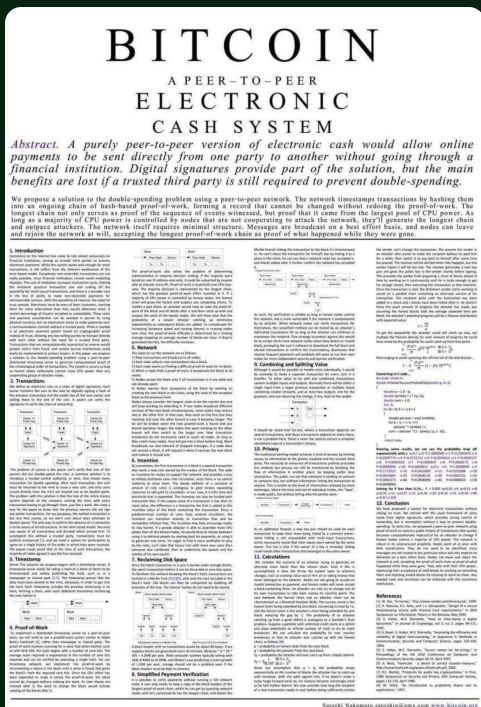


Když node hledá node

Peer discovery a skrytá pravidla propagace transakcí

Matěj Husák – Trezor

Všechno začalo 9 stránkami



9

stránek celkem

Bitcoin whitepaper, 2008

Systém, který položil základy kryptoměn.

Co whitepaper popisuje

- popis Bitcoinu jako systémového celku
- proof-of-work, řetězení bloků, atd.

Co naopak neobsahuje:

- implementační detaily



Jak se nody vlastně najdou?

Whitepaper řeší

- **Elektronické podpisy** a transakce
- **Proof-of-Work** a konsensus
- **Řetězení bloků** a řešení forků
- **Incentivy** (odměna a poplatky)
- **SPV** (zjednodušené ověřování)

Neřeší → věc implementace

Peer discovery (DNS seeds, addr zprávy)

- **Formáty P2P zpráv** (inv, getdata, tx)
- **Relay policy** (co node přepostí a zahodí)
- **Mempool** a konkrétní fee pravidla

Tohle všechno = věc konkrétního softwaru



Consensus je sdílený. Zbytek je volba.

Rozdíl mezi protokolem a pravidly softwaru

- **Consensus pravidla** (co je platný blok) = pevná, sdílená všemi.
- **Relay policy** (co node přeposílá, koho hledá) = věc konkrétní implementace.
- **Různé nody = různá pravidla** — a to je pro zdraví celé sítě v pořádku.

