast the latest stored transaction to get the payment.

VII. Results and Evaluation

Figure 5 shows our completed proof-of-concept system. The small display on the smart socket system shows the bitcoin symbol, indicating that the system is currently in payed mode. The total elapsed time from plug in of the smart cable to the first completed payment was 27 seconds. The boot time for the Raspberry Pi was approximately 24 seconds and then 2-3 seconds for negotiating and completing the payment. In our test scenario, the cable performed multiple payments[2] while toasting several batches of toast. The 1300 W toaster used approximately 40 Wh of energy per batch of toast slices.

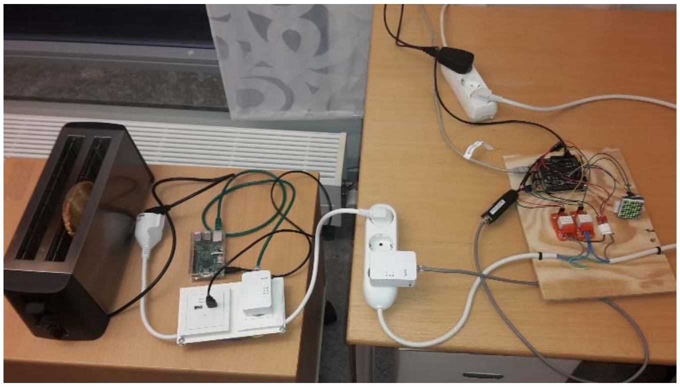


Fig. 5. The prototype implementation. The smart cable to the left connected plugged into the smart socket system to the right.

Based on the proof-of-concept implementation, we can confirm that using bitcoin for thing-to-thing payments is a feasible solution and that the concept works. However, a real implementation of a smart cable and socket would need to be more carefully designed and implemented. The remainder of this chapter goes through some technical issues and trust considerations as well as suggestions on how to improve them.

A. Technical issues

Technically, the prototype system is too large to be practical and user friendly. In a more realistic system, smaller custom-made components would be needed.

The Raspberry Pi system in the cable is also too slow. It uses an ordinary Linux operating system and the time to initiate communication is dominated by the time to boot the system,

[2] The bitcoin addresses used: 1AQuzcmm9XimkVeGFfK7s1hHCYy7VffW7U (cable) and 1LkE18xg6ZMA82rBqeycc3K26KntdWmdDF (socket).

24 seconds. However, using a smaller embedded system, which can boot in a fraction of a second, would solve both the size and speed limitations of the proof-of-concept system. There should be no problem utilizing an embedded controller based on, for example, an ARM, AVR, or MIPS architecture, if supporting libraries for communication and the bitcoin crypto operations can be accommodated.

One problem in the current hardware design is that if a high-powered appliance is connected, the socket might enter lockout mode and turn off the power before a payment has been settled. This, depends in part on how fast the payment process can be completed. The problem can be solved by adding an extra relay in the smart cable. An additional relay would let the cable turn on the electricity when the payment has completed.

In our design, we rely on a third-party API for bitcoin transactions. This dependency might be problematic, for example, if the third-party API shuts down. It is possible to directly interact and participate in the bitcoin peer-to-peer network but this is typically slightly more complex and require more storage and computational power. However, there is an intermediate alternative, where all third-part API functionality except storage and indexing (to find spendable outputs) of the blockchain is moved to the smart socket.

B. Trust considerations

Trust is of paramount importance when dealing with payments. In the basic protocol, we chose to let payments be in advance. Hence, the socket can fool the cable by accepting payment but still not go into payed mode. The reverse problem could occur if power was delivered before payment – the cable can receive energy and then skip payment. We believe these problems can be minimized by letting transactions be small and if fraudulent behavior is detected the cable can refuse to pay, and the socket can refuse to provide. In practice, it might be necessary for the cable to also include a current sensor to be able to verify the amount of delivered electricity. Furthermore, one of the reasons for using advance payments is that the user should be able to unplug the appliance (cable) at any time without negatively affecting the socket.

