



شبکه‌ی بیت‌کوین

معماری شبکه‌ی همتا-به-همتا

بیت‌کوین به صورت یک شبکه با معماری همتا-به-همتا (peer-to-peer: P2P) روی اینترنت ساخته شده است. در عبارت دیگر، توپولوژی این شبکه از لحاظ موقعیت همسان (همتا) هستند، و هیچ گرهی «خاص» نیست و بر دیگران بزرگ ندارد؛ به عبارت دیگر، توپولوژی این شبکه «تخت» است، هیچ سلسله مراتبی وجود ندارد، و هر گره تعدادی اتصال پاده گرهای دیگر دارد. در این معماری وظایف شبکه بین همه گرهاتوزیع می‌شود؛ هیچ گرهی نقش «سروریس دهنده» ندارد و شبکه غیر مرکز است. در یک شبکه‌ی P2P، گره‌ها هم‌مان هم به دیگران سرویس می‌دهند و هم از دیگران سرویس می‌گیرند؛ در واقع، انگیزه‌ی گره‌ها از مشارکت در شبکه «بده-بستان متقابل» است. شبکه‌های P2P ذاتاً انعطاف‌پذیر، غیر مرکز و یاز هستند. بر جسته ترین نمونه از معماری P2P همان اینترنت در روزهای اولیه ظهور آن است که گره‌های این پروتکل اینترنت (همچنان ماهیت توپولوژی-تخت خود را حفظ کرده است. جدای از بیت‌کوین، بزرگترین و موفق‌ترین کارهای فناوری‌های P2P اشتراک‌گذاری فایل و موسیقی است، که نپستر (شبکه‌ی اشتراک موسیقی) پیشگام آن و بیت‌کوین (شبکه‌ی اشتراک فایل) جدیدترین و موفق‌ترین نمونه‌های آن هستند.

معماری P2P شبکه‌ی بیت‌کوین چیزی بیش از یک انتخاب فناوری است. بیت‌کوین اساساً به عنوان یک بمن نقدینگی دیجیتال P2P طراحی شد، و معماری این شبکه علاوه بر آن که انعکاسی از خصوصیات کلیدی آن محسوب می‌شود، ویزگی بنیادی آن را نیز تشکیل می‌دهد. عدم مرکز یک اصل بنیادی در طراحی بیت‌کوین است که فقط یک شبکه‌ی P2P تخت، غیر مرکز، و مبتنی بر اجماع قابل دستیابی است.

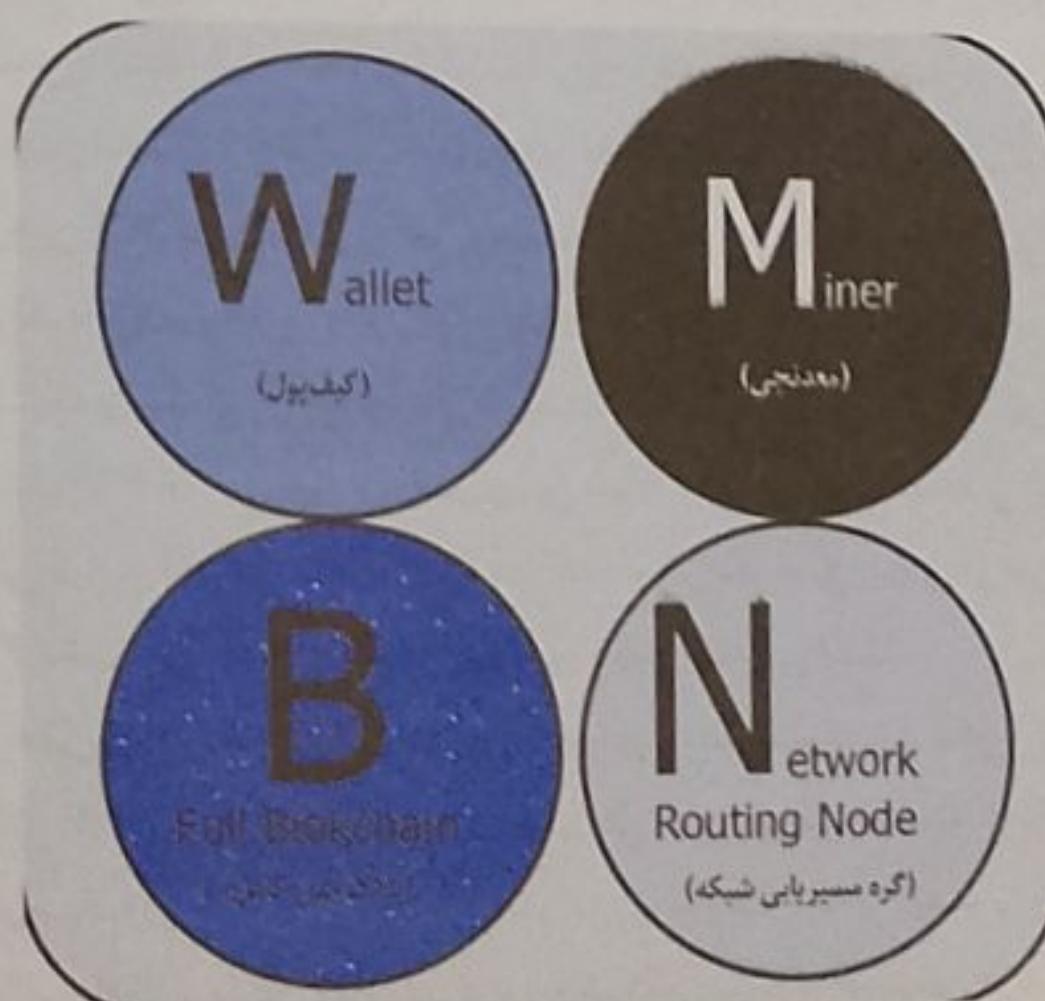
به مجموعی گره‌هایی که پروتکل P2P بیت‌کوین را اجرامی کنند، شبکه‌ی بیت‌کوین (bitcoin network) گفته می‌شود. علاوه بر پروتکل P2P بیت‌کوین، پروتکل‌های دیگر مثل استراتوم نیز هستند که برای معدنکاوی و استخراج بیت‌کوین و کیف‌پول‌های سبک وزن یا موبایل به کار می‌روند. این پروتکل‌های جانبی توسط سرویس دهنده‌های مسیریاب دروازه ارائه می‌شوند که با استفاده از پروتکل P2P بیت‌کوین به شبکه‌ی بیت‌کوین دسترسی دارند و این شبکه را به گره‌های که پروتکل‌های دیگر را اجرا می‌کنند، گسترش می‌دهند. برای مثال، سرویس دهنده‌های استراتوم گره‌های استخراج کننده‌ی استراتوم را از طریق پروتکل استراتوم را به شبکه‌ی اصلی بیت‌کوین متصل کرده و بین پروتکل استراتوم و پروتکل P2P

بیت‌کوین یک پُل ارتباطی برقرار می‌کنند. وقتی از اصطلاح «شبکه‌ی بیت‌کوین گسترش یافته» استفاده می‌کنیم، منظورمان کل شبکه‌ای است که پروتکل P2P بیت‌کوین، پروتکل‌های استخراج-گروهی، پروتکل استراتوم، و دیگر پروتکل‌های متصل‌کننده‌ی اجزای سیستم بیت‌کوین را در بر می‌گیرد.

انواع گره و نقش‌های آن

هر چند گره‌های شبکه‌ی P2P بیت‌کوین همسان (همتا) هستند، ولی بسته به وظایف و کارکردهایی که انجام می‌دهند، نقش‌های مختلفی به خود می‌گیرند. یک گره بیت‌کوین (bitcoin node) مجموعه‌ای از چهار کارگرد است: مسیریابی، پایگاه داده‌ی بلاک‌چین، استخراج (معدنکاری)، و سرویس کیف‌پول. همان طور که در شکل ۱-۸ نشان داده شده، یک گره کامل همه‌ی این چهار کارگرد را انجام می‌دهد. تمامی گره‌های بیت‌کوین در کارگرد مسیریابی مشارکت می‌کنند، و می‌توانند کارکردهای دیگر را هم داشته باشند. تمام گره‌ها اعتبارسنجی و انتشار (توزیع) تراکنش‌ها و بلاک‌ها را انجام می‌دهند، و دیگر گره‌های شبکه را شناسایی کرده و با آنها اتصال (ارتباط P2P) برقرار می‌کنند. در شکل ۱-۸، کارگرد مسیریابی با دایره‌ای که حرف «N» در آن نقش بسته، مشخص شده است.

برخی گره‌ها یک کپی کامل و بهروز از بلاک‌چین نگه می‌دارند؛ به این قبیل گره‌ها گره کامل (full node) گفته می‌شود. یک گره کامل می‌تواند به طور مستقل و بدون نیاز به همکاری گره‌های دیگر هر تراکنشی را اعتبارسنجی (verify) کند. برخی گره‌ها فقط یک زیرمجموعه از بلاک‌چین رانگه می‌دارند و با استفاده از روشی موسوم به اعتبارسنجی پرداخت ساده (simplified payment verification)، یا SPV، تراکنش‌هارا اعتبارسنجی می‌کنند؛ به اینها گره SPV یا گره سبک وزن گفته می‌شود. در شکل ۱-۸، کارگرد نگهداری و بهروزرسانی بلاک‌چین با دایره‌ای که حرف «B» در آن نقش بسته، نشان داده شده است. گره‌های SPV این دایره با حرف «B» را ندارند، که نشان می‌دهد که نیازی کاملاً از بلاک‌چین در آنها نگهداری نمی‌شود (شکل ۳-۸ را بینید).



شکل ۱-۸ یک گره شبکه‌ی بیت‌کوین با تمامی چهار کارگرد آن: کیف‌پول، معدنچی، پایگاه داده‌ی بلاک‌چین کامل، و مسیریاب شبکه.

گره‌های معنچی با استفاده از سخت‌افزارهای تخصصی (برای حل الگوریتم «اثبات-کار»، PoW) برای ایجاد بلاک‌های جدید با یکدیگر رقابت می‌کنند. برخی گره‌های معنچی گره کامل نیز هستند (یعنی یک کپی کامل و بهروز از بلاک‌چین نگه می‌دارند)، ولی برخی دیگر گره‌های سبک وزنی هستند که در فرآیندی موسوم به استخراج گروهی مشارکت می‌کنند و برای دسترسی به امکانات گره کامل به یک سرویس دهنده‌ی گروهی وابسته هستند. در شکل ۱-۸، کارکرد معنکاوی و استخراج بیت‌کوین با دایره‌ای که حرف «M» در آن نقش بسته، نمایش داده شده است.

کیف‌پول می‌تواند جزوی از یک گره کامل باشد؛ مشتری‌های بیت‌کوین که روی کامپیوترهای رومیزی اجرا می‌شوند، معمولاً کیف‌پول هم دارند. اما امروزه بخش بزرگی از کیف‌پول کاربران از نوع گره SPV است، به خصوص آنها بی که روی دستگاه‌های باقدرت پردازش کمتر (مثل تلفن‌های هوشمند) اجرا می‌شوند. در شکل ۱-۸، کارکرد کیف‌پول با دایره‌ای که حرف «W» در آن نقش بسته، مشخص شده است.

شبکه‌ی بیت‌کوین گسترش یافته

شکل ۲-۸ رايج‌ترین انواع گره در «شبکه‌ی بیت‌کوین گسترش یافته» را نشان می‌دهد.

شبکه‌ی اصلی بیت‌کوین، که پروتکل P2P بیت‌کوین را اجرا می‌کند، از حدود ۵۰۰۰ تا ۸۰۰۰ گره شونده تشکیل می‌شود که ویرایش‌های مختلفی از مشتری مرجع بیت‌کوین (هسته‌ی بیت‌کوین) را اجرا می‌کنند؛ چند صد گره دیگر هم در این شبکه حضور دارند که پیاده‌سازی‌های دیگری از پروتکل P2P بیت‌کوین (مانند بیت‌کوین کلامیک، bcoin، btcd، LibbitcoinJ، BitcoinJ، و Libcoin) را اجرا می‌کنند. درصد کوچکی از این گره‌های شبکه‌ی بیت‌کوین نامحدود، P2P بیت‌کوین گره استخراج (یا معنکاوی) نیز هستند، که تراکنش‌های اعتبارسنجی می‌کنند، بلاک‌های جدید تولید می‌کنند، و در فرآیند استخراج بیت‌کوین‌های جدید رقابت می‌کنند. تعداد زیادی از شرکت‌های بزرگ با راه‌اندازی گره‌های کیف‌پول هم ندارند. این گره‌ها به عنوان مسیریاب‌های لبه‌ی شبکه عمل می‌کنند، و اجازه می‌دهند سرویس‌های مختلف دیگر (تعییر ارز، کیف‌پول، کاوشگر بلاک، پردازش پرداخت‌های تجاری) روی آنها ساخته شوند.

