Mempool as a Battleground:

RBF Pinning, package relay, v3, ephemeral anchors

glozow

Today

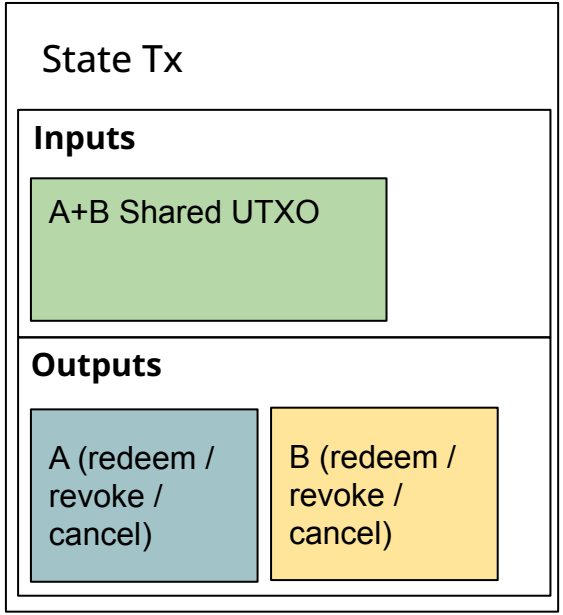
- The Problem
- Current Options
- Solution Part 1: Package Relay
- Solution Part 2: v3
- Solution Part 3: Ephemeral Anchors

The Problem

L2 == awesome

Sign now, broadcast later.

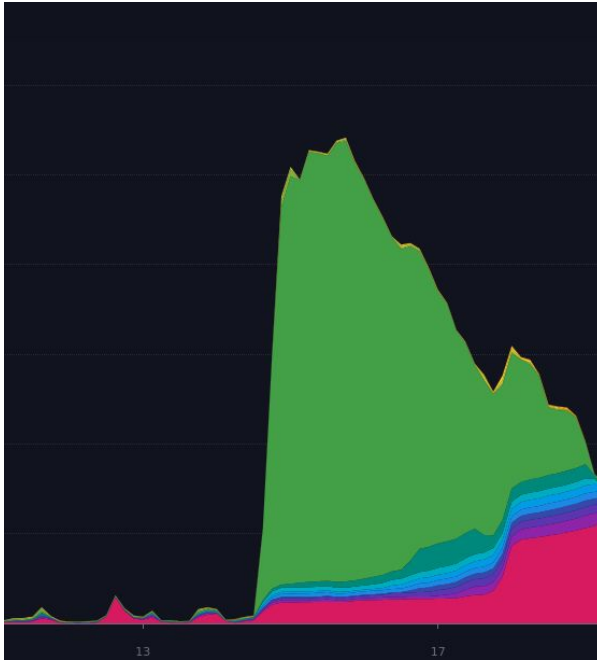
- do more stuff, put less on-chain
- privacy™, scalability™



L2 == awesome

Sign now, broadcast later.

- do more stuff, put less on-chain
- privacy™, scalability™



The problem

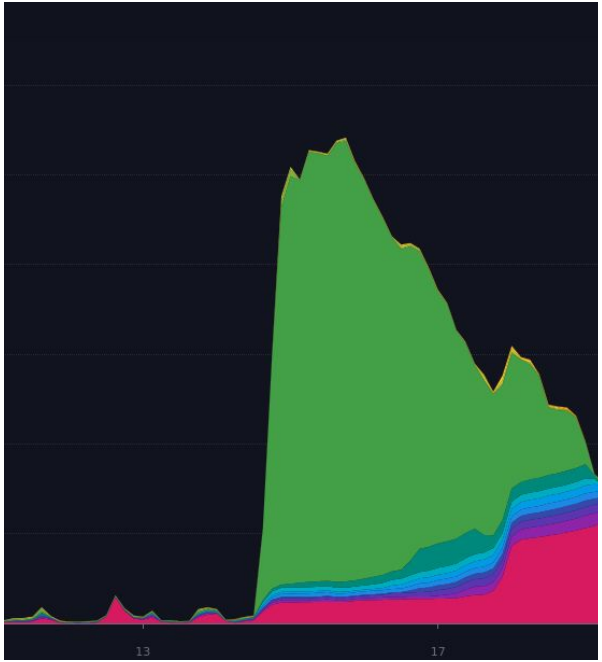
Typically, when you sign, you decide fees.

- A lot can change between sign and broadcast.
- This tx is shared with someone untrusted.

Current Options

Predict the Fee Using Your Magic Crystal Ball

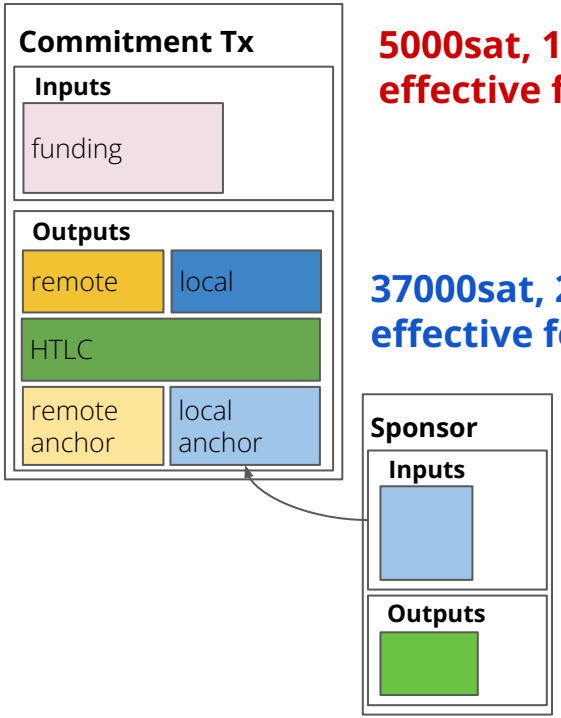
both likely



Related methods:
- Overshoot the feerate
- Sign multiple transactions at different feerates

(Bastien Teinturier: https://lists.linuxfoundation.org/pipermail/lightning-dev/2022-October/003729.html)

Attach a fee-bumping child (CPFP)

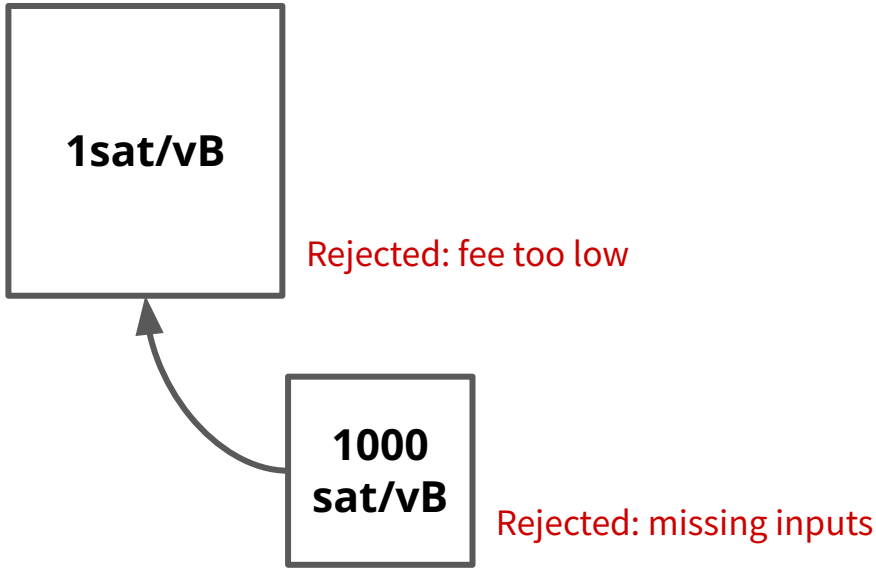


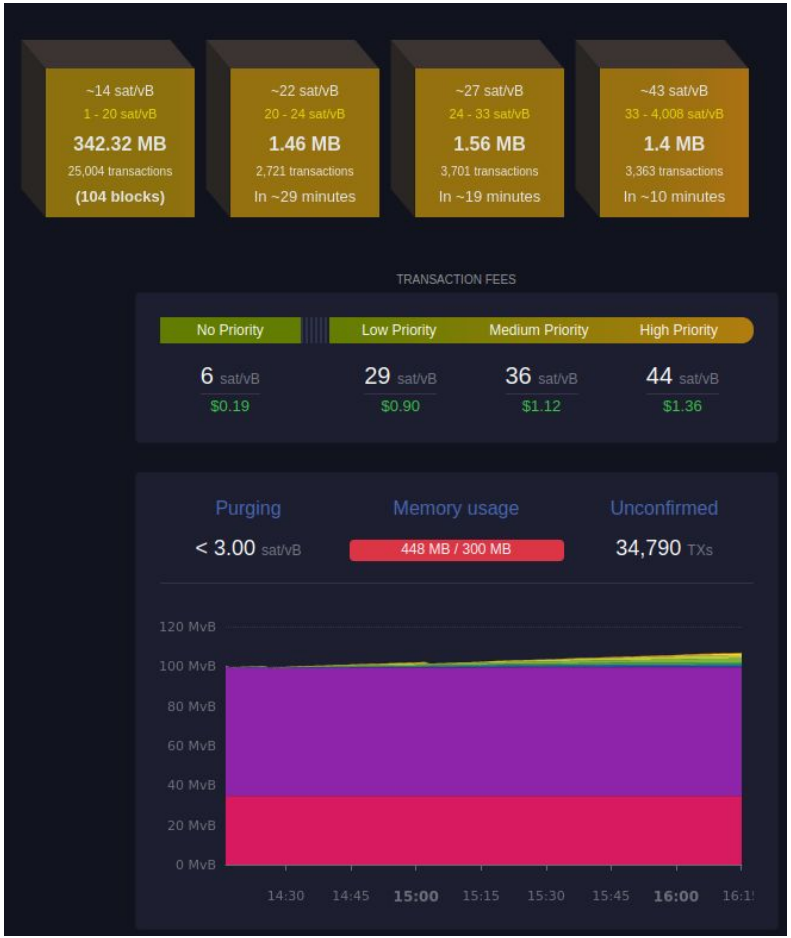
5000sat, 1000vB (5sat/vB)
effective feerate: 35sat/vB