Zastoupení implementací

Celkem ~23 800 nodů



77 % — Bitcoin Core



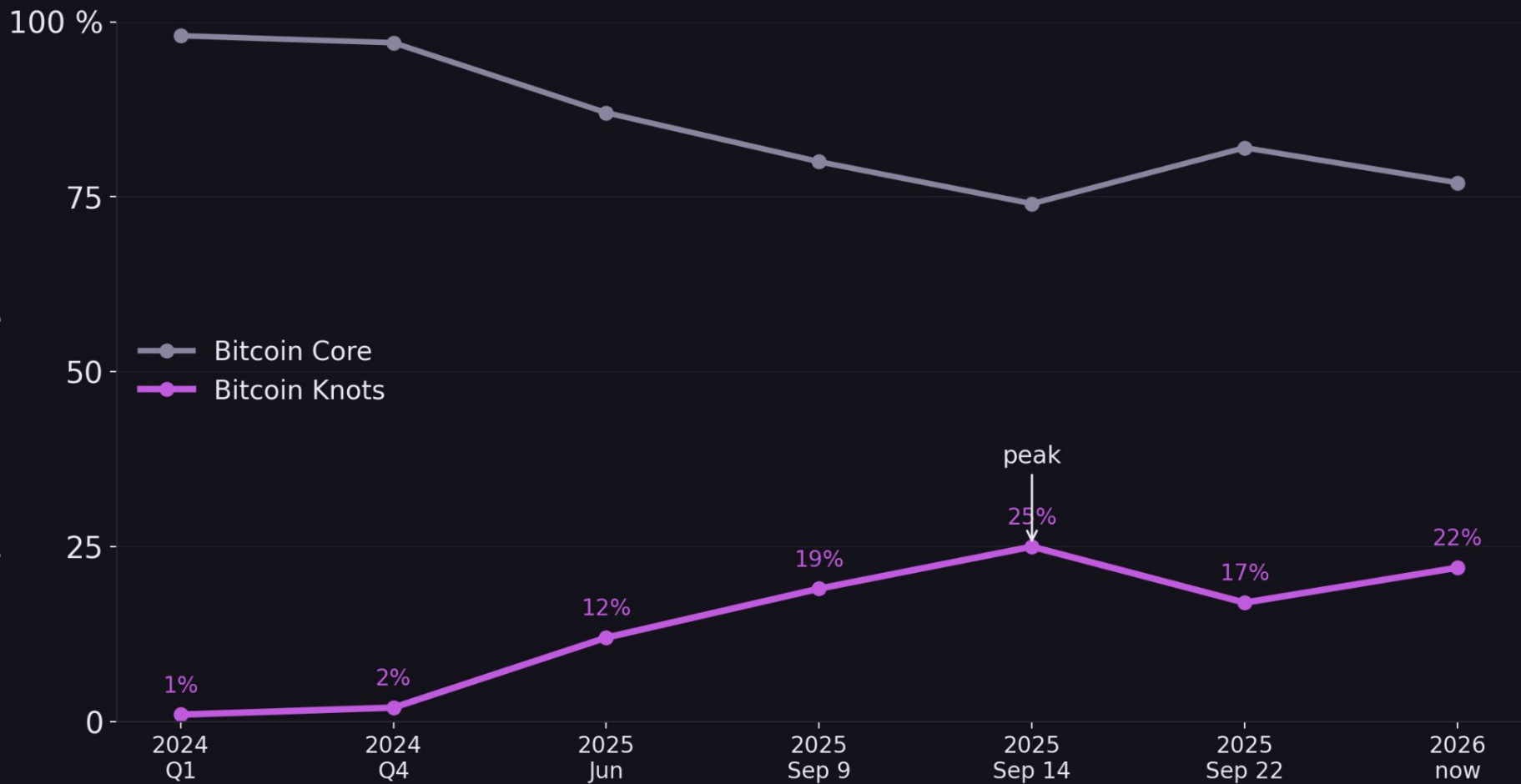
22 % — Bitcoin Knots



1 % — ostatní

Podíl node implementací v čase

podíl dosažitelných nodů





Musíme se vetřít na párty

1. Najít párty

Musím tu párty najít – zjistit, kde vůbec je

2. Zjistit přístup

Zjistit, jestli mě tam pustí – a jestli chtějí.

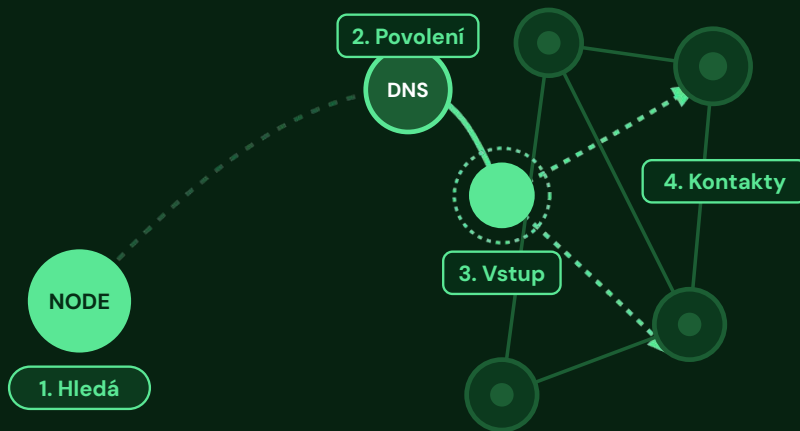
3. Vstoupit

Když jo – jdu dovnitř, nebo se ptám znova pokud mě vyghostujou.

4. Získat kontakty

A hned si najdu další kontakty – ať ji příště nehledám znovu

Node to dělá úplně stejně – jen místo jmen používá IP adresy





Ale kde ta párty vlastně je?

Čerstvý node nezná jedinou adresu.