شبکه‌ی بیت‌کوین گسترش یافته (extended bitcoin network) به شبکه‌ای گفته می‌شود که علاوه بر شبکه‌ی اصلی بیت‌کوین [مجموعه‌ی گره‌هایی که پروتکل P2P بیت‌کوین را اجرا می‌کنند]، گره‌های دیگر که پروتکل‌های تخصصی را اجرا می‌کنند، را نیز در بر می‌گیرد. تعدادی سرویس دهنده موسوم به سرویس دهنده‌ی گروهی (pool server) و دروازه‌ی پروتکل (protocol gateway) نیز وجود دارند که به شبکه‌ی P2P بیت‌کوین هستند و گره‌های اجراینده‌ی پروتکل‌های دیگر را به آن متصل می‌کنند. این گره‌ها اکثرًا گره‌های استخراج گروهی (pool mining) [فصل ۱۰ را بینید] و مشتری‌های کیف‌پول سبک وزن هستند، که بدون یک کپی کامل از بلاک‌چین کار می‌کنند. در شکل ۳-۸ شبکه‌ی بیت‌کوین گسترش یافته و انواع گره‌ها و پروتکل‌های مختلف آن را می‌بینید.

شبکه‌ی بازپخش بیت‌کوین

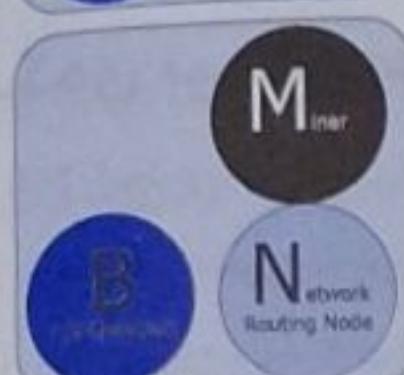
هر چند شبکه‌ی P2P بیت‌کوین نیازهای کلی طیف وسیعی از انواع مختلف گره را تأمین می‌کند، زمان اختفای آن [مدت زمانی که طول می‌کشد تا شبکه به یک درخواست پاسخ دهد] برای نیاز تخصصی گره‌های استخراج بیت‌کوین بیش از حد زیاد است. معنچیان بیت‌کوین درگیر یک رقابت حساس هستند تا مساله‌ی PoW (اثبات-کار) را حل کنند و بلاک‌چین

**مشتری مرجع (هسته‌ی بیت‌کوین)**

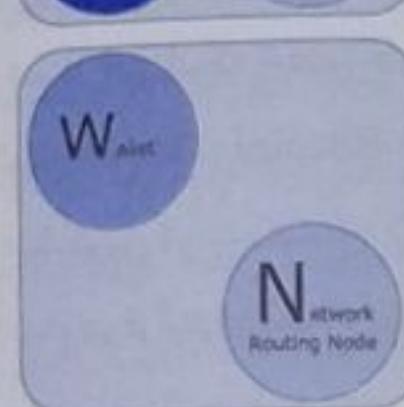
شامل کیفیبول، معدنجی، پایگاه داده‌ی بلاکچین کامل، و گره مسیریابی شبکه روی شبکه P2P بیت‌کوین.

**گره بلاکچین کامل**

شامل پایگاه داده‌ی بلاکچین کامل، و گره مسیریابی شبکه روی شبکه P2P بیت‌کوین

**معدنجی انفرادی (تکرو)**

شامل کارگردان معدنکاوی (استخراج) به اصلاحه‌ی یک کپی کامل از پایگاه داده‌ی بلاکچین و گره مسیریابی شبکه P2P بیت‌کوین.

**کیفیبول سبکوزن (SPV)**

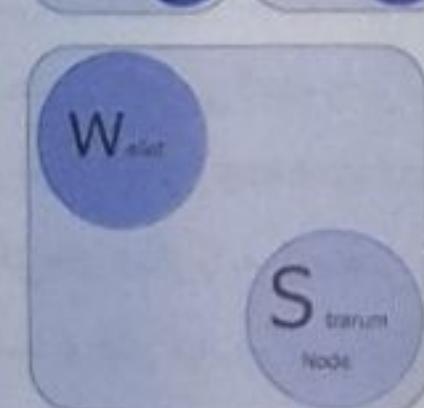
شامل کیفیبول و گره مسیریابی روی شبکه P2P بیت‌کوین، بدون یک کپی از بلاکچین

**سرویس‌دهنده‌های پروتکل گروهی**

مسیریاب‌های دروازه برای اتصال شبکه P2P بیت‌کوین به گره‌هایی که پروتکل‌های دیگر را اجرا می‌کنند، مثل گره‌های استخراج-گروهی یا گره‌های استراتوم.

**گره‌های استخراج (معدنکاوی)**

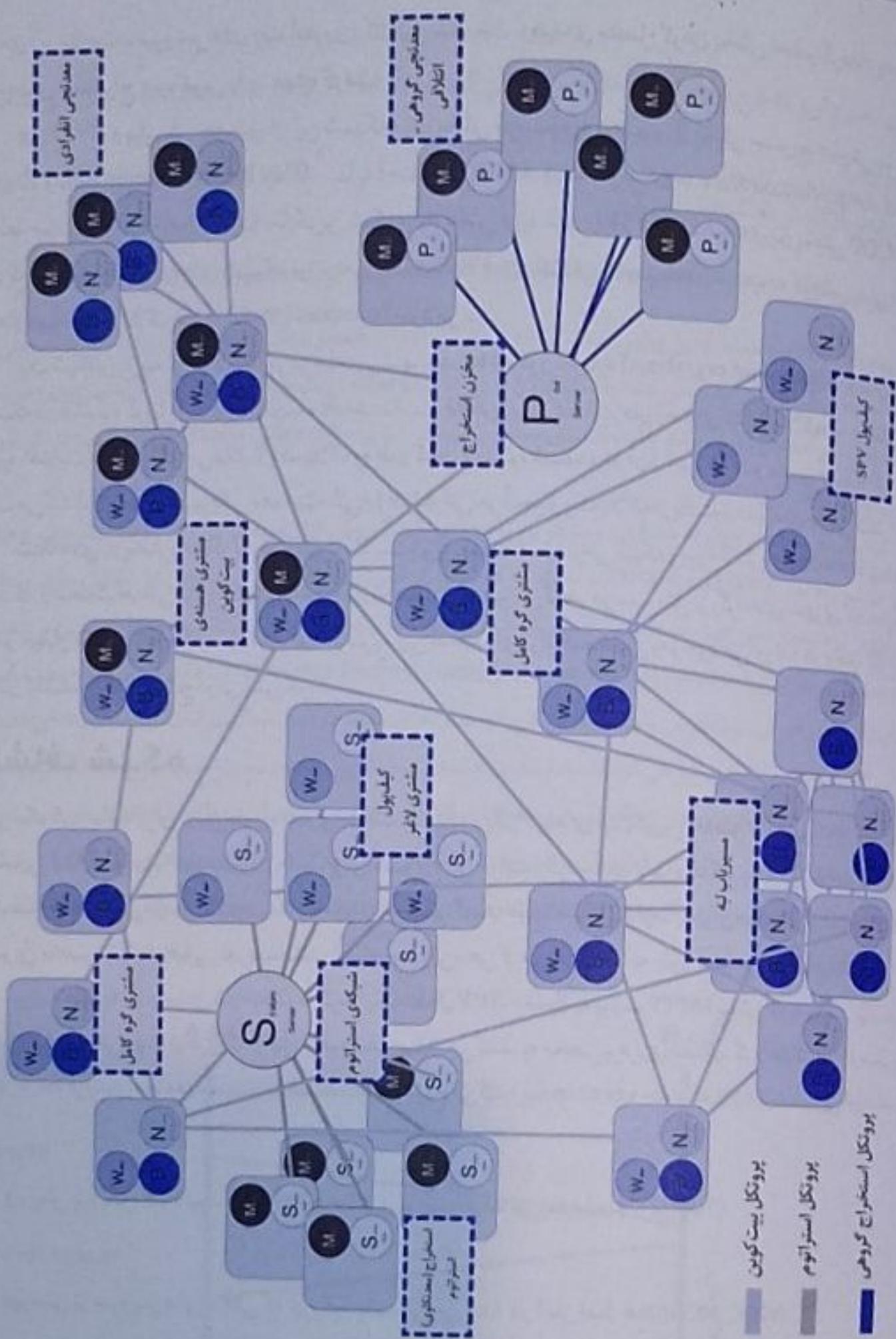
شامل کارگردان استخراج، بدون یک کپی از بلاکچین، با گره پروتکل استراتوم (S) یا گره پروتکل استخراج (P).

**کیفیبول سبکوزن (SPV) استراتوم**

شامل کیفیبول و گره مسیریابی شبکه روی پروتکل استراتوم، بدون یک کپی از بلاکچین

شکل ۲-۸ انواع گره در شبکه‌ی بیت‌کوین گسترش یافته.

را توسعه دهند [فصل ۱۰ را ببینید]. اگر چه معدنجیان بیت‌کوین معمولاً به صورت گروهی در این رقابت شرکت می‌کنند، ولی بایستی فاصله‌ی زمانی بین انتشار یک بلاک برنده و شروع دور بعدی مسابقه را به حداقل برسانند. در عملیات استخراج بیت‌کوین، زمان اختلافی شبکه (network latency) تأثیر مستقیم بر حاشیه‌ی سود دارد.



شکل ۲-۸ ارتباط انواع گره، دروازه و پروتکل در شبکه‌ی بیت‌کوین گسترش‌یافته.

یک شبکه‌ی بازپخش بیت‌کوین (Bitcoin Relay Network) شبکه‌ای است که نلاش می‌کند زمان اختفای انتقال بلاک‌ین معدنچیان را به حداقل برساند. اولین شبکه‌ی بازپخش بیت‌کوین [در <http://www.bitcoinrelaynetwork.org/>] در سال ۲۰۱۵ توسط یک برنامه‌نویس هسته‌ی بیت‌کوین به نام مت کورا (Metacoin) ابداع شد تا همزمان سازی سریع بلاک‌های بین گره‌های معدنچی را بازمان اختفای پسیار پایین ممکن سازد. این شبکه که از چندین گره تخصصی پراکنده در سرتاسر دنیا به میزبانی

شبکه‌ی ذیرساخت «سرمیس‌های وب آمازون» تشکیل شده بود، وظیفه‌ی متصل کردن بخش اعظم گره‌های معدنچی و گروه‌های استخراج بیت‌کوین را بر عهده گرفت.

در ۲۰۱۶، ویرایش جدیدی از این شبکه‌ی بازپخش موسوم به موتور بازپخش سریع اینترنتی بیت‌کوین (Fast Internet Bitcoin Relay Engine، یا به اختصار FIBRE، آدرس <http://bitcoinfibre.org>)، که آن هم توسط ملت کورالو ساخته شده بود] جایگزین شبکه‌ی بازپخش اولیه شد. FIBRE یک شبکه‌ی بازپخش UDP-محور است که بلاک‌های گره‌های شبکه توزیع می‌کند. FIBRE برای کاهش حجم داده و در نتیجه کاهش زمان انتقالی شبکه از بهینه‌سازی بلاک فشرده (compact block) سود می‌برد.

یک شبکه‌ی بازپخش دیگر (که در مرحله‌ی پیشنهاد است) فالکون نام دارد [<http://www.falcon-net.org/about>]؛ این شبکه در دانشگاه کورنل [نیویورک] توسعه داده شده است. فالکون برای کاهش زمان انتقالی انتشار بلاک‌ها، به جای روش «ذخیره-هدایت» [که منتظر می‌ماند تا یک بلاک به طور کامل وارد گردد و در آن ذخیره شود، و سپس آن را به گره بعدی هدایت می‌کند]، از روش «مسیریابی-هدایت-آنی» [انتشار آنی هر آنچه از یک بلاک دریافت شده] استفاده می‌کند.

شبکه‌های بازپخش جایگزین شبکه‌ی P2P بیت‌کوین نیستند؛ آنها نوعی شبکه‌ی رو-گذاری (overlay network) هستند که ارتباطات اضافی برای گره‌های بانیازهای تخصصی فراهم می‌کنند. درست مثل بزرگراه‌های شهری که جایگزین کوچه و خیابان نمی‌شوند، بلکه مسیرهای کوتاه و سریع بین نقاط دارای ترافیک سنگین به وجود می‌آورند؛ در واقع، اگر کوچه و خیابان نباشد، بزرگراه به هیچ دردی نمی‌خورد.

اکتشاف شبکه

وقتی یک گره جدید برای اولین بار راه‌اندازی می‌شود، باید دیگر گره‌های بیت‌کوین حاضر در شبکه را کشف و شناسایی کند تا بتواند با آنها مشارکت کند. برای شروع این فرآیند، گره جدید باید دستکم یک گره حاضر در شبکه را کشف کرده و به آن متصل شود. مکان جغرافیایی سایر گره‌ها هیچ اهمیتی ندارد؛ به بیان دیگر، توپولوژی شبکه بیت‌کوین به صورت جغرافیایی تعریف نشده است. بنابراین، هر گرهی می‌تواند به طور تصادفی انتخاب شود.