37000sat, 200vB (185sat/vB)
effective feerate: 35sat/vB

Attach a fee-bumping child (CPFP)

Caveat: parent must meet mempool min feerate



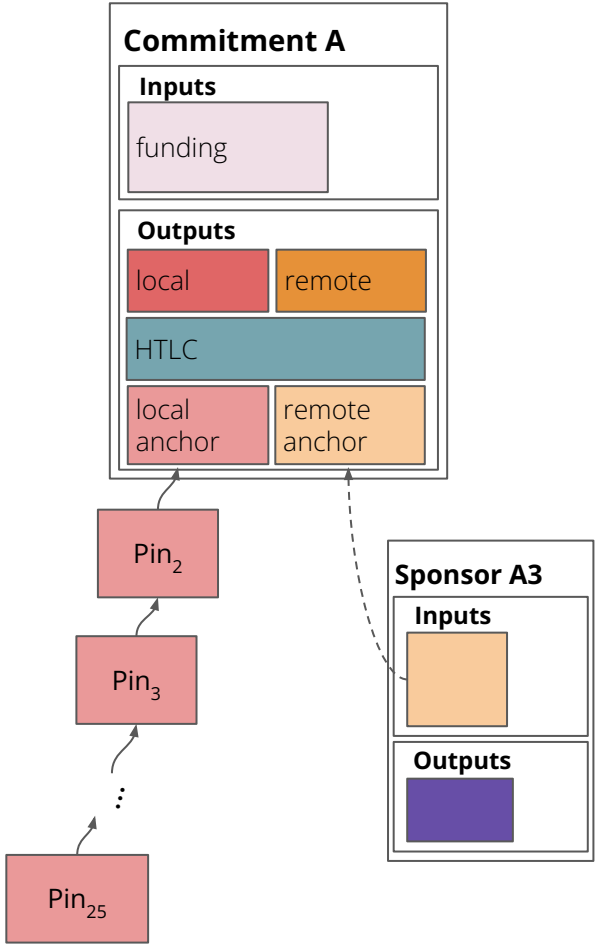


mempool.space/ 2023-03-08

Attach a fee-bumping child (CPFP)

Ugliness of anchor outputs:

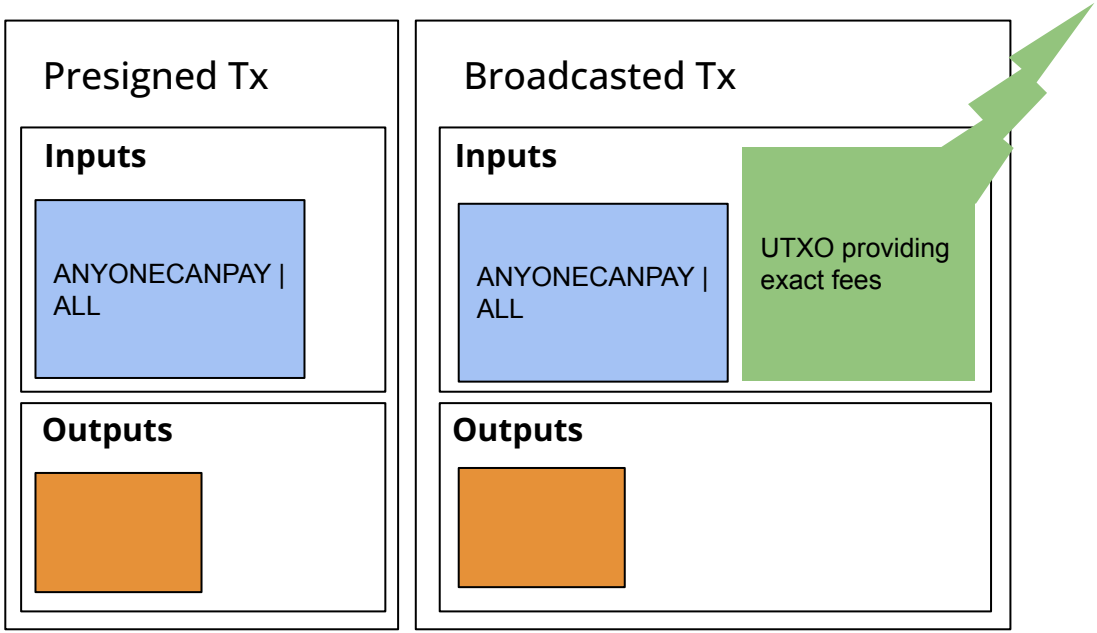
- Needs one for each participant
- Various hacks needed to avoid pinning
  - Other outputs can't be spent (CSV 1)
  - Needs CPFP carve out (2-party only)
- Shaved off from channel balance (cannot be dust)
  - blocker for eltoo
  - Creates low-value UTXOs (unless cleaned up)