1. Adresy z minula (peers.dat)

„Vím, kde se scházejí“ — historie uložená na disku



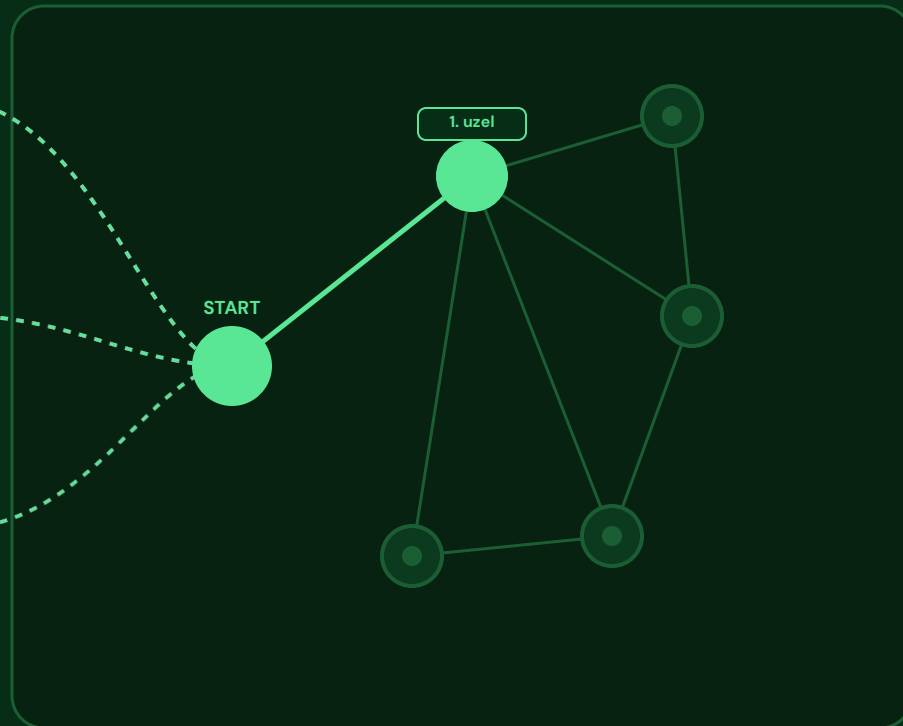
2. DNS seeds (fresh-default)

Zeptám se známého, kdo dnes pořádá akce.
DNS hard coded přímo v implementaci.
Původně IRC (odstraněno v 0.8.2.)



3. Hardcoded fallback

Nouzový seznam pevně zapsaný v kódu





Sedm jmen v Coru, devět v Knots

Kdo tě na párty zve: DNS seeds v kódu

Bitcoin Core (7)

bluematt.me
jonasschnelli.ch
petertodd.net **jen Core**
sprovoost.nl
emzy.de
wiz.biz
achownodes.xyz

Bitcoin Knots (9)

sipa.be **jen Knots**
bluematt.me
dashjr **jen Knots** **hardfork**
haf.ovh **jen Knots** **hardfork**
jonasschnelli.ch
sprovoost.nl
emzy.de
wiz.biz
achownodes.xyz

I seznam „koho se zeptat na cestu” je **volba implementace**. Knots přidává seedy podporující hardfork seeding (BIP-110).



Potřesení rukou

P2P handshake a vyjednávání vlastností spojení

1. Výměna version zprávy

Version, services (segwit, full node), timestamp, addr_recv, addr_from, nonce, user_agent, start_height, relay

2. Společná řeč

Domluví se na nižší společné verzi protokolu. Díky tomu si bez problémů rozumí staré i nové implementace.

3. Přísná pravidla protokolu

Chováš se špatně? Uzel tě začne přehlížet a přednostně odpojí (discouragement místo tvrdého banu).

Handshake sequence



Dokud se nepředstavíš a nepotvrdíte si podmínky, druhá strana **neřekne ani slovo** a nepošle žádná data o transakcích či blocích.



„Koho dalšího znáš?“

Budování okruhu známých a správa adres v paměti (addrman)

1. Gossip (getaddr → addr / addrv2)

Nody si mezi sebou neustále posílají a přeposílají adresy ostatních známých účastníků sítě.

2. Databáze adres (addrman)

Node si udržuje vlastní paměťovou databázi adres s historickými informacemi a statistikou dostupnosti.

3. Buckety (ochrana proti sybil útoku)

Adresy jsou roztržděné do samostatných přihrádek na základě IP rozsahu a klíče. New | Tried.

Cíl: Pár zamořených nebo útočných adres ze stejné sítě nesmí zabrat celou tabulku sítě.