Using a digital currency means that there is also the risk of double spending the coins or that one party tries to trick the other party to accept bogus transactions. Hence, the cable needs to verify that the transactions it is signing has not been manipulated by the socket. For example, since a third-party API assists in creating the transactions, the socket could prepare a very high value transaction but rewrite some information to make the value appear to be low. To verify the transaction, the cable must confirm that the binary data represents the correct amount to pay before signing it. An alternative solution would be to let the socket forward the raw information to the cable and let the cable create the binary data and sign it.

On the other hand, the socket also needs to make sure that the signed transaction delivered from the cable is sound. It needs to verify the value, the signature, and that the transaction output address equals the socket address. For example, the cable can try to fool the socket by creating an entirely new transaction with an output going back to the cable address or just change the value so the total amount does not add up. Such an erroneous transaction will not be mined into the blockchain. The socket also needs to make sure the transaction is not an old one trying to double spend previously spent outputs. For this, online access to the bitcoin network is a requirement; Offline transactions are possible, but not really feasible from a security point of view.

A final trust issue concerns how transactions are handled in the bitcoin network. Transactions are not valid directly but needs to be confirmed before they can be trusted. This can take several minutes or even hours. In our implementation, the socket assumes that a valid-looking transaction accepted by the third-party API will eventually be confirmed. For long-lasting sessions, it could be wise to include a confirmation check by the socket to verify that the payments have been included in the blockchain.

VIII. Obstacles for Widespread Adoption

Although our proof-of-concept implementation of ubiquitous thing-to-thing payments works as expected we have identified several areas that has to be taken into consideration, and should be improved, before the technology is ready for widespread adoption.

A. Excessive Transaction Fees and Scalability

One obstacle is the relatively high transaction fee for micro-payments in the bitcoin network. This fee is independent of the value of a transaction, which means that transferring smaller values are relatively more expensive. In January 2017, transferring $1 would result in an added fee of at least 10%.

Our single-fee improved micro-payment protocol, presented in the previous section, mitigates somewhat the cost of transactions by aggregating multiple smaller transactions into a larger one. However, if the cable disconnects, the socket will be forced to carry out the transaction regardless of the aggregated amount. Thus, for short-lived or low-value sessions, the fee can still be prohibitive. See Section IX for related work on micro-payments.

The fee structure is currently hardcoded into the bitcoin core reference implementation and is also related to the scalability issues bitcoin faces. The bitcoin network has a limited transaction capacity since the maximum size of a bitcoin block is fixed to one megabyte and one new block is added to the block chain approximately every 10 min, resulting in a typical transaction rate of 3 transactions per second [6]. If the demand for transactions is more than can be handled, the miners (creators of blocks) will typically prioritize the transactions that has the highest transaction fee per byte ratio–thus maximizing their own revenues when mining blocks. This means that solving the scalability issues, i.e. allowing more transactions per second, will probably lower the fees. Scalability is currently a widely-discussed issue in the bitcoin community with several proposed solutions [6].

B. Further Obstacles

Other identified obstacles are related to standardization and usability. For thing-to-thing payments to become ubiquitous, the payment protocol needs to be standardized. It would be a big usability issue if the cable you bought was incompatible with the socket you want to use.

Furthermore, the handling of accounts used by your things should ideally be standardized and easy. If a user owns multiple smart things, each thing should typically have its own account from a security perspective. However, if we accidently break a device we do not want to lose the money protected by the private key inside the broken device. Therefore, we would need to store all account keys in a separate system as well, maybe similar to the already existing bitcoin wallet services. Furthermore, devices could be delivered with pre-configured bitcoin address keys, which could be transferred to your wallet either electronically or manually using, for example, QR codes hidden under security labels, like the ones on lottery scratch cards, or they could allow you to manually enter address key(s) generated by your wallet.

IX. Related Work on Micro-Payments

Our single-fee protocol for micro-payments is based on the same principle as the earlier suggested payment channel [4][5] micro-payment protocols. Like our protocol, the idea of the payment channel protocols is also to prepare and sign incrementally larger valued transactions until the receiver broadcasts the transaction to receive all aggregated payments. One difference, however, is that before any payment transactions are created the sender transfers an amount of bitcoin using a multi-signature transaction to safely store and reserve the maximum value in the bitcoin blockchain. The output of the multi-signature transaction can only be spent if it is signed by both the sender and the receiver. Hence, the incremental transactions containing the micro-payments must be signed by both the sender and the receiver. Since the reserved amount is indefinitely locked in a multi-signature transaction a second refund transaction is created, as a safeguard, before the first payment transaction is signed. This refund transaction, dated sometime in the future, is signed by both parties and basically refunds the locked transaction back to the sender. The refund transaction is useful if the connection between the sender and receiver is lost or the receiver never broadcasts the payment.