ANYONECANPAY to increase inputs

Sign transaction with SIGHASH_ANYONECANPAY

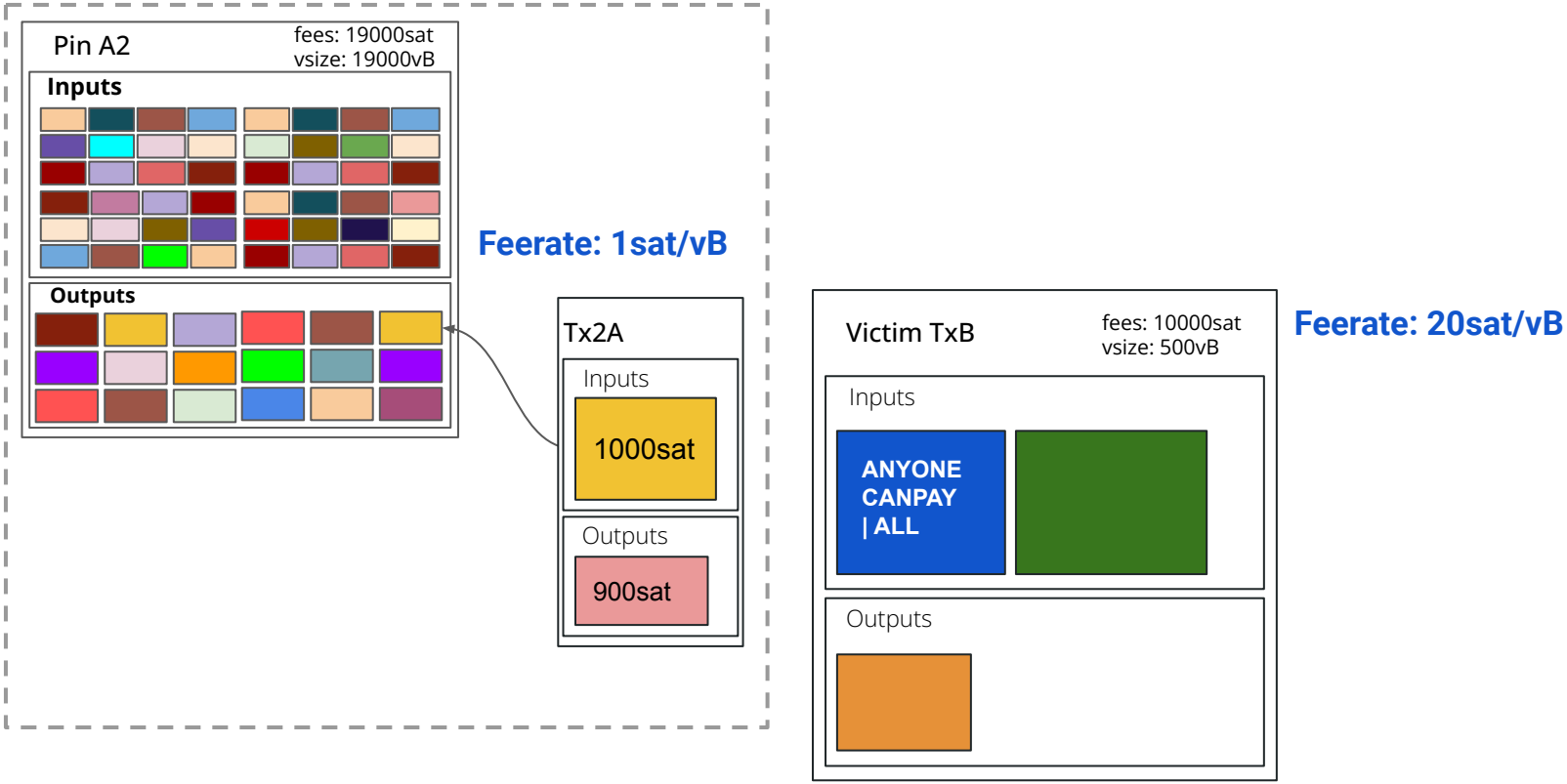
Adjust fees by adding inputs



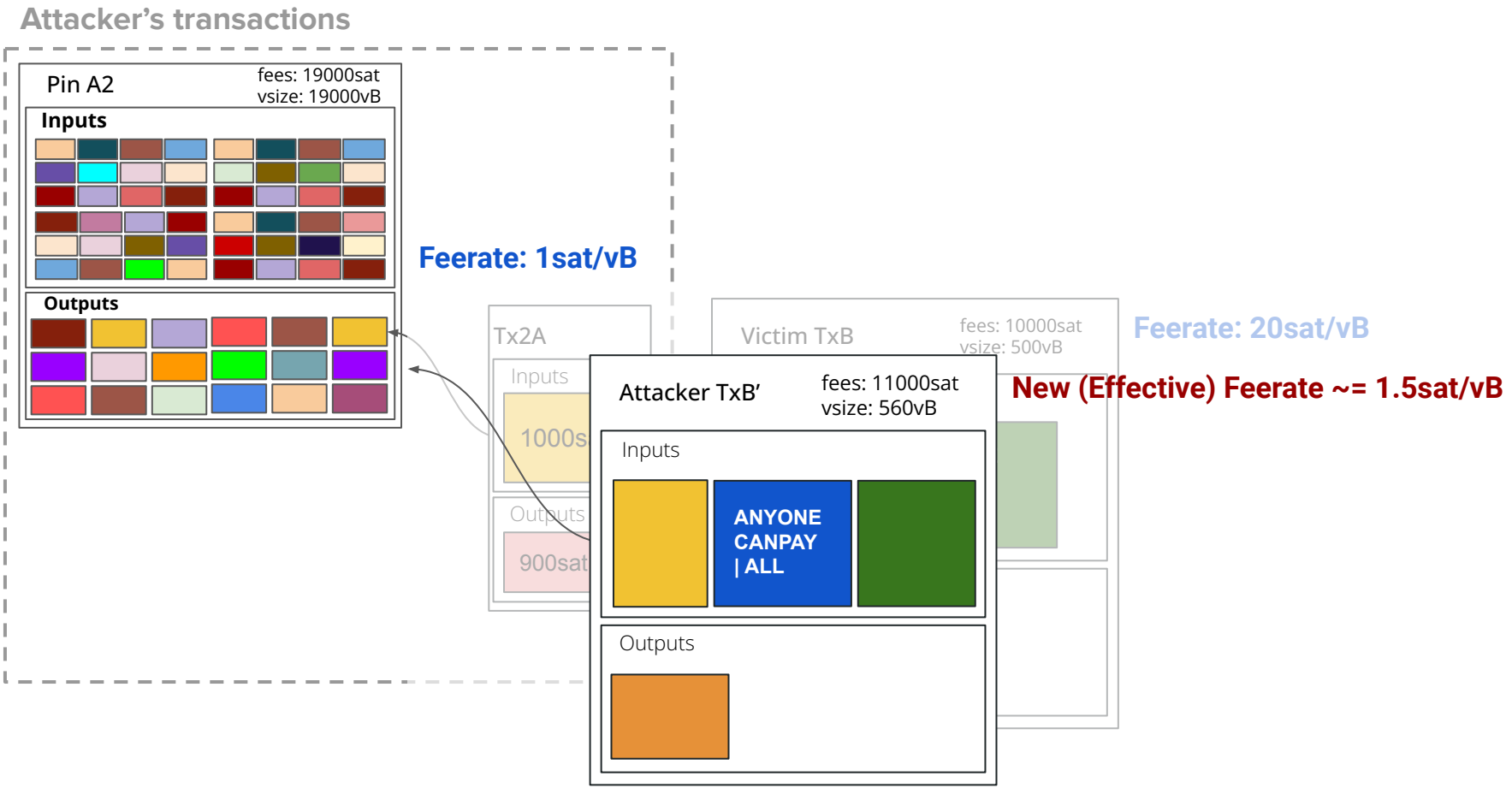
(Revault: https://github.com/revault/practical-revault/blob/master/transactions.md#cancel_tx)

ANYONECANPAY -> anyone can RBF

Attacker's transactions



ANYONECANPAY -> anyone can RBF




TO GET INTO THE BLOCK
ANYONECANPAY FOR ME
AND FOR MY NEW ANCESTOR

Pinning Attack: a type of censorship in which attacker takes advantage of mempool policy limitations to prevent a tx from getting mined or entering a mempool

Pinning Attack: a type of censorship in which attacker takes advantage of mempool policy limitations to prevent a tx from getting mined or entering a mempool

attacker isn't paying fair price

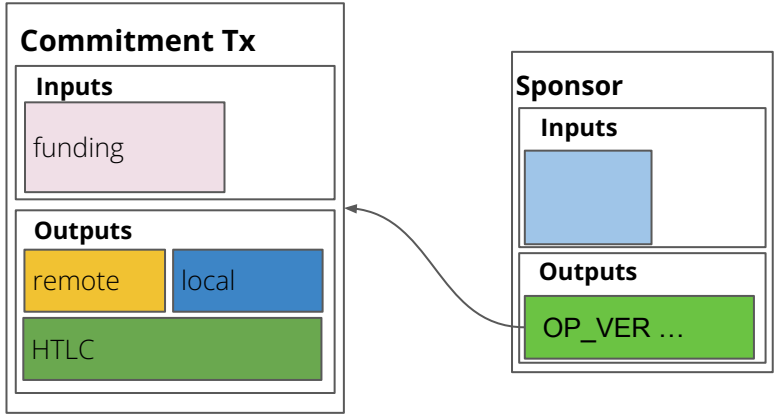
mempool should provide a **fair (fee-based) market for block space**

“Can we Soft Fork it out?”

Proposal

Similar to CPFP: Transaction Sponsors soft fork

- no anchor outputs
- “anyone can bump”
- many similar limitations to CPFP
  - everything still needs to be CSV 1
  - package limit pinning
    (sponsor-sponsee ~= parent-child)
- needs soft fork

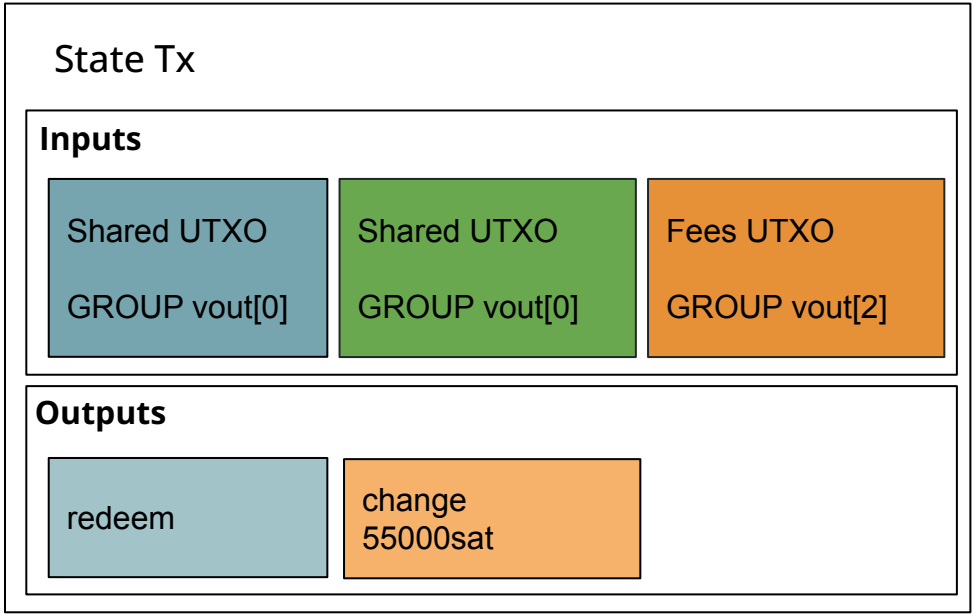


(Jeremy Rubin: https://gist.github.com/JeremyRubin/92a9fc4c6531817f66c2934282e71fdf)

Proposed

SIGHASH_GROUP / Signature bundles soft fork

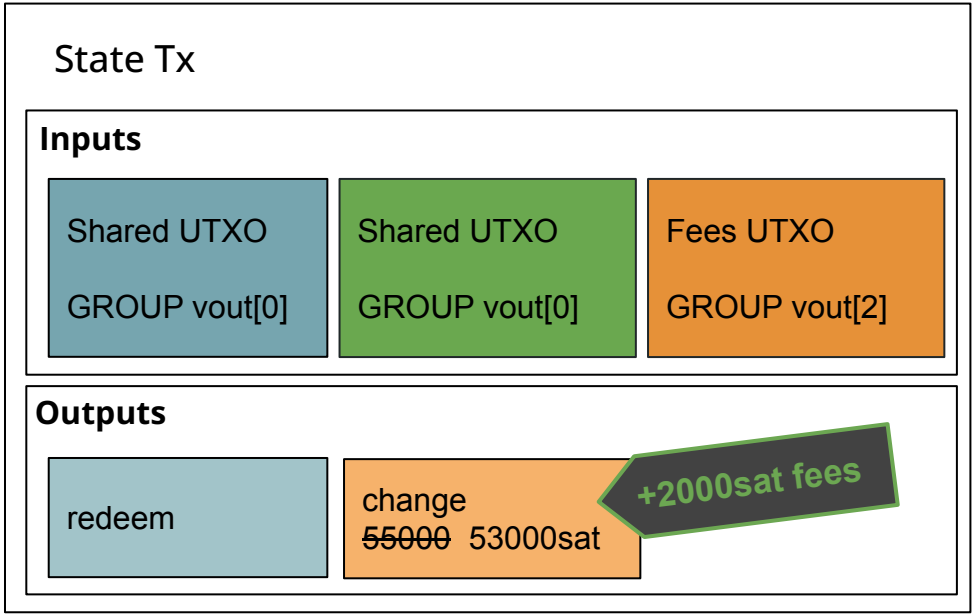
Instead of signing all/single/none of the outputs, specify a range



SIGHASH_GROUP / Signature bundles soft fork

Instead of signing all/single/none of the outputs, specify a range

Adjust fees simply by modifying change output amount

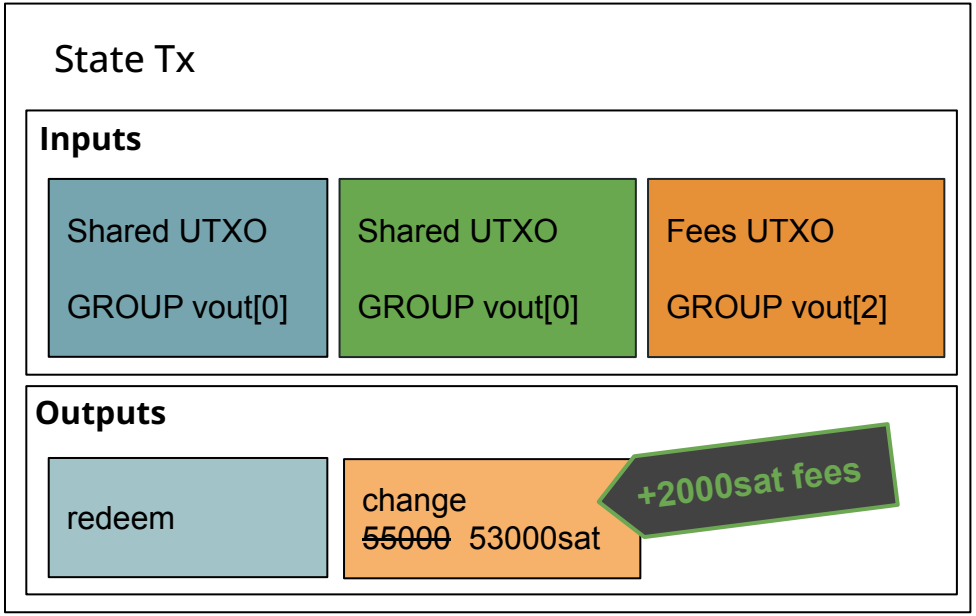


Anthony Towns: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2021-July/019243.html
Rusty Russell: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2018-April/015862.html