Struktura addrman (Buckety)^{New & Tried tables}

P1
Peer A

P2
Peer B

P3
Peer C

Bucket 0x01
192.168.1.1
10.0.0.42

Bucket 0x02
172.16.0.5
bad.node.ip

Bucket 0x03
84.200.2.1
[empty]

Bucket 0x04
45.33.32.156
212.83.1.9

Bucket 0x05
bad.node.ip2
[empty]

Bucket 0x06
93.184.216.34
198.51.100.1

NODE

Izolované přihrádky štipí zbytek tabulky



yarn crawl | Bitcoin node scanner

```
→ bitcoin-node-scanner git:(main) ✗ yarn crawl
93.186.2.15:8333      52ms    966573  /Satoshi:31.1.0/
80.219.131.189:8333  60ms    966573  /Satoshi:28.1.0/
78.80.86.20:8333     59ms    966573  /Satoshi:31.1.0/
164.152.163.102:8333 127ms   966573  /Satoshi:29.3.0/
45.152.160.19:8333   117ms   966573  /Satoshi:29.4.0/
95.214.148.216:8333  126ms   966573  /Satoshi:31.1.0/
178.142.223.253:8333 50ms    966573  /Satoshi:31.0.0/
93.130.207.36:8333   45ms    966573  /Satoshi:31.1.0/
189.154.88.127:8333  179ms   966573  /Satoshi:31.0.0/
98.73.172.33:8333    131ms   966573  /Satoshi:31.1.0/
147.93.3.56:8333     118ms   961639  /Satoshi:29.3.0/Knots:20260508/
67.230.150.224:8333  1153ms  966573  /Satoshi:27.0.0/
72.0.173.134:8333    228ms   966573  /Satoshi:28.1.0(FutureBit-Apollo-Node)/
154.53.63.218:8333   170ms   966573  /btcwire:0.5.0/btcd:0.26.2/
37.230.168.216:8333  67ms    966573  /Satoshi:31.0.0/
97.102.235.168:8333  151ms   966573  /Satoshi:29.0.0/
31.201.110.138:8333  98ms    966573  /Satoshi:29.1.0/
212.110.171.118:8333 44ms    966573  /Satoshi:0.13.1/
66.219.196.170:8333  161ms   966573  /Satoshi:28.0.0/
157.211.242.28:8333  378ms   966573  /Satoshi:31.1.0/
213.109.236.129:8333 36ms    966573  /Satoshi:24.99.0/
89.161.26.230:8333   59ms    966573  /Satoshi:28.1.0/
85.208.117.15:8333   251ms   966573  /Satoshi:28.0.0/
104.250.145.186:8333 176ms   966573  /Satoshi:30.2.0/
80.227.241.154:8333  195ms   966573  /Satoshi:28.1.0/
148.251.237.73:8333  18ms    966573  /Satoshi:30.0.0/
^C
Aborting crawl...

Done in 28.5s
  scanned: 311
 reachable: 247
```

GitHub





8 kamarádů, které si vybírám sám

Topologie peer-to-peer spojení a ochrana proti obklíčení uzlu

Typy spojení (Bitcoin Core defaults)

~8 **Outbound**: Vybírá si můj uzel sám (aktivní)

~117 **Inbound**: Přijímám od ostatních (pasivní)

Eclipse Attack (Hrozba)

Útočník tě obklíčí samými svými uzly a izoluje tě od zbytku pravdivé sítě.

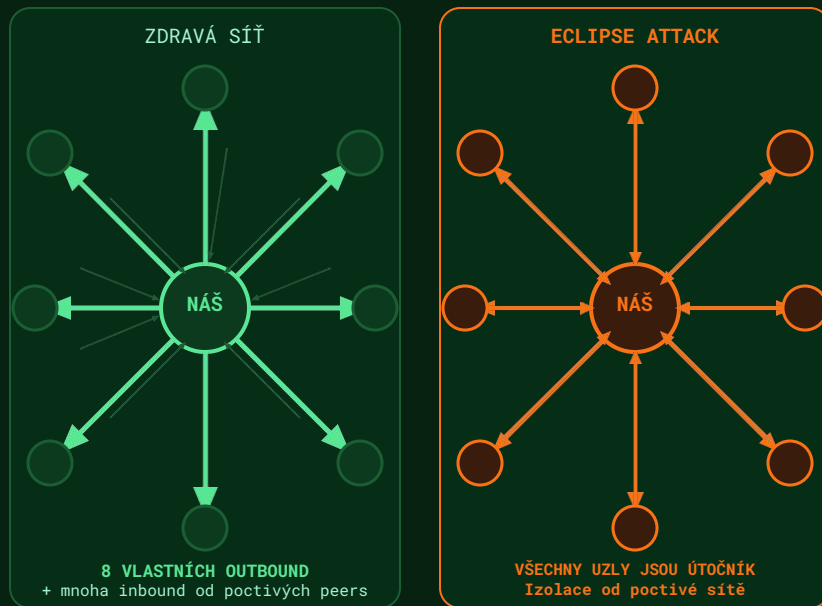
Obrana: Oddělení inbound/outbound, **feeler** a **anchor** spojení.