برای اتصال به یک همتای شناخته شده، گره‌ها یک اتصال TCP، معمولاً به پورت ۸۳۳۳ (پورتی که به پورت بیت‌کوین معروف است)، یا هر پورت دیگری که از قبل معلوم شده، برقرار می‌کنند. به محض برقراری اتصال، گره آغازکننده با ارسال یک پیام *version* فرآیندی موسوم به دستداد (handshake) را شروع می‌کند؛ پیام *version* حاوی اطلاعات شناسایی اولیه است:

nVersion

ویرایش پروتکل P2P بیت‌کوین که این مشتری با آن «حرف می‌زند» (مثلاً 70002).

nLocalServices

فهرستی از سرویس‌های محلی که این گره پشتیبانی می‌کند؛ در آغاز فقط *NODE_NETWORK*.

nTime

زمان فعالی.

addrYou

آدرس IP گره دور از دید این گره.

addrMe

آدرس IP گره محلی از دید خود این گره.

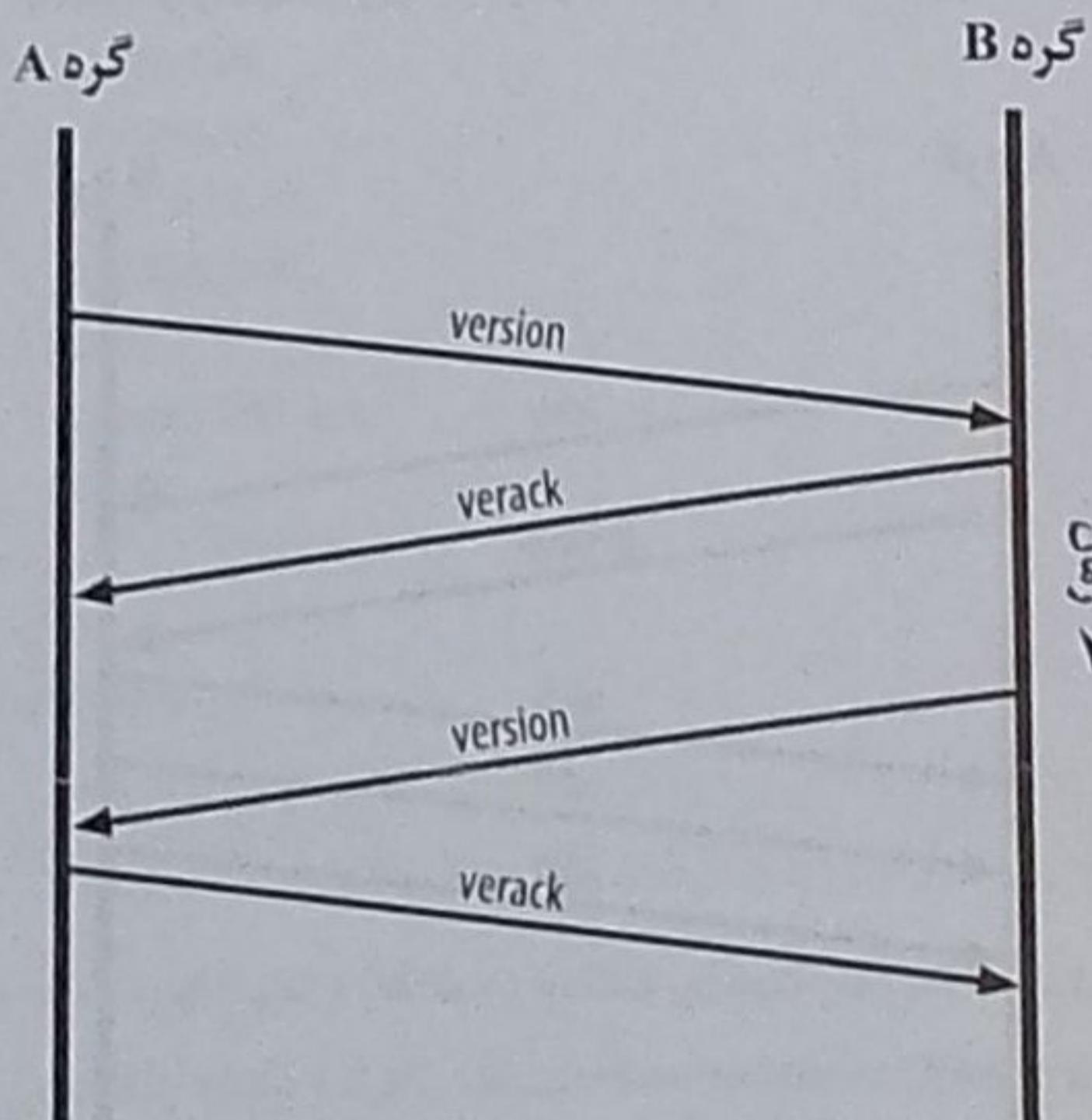
زیر-ویرایش، که نوع نرم افزار در حال اجرا در این گره (مثالاً /Satoshi:0.9.2.1/) را نشان می‌دهد.

BestHeight
ارتفاع بلاک در بلاک چین این گره.

[برای دیدن نمونه‌ای از پیام `version` به <http://bit.ly/1qlsC7w> بروید.] آنگاه کنید. افرآیند دستدار را در شکل ۴-۸ مشاهده می‌کنید.

پیام `version` همیشه اولین پیامی است که از یک همتای (گره) به همتای دیگر فرستاده می‌شود. وقتی گره محلی یک پیام `version` از گره دور دریافت می‌کند، بانگاه کردن به فیلد `nVersion` آن متوجه می‌شود آیا با آن همتای دور سازگار هست یا خیر. اگر همتای دور سازگار باشد، همتای محلی با ارسال یک پیام `verack` پیام `version` آن را تصدیق کرده و یک اتصال برقرار می‌کند.

اما یک گره جدید چگونه همتایان بیت‌کوین را پیدا می‌کند؟ روش اول استفاده از نوع خاصی سرویس دهنده‌ی DNS، موسوم به بذر DNS (DNS seed)، است که فهرستی از آدرس IP گره‌های بیت‌کوین در اختیار دارد. برخی از این سرویس‌دهنده‌ها حتی فهرست آدرس IP استاتیک گره‌های شتونده را نیز ارائه می‌کنند. اغلب این سرویس‌دهنده‌ها از پیاده‌سازی خاصی از برنامه‌ی BIND (Berkeley Internet Name Daemon) استفاده می‌کنند که به کمک یک زبان اینترنتی یا یک گره بیت‌کوین باسابقه (گرهی که مدت‌های مديدة در حال فعالیت است) مجموعه‌ای تصادفی از آدرس IP گره‌های بیت‌کوین بر می‌گرداند. مشتری هسته‌ی بیت‌کوین حاوی نام پنج سرویس دهنده از این بذرها DNS است. تنوع مالکیت و تنوع پیاده‌سازی بذرها DNS مختلف باعث شده تا فرآیند اتصال اولیه به شبکه می‌باشد. بیت‌کوین به سطح بالایی از ثبات و اطمینان دست یابد. چگونگی استفاده از سرویس دهنده‌های بذر DNS در مشتری هسته‌ی بیت‌کوین به وسیله‌ی گزینه‌ی dnsseed-کنترل می‌شود؛ این گزینه در حالت پیش‌فرض مقدار ۱ (استفاده از بذر DNS) دارد.

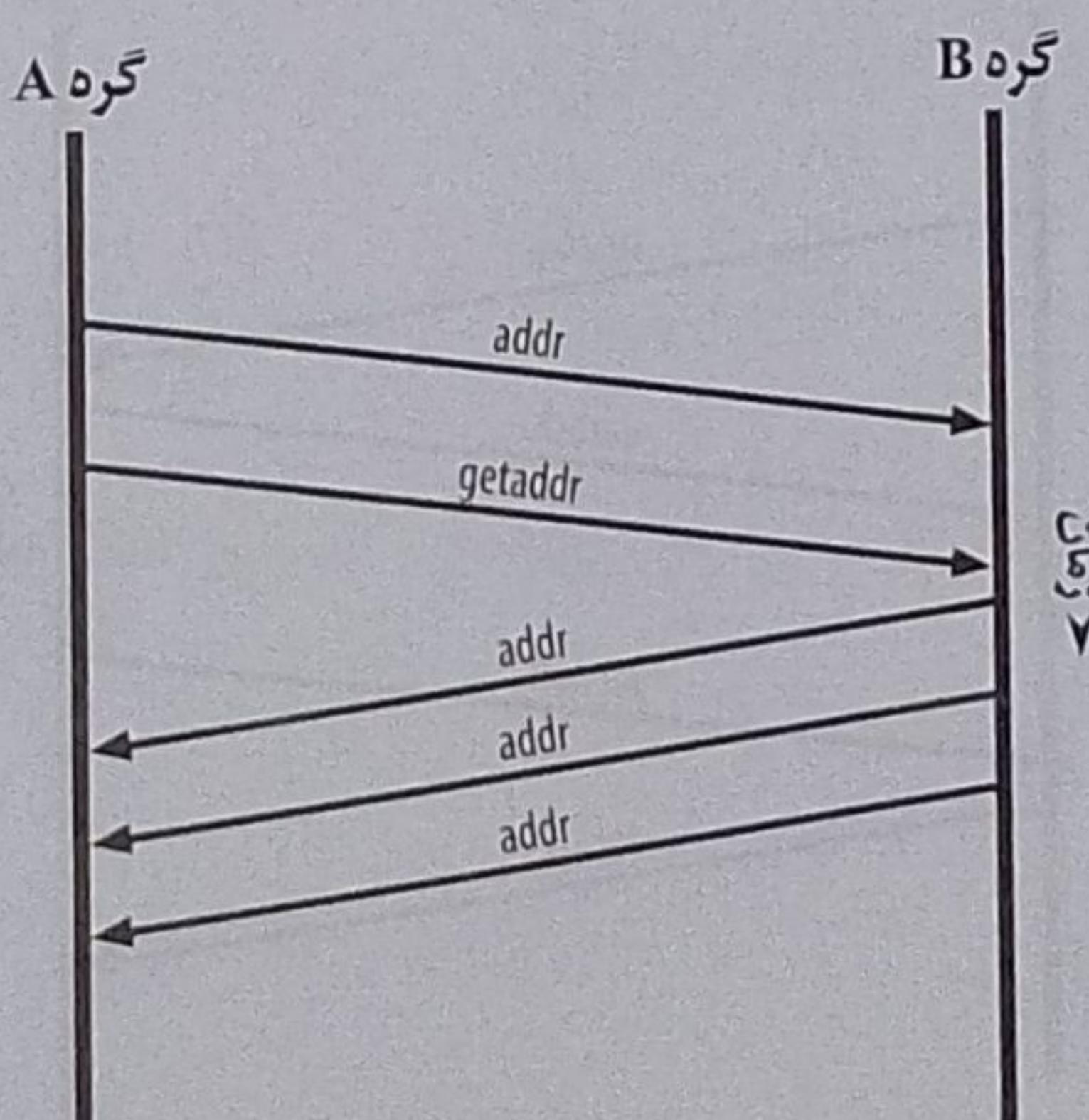


شکل ۴-۸ دستداد آغازین بین دو همتا.

در روش دیگر، به گرهی که برای اولین بار می‌خواهد به شبکه‌ی بیت‌کوین متصل شود، آدرس IP حداقل یک گره بیت‌کوین داده می‌شود؛ بعد از این اتصال اولیه، گره مزبور می‌تواند اطلاعات لازم برای شناسایی گره‌های بعدی و اتصال به آنها را از این گره به دست آورد. این کار با استفاده از آرگومان خط-فرمان `seednode`-انجام می‌شود. بعد از استفاده از این بذر برای شناسایی اولیه، این مشتری ارتباط خود با آن گره را قطع کرده و به گره‌هایی که به تازگی شناسایی کرده است، متصل می‌شود.

همین‌که یک یا چند اتصال برقرار شد، گره جدید یک پیام `addr` حاوی آدرس IP خود را به همسایگانش می‌فرستد. آن همسایه‌ها به توبه‌ی خود این پیام `addr` را به همسایه‌های دیگر می‌فرستند، تا مطمئن شوند گره جدید هر چه بیشتر و بهتر در شبکه شناخته می‌شود. علاوه بر آن، گره جدید می‌تواند با ارسال پیام `getaddr` به همسایگان خود، فهرستی از آدرس IP همتایان دیگر در شبکه را از آنها طلب کند. بدین ترتیب، یک گره می‌تواند ضمن یافتن همتایان شبکه، حضور خود را نیز اعلام کند تا دیگران بتوانند وی را پیدا کنند. این فرآیند که به پروتکل اکتشاف آدرس معروف است، در شکل ۵-۸ نشان داده شده است.