SIGHASH_GROUP / Signature bundles soft fork

Instead of signing all/single/none of the outputs, specify a range

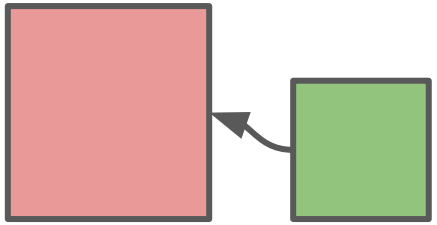
Adjust fees simply by modifying change output amount



Solutions Categorized

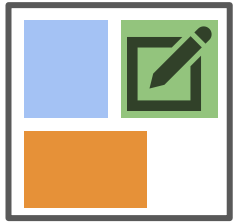
Broadcast As Is

- Magical fee prediction
- Sign multiple txns



Add Sponsor Tx

- CPFP
- Transaction Sponsors (soft fork)



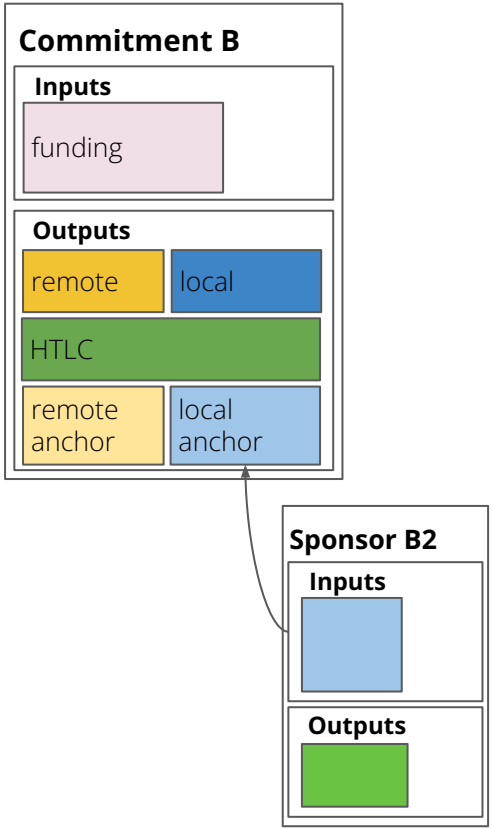
Modify the Tx Itself

- ANYONECANPAY
- SIGHASH_GROUP (soft fork)

Solution Part 1:
Package {CPFP, RBF, Relay}

Package {CPFP, RBF, Relay}

- ✅ 0 fees or 1sat/vB on shared tx (commitment)
- ✅ add fees at broadcast time



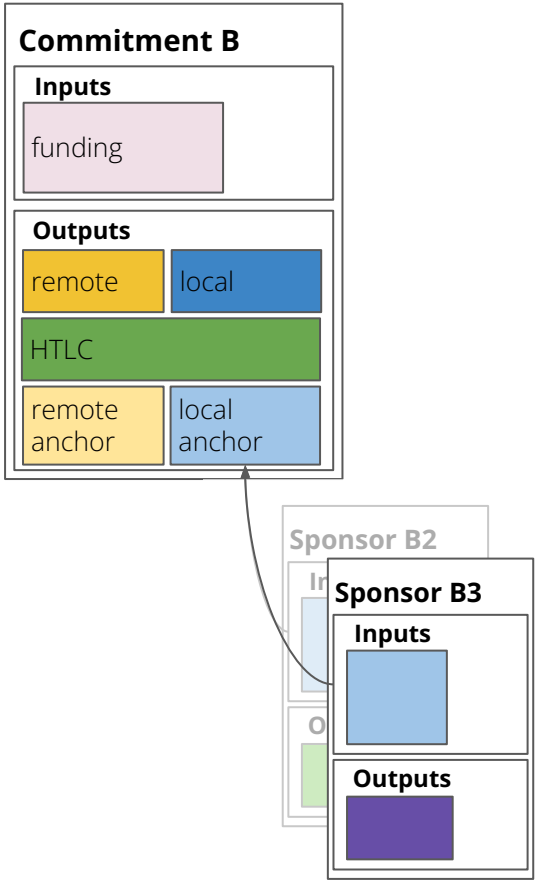
Package {CPFP, RBF, Relay}

✅ 0 fee or 1sat/vB on shared tx (commitment)

✅ add fees at broadcast time

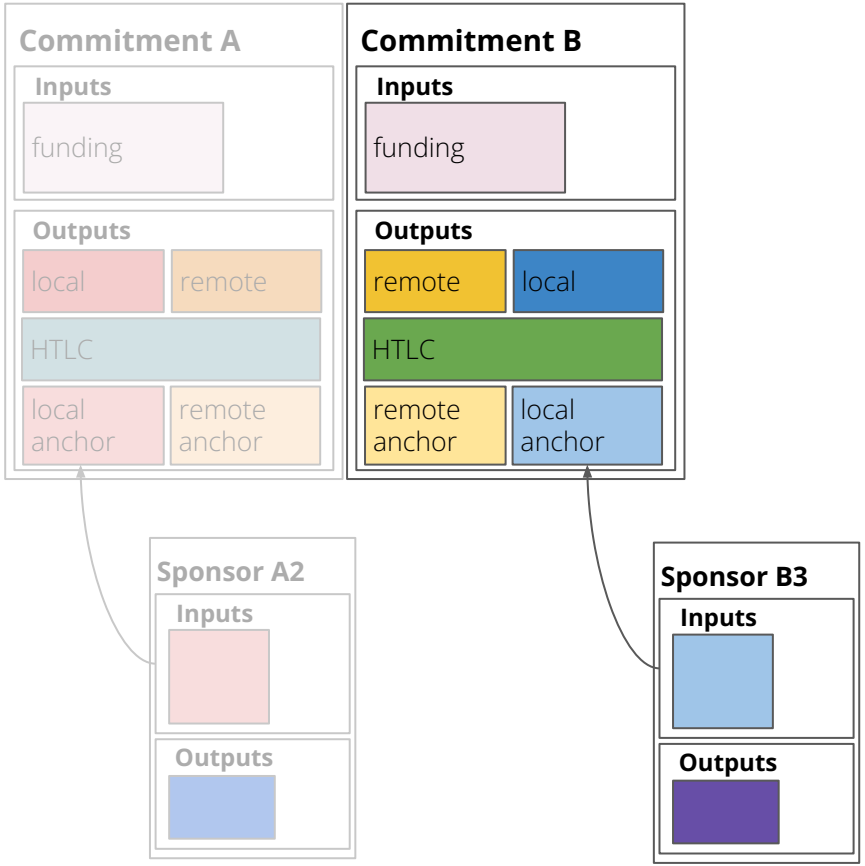
✅ bump feerate by RBFing the child

✅ package relay protocol changes make propagation more reliable



Package {CPFP, RBF, Relay}

- ✅ 0 fees on shared tx (commitment)
- ✅ add fees at broadcast time
- ✅ bump feerate by RBFing the child
- ✅ package relay protocol changes make propagation more reliable
- ✅ if conflicting tx exists, child fees count for RBF fee-related rules