Ochranné mechanismy

Anchors: Pamatuje si 2 prověřené outbound adresy i po restartu.

Feelers: Krátkodobá testovací spojení pro ověření nových adres z addrman.

Normální stav vs. Eclipse Attack





**Dobře. Node teď zná jiné nody. Co si
ale posílají?**

Přechod od objevování uzlů k propagaci transakcí



Transakce neletí rovnou k těžaři

Šíření transakce v P2P síti (gossip) a announce-first princip

1. Skok po skoku (gossip)

Transakce putuje sítí postupně od jednoho nodu k druhému. Nikdo ji neposílá přímou cestou minerovi.

2. Třífázový handshaking

inv: „Mám tohle TXID“ (nabídka)

getdata: „Neznám, pošli mi ho“ (žádost)

tx: „Tady je celá transakce“ (data)

3. Efektivita a šetření pásma

Nejdřív se oznamuje jen ID (announce-first), nikoliv celá transakce.

Cíl: Šetří se datový tok — žádný uzel nedostane stejnou transakci dvakrát.

Detailní průběh mezi dvěma sousedy



Krok 1: inv (Inventory)

B → C: „Mám tx 0a3f...“ (pouze Hash/ID)

Krok 2: getdata

C → B: „Pošli mi kompletní tx 0a3f...“

Krok 3: tx (Transaction)

B → C: [Plná data transakce (vstupy, výstupy)]



Mýtus: víc spojení = rychlejší transakce

Šíření transakce v P2P síti, Poisson delay a privacy mechanizmy

Intuice vs. Realita

Intuice říká: víc peerů = dřív u cíle. Není to tak. Gossip je záměrně zpomalený pomocí náhodných dávek (Poisson delay).

Ochrana soukromí

Dělá se to kvůli soukromí: aby nešlo ze sítě snadno vysledovat, od kterého konkrétního uzlu transakce původně vyšla.

Proč existují „sniffer“ nody?

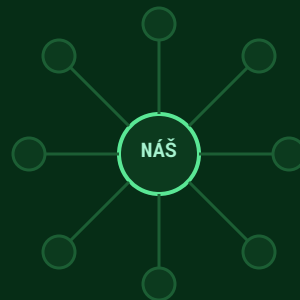
Víc spojení = lepší posлуouchání, ne rychlejší vysílání vlastní transakce.

Síť je rovnostářská schválně — vysoký počet spojení tě při odesílání nijak nezvýhodní.

Aha! moment: Uzly s tisíci spojení slouží výhradně k monitorování sítě.

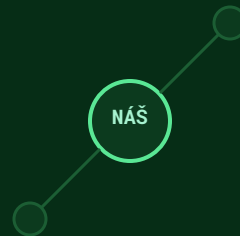
Porovnání rychlosti propagace

HODNĚ SPOJENÍ



8+ outbound peerů

MÁLO SPOJENÍ



pouze 2 peery



Čas propagace je v obou případech SKORO STEJNÝ

Poisson delay vyrovnává rychlost šíření transakcí skrz napříč sítí.



Žádný „globální mempool“ neexistuje

Lokální mempooly, minrelaytxfee a individuální pravidla uzlů

Lokální stav, ne globální síť

Každý node si drží **vlastní mempool**. Neexistuje žádná centrální databáze nepotvrzených transakcí.

Individuální pravidla validace

Každý má vlastní názor na to, co je **platná tx a co spam**. Co jeden uzel přijme, druhý může okamžitě zahodit.

