Kriptovaluták elméleti háttere

Ladóczki Bence
docens
Távközlési és Mesterséges Intelligencia Tanszék (TMIT)
Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar (VIK)
ladoczki.bence@vik.bme.hu

2026. november 3.

1 P2P hálózat és blokkterjesztés

A blokklánc csomópontjai központi szerver nélkül, egyenrangú (peer-to-peer, P2P) kapcsolatokon keresztül cserélnek adatot. A hálózati réteg feladata, hogy a tranzakciók, blokkok és egyéb protokollüzenetek eljussanak a többi csomóponthoz; az üzenetek érvényességéről és a kanonikus láncról ettől külön a konszenzus szabályai döntenek. Blokkterjesztéskor a javasló vagy bányász elküldi az új blokkot néhány társának, azok pedig továbbadják a hálózaton. A terjedés sebessége biztonsági szempontból is számít: amíg a csomópontok nem értesülnek ugyanarról a blokkról, átmenetileg eltérő láncfejeket láthatnak, PoW-rendszerben pedig a későn érkező blokk növelheti az elavuló blokkok arányát.

Az Ethereum konszenzusrétege a gyors terjesztést igénylő üzenetekhez GossipSub-alapú pletykahálózatot használ. Egy csomópont az üzenetet a kapcsolati hálózatának egy részével osztja meg; a társak továbbítják, illetve jelzik, ha egy üzenet hiányzik, és szükség esetén külön kérik le. A teljes blokk egyetlen nagy üzenetként való küldése egyszerű, de két hátránya van: az egyes csomópontok csak a teljes üzenet megérkezése után tudják továbbküldeni, és az egymással párhuzamosan érkező másolatok fölösleges sávszélességet használhatnak. A következő eljárások ezeket a késleltetési, sávszélesség- és adat-elérhetőségi problémákat különböző módon kezelik.

Snappy: tömörítés az üzenetküldésben. Az adatokat a hálózaton gyakran tömörített formában küldik. A Snappy egy gyors, általános célú tömörítési formátum: csökkentheti az átvitt bájtok számát, de önmagában nem változtatja meg, hogy az üzenet miként jut el a csomópontokhoz. Az EIP-706 a DevP2P üzeneteinek Snappy-tömörítését írta le; a konszenzusrétegi GossipSub-forgalomban pedig minden üzenetet külön tömörítenek. Egy 2026-os kutatási prototípus a tömörítési előzmény megőrzésével a szomszédos üzenetek közti ismétlődéseket is kihasználta. A Teku három csomópontjával végzett, 40 perces próbában a mért forgalom 1349,6 MB-ról 594,7 MB-ra csökkent. Ez egy korlátozott prototípusmérés, nem minden kliensre vagy hálózati üzenettípusra érvényes garancia [1, 2].

SPREAD: terjesztés a küldő elrejtésével. A pletykahálózatban a tranzakciót elsőként továbbító csomópont személye a közeli társai számára felismerhető lehet. Ez lehetőséget adhat a tranzakció küldőjének vagy a validátor helyének feltérképezésére. A SPREAD olyan kutatási javaslat, amely véletlen séta beiktatásával nehezítené meg a forrás azonosítását, miközben a következő ugrásokat a hálózati késleltetés figyelembevételével választja meg. A cél a küldő adatvédelmének és a gyors terjesztésnek az összehangolása. A SPREAD tervezett protokollkiegészítés; nem az Ethereum jelenlegi, általánosan használt terjesztési módja [3].

CHAMP: megbízhatóbb kapcsolatok a tranzakció-várólistához. A CHAMP nem blokkterjesztési eljárás, hanem az Ethereum végrehajtási rétegének tranzakció-várólistáját (mempool) kiszolgáló társak kiválasztását javítja. A csomópontok kapcsolati helyei korlátozottak, és a véletlenszerű társcserére azért van szükség, hogy a hálózat ne záródjon be a régi résztvevők köré, illetve ne váljon könnyűvé egy csomópont elszigetelése. A CHAMP a társcsere mellett külön védett helyeket tart fenn azoknak a társaknak, amelyek hasznos tranzakciókat juttattak el a csomóponthoz; a hasznosság egy részét on-chain, vagyis a láncon ténylegesen megjelent és véglegesedett tranzakciók alapján méri. A leírás szerint a bekerülésalapú védelem már megjelent a go-ethereum egyik kiadásában, a kéréskésleltetés szerinti védelem pedig még fejlesztés alatt állt a cikk közzétételekor [4]. Ez a megoldás a tranzakciók terjedésének megbízhatóságát támogatja, de nem garantálja, hogy minden tranzakciót beillesztenek egy blokkba.

Szegmentált terjesztés és az EIP-8411. A szegmentált terjesztés a nagy blokkot kisebb darabokra bontja. A fogadó csomópont minden darabot külön ellenőrizhet és azonnal továbbküldhet, nem kell megvárnia a teljes blokkot. Így a különböző hálózati ugrások részben átfedésbe kerülnek, a továbbítás csomópontonkénti késleltetése pedig csökkenhet. Az EIP-8411 tervezet az Ethereum végrehajtási blokkjának darabolt pletykaterjesztését írja le: a blokképítő egy Merkle-fa gyökerével kötelezi el magát a darabok mellett, a fogadók pedig a gyökérhez tartozó Merkle-bizonyítékkal ellenőrizhetik az egyes darabokat. A tervezet 64 darabot határoz meg. Fontos, hogy az EIP-8411 jelenleg tervezet, és az EIP-7732 ePBS-javaslatára épül [5].