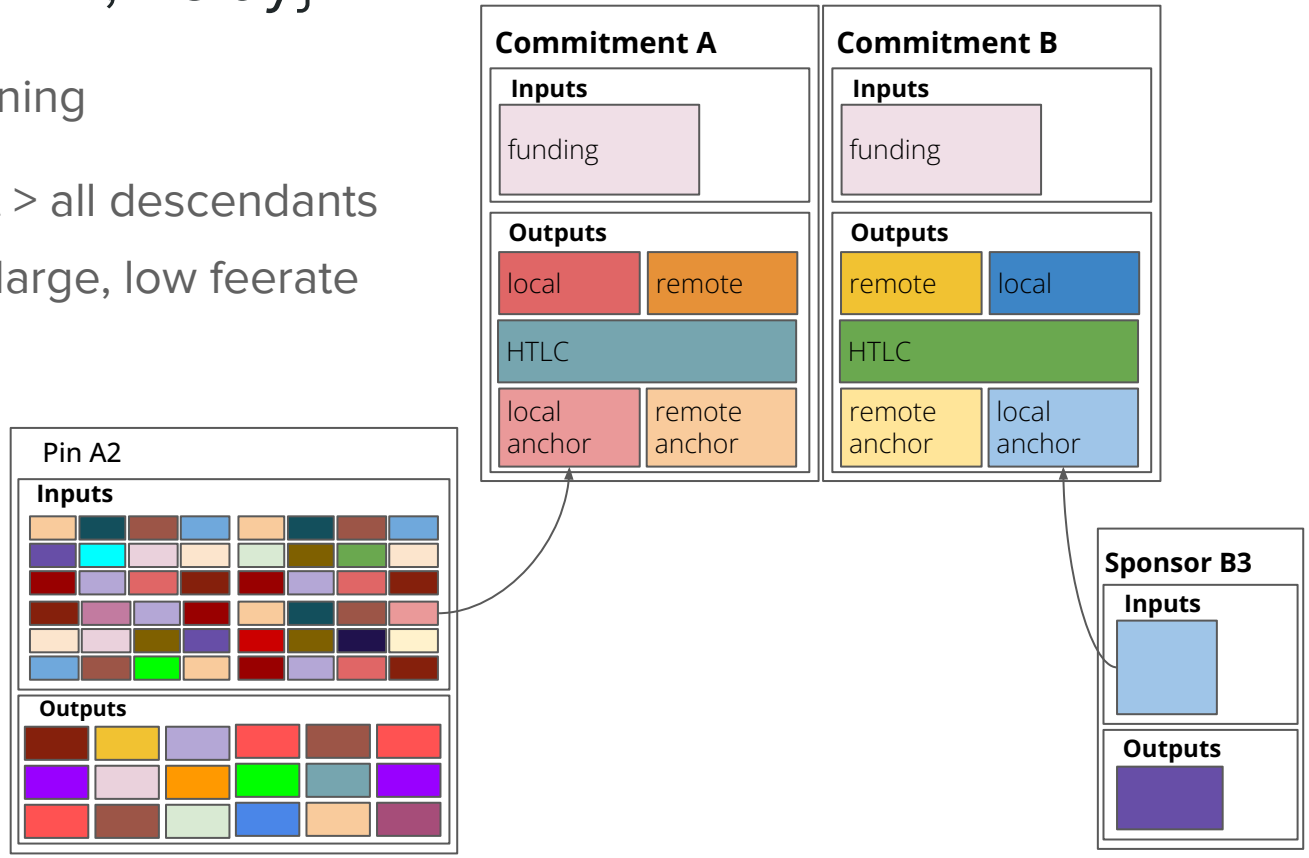


Package {CPFP, RBF, Relay}

❌ Caveat: “Rule 3” Pinning

replacement fees must > all descendants

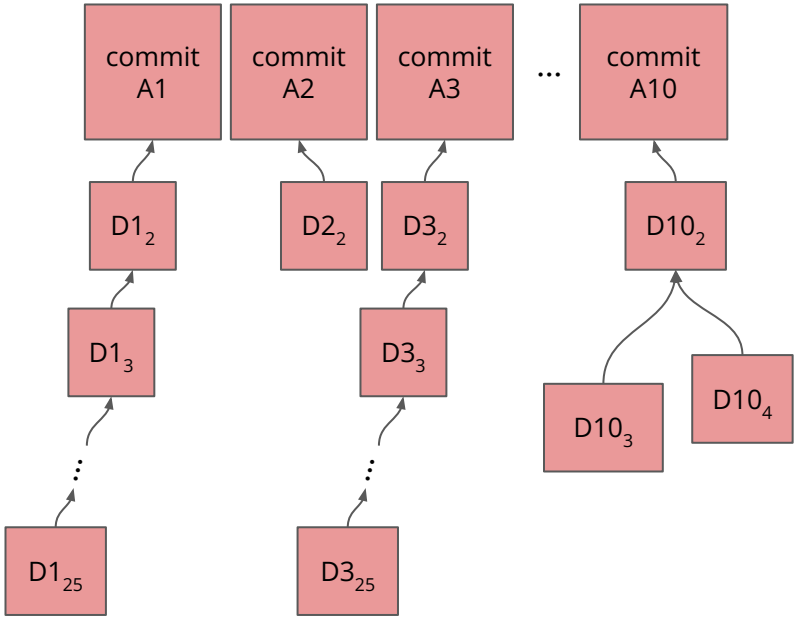
descendant(s) may be large, low feerate

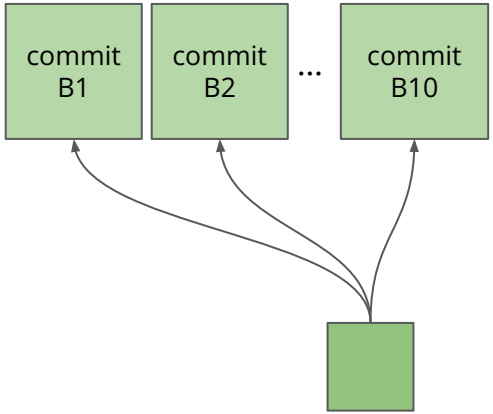


Package {CPFP, RBF, Relay}

❌ Caveat: “Rule 5” Pinning

can’t replace more than 100 at a time. batching is dangerous





“Ah ok, we just need fix RBF”

– some idiot, January 2022

https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2022-January/019817.html
Greg Sanders: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2022-May/020458.html

Solution for this:
Let’s add an incentive compatibility rule to RBF!

Let’s add an incentive compatibility rule to RBF!

Already in Mempool

Conflicting Tx
(15sat/vB)

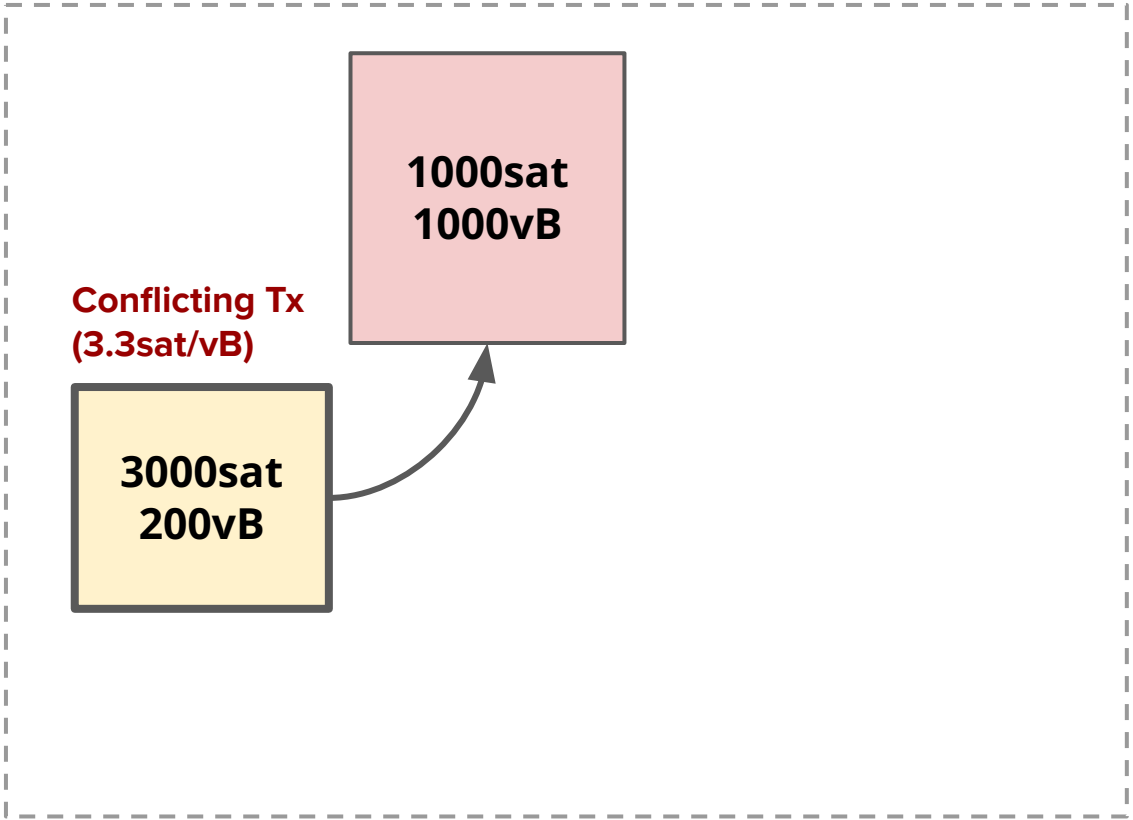
3000sat
200vB

Replacement Tx
(10sat/vB)

1000sat
100vB

Let’s add an incentive compatibility rule to RBF!

Already in Mempool

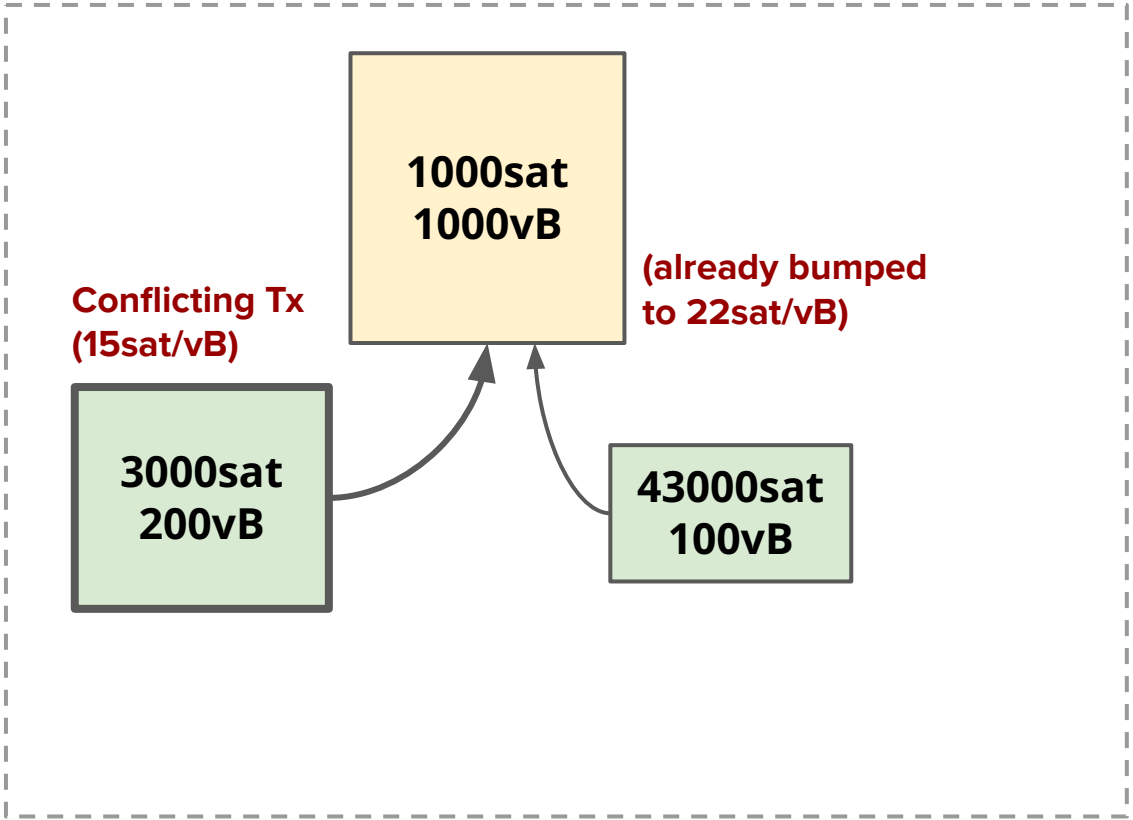


Replacement Tx
(10sat/vB)

1000sat
100vB

Let’s add an incentive compatibility rule to RBF!