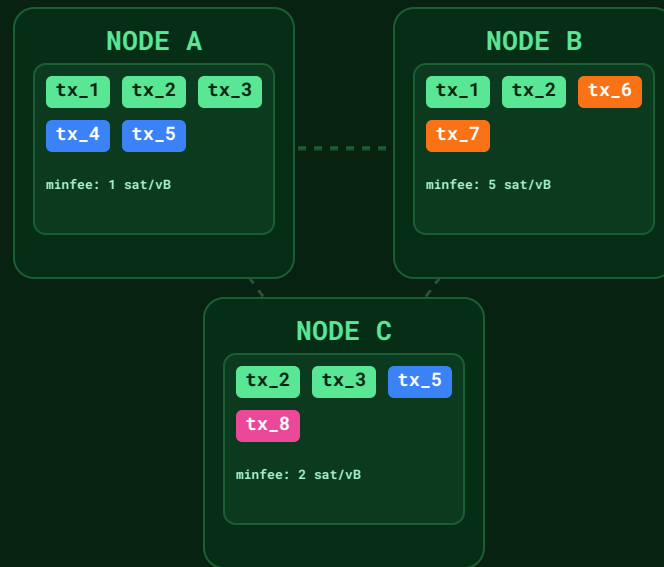
Filtr minrelaytxfee

minrelaytxfee slouží jako první bezpečnostní a ekonomický prahový filtr.

Co neprojde přes nastavený poplatkový práh, ten node **dál nepošle**.

Aha! moment: Mempool je ryze lokální záležitost, obsahy se napříč sítí liší.

Srovnání obsahu mempoolů v síti



Částečný překryv: pouze tx_2 mají všechny 3 uzly



mempool.space = okno do jednoho nodu

Proč populární explorer nepředstavuje centrální pravdu

Nástroj, který všichni známe

Oblíbený explorer zobrazuje mempool **jediného uzlu**. „Globální mempool“ je pouhá iluze čistého uživatelského rozhraní.

Architektura v pozadí

Bitcoin Core node uchovává data, **Electrum server** vyřizuje vyhledávání adres a rozhraní (frontend) vše vizualizuje.

Rozdílná pravidla (Relay Policy)

Jiný uzel v síti = jiná relay policy = **odlišný obsah mempoolu**.

Co mempool.space zobrazuje jako nepotvrzené, jiný node nemusel vůbec přijmout.

Aha! moment: relaymap dělá opak — místo jednoho nodu ukazuje rozmanitost celé sítě.

Co se skrývá za rozhraním?

1 KONKRÉTNÍ NODE

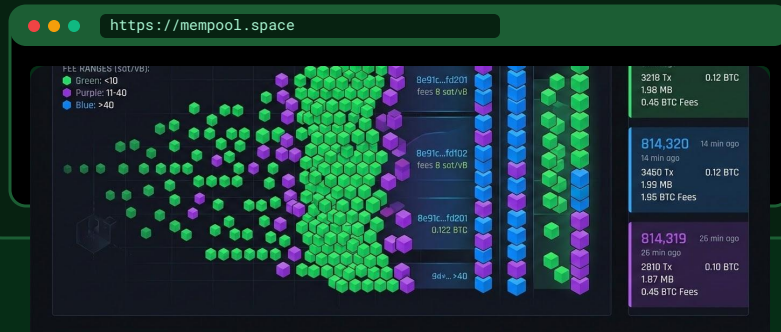
Bitcoin Core
Vlastní relay policy

Electrum Server
Adresní lookup & index

Mempool API



JSON Data / WebSockets





RBF: na tenhle parametr jsme zapomněli

Replace-By-Fee a pravidla pro bezpečné nahrazování transakcí v mempoolu

Replace-By-Fee (RBF)

Nahrazení nepotvrzené tx verzí s vyšším poplatkem (fee).
Umožňuje zrychlit confirmaci zaseknuté transakce.

Háček: Incremental Relay Fee

Přírůstek poplatku musí překonat **incrementalrelayfee**. Nestačí přidat jen 1 satoshi, rozdíl musí být dostatečně velký.

Mechanismus nahrazení (RBF)

PŮVODNÍ TX

tx_orig

Fee: 2 sat/vB
Total: 280 sat

STUCK IN MEMPOOL

RBF NÁHRADA

tx_replacement

Fee: 2 sat/vB (Base)
280 sat

+ INCREMENTAL
+5 sat/vB (Δ fee)
New: 7 sat/vB



Podmínka přijetí uzlem:

$\text{New_Fee} \geq \text{Old_Fee} + \text{incrementalrelayfee}$
(Rozdíl Δ fee musí pokrýt náklady na relay)



Co jsou ty skrytá pravidla?

