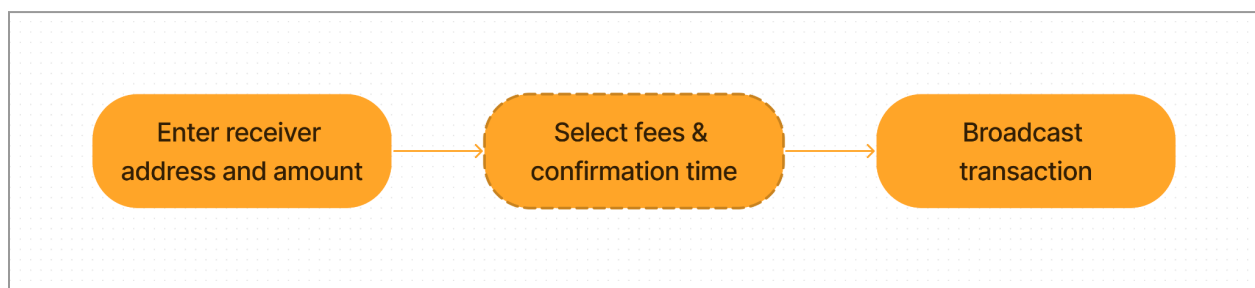


Thought process

Ideally, a payjoin user flow would be indistinguishable from one for a regular bitcoin sending flow. The primary reason a regular sender flow has to change is that the tx-size necessarily changes during the payjoin process.

Additionally, the both fee-amount and fee-rates may change during the course of the payjoin iterations. For the purpose of creating the sender flow, we will assume that the receiver only contributes UTXOs but does not contribute to the fees since this allows an automated, seamless process.

Let's explore these reasons:



A simplified bitcoin transaction flow

While it is abstracted away into things a user actually cares about, step 2 actually entails selecting a fee-rate based on a predetermined tx-size (in step 1). Since $\text{fee-amount} = \text{fee-rate} \times \text{tx-size}$, selecting a fee-amount involves assuming/knowing a tx-size and calculating the resulting fee-rate, which is used to estimate confirmation time.

For payjoins this information will be inaccurate at first, since tx-size will increase once receiver contributes their UTXOs and adds signature. This may result in a longer confirmation time than expected. BIP-78 does not allow the sender to defer/bump fees until after the receiver contributions are made, since initial PSBT shared with the receiver must be [broadcastable](#). There has to be a mechanism for the sender to estimate/control the confirmation time as well as the fees they pay during the transaction in advance. If the sender tried to fee-bump to maintain desired confirmation time (thus changing any inputs/outputs) AFTER the receiver has

contributed UTXOs and signed, the transaction needs the receiver signature again for the transaction to be broadcastable.

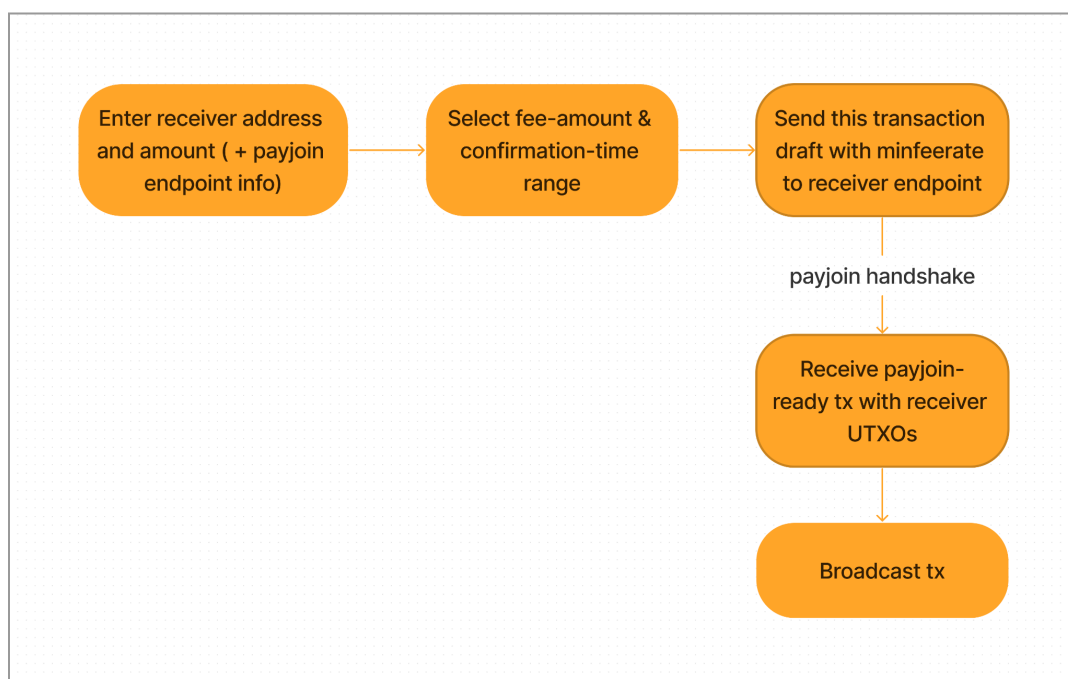
In short, the sender flow outlined here asks the user to choose a fee-range instead of a fee-amount (or fee-rate) while keeping the rest of the user flow almost exactly the same. We use it to set 3 optional parameters specified in BIP-78, which can be used to construct a simple but effective payjoin implementation.

minfeerate: The fee-rate chosen by the user can be set as the minfeerate parameter. This is also used to calculate the lower bound of the fee-range.

themaxadditionalfeecontribution: This parameter is calculated as described in BIP-78 and used to calculate the higher bound of the fee-range.

additionalfeeoutputindex: This is used to designate the 'fee-output' that the receiver will deduct the fees being paid over and above the minimum.

With this, a simplified PayJoin transaction flow would look like this:



Conversation archive

Note: This is the previous version of the content retained for archive and the comments it contains.

In short, the sender flow outlined here asks the user to choose a fee-range instead of a fee-amount (or fee-rate) while keeping the rest of the user flow almost exactly the same. We use it to set 3 optional parameters specified in BIP-78, which can be used to construct a simple but effective payjoin implementation.

minfeerate: The fee rate of the higher end of this fee-range is used to construct the PSBT, while the lower end is used to set the parameter called *minfeerate* parameter.

themaxadditionalfeecontribution: The delta of this fee range is used to calculate the *themaxadditionalfeecontribution* parameter, varying values of which allows retention of the fast, medium, slow confirmation time UI.

additionalfeeoutputindex: This delta is also used to select the UTXOs, which needs to be set as the *additionalfeeoutputindex* parameter as needed.