اگر یک گره می‌خواهد مسیرهای متنوعی به سمت شبکه‌ی بیت‌کوین داشته باشد، باید دستکم به چند همتای متفاوت متصل شود. این مسیرها دائمی و قابل اطمینان نیستند [گره‌ها از شبکه قطع شده و دوباره به آن متصل می‌شوند]، بنابراین گره جدید باید به اکتشاف گره‌های دیگر ادامه دهد تا بتواند ضمن حفظ اتصال خود به شبکه‌ی بیت‌کوین، [همان طور که خودش زمانی از دیگران برای اتصال به شبکه کمک گرفت] در این راه به گره‌های جدید دیگر کمک کند. برای اتصال اولیه به شبکه فقط یک اتصال کفايت می‌کند، چون همان گره اول این گره جدید را به همتایان خود معرفی کرده و زمینه را برای شناسایی بیشتر آن در شبکه فراهم می‌کند. همچنین متصل شدن به بیش از پنج یا شش گره غیرضروری است و نتیجه‌ای جز اتلاف منابع شبکه نخواهد داشت. بعد از راه‌اندازی اولیه، گره جدید آخرین اتصال‌های موفق خود را به سرعت برقرار کند. اگر در راه‌اندازی دوباره هیچ یک از آن گره‌های همتا به درخواست اتصال این گره پاسخ نداده، گره جدید مجبور است، با استفاده از پیام `version`، فرآیند راه‌اندازی اولیه را از نو اجرا کند.



شکل ۵-۸ انتشار و اکتشاف آدرس.

برای دیدن فهرست اتصال‌های همتا در گرهی که در حال اجرای مشتری هسته‌ی بیت‌کوین است، می‌توانید از فرمان `getpeerinfo` استفاده کنید:

```
$ bitcoin-cli getpeerinfo
```

```
[
```

```

  [
    {
      "addr" : "85.213.199.39:8333",
      "services" : "00000001",
      "lastsend" : 1405634126,
      "lastrecv" : 1405634127,
      "bytessent" : 23487651,
      "bytesrecv" : 138679099,
      "conntime" : 1405021768,
      "pingtime" : 0.00000000,
      "version" : 70002,
      "subver" : "/Satoshi:0.9.2.1/",
      "inbound" : false,
      "startingheight" : 310131,
      "banscore" : 0,
      "syncnode" : true
    },
    {
      "addr" : "58.23.244.20:8333",
      "services" : "00000001",
      "lastsend" : 1405634127,
      "lastrecv" : 1405634124,
      "bytessent" : 4460918,
      "bytesrecv" : 8903575,
      "conntime" : 1405559628,
      "pingtime" : 0.00000000,
      "version" : 70001,
      "subver" : "/Satoshi:0.8.6/",
      "inbound" : false,
      "startingheight" : 311074,
      "banscore" : 0,
      "syncnode" : false
    }
  ]

```

برای کنار گذاشتن مدیریت خودکار گره‌های همسایه (همتاها) و تعیین فهرست آدرس‌های IP دلخواه می‌توانید از گزینه‌ی خط-فرمان `<seednode=IPAddress>`-استفاده کرده و یک یا چند آدرس IP را مشخص کنید. اگر از این گزینه استفاده شود، گره بیت‌کوین فقط به همان آدرس‌های IP مشخص شده متصل خواهد شد و دیگر تلاشی برای اکشاف و شناسایی خودکار گره‌های همتا نخواهد کرد.

اگر روی یک اتصال ترافیک وجود نداشته باشد، گره‌های دو طرف این اتصال تلاش می‌کنند با ارسال منظم یک پیام آن را برقرار نگه دارند. اگر یک گره برای مدت بیش از ۹۰ دقیقه به این پیام‌ها پاسخ ندهد، فرض برآن گذاشته می‌شود که اتصال آن از شبکه قطع شده و همتای جدیدی جستجو می‌شود. با این روش، شبکه‌ی بیت‌کوین به طور پویا خود را با قطع و وصل گره‌ها و اختلالات شبکه سازگار می‌کند، و قادر است به صورت طبیعی و بدون هر گونه کنترل مرکزی بزرگ و کوچک شود.

گره کامل

گره کامل (full node) گرهی است که یک نسخه‌ی کامل و پیروز از بلاک‌چین با تمامی تراکنش‌های آن نگه می‌دارد. اگر بخواهیم دقیق تر باشیم، باید آن را «گره کامل بلاک‌چین» بنامیم. در سال‌های اولیه‌ی بیت‌کوین، همه‌ی گره‌ها گره کامل بودند، ولی امروزه فقط گره‌هایی که مشتری هستند بیت‌کوین را اجرا می‌کنند، گره کامل بلاک‌چین هستند. با این حال، در دو سال گذشته، انواع جدیدی از مشتری بیت‌کوین وارد بازار شده‌اند که یک کمی کامل از بلاک‌چین نگه نمی‌دارند و فقط یک مشتری بیت‌کوین سیکریزن را اجرا می‌کنند در قسمت بعد به طور مفصل درباره‌ی این نوع گره‌ها صحبت خواهیم کرد.

گره‌های کامل بلاک‌چین یک نسخه‌ی کامل و پیروز از بلاک‌چین بیت‌کوین با تمام تراکنش‌های آن نگه می‌دارند؛ این گره‌ها قادر هستند به طور مستقل تمام تراکنش‌های بیت‌کوین را از اولین بلاک (موسم به بلاک زاینده) تا آخرین بلاک تا حتمتله در شبکه بازمی‌گردانند. یک گره کامل بلاک‌چین می‌تواند هر تراکنش را باید از کتابه متبع یا کمک (اطلاعات) گره‌های دیگر به طور مستقل و با مسئولیت کامل اعتبارسنجی کند. گره‌های کامل بلاک‌چین برای اعتبارسنجی تراکنش‌های جدید و ثبت آنها در بلاک‌چین محلی خود فقط به دریافت پیروزمانی درباره‌ی بلاک‌های جدید از شبکه بیت‌کوین نیاز دارند.

راه‌اندازی و ادغامی یک گره کامل بلاک‌چین بالاترین سطح از تجربه‌ی بیت‌کوین است: اعتبارسنجی مستقل تمام تراکنش‌ها، بدون نیاز به کمک، اعتماد پاهمکاری سیستم‌های دیگر. فهمیدن این که در حال اجرای چه نوعی از هسته‌ی بیت‌کوین هست ساده است. یک گره کامل به بیش از ۲۵۰ کیگابایت نصایر دیگر برای ذخیره‌سازی بلاک‌چین نیاز دارد. اگر نسخه‌ی بیت‌کوین شما فضای زیادی اشغال کرده و برای هماهنگ شدن با شبکه به دویا سه روز زمان نیاز داشته، بدانید که در حال اجرای یک گره کامل هستید. این بهایی است که باید برای استقلال کامل و رهایی از سلطه‌ی اقتدار مرکزی پردازید. پاده‌سازی‌های مختلفی از مشتری بلاک‌چین کامل بیت‌کوین وجود دارد که بازبان‌های برنامه‌نویسی و معماری‌های نرم‌افزار مختلف توسعه داده شده‌اند. با این حال، محبوب‌ترین پاده‌سازی بلاک‌چین کامل همان «مشتری هسته‌ی بیت‌کوین» مرجع است، که به تمام مشتری‌سازی‌های نیز شاخته می‌شود. بیش از ۷۵ درصد از گره‌های شبکه بیت‌کوین و پراش‌های مختلف هسته‌ی بیت‌کوین را اجرا می‌کنند. این مشتری با عنوان «Satoshi» در فیلد `subver` (زیر-ویرایش) پیام `version` مخصوص می‌شود، مثل رشته‌ی `Satoshi:0.8.6/Satoshi` که در خروجی فرمان `getpeerinfo` مثال قبل دیدید.

مبادله‌ی «دفتر دارایی»

اولین کاری که یک گره کامل بعد از اتصال به گره‌های همتا انجام می‌دهد، تلاش برای ساختن یک بلاک‌چین کامل است. اگر این یک گره کاملاً جدید باشد، فقط یک بلاک را می‌شناسد: بلاک زاینده. بلاک زاینده (genesis block) به صورت ثابت در نرم‌افزار مشتری قرار داده شده است. این گره جدید با شروع از بلاک ۰ (#(بلاک زاینده)، صدها هزار بلاک را بازگیری می‌کند تا بلاک‌چین خود را با شبکه همگام کرده و یک بلاک‌چین کامل داشته باشد.

فرآیند همگام‌سازی بلاک‌چین با پیام `version` شروع می‌شود، چون این پیام حاوی فیلد `BestHeight` است که ارتفاع بلاک‌چین (تعداد بلاک‌ها) در یک گره را نشان می‌دهد. این گره با گرفتن پیام `version` از همتاها مختلف، می‌تواند

متوجه شود که آنها چند بلاک دارند، و سپس این تعداد را با بلاک‌هایی که در بلاک‌چین خودش دارد، مقایسه می‌کند. گره‌های همتایک پیام `getblocks` مبادله می‌کنند که حاوی درهم (اثرانگشت) بالاترین بلاک در بلاک‌چین محلی آنها است. یکی از گره‌ها بلافاصله متوجه می‌شود که درهم دریافت شده متعلق به یکی از بلاک‌هایی است که در بالای بلاک‌چین وی قرار ندارد، بلکه متعلق به یک بلاک قدیمی است، بنابراین نتیجه می‌گیرد که بلاک‌چین محلی او بزرگتر از بلاک‌چین همسایگانش است.

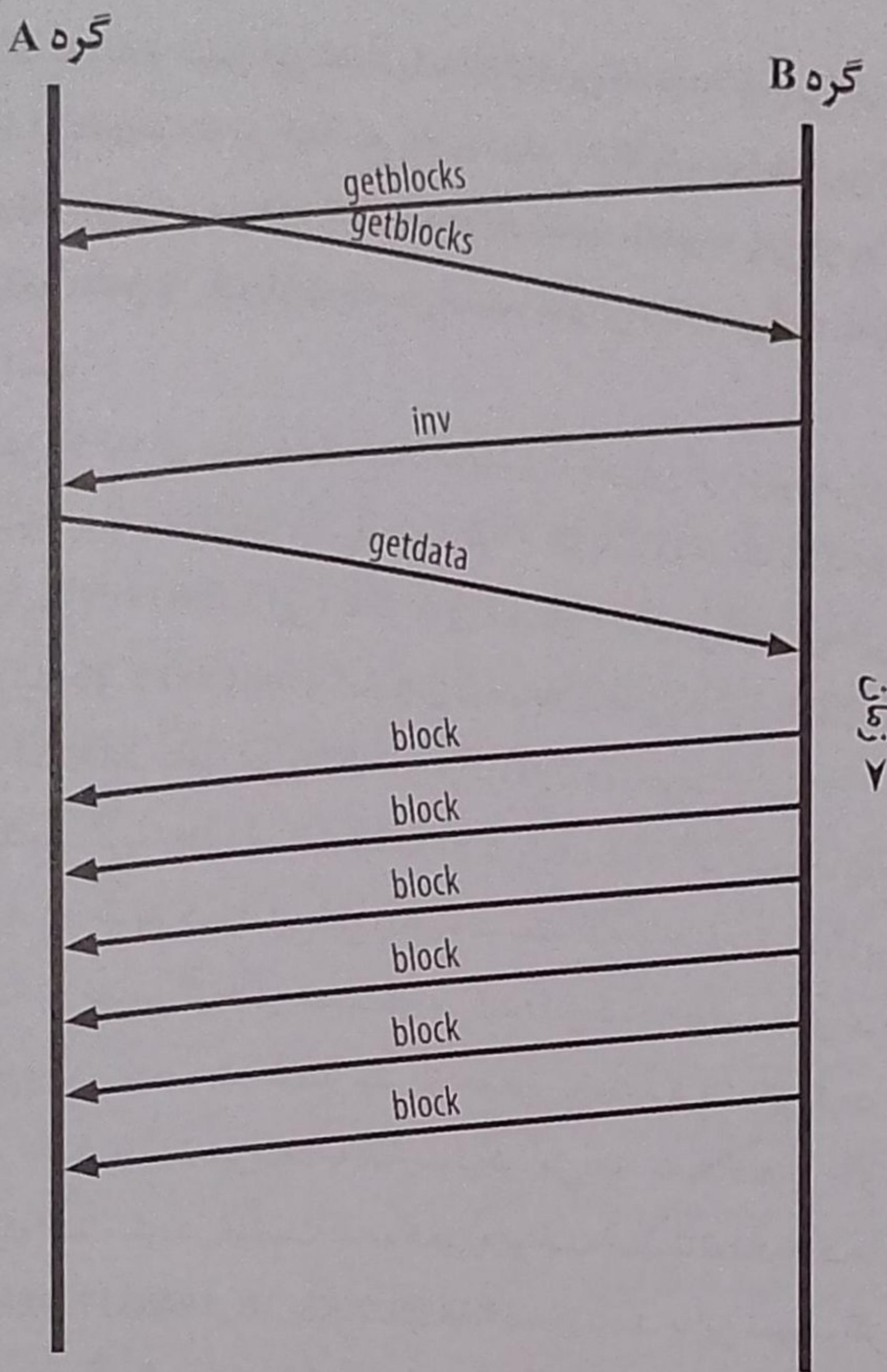
بلاک‌چین بزرگتر یعنی بلاک‌های بیشتر (ارتفاع بلاک بیشتر)، بنابراین گره مجبور می‌تواند متوجه شود که گره‌های دیگر کدام بلاک‌هارا ندارند و باید به آنها فرستاده شوند. این گره 50° بلاک نخست بلاک‌چین خود را شناسایی گردد و درهم آنها را با استفاده از یک پیام ۱۷۷ (دفتر دارایی) به گره‌های همسایه ارسال می‌کند. گره‌هایی که این بلاک‌هارا ندارند، می‌توانند با ارسال یک سری پیام‌های `getdata`، و گنجاندن درهم بلاک‌هایی که نیاز دارند (همان درهم‌هایی که از طریق پیام ۱۷۷ دریافت کردند)، اطلاعات کامل این بلاک‌هارا از گره دارند درخواست کنند.