Přechod od objevování uzlů k propagaci transakcí



Relay pravidla

Klíčový slepý úhel v P2P komunikaci Bitcoinových uzlů

Feefilter signalizuje minrelaytxfee ✓

Uzel běžně oznamuje minimální poplatek pro transakce, které je ochoten přijmout a přeposlat.

Ale incrementalrelayfee NE ✗

Neexistuje žádný P2P protokolový signál pro požadovaný přírůstek poplatku při RBF.

→ RBF náhradu posílám naslepo

Nevím, jestli má můj peer stejný práh jako já. Když ho má vyšší, tiše mi ji odmítne — a já se to nedozvím.

Slepý bod P2P protokolu

NODE A (TY)

min fee ✓
Signál: 1 sat/vB



RBF tx
(naslepo)

NODE B (PEER)

min fee ✓
Přijímá: 1 sat/vB

Skrytý práh:
+3 sat/vB incremental
Pokud RBF nesplní
práh, tx se ticho
zahodí!

Tiché odmítnutí: Node A neví, zda Node B náhradu přijal



Co s tím udělat?

Řešení slepého bodu P2P komunikace a poctivý pohled na realitu v síti

Návrh řešení

- **Signalizovat přes P2P i incrementalrelayfee**

Rozšířit protokol o explicitní požadavek na přírůstek poplatku.

- **Aktivní vyhledávání vhodných peerů**

Uzel si najde peery s odpovídajícím prahem — zpět k principu „node hledá node“.

- **Souvislost s BIP 153 (SENDTEMPLATE)**

Širší rámec pro efektivní vyjednávání struktury a pravidel mempoolu.

Rozšířený feefilter (Návrh)

NODE A (TY)

min fee ✓
Signál: 1 sat/vB

incremental fee ✓
Signál: +1 sat/vB
(BIP rozšíření)

RBF tx
(na jistotu)

NODE B (PEER)

min fee ✓
Přijímá: 1 sat/vB

incremental fee ✓
Požadavek: +1 sat/vB
Konec blind spotu

Poctivá protiváha / Protiargument (Murch):

Po backportu do Bitcoin Core 29.1 už **>30 % nodů v síti akceptuje nižší incremental feerate**.

V praxi tak RBF náhrada skoro vždy úspěšně projde i bez nového protokolového signálu.



Node už není sám — Párty teprve začíná

Propojená síť

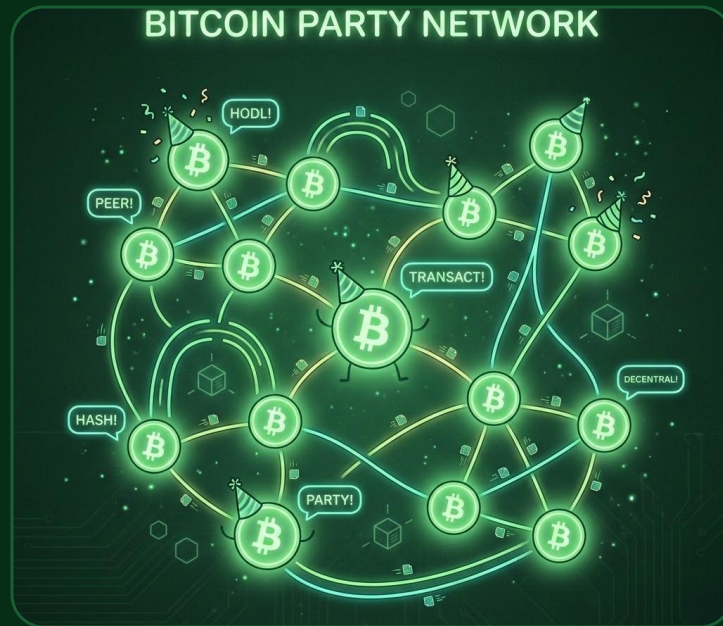
Uzel, co na začátku nikoho neznal, teď ví, jak si najít sousedy a efektivně komunikovat.

Evoluce protokolu

I po 15 letech se dá základní síťová vrstva zlepšovat a posouvat dál.

Stálý vývoj

Bitcoin není hotový. Vývoj P2P vrstvy je neustálý proces.





Díky! Otázky?