In comparison with our protocol, the payment channel requires two transactions and two fees while our protocol only requires one transaction and one fee. Also, if no payments are made the payment channel still requires two transactions and locks the reserved amount for some time. While being more expensive, the upside is that it is better protected from fraud. Nevertheless, we believe our less complex, single-fee protocol works well for smaller amount of payments where there is less risk of fraud.

X. Conclusions

An important enabler technology for the Internet of Things (IoT) era is to allow “things” to automatically and ubiquitously make payments to other “things” without any human intervention. To this end we have successfully implemented a proof-of-concept smart cable that pays a smart socket for delivering electricity.

Although allowing IoT devices, or any-thing, to pay for their own power the technology is more general and could be used by an electric car to refuel itself. It could allow for IoT devices to rent computational capacity or allow washing machines to order detergent. Hence, our proof-of-concept implementation shows that trustless, autonomous, and ubiquitous thing-to thing micro-payments is no longer a future technology.

The payments are done using bitcoins, which is an electronic blockchain-based currency, where transactions are validated by the sender when they have been signed by the private key of the paying address (account). Hence, the cable has the means to pay for itself using its private key to sign transactions which transfers bitcoins to the receiver’s bitcoin address. In this case the sender is the smart cable and the receiver is the smart socket. Using bitcoin technology has the potential of being cheaper, more anonymous, and privacy protecting compared to traditional credit card-based payments.

A bit surprising, one serious limitation we found was the high bitcoin transaction fee for micro-payments. To reduce this cost, we implemented a single-fee micro-payment protocol which aggregates multiple smaller transaction into one larger transaction, thus reducing cost. Although our thing-to-thing implementation is working we have identified and addressed several problematic areas that have to be taken into consideration, and should be improved. Still, the bitcoin technology is a promising payment solution given that scalability and transaction cost problems can be solved.

References

[1] M. T. Alam, H. Li, and A. Patidar, "Bitcoin for smart trading in smart grid," The 21st IEEE International Workshop on Local and Metropolitan Area Networks, Beijing, 2015, pp. 1-2.

[2] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer and S. Welten, "Have a snack, pay with Bitcoins," IEEE P2P 2013 Proceedings, Trento, 2013, pp. 1-5.

[3] Bitcoin project, “Bitcoin – open source P2P money,” 2017. [Online]. Available: http://bitcoin.org

[4] Bitcoin wiki, “Contract example 7: Rapidly adjusted (micro)payments to a pre-determined party” [Online] Available: https://en.bitcoin.it/wiki/Contract#Example_7:_Rapidly-adjusted_.28micro.29payments_to_a_pre-determined_party

[5] Bitcoin wiki, “Payment channels,” [Online] Available: https://en.bitcoin.it/wiki/Payment_channels

[6] Bitcoin wiki, “Scalability FAQ”, [Online] Available: https://en.bitcoin.it/wiki/Scalability_FAQ

[7] IBM, “Device democracy- Saving the future of the Internet of Things,” IBM Institute for Business Value, Whitepaper, 2015.

[8] IBM, “Empowering the edge - practical insights on a decentralized Internet of Things,”, IBM Institute for Business Value, Whitepaper, 2015

[9] M. Mihaylov, S. Jurado, N. Avellana, K. Van Moffaert, I. M. de Abril, and A. Nowé, "NRGcoin: Virtual currency for trading of renewable energy in smart grids," 11th International Conference on the European Energy Market (EEM14), Krakow, 2014, pp. 1-6.

[10] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[11] T. Wolf, et al., “ChoiceNet: toward an economy plane for the Internet”. ACM SIGCOMM Computer Communication Review, 44(3):58–65, July 2014.

[12] A. Zohar, “Bitcoin: under the hood,” Commun. ACM 58, 9, 104-113, August 2015