برای مثال، فرض کنید یک گره تازهوارد است و فقط بلاک زاینده را دارد. این گره یک پیام `1nv` از گره‌های همسایه دریافت می‌کند که حاوی درهم 50° بلاک ابتدایی این زنجیره است. گره تازهوارد با ارسال پیام‌های `getdata` شروع به درخواست کردن این بلاک‌ها از تمامی گره‌های همسایه می‌کند؛ این گره درخواست‌های خود را بین تمام همسایه توزیع می‌کند تا مطمئن شود هیچ یک از آنها تحت فشار بیش از حد قرار نخواهد گرفت. این گره در هر لحظه تعداد بلاک‌های «در دست انتقال» به ازای هر اتصال را ثبت می‌کند؛ به عبارت دیگر، می‌داند چه بلاک‌هایی را از کدام همتا درخواست کرده ولی هنوز آنها را دریافت نکرده است. گره درخواست‌کننده همواره موازن است تا تعداد بلاک‌های «در دست انتقال» از حد تعیین شده با پارامتر `MAX_BLOCKS_IN_TRANSIT_PER_PEER` تجاوز نکند. با این تمهد، اگر بلاک‌هایی که یک گره نیاز به بارگیری دارد، خیلی زیاد باشد، آهنگ انتقال بلاک کنترل شده شبکه دچار اضافه بار نخواهد شد؛ به بیان دیگر، این گره فقط زمانی بلاک‌های جدید درخواست خواهد کرد که بلاک‌های قبلی را دریافت کرده باشد. هر بلاک پس از دریافت شدن (به روشی که در فصل ۹ توضیح خواهیم داد) به بلاک‌چین اضافه می‌شود. پسا از این تدریجی بلاک‌چین، بلاک‌های بیشتری درخواست و دریافت می‌شوند، و این فرآیند تا همگام شدن بلاک‌چین این گره باقیهای شبکه ادامه می‌یابد.

هر بار که یک گره برای مدتی از شبکه‌ی بیت‌کوین قطع شده و سپس دوباره به آن متصل می‌شود، فرآیند مقایسه‌ی بلاک‌چین محلی با گره‌های همتا و دریافت بلاک‌هایی که در این بلاک‌چین وجود ندارند، انجام می‌شود. صرفنظر از این که یک گره چه مدتی غایب بوده و چه تعداد بلاک را در بلاک‌چین محلی خود ندارد، این فرآیند همیشه با ارسال پیام `getblocks` و دریافت یک پاسخ ۱۷۷ شروع شده، و با درخواست و بارگیری بلاک‌های غایب (با یک سری پیام‌های `getdata`) ادامه خواهد یافت. شکل ۸-۶ فرآیند همگام‌سازی بلاک‌چین و پروتکل انتشار بلاک را نشان می‌دهد.

گره «اعتبارسنجی پرداخت ساده» (SPV)

آشکار است که همه‌ی گره‌های توانند یک کپی کامل و به روز از بلاک‌چین داشته باشند. برخی از مشتری‌های بیت‌کوین برای اجرارویی دستگاه‌های دارای محدودیت سخت افزاری (مثل تلفن هوشمند، تبلت و سیستم‌های نهفته) طراحی شده‌اند. در این قیل دستگاه‌ها از روشی موسوم به SPV، اعتبارسنجی پرداخت ساده (simplified payment verification)، استفاده می‌شود که به آنها اجازه می‌دهد کار خود را بدون نگهداری یک نسخه‌ی کامل از بلاک‌چین پیش ببرند، و به آنها مشتری سیکوریت نیز گفته می‌شود. با اوج گرفتن محبوبیت بیت‌کوین، گره SPV در حال تبدیل شدن به رایج‌ترین نوع گره بیت‌کوین (مخصوصاً برای آنها) است که فقط به کیف‌پول بیت‌کوین نیاز دارد.



شکل ۸-۶ همگام‌سازی بلاک‌چین محلی با درخواست و دریافت بلاک‌ها از یک همتا.

گره‌های SPV فقط سرآیند بلاک‌ها را بارگیری می‌کنند و تراکنش‌های درون آنها را دریافت نمی‌کنند. زنجیره‌ی حاصل معمولاً ۱۰۰۰ بار کوچکتر از بلاک‌چین کامل است. از آنجا که این گره‌ها چیزی درباره‌ی تمامی تراکنش‌های روی شبکه نمی‌دانند، نمی‌توانند یک تصویر کامل از UTXO های موجود برای خرج کردن داشته باشند. گره‌های SPV برای اعتبارسنجی تراکنش‌ها از روشی متفاوت استفاده می‌کنند که در آن یک گره برای تأمین اطلاعات مورد نیاز درباره‌ی بخش‌های مرتبط بلاک‌چین به گره‌های همسایه اتکا می‌کند.

برای مقایسه، گره کامل را می‌توان مانند جهانگردی دانست که برای گشت و گذار در یک شهر غریب از نقشه‌ای دقیق حاوی جزئیات تمام خیابان‌ها و مکان‌های آن شهر استفاده می‌کند، در حالی که گره SPV مثل جهانگردی است که فقط خیابان اصلی شهر را می‌شناسد و برای گشت و گذار در شهر مسیر را به صورت تصادفی از عابران غریبه می‌پرسد. هر چند هر دو جهانگرد در نهایت می‌توانند به وجود یا عدم وجود یک خیابان خاص پی ببرند (یکی به کمک نقشه و دیگری با پرس و جو از عابران)، ولی جهانگرد بدون نقشه هیچ چیز دیگری در مورد آن خیابان نمی‌داند، واز وجود خیابان‌های دیگر هم بی‌اطلاع است. برای مثال، وقتی این جهانگرد به خیابانی به نام «گلبرگ هشتم» می‌رسد، مطلقاً نمی‌داند آیا این همان خیابانی است که به دنبال آن می‌گردد، و آیا خیابان‌های دیگری نیز با نام «گلبرگ هشتم» در آن شهر وجود دارند یا خیر. جهانگرد بدون نقشه فقط باید از تعداد کافی افراد آدرس بپرسد و امیدوار باشد اورا گمراه نکنند و جیش را نزنند.

در روش SPV، اعتبارسنجی تراکنش‌ها به جای ارتفاع آنها، با ارجاع به عمق تراکنش در بلاک چین انجام می‌شود. در حالی که در یک گره کامل زنجیره‌ای کاملاً اعتبارسنجی شده متشکل از صدها هزار بلاک و تراکنش وجود دارد، و سابقه‌ی یک تراکنش (در زمان) را می‌توان تا خود بلاک زاینده تعقیب کرد، گره‌های SPV فقط زنجیره‌ی بلاک‌ها (ونه همه‌ی تراکنش‌ها) را اعتبارسنجی کرده و این زنجیره را به تراکنش مورد نظر لینک می‌کنند.

همه‌ی مثال، وقتی یک گره کامل بخواهد تراکنشی را در بلاک #300,000 اعتبارسنجی کند، تمامی سیصد هزار بلاک زیرین آن را تا بلاک زاینده لینک کرده و یک پایگاه داده از UTXO‌های موجود در این بلاک‌ها می‌سازد، و با بررسی این که زیرین (های) ارجاع شده در این تراکنش خرج نشده‌اند، اعتبار آن را تأیید می‌کند. اما یک گره SPV نمی‌تواند معتبر بودن UTXO را تأیید یا تکذیب کند. به جای آن، گره SPV با استفاده از یک مسیر مرکل (merkle path) [فصل بعد را بینید] یک لینک بین این تراکنش و بلاکی که در آن قرار دارد، برقرار خواهد کرد. سپس، این گره SPV صبر می‌کند تا شش بلاک روی پلاک حاوی تراکنش (بلاک #300,000) مورد نظر ابانته شوند (در این مثال، بلاک‌های #300,001 تا #300,006)، و با اطمینان از عمق کافی آن در زیر بلاک‌های #300,001 تا #300,006، این تراکنش را اعتبارسنجی می‌کند. به بیان دیگر، گره SPV از این واقعیت که دیگر گره‌های شبکه بلاک #300,000 را پذیرفته‌اند و سپس کار لازم برای تولید شش بلاک جدید روی آن را انجام داده‌اند، به طور غیرمستقیم نتیجه می‌گیرد که آن تراکنش حاوی UTXO‌های خرج شده نبوده است.

اگر تراکنشی در یک بلاک وجود نداشته باشد، نمی‌توان وجود آن را به یک گره SPV قالب کرد، چون گره SPV با درخواست «اثبات مسیر مرکل» و با اعتبارسنجی PoW در زنجیره‌ی بلاک‌ها از وجود یک تراکنش مطمئن می‌شود. با این حال، وجود یک تراکنش را می‌توان از دید یک گره SPV «مخفى» کرد. به عبارت دیگر، یک گره SPV می‌تواند وجود یک تراکنش را به طور قطع و یقین اثبات کند، ولی نمی‌تواند یقین پیدا کند که یک تراکنش (مثل تراکنشی که یک UTXO واحد را دوبار خرج کرده) وجود ندارد، چون سابقه‌ی کل تراکنش‌های شبکه را در اختیار ندارد. از این نقطه ضعف می‌توان برای از کار انداختن یک گره SPV (حمله‌ی DoS) یا حمله‌ی خرج-دوباره به آن استفاده کرد. برای دفاع در مقابل این حمله‌ها، یک گره SPV باید هر بار به طور تصادفی به گره‌های متفاوتی متصل شود تا احتمال اتصال به حداقل یک گره درستکار را افزایش دهد. البته همین نیاز به اتصال تصادفی به گره‌های همسایه، گره‌های SPV را در مقابل حملات افزایش‌بندی شبکه (network partitioning) یا حملات سیبل (Sybil) آسیب‌پذیر می‌کند. این حملات گره SPV را مجبور می‌کنند، به چای گره‌های درستکار یا شبکه‌ی بیت‌کوین واقعی، نادانسته به یک گره یا شبکه‌ی جعلی متصل شود.

در عمل، یک گره SPV که اتصالات کافی به شبکه‌ی بیت‌کوین داشته باشد، از امنیت کافی برخوردار است. در واقع، این گره‌ها می‌توانند به توازن مناسبی بین منابع مورد نیاز، عملی بودن و امنیت دست پیدا کنند. با این حال، برای داشتن امنیت خلل‌ناپذیر هیچ چیز جای داشتن یک نسخه‌ی کامل از بلاک چین را نمی‌گیرد.

گره بلاک چین کامل برای اعتبارسنجی یک تراکنش و فهمیدن این که UTXO (های) ارجاع شده در ورودی (های) آن خرج نشده هستند، کل زنجیره‌ی بلاک چین شامل صدها هزار بلاک واقع در زیر بلاکی که این تراکنش در آن قرار دارد (تا اولین بلاک: بلاک زاینده)، را جستجو می‌کند. در طرف مقابل، گره‌های SPV فقط به این توجه می‌کنند که آن بلاک زیر چند بلاک دیگر قرار گرفته است.

گره‌های SPV برای گرفتن سرآیند بلاک‌ها (به جای getblocks) از پیام getheaders استفاده می‌کنند. در واکنش به این درخواست، گره پاسخ‌دهنده ۲۰۰۰ سرآیند بلاک را در یک پیام headers به آن بر می‌گرداند. این فرآیند هیچ تفاوت ماهوی با دریافت بلاک‌های کامل توسط یک گره بلاک چین کامل ندارد. گره‌های SPV همچنین برای فیلتر کردن استریم بلاک‌ها و تراکنش‌های آینده (که از طرف گره‌های همتا به آنها فرستاده می‌شوند) از یک فیلتر مخصوص استفاده می‌کنند.