Already in Mempool

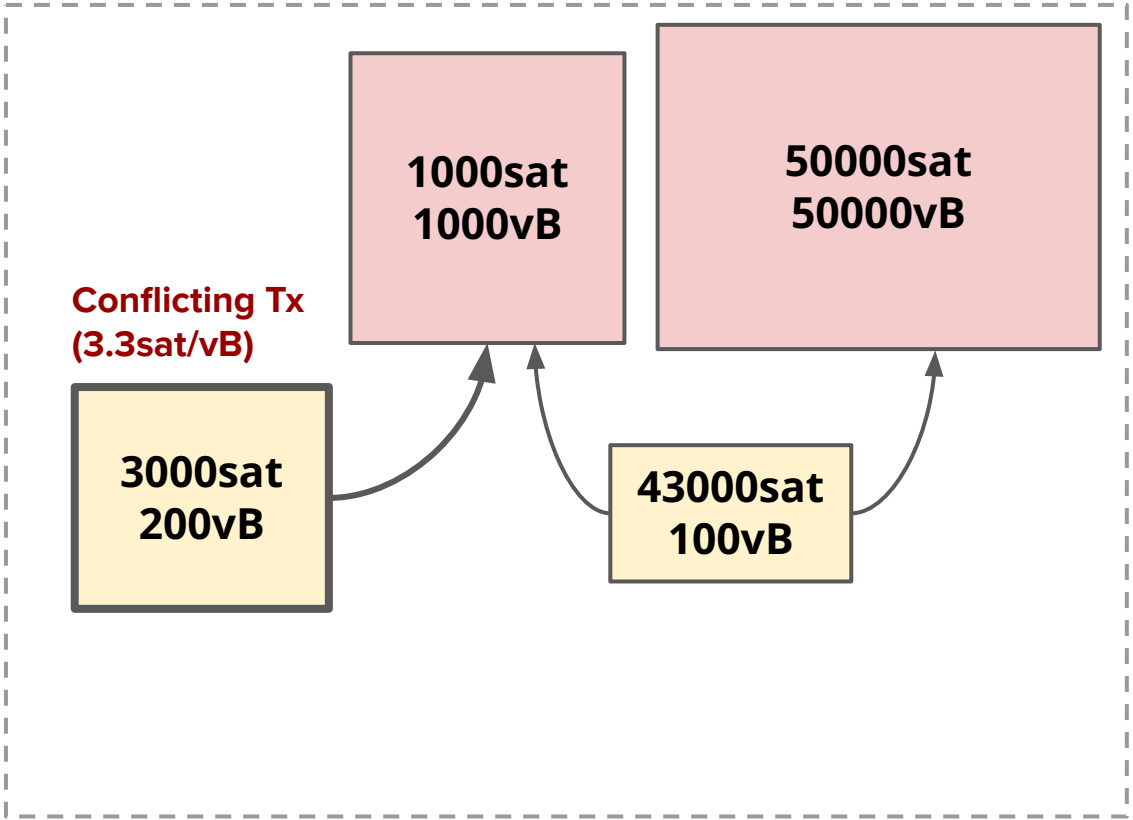


Replacement Tx
(10sat/vB)

1000sat
100vB

Let’s add an incentive compatibility rule to RBF!

Already in Mempool

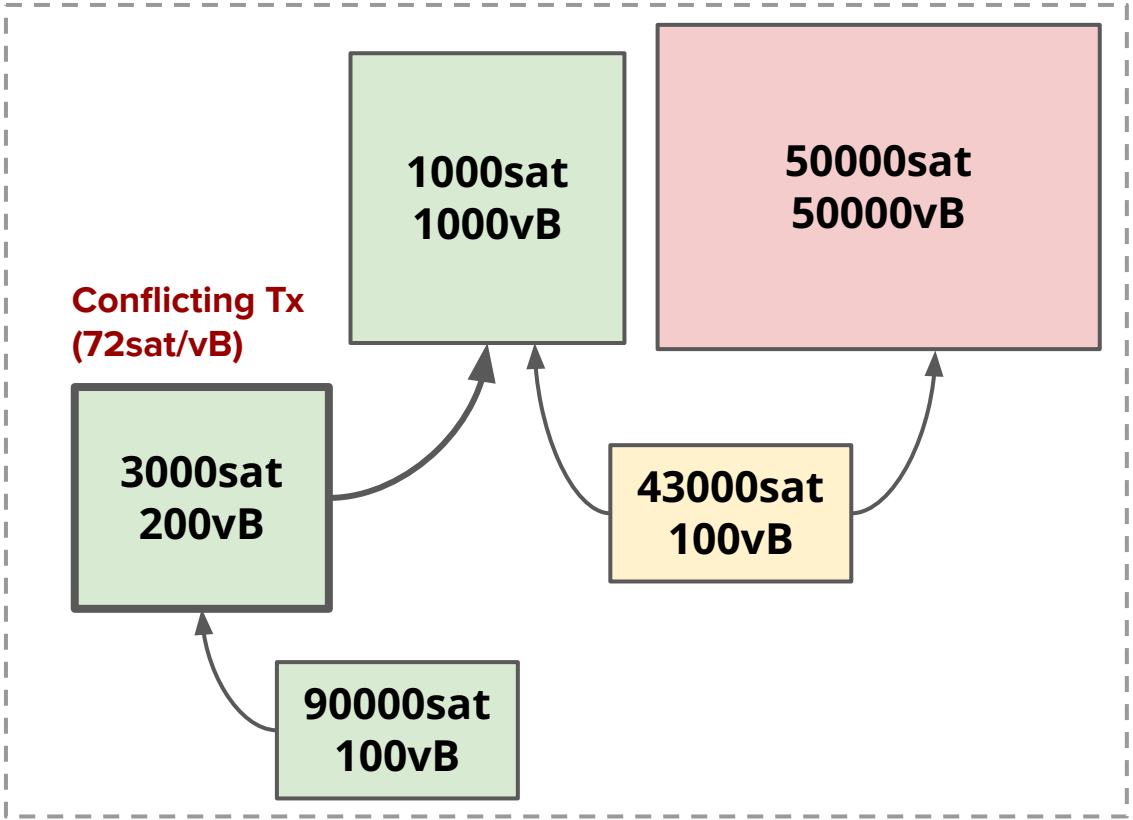


Replacement Tx
(10sat/vB)

1000sat
100vB

Let's add an incentive compatibility rule to RBF!

Already in Mempool

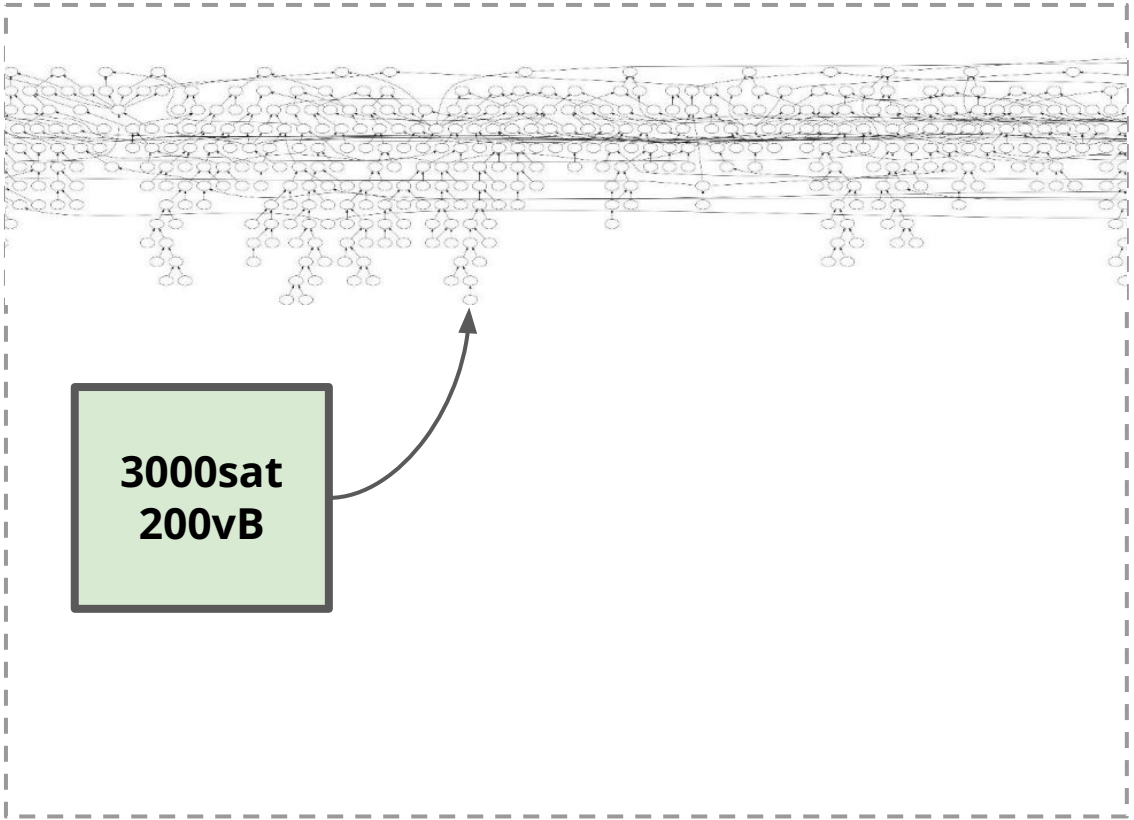


Replacement Tx
(10sat/vB)

1000sat
100vB

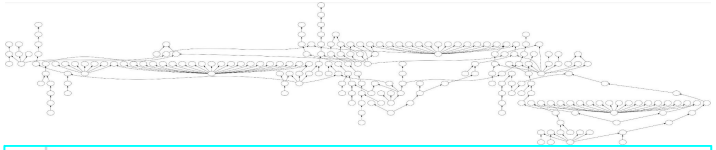
Let’s add an incentive compatibility rule to RBF!

Already in Mempool



Replacement Tx
(10sat/vB)

1000sat
100vB



Murch
@murchandamus

The above cluster was composed of 219 unconfirmed txs. I just found another cluster of 881 linked unconfirmed txs. 😆

If you work on coin selection, please take an input's full ancestry into account when evaluating the viability of unconfirmed inputs during transaction building.