Az EIP-8411-hez kapcsolódó kutatási mérések egy 500 csomópontos, szimulált hálózaton, 1 MiB-os blokk és 50 Mbit/s feltöltési sebesség mellett hasonlították össze az eljárás változatait. A darabolt változatok egyike a csomópontok feléig körülbelül 0,75 másodperc alatt jutott el, míg az egyben küldött üzenetnek ehhez körülbelül 5 másodpercre volt szüksége. Ezek a számok a megadott szimulációra érvényesek; nem tekinthetők a főhálózat teljesítménygaranciájának [6].

SEACAST: közös eljárás teljes és mintavételezett terjesztésre. A SEACAST neve a szegmentált, kódolt, adat-elérhetőséget mintavételező terjesztésre utal (Segmented, Encoded, Availability-sampled broadCAST). A csomópontok nem feltétlenül kapják meg a teljes adatot: a teljes blokkra igényt tartó csomópontok minden szükséges darabot kérnek, míg az adat-elérhetőség mintavételezését végzők csak a kijelölt mintákat. A darabok hibajavító kóddal egészíthetők ki, így egy csomópontnak elegendő számú, de nem feltétlenül előre meghatározott darabot kell megkapnia az adat helyreállításához. A darabokhoz külön ellenőrzési bizonyíték tartozhat. A SEACAST a teljes blokk terjesztését és az adat-elérhetőségi minták célzott kézbesítését egy általánosabb rendszerben kezeli; specifikációja és mérései kutatási munka, nem bevezetett Ethereum-protokoll [7].

FullDAS és RowDAS: adat-elérhetőség és elosztott helyreállítás. A FullDAS a nagyobb adatmennyiségek adat-elérhetőségi mintavételezésére és hálózati szétosztására kidolgozott kutatási terv. A kétdimenziós hibajavító kód az adatot sorok és oszlopok mentén rendezi el. A csomópontok csak a rájuk kiosztott részeket tárolják, és a szomszédos részekből helyi helyreállítást is végezhetnek. Ez a tervezés a mintavételezés mellett azt célozza, hogy a hálózaton belüli javítás több csomópont közreműködésével állítsa helyre a hiányzó adatot [8].

A RowDAS a PeerDAS oszlopalapú elosztását egy soralapú terjesztési tengellyel egészítené ki. A kijelölt sorhálózat tagjai megosztanák egymás között a blobok sorainak helyreállítási feladatát, így nem minden nagy kapacitású csomópontnak kellene minden hiányzó adatot külön rekonstruálnia. Az EIP-8371 ezt a megközelítést írja le; jelenleg tervezet. A hozzá közölt mérések prototípuson, mesterséges hálózati terhelés mellett készültek, és nem jelentik a főhálózaton mért teljesítményt [9, 10].

Mit oldanak meg ezek az eljárások? Az eljárások eltérő rétegekre és célokra valók. A Snappy a továbbított bájtok számát csökkentheti; a szegmentálás a nagy üzenet továbbítási késleltetését mérsékelheti; a SPREAD a küldő azonosítását nehezíti; a CHAMP a tranzakció-várólistát kiszolgáló kapcsolatok minőségét javítja; a FullDAS, a RowDAS és a SEACAST pedig adat-elérhetőségi mintavételezést, részadat-terjesztést vagy helyreállítást céloz. Ezek nem egymást helyettesítő megoldások: együttműködésük a hálózati szabályok, az ellenőrzési bizonyítékok, a sávszélesség és az esetleges hibás vagy adatot visszatartó csomópontok kezelésének összehangolását igényli.

Hivatkozások

[1] A. Nashatyrev, *Snappy with a memory: 40% less gossip traffic*, Ethereum Research, 2026. https://ethresear.ch/t/26078

[2] Ethereum Improvement Proposals, *EIP-706: DEVp2p snappy compression*. https://eips.ethereum.org/EIPS/eip-706

[3] Ethereum Research, *SPREAD: Extending GossipSub with Efficient Anonymous Dissemination*, 2026. https://ethresear.ch/t/25343

[4] Ethereum Research, *CHAMP: Hardening the Mempool with CHain-Anchored, Multi-dimensional Peer Protection*, 2026. https://ethresear.ch/t/26074

[5] Ethereum Improvement Proposals, *EIP-8411: Fast Execution Payload Broadcast*. Tervezet. https://eips.ethereum.org/EIPS/eip-8411

[6] C. Kiraly et al., *EIP-8411: what segmented payload diffusion is made of*, Ethereum Research, 2026. https://ethresear.ch/t/26025

[7] C. Kiraly, *SEACAST: fast, lean, sampled broadcast for Ethereum, with an executable spec and a formal model*, Ethereum Research, 2026. https://ethresear.ch/t/26167

[8] C. Kiraly et al., *FullDAS: towards massive scalability with 32MB blocks and beyond*, Ethereum Research, 2024. https://ethresear.ch/t/19529

[9] Ethereum Improvement Proposals, *EIP-8371: RowDAS – Distributed Blob Reconstruction*. Tervezet. https://eips.ethereum.org/EIPS/eip-8371

[10] C. Kiraly, *RowDAS (EIP-8371): Distributed Blob Reconstruction, measured*, Ethereum Research, 2026. https://ethresear.ch/t/25897