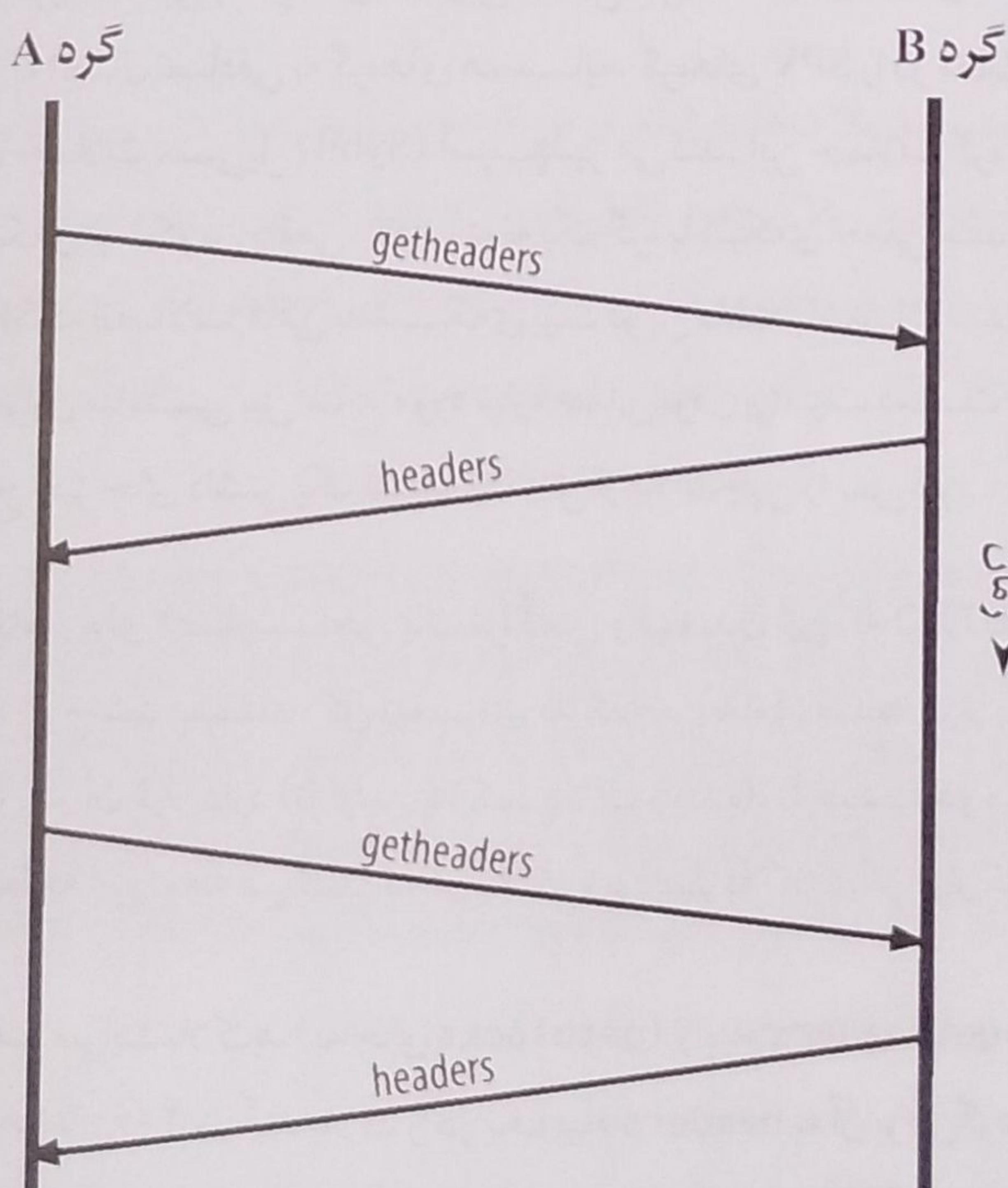
تراکنش‌های خاصی که یک گره SPV به آنها علاقه دارد، از طریق درخواست `getdata` بازیابی می‌شوند؛ گره هم‌تا در پاسخ به این درخواست، تراکنش‌های خواسته شده را از طریق یک پیام `tX` به گره SPV برمی‌گرداند. شکل ۷-۸ فرآیند همگام‌سازی سرآیند بلاک‌ها را نشان می‌دهد.

از آنجاکه گره‌های SPV به تراکنش‌های خاصی نیاز دارند تا آنها را به صورت گزینشی اعتبارسنجی کنند، احتمال به خطر افتادن حریم خصوصی در آنها وجود دارد. بر خلاف گره‌های بلاک‌چین کامل که کل تراکنش‌های هر بلاک را جمع‌آوری می‌کنند، درخواست‌های گزینشی یک گره SPV برای بازیابی داده‌های خاص می‌تواند به لطف رفتن ناخواسته‌ی آدرس‌های کیف‌پول وی بینجامد. برای مثال، یک شخص ثالث می‌تواند با پایش ترافیک شبکه تمامی تراکنش‌های درخواست‌شده توسط یک گره SPV را ردیابی کند و از این اطلاعات برای یافتن ارتباط منطقی بین آدرس‌های بیت‌کوین با کاربر آن کیف‌پول استفاده کرده و حریم خصوصی وی را نقض کند.

کمی بعد از معرفی گره‌های SPV/سبک‌وزن، برنامه‌نویسان بیت‌کوین برای مقابله با خطر نقض محترمانگی گره‌های SPV، یک ویژگی جدید به نام فیلتر بلوم به شبکه‌ی بیت‌کوین اضافه کردند. فیلتر بلوم از طریق ساز و کاری که برای فیلتر کردن [به جای الگوهای ثابت و مشخص] از الگوهای مبتنی بر احتمالات استفاده می‌کند، به گره‌های SPV اجازه می‌دهد زیرمجموعه‌ای از تراکنش‌هارا بازیابی کنند، بدون این که معلوم شود دقیقاً کدام آدرس‌هارا می‌خواهند.

فیلتر بلوم

فیلتر بلوم [که در ۱۹۷۰ توسط بارتون هاوارد بلوم ابداع شد] یک فیلتر جستجوی مبتنی بر احتمالات است که الگوی جستجوی مطلوب را بدون بیان دقیق آن توصیف می‌کند. فیلترهای بلوم ابزاری مناسب برای جستجو بر اساس الگوهای دلخواه بدون به خطر انداختن حریم خصوصی (محترمانگی) کاربر هستند. گره‌های SPV از این نوع فیلتر برای درخواست تراکنش‌های دلخواه (از گره‌های همسایه) استفاده می‌کنند، بدون آن که مشخص کنند دقیقاً دنبال چه آدرس، کلید یا تراکنشی می‌گردند.



شکل ۷-۸ همگام‌سازی سرآیند بلاک‌ها در گره SPV.

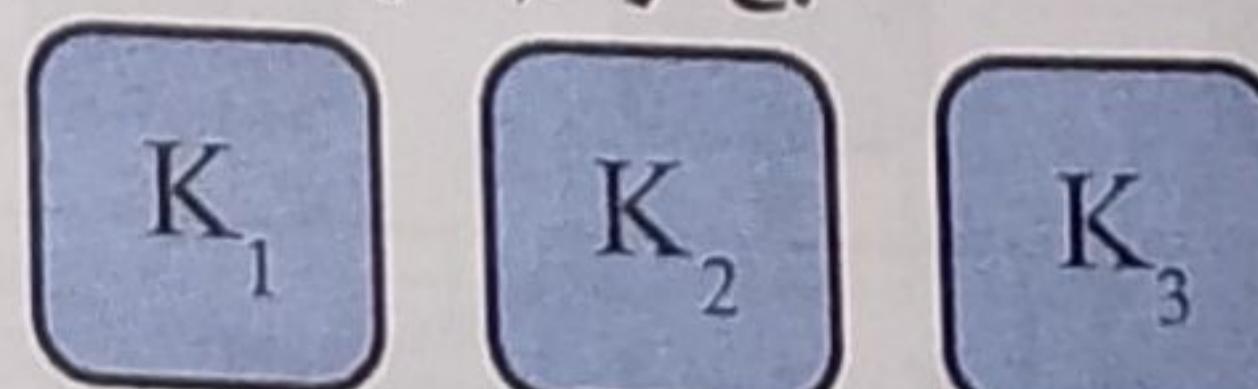
در مثال قسمت قبل دیدیم که جهانگرد بدون نقشه‌ی ما دنبال آدرس «گلبرگ هشتم» می‌گشت. اگر او این آدرس را به طور مستقیم از عابران غریبه بپرسد، ناخواسته مقصد خود را فاش کرده است. پرسشی از نوع فیلتر بلوم می‌تواند چیزی شبیه این باشد: «آیا در این اطراف خیابانی می‌شناسید که نام آن با گل شروع شود؟» این پرسش نسبت به «آیا مسیر رسیدن به خیابان گلبرگ هشتم را می‌دانید؟» اطلاعات خیلی کمتری از مقصد جهانگرد فاش می‌کند. در این روش، میزان اطلاعاتی که برای دیگران فاش می‌کنید، کاملاً به خود شما بستگی دارد، مثلاً «آیا خیابانی می‌شناسید که نام آن با گل شروع شود؟» [جزئیات کمتر]، یا «آیا خیابانی می‌شناسید که نام آن به برگ ختم شود؟» [جزئیات بیشتر]. توجه کنید که کمتر به کار می‌برید، آدرس‌های بیشتری دریافت می‌کنید و احتمال فاش شدن مقصد شما کمتر خواهد بود، ولی در عین حال باید اطلاعات بیشتری را پردازش کنید. از طرف دیگر، اگر الگوی دقیق‌تری به کار ببرید، نتایج کمتری دریافت می‌کنید، ولی احتمال به خطر افتادن حریم خصوصی شما بیشتر خواهد شد.

فیلتر بلوم به گره SPV اجازه می‌دهد الگوی جستجوی تراکنش‌هارا با تأکید بر دقت یا محرمانگی تنظیم کنید. هر چه یک فیلتر بلوم مشخص‌تر باشد، نتایج دقیق‌تری تولید خواهد کرد، ولی احتمال فاش شدن الگوی مطلوب آن (و در نتیجه لو رن آدرس‌هایی که در کیف‌پول کاربر وجود دارند) نیز افزایش می‌یابد. یک فیلتر بلوم مبهم‌تر تراکنش‌های بیشتری به گره SPV بر می‌گردد که بسیاری از آنها نامرتبط هستند، ولی به آن اجازه می‌دهند حریم خصوصی خود را بهتر حفظ کند.

طرز کار فیلتر بلوم

فیلتر بلوم به صورت یک آرایه با طول متغیر از N بیت باینری (رقم دیجیتال) و تعدادی تابع دَرهم‌ساز M پیاده‌سازی می‌شود. این توابع دَرهم‌ساز (که تعداد آنها متغیر است) طوری طراحی شده‌اند که خروجی آنها همواره بین ۱ تا N (متناظر با طول آرایه‌ی بیت‌های باینری) باشد. این توابع به صورت قطعی و ثابت تولید می‌شوند، به طوری که پیاده‌سازی یک فیلتر بلوم در هر گره همیشه از توابع دَرهم‌ساز یکسانی استفاده می‌کند و (به ازای یک ورودی مشخص) خروجی ثابتی خواهد داشت. با انتخاب طول‌های مختلف برای آرایه‌ی بیت فیلتر بلوم (N) و تعداد مختلف توابع دَرهم‌ساز (M) می‌توان سطح دقت و محرمانگی این فیلتر را به دلخواه تنظیم کرد. در شکل ۸-۸ یک فیلتر بلوم بسیار ساده با آرایه‌ای به طول ۱۶ بیت و سه تابع دَرهم‌ساز نشان داده شده است. در آغاز به تمامی بیت‌های آرایه‌ی N مقدار ۰ داده می‌شود. برای اضافه کردن یک الگو به فیلتر بلوم، این الگو به ترتیب نوسط توابع M دَرهم‌سازی می‌شود. اعمال تابع دَرهم‌ساز اول به ورودی (الگوی مورد نظر) یک عدد باینری بین ۱ و N تولید می‌کند. به بیت متناظر با این عدد در آرایه‌ی N (که از ۱ تا N شماره‌گذاری شده‌اند) مقدار ۱ داده شده و بدین ترتیب خروجی می‌شود.