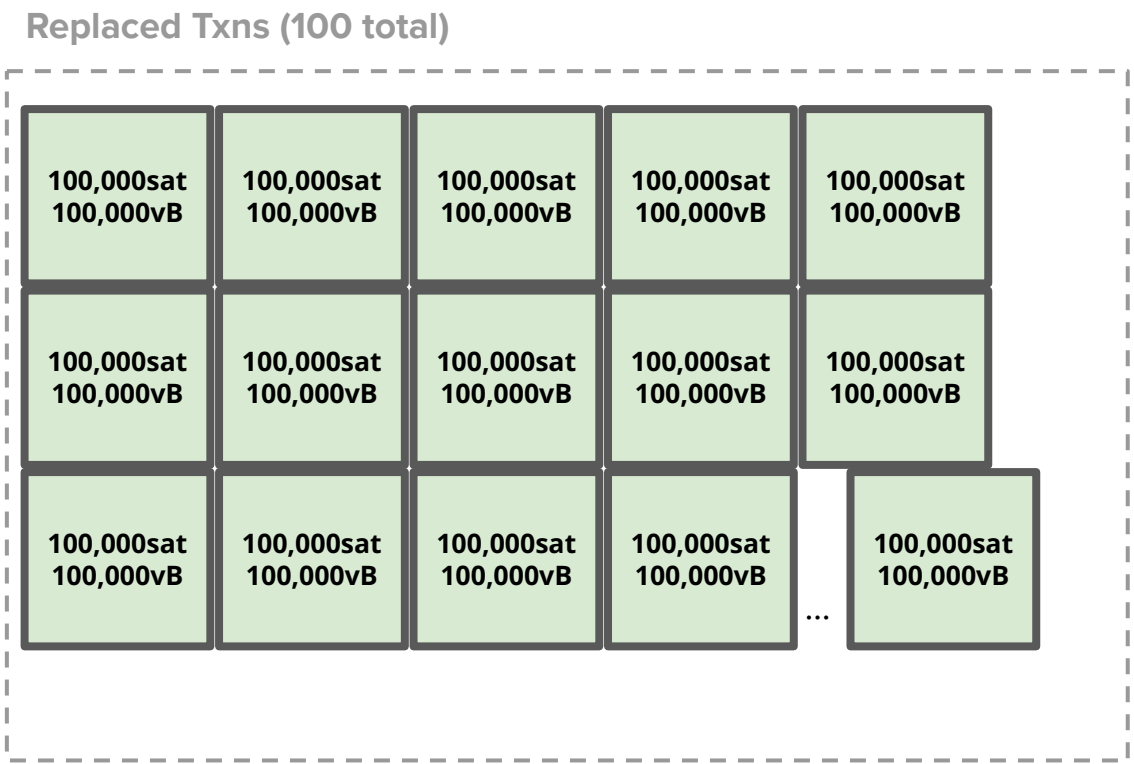
https://twitter.com/murchandamus/status/1352850189686599680?s=20

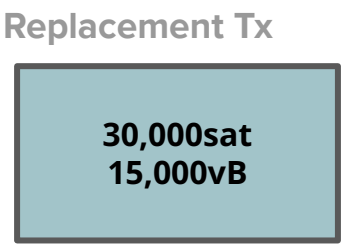
Before you say “can we get rid of Rule 3 entirely?”

“Replacement’s feerate and incentive compatibility score must increase by 2x”

Before you say “can we get rid of Rule 3 entirely?”

“Replacement’s feerate and incentive compatibility score must increase by 2x”

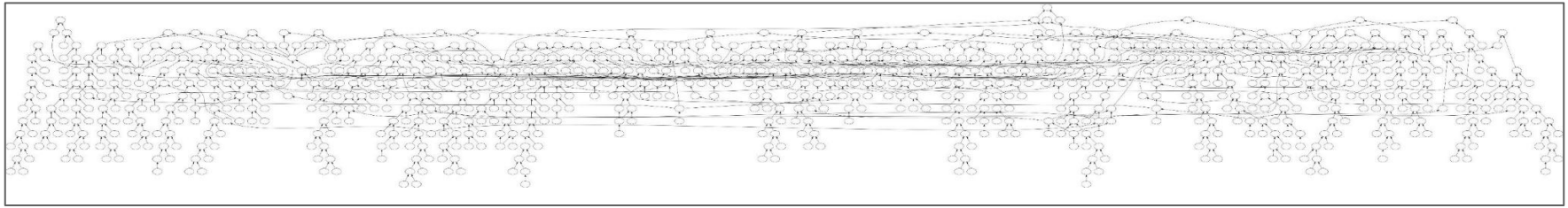


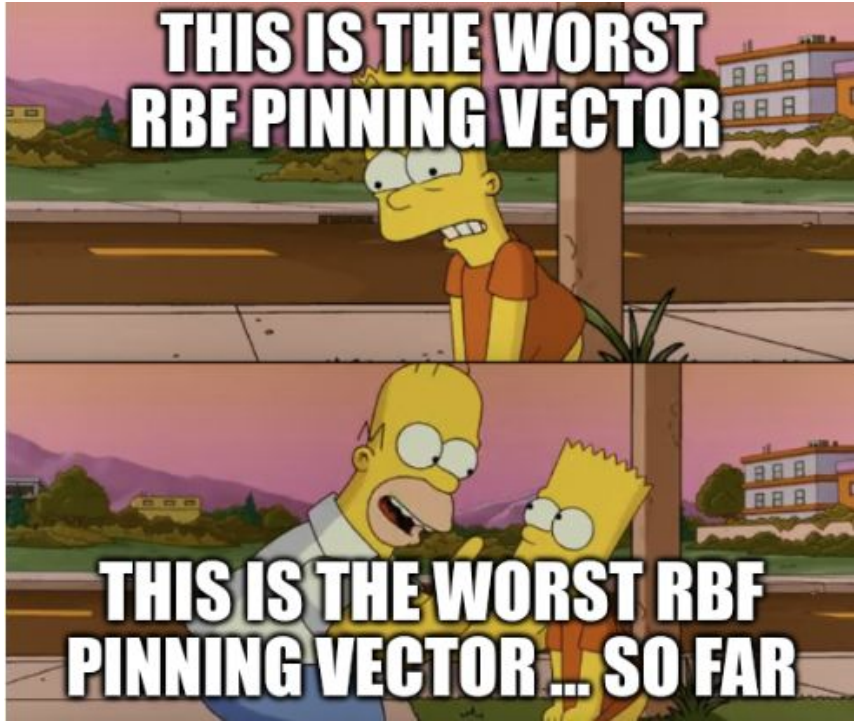


30,000sat paid
10,015,000vB relayed
= 0.003sat/vB

Solution Part 2: v3 to fix pinning

Takeaway: we allow these, even though we can’t properly handle them.

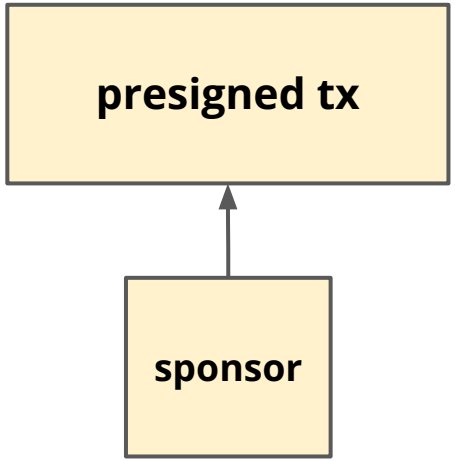



THIS IS THE WORST
RBF PINNING VECTOR
THIS IS THE WORST RBF
PINNING VECTOR ... SO FAR

@instagibbs

V3 Rules

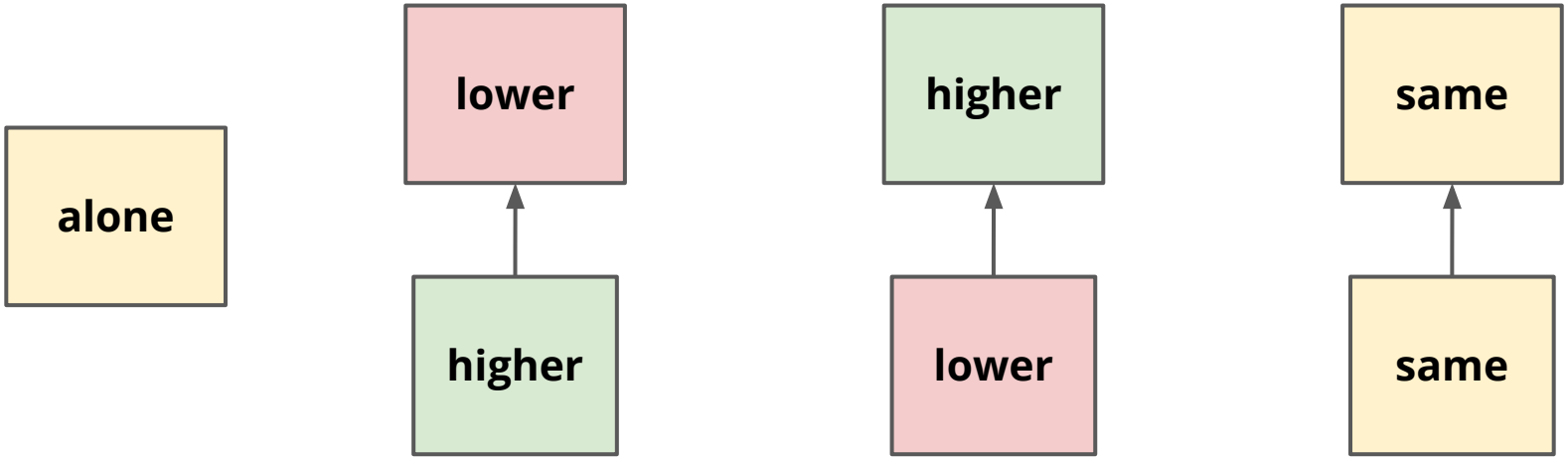
- 1 parent 1 child only
- child can’t be more than 1000vB
- (unconfirmed) v3 must spend v3
- (unconfirmed) non-v3 must spend non-v3
- v3 signals replaceability