۳ تابع دَرهم‌ساز



خروجی تابع دَرهم‌ساز
۱۶ تا ۱

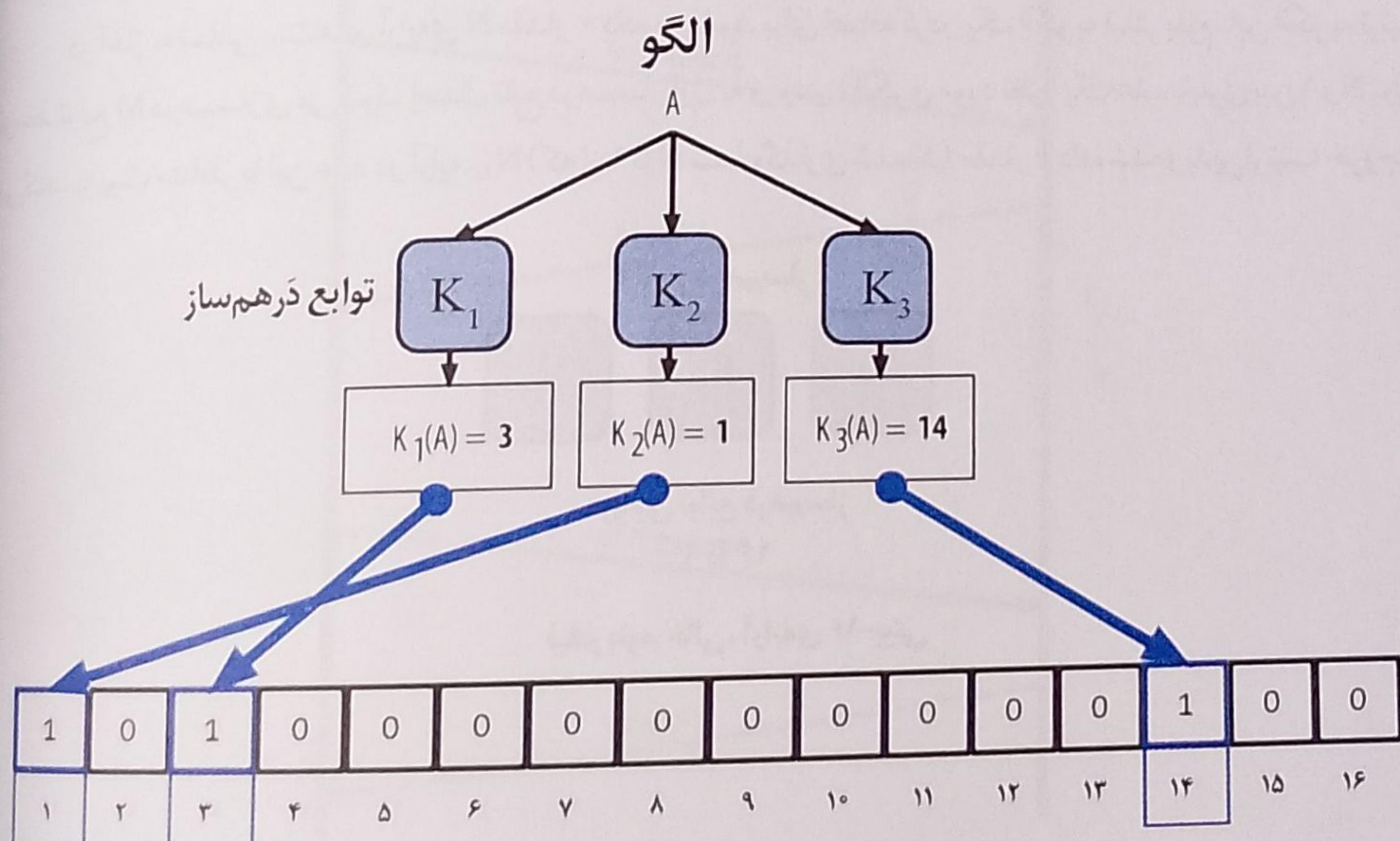
فیلتر بلوم خالی، آرایه‌ی ۱۶-بیتی

۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	

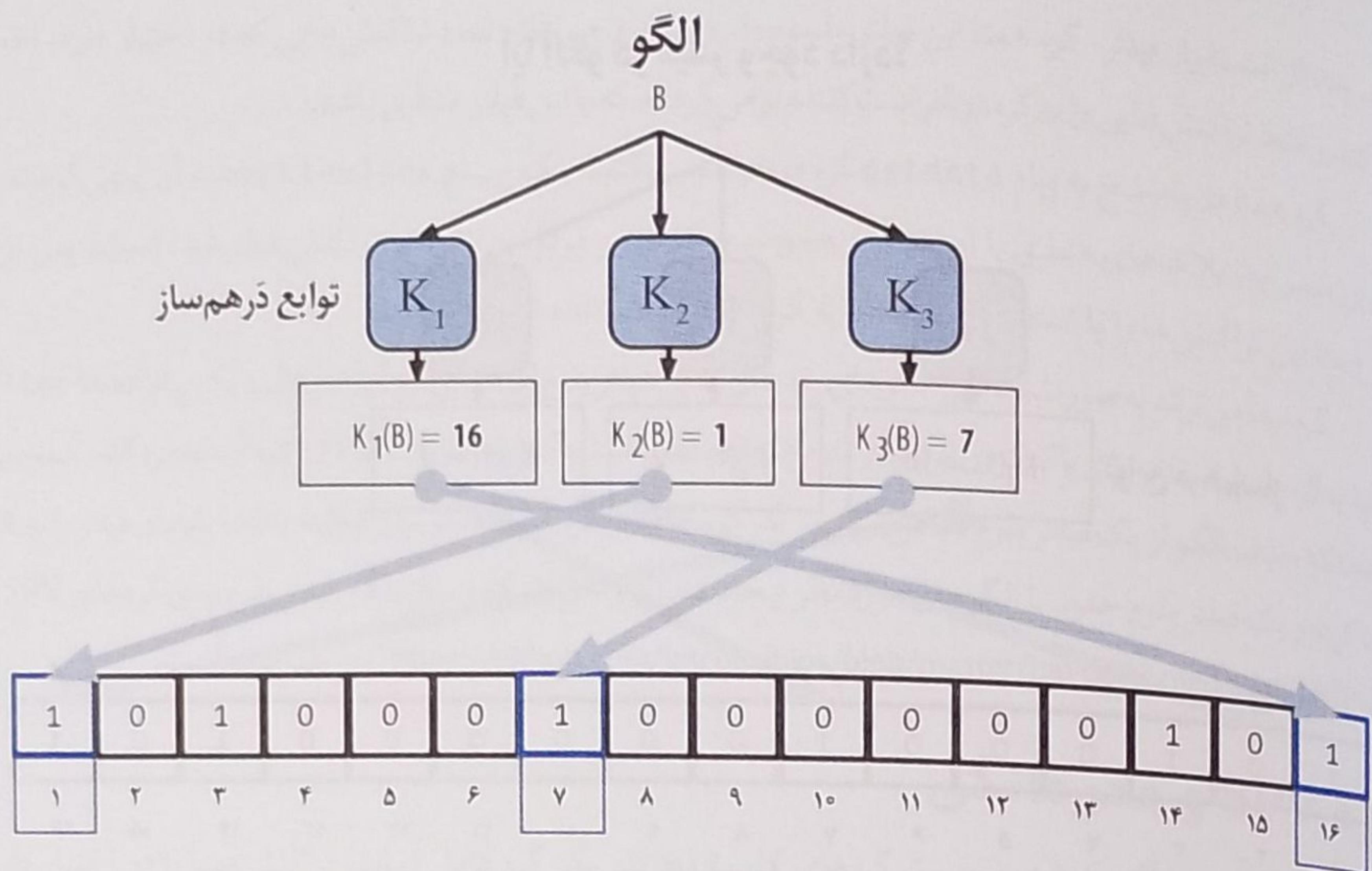
شکل ۸-۸ فیلتر بلوم بسیار ساده با فیلد ۱۶-بیتی و سه تابع دَرهم‌ساز.

تابع دَرْهَم ساز ثبت می‌شود. این کار با توابع دَرْهَم ساز بعدی نیز تکرار شده و نتیجه در آرایه‌ی N ثبت می‌شود. همین که تمامی توابع دَرْهَم ساز M روی ورودی اعمال شدند، الگوی جستجو به صورت M بیت که در آرایه‌ی N مقدار آنها از ۰ به ۱ تغییر یافته، «ثبت» خواهد شد. در شکل ۹-۸ این فرآیند برای الگوی ورودی «A» به فیلتر بلوم شکل ۸-۸ نشان داده شده است. اضافه کردن الگوهای بعدی به همین سادگی است: الگوی ورودی به ترتیب توسط توابع M دَرْهَم سازی شده و به بیت متناظر با این عدد در آرایه‌ی N مقدار ۱ داده می‌شود. توجه کنید که با پُر شدن فیلتر بلوم با الگوهای بعدی، ممکن است خروجی یکی از توابع دَرْهَم ساز M با بیتی که از قبل مقدار ۱ دارد، مصادف شود؛ در این حالت مقدار آن بیت تغییر نخواهد کرد. در واقع، هر چه الگوهای بیشتری در یک فیلتر بلوم ثبت شوند، آن فیلتر به سمت اشباع شدن با بیت‌های ۱ پیش می‌رود و «دقت» فیلتر کاهش می‌یابد. به همین دلیل است که فیلتر بلوم یک ساختمان داده‌ی مبتنی بر احتمالات خوانده می‌شود: هر چه تعداد الگوهای ورودی بیشتر باشد، دقت فیلتر کمتر است. دقت یک فیلتر بلوم به تعداد الگوهای اضافه شده نسبت به اندازه‌ی آرایه‌ی بیت (N) و تعداد توابع دَرْهَم ساز (M) بستگی دارد. یک آرایه‌ی بیت بزرگتر و تعداد بیشتر توابع دَرْهَم ساز می‌تواند الگوهای بیشتری را با دقت بالاتر ثبت کند. یک آرایه‌ی بیت کوچکتر یا تعداد کمتر توابع دَرْهَم ساز الگوهای کمتری را ثبت می‌کند و دقت پایین‌تری تولید خواهد کرد. شکل ۸-۸ فرآیند اضافه کردن دومین الگوی ورودی «B» به فیلتر بلوم شکل ۸-۸ را نشان می‌دهد.

برای تشخیص این که یک الگو در فیلتر بلوم وجود دارد یا خیر، آن الگو به ترتیب توسط توابع M دَرْهَم سازی شده و عدد حاصل با بیت متناظر در آرایه‌ی N مقایسه می‌شود. اگر تمام خروجی‌های توابع دَرْهَم ساز با بیت‌هایی که مقدار ۱ دارند، منطبق شوند، آنگاه/حتماً این الگو در فیلتر بلوم وجود دارد. اما این نتیجه‌گیری به هیچ وجه قطعی نیست، چون ممکن است مقدار ۱ در این بیت حاصل همپوشانی با الگوهای دیگر باشد، ولی به هر حال محتمل است. به بیان ساده، مثبت بودن این آزمون فقط به معنای «شاید، بله» است. در شکل ۱۱-۸ فرآیند آزمایش وجود الگوی «X» در فیلتر بلوم شکل ۸-۸ را مشاهده می‌کنید؛ بیت‌های متناظر با دَرْهَم سازی این الگو همگی مقدار ۱ دارند، بنابراین احتمال انطباق آن با الگوی فیلتر بلوم مثبت است و شاید این الگو واقعاً در فیلتر وجود داشته باشد.



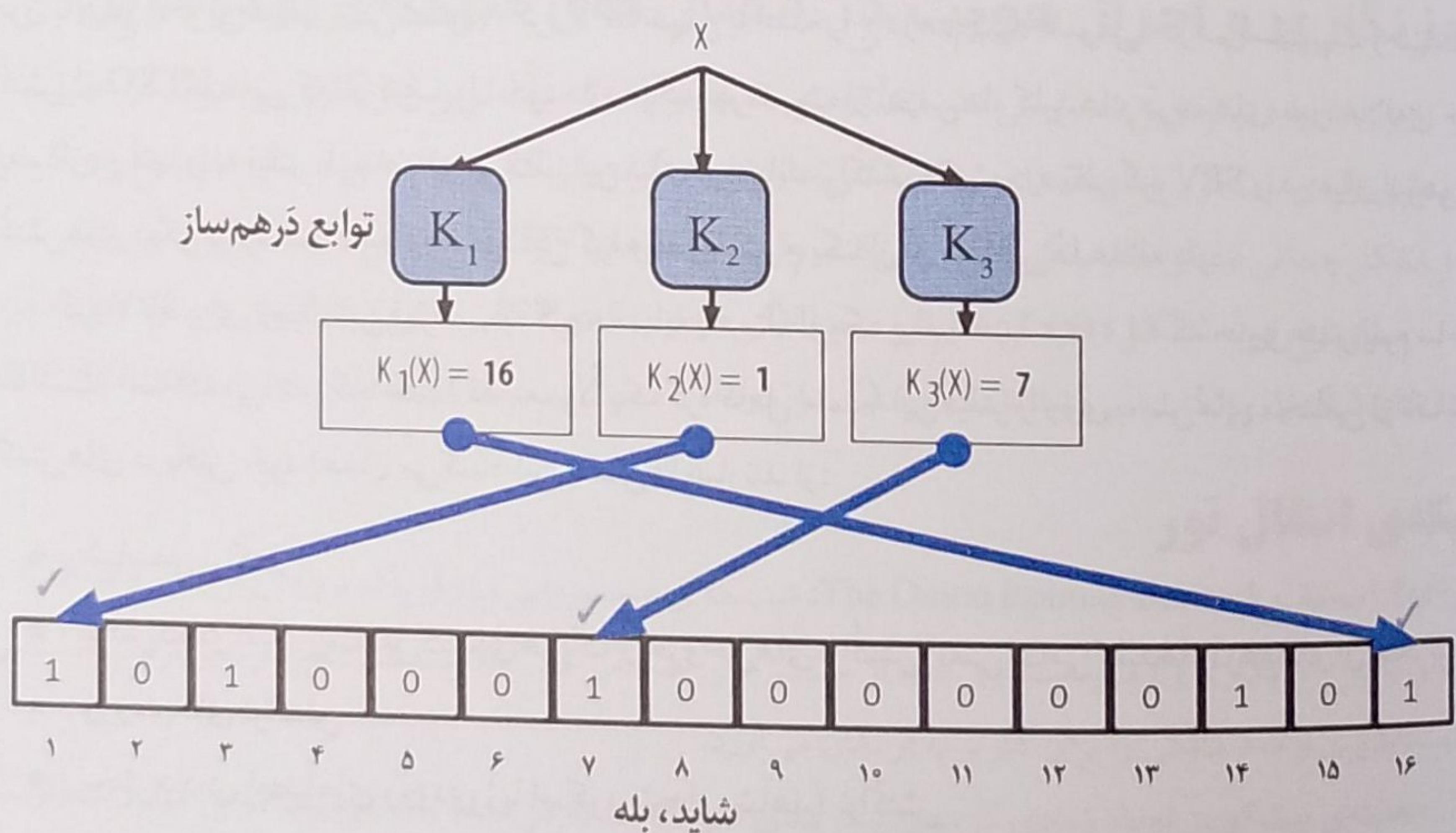
شکل ۹-۸ اضافه کردن الگوی «A» به فیلتر بلوم.



شکل ۱۰-۸ اضافه کردن الگوی دوم «B» به فیلتر بلوم.

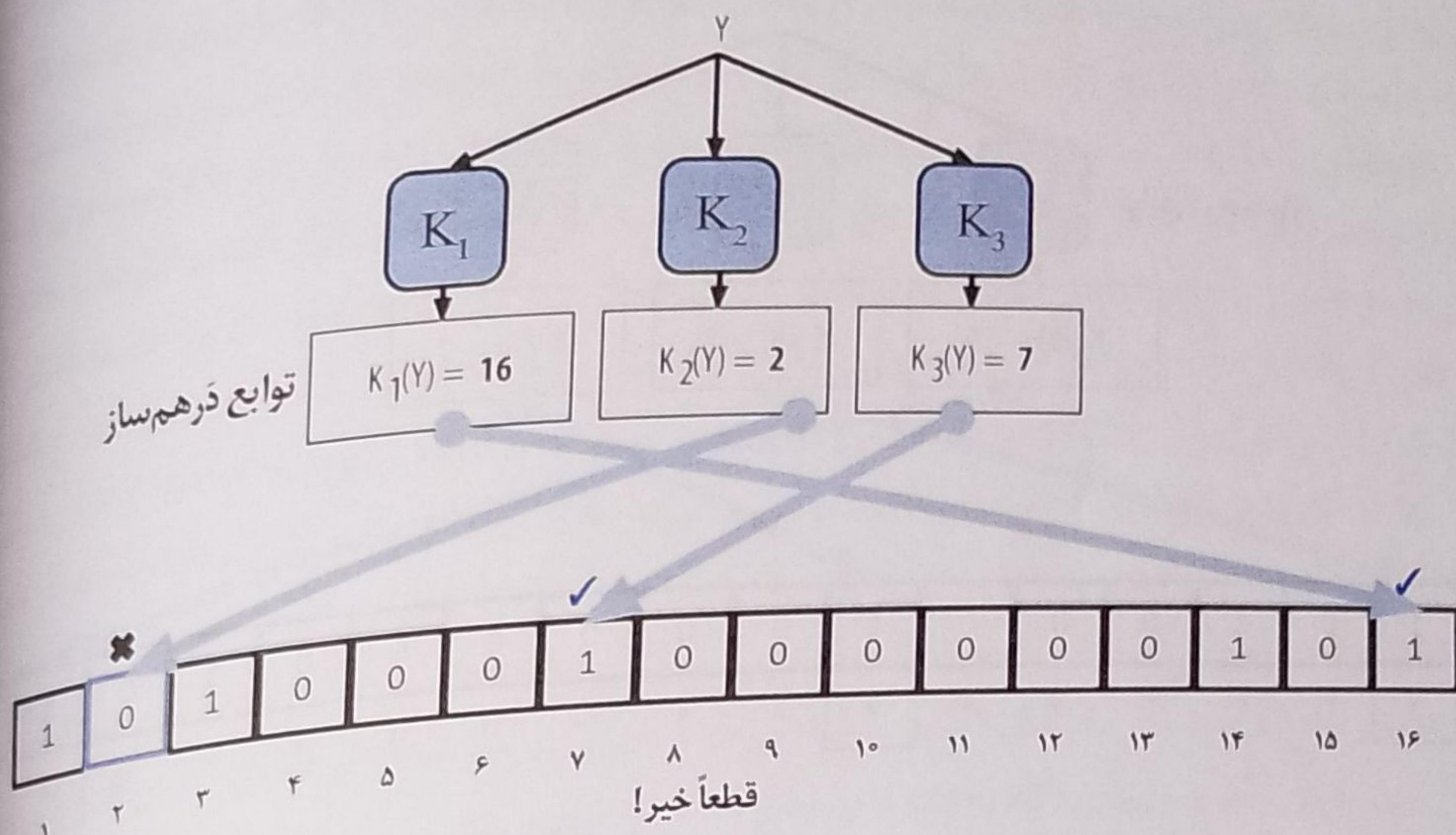
از سوی مقابل، اگر حتی یکی از بیت‌های متناظر با خروجی توابع درهم‌ساز در آرایه‌ی N فیلتر بلوم مقدار \circ داشته باشد، به طور قطع می‌توان نتیجه گرفت که آن الگو در فیلتر وجود ندارد. نتیجه‌ی منفی دیگر یک احتمال نیست، بلکه یقین است. به بیان ساده، منفی شدن این آزمون فقط می‌تواند معنای «قطعاً خیر!» داشته باشد. در شکل ۱۰-۸ فرآیند آزمایش وجود الگوی «Y» در فیلتر بلوم شکل ۸-۸ نشان داده شده است؛ همان‌طور که می‌بینید، یکی از بیت‌های متناظر با درهم‌سازی این الگو مقدار \circ دارد، پس این الگو یقیناً در فیلتر وجود ندارد.

آیا الگو در فیلتر وجود دارد؟



شکل ۱۱-۸ آزمایش تشخیص وجود الگوی «X» در فیلتر بلوم. از آنجا که این فقط نتیجه‌ای با احتمال مثبت است، باید آن را معنای «شاید» تلقی کرد.

آیا الگو در فیلتر وجود دارد؟



شکل ۱۲-۸ آزمایش تشخیص وجود الگوی «Y» در فیلتر بلوم. از آنجا که نتیجه به طور قطع منفی است، معنای آن را باید «قطعًا خیر!» دانست.

استفاده از فیلتر بلوم در گره‌های SPV

گره‌های SPV از فیلتر بلوم برای پالایش تراکنش‌ها (و بلاک‌هایی که این تراکنش‌ها در آن قرار دارند) و انتخاب تراکنش‌هایی که نیاز دارند، استفاده می‌کنند، بدون این که آدرس‌ها یا کلیدهای مورد نظر خود را فاش کنند. از آنجا که گره SPV کار خود را با یک فیلتر بلوم «خالی» شروع می‌کند، در این مرحله فیلتر آن هیچ تراکنشی را برنمی‌گرداند (چون با هیچ الگویی منطبق نمی‌شود). گره SPV سپس با استخراج دارهم کلید عمومی، دارهم اسکریپت و شناسه‌ی تراکنش از UTXO هایی که در کیف‌پول خود دارد، یک فهرست از آدرس‌ها، کلیدها و دارhem‌های مورد علاقه‌ی خود می‌سازد و آنها را به فیلتر بلوم اضافه می‌کند. این فیلتر می‌تواند تراکنش‌های مورد نظر گره SPV را در میان ابوهی از تراکنش‌های دیگر برگرداند، تا معلوم نشود این گره واقعاً به کدام یک از آن تراکنش‌ها علاقه دارد.

گره SPV برای ارسال این فهرست به گره همسایه (همتا) از یک پیام `filterLoad` که حاوی فیلتر بلوم ساخته شده است، استفاده می‌کند. گره همتا (که معمولاً یک گره کامل است) این فیلتر را روی بخش‌های مختلفی از تک نک تراکنش‌های دریافتی خود إعمال می‌کند؛ این بخش‌ها عبارتند از:

- شناسه‌ی تراکنش
- بخش داده‌ی اسکریپت قفل‌کننده‌ی هریک از خروجی‌های تراکنش (یعنی تمامی کلیدها و دارhem‌های آن اسکریپت)
- ورودی‌های تراکنش
- بخش داده‌ی امضاهای ورودی (یا اسکریپت‌های شاهد) تراکنش

با این روش، فیلتر بلوم می‌تواند دارhem کلیدهای عمومی، اسکریپت‌ها، مقادیر `OP_RETURN`، کلیدهای عمومی موجود در امضاهای، یا هر چیزی را که در آینده به قراردادهای هوشمند یا اسکریپت‌های پیچیده اضافه شود، شناسایی

بعد از استقرار فیلتر، گره هم‌تا این فیلتر بلوم را روی خروجی‌های تمام تراکنش‌هایی که در اختیار دارد، اعمال کند. می‌کند و فقط تراکنش‌هایی را به گره درخواست‌کننده بر می‌گرداند که با این فیلتر منطبق باشند. گره هم‌تا در پاسخ به پیام `getdata` گره درخواست‌کننده یک پیام `merkleblock` به آن بر می‌گرداند که حاوی سرآیند بلاک‌های منطبق با این فیلتر و همچنین یک مسیر مرکل به ازای هر تراکنش فیلتر شده است. پس از آن گره هم‌تا این تراکنش‌ها را با استفاده از پیام `t` به گره درخواست‌کننده ارسال می‌کند. گره مبدأ می‌تواند به صورت تعاملی الگوهای جدیدی به این فیلتر بلوم اضافه کرده و آن را در قالب یک پیام `filterload` گره مبدأ می‌تواند به گره همتا بفرستد. برای پاک کردن این فیلتر بلوم هم فقط کافی است یک پیام `filterclear` به آن مخابره کند. همچنین، جدید به گره همتا بفرستد. اگر گره SPV دیگر به یک الگونیاز نداشته باشد، باید آن فیلتر را به کلی از آنجاکه حذف الگواز یک فیلتر بلوم ممکن نیست، پروتکل شبکه وساز و کار فیلتر بلوم برای گره‌های SPV در پاک کرده و یک فیلتر بلوم جدید با الگوهای مورد نظر ایجاد کند. پروتکل شبکه وساز و کار فیلتر بلوم برای گره‌های SPV در [https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki] BIP-37 تعریف شده است.

گره SPV و محربانگی

گره‌های SPV محربانگی کمتری نسبت گره‌های کامل دارند. یک گره کامل تمامی تراکنش‌های را در اختیار دارد، بنابراین هیچ اطلاعاتی درباره‌ی آدرس‌هایی که در کیف‌پول خود به کار می‌برد، افشا نمی‌کند. از طرف دیگر، گره‌های SPV فهرستی فیلتر شده از تراکنش‌های مرتبط با آدرس‌هایی که در کیف‌پول خود دارند، دریافت می‌کنند، و در نتیجه مشکلات بیشتری برای حفظ حریم خصوصی (محربانگی) خود خواهند داشت.

فیلتر بلوم ابزاری است برای کاستن از مخاطرات این وضعیت. بدون فیلتر بلوم، یک گره SPV باید فهرست آدرس‌های مورد نظر خود را به صراحةً فاش کند، و خطر نقض حریم خصوصی را به جان بخرد. با این حال، حتی با وجود فیلتر بلوم، دشمنان یک کاربر می‌توانند ترافیک گره SPV وی را پایش کرده یا به طور مستقیم به عنوان یک گره همتا در شبکه‌ی P2P بیت‌کوین به او متصل شوند و با جمع‌آوری اطلاعات کافی در طول زمان، آدرس‌های کیف‌پول این گره SPV را به دست آورند.

رمزنگاری و احراز هویت در اتصال‌های شبکه‌ی بیت‌کوین

اکثر کاربران تازه‌وارد بیت‌کوین تصور می‌کنند اتصال‌های شبکه در یک گره بیت‌کوین رمزنگاری می‌شوند. در حقیقت، در پیاده‌سازی اولیه‌ی بیت‌کوین، اتصال‌های شبکه به کلی عاری از هر گونه رمزنگاری بودند. هر چند این موضوع برای گره‌های کامل مشکل چندانی از نظر محربانگی ایجاد نمی‌کند، ولی برای گره‌های SPV یک مشکل بزرگ محسوب می‌شود. برای رفع این مشکل و افزایش امنیت و محربانگی در شبکه‌ی P2P بیت‌کوین، دوروش وجود دارد که هر دو اتصال‌های شبکه را رمزنگاری می‌کنند: شبکه‌ی انتقال تور، و احراز هویت و رمزنگاری همتا-به-همتا با BIP-150/151.

شبکه‌ی انتقال تور

تور (Tor) [مخفف The Onion Routing network]: شبکه‌ی مسیریابی پیاز (Onion Routing network) یک پروژه‌ی نرم‌افزار و شبکه است که برای