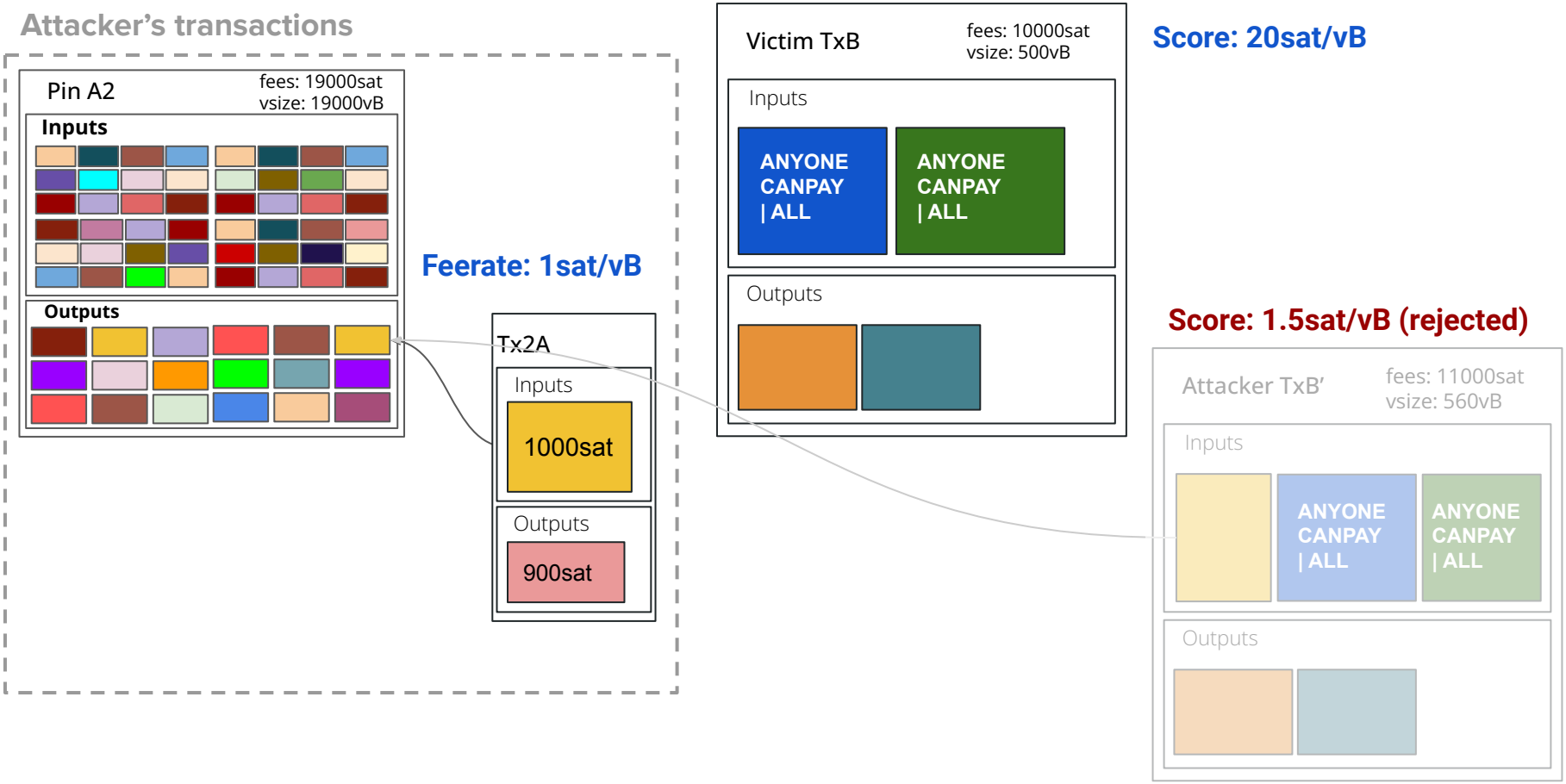
Incentive compatibility score

Cluster can’t be larger than 2, so it’s just = min(self feerate, ancestor feerate)

Pretty easy to show this is always correct:



ANYONECANPAY Replacement Pinning

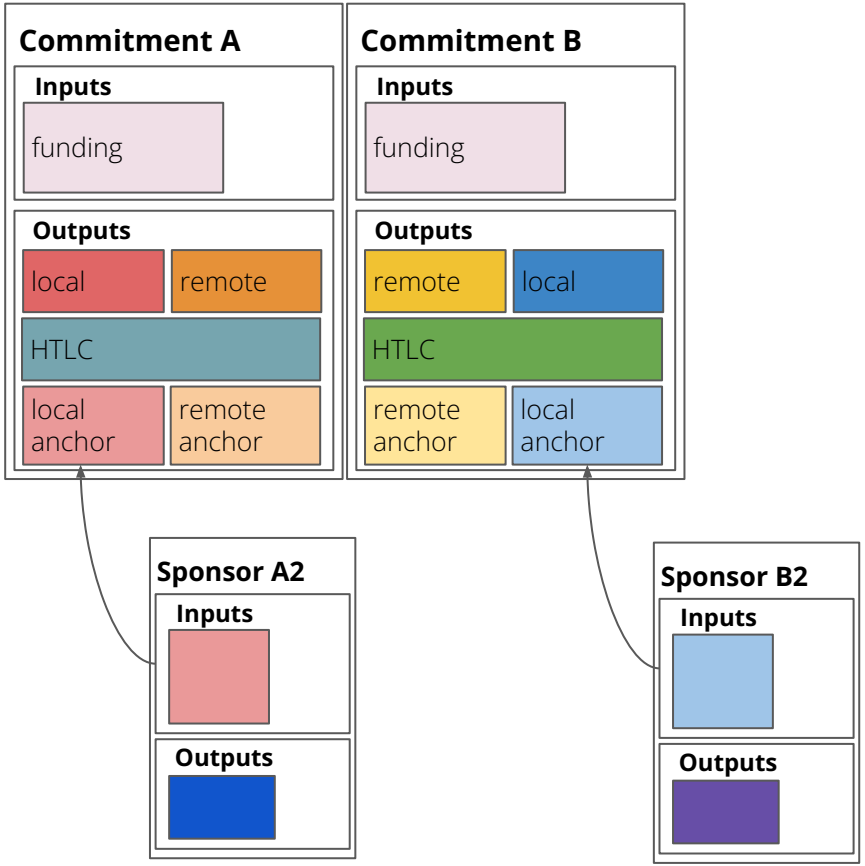


Rule 3 Pinning

Limiting the size of an attached tx == limiting the economic damage your counterparty can do to you

Need A or B to confirm?

1. Decide confirmation target, feerate is ***f***
2. Assuming both txns have size ***s*** vB, add fees of ***f * (s + 1000)*** to fee-bumping child
3. Broadcast commitment tx + child
4. If no confirmation, must be because feerate too low. RBF the child



“Cute, but is this incentive compatible?”

Miner Benefits
(if users use it)

🧠 DoS-resistant, generally computationally cheap to handle

🧠 Can assess incentive compatibility quite easily

User Benefits
(if network adopts it)

✅ No difference between tx signaling and its ancestor signaling

✅ Any RBF requires an incentive compatibility score increase

✅ Just broadcast, no need to monitor mempools to see if you need to pay extra to RBF

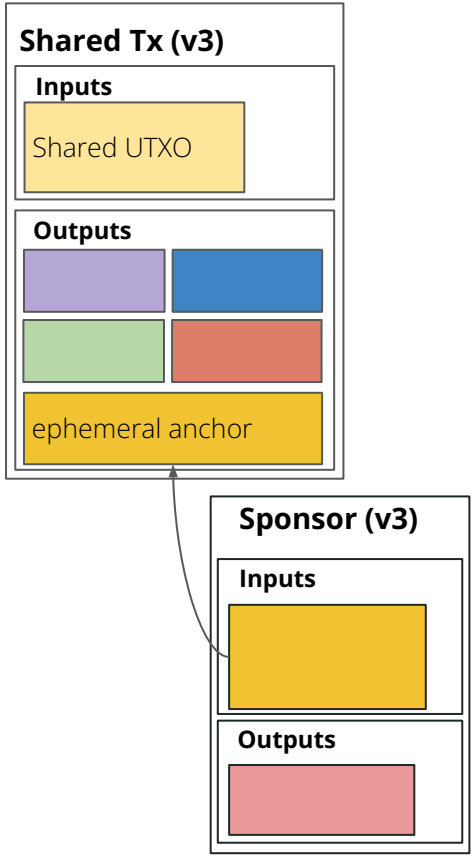
- ✅ Rule 5 pinning severity reduced by 24x
- ✅ Rule 3 pinning severity reduced by 100x

Solution Part 3: Ephemeral Anchors

(Greg Sanders)

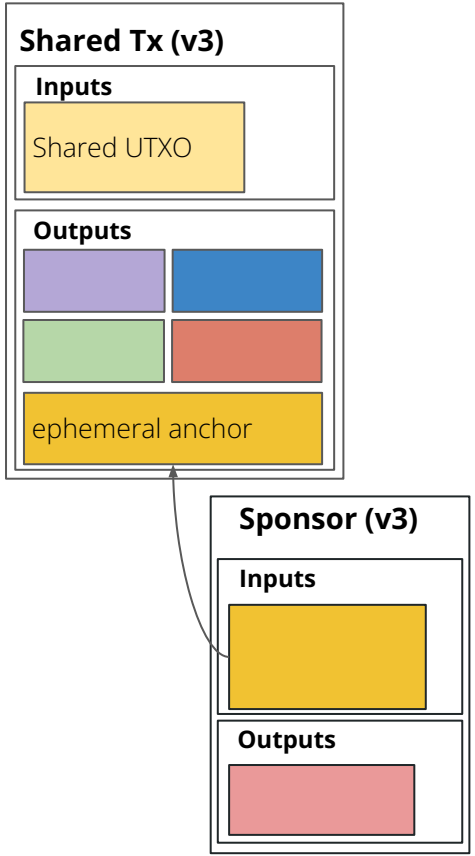
Ephemeral Anchor Rules

- parent:
  - 0 fee, so it must be bumped
  - 1 OP_TRUE output to attach fee-bumping child (“anchor”)
  - anchor output **can be any value** (including 0)
  - v3: only 1 child allowed
- child:
  - spends the anchor (“ephemeral”)
  - v3: only 1 parent allowed



Ephemeral, 0-value, Anchor Outputs

- ✅ Anyone can bump the tx
  - ✅ Watchtowers don't need keys
  - ✅ Works for transactions shared between N>2 parties
- ✅ Exactly 1 anchor output. That output *must* be spent.
  - ✅ Smaller tx size
  - ✅ CPFP carveout can be phased out
  - ✅ Don't need 1 CSV for the other outputs
- ✅ No need to shave value off channel balance (wen eltoo?)



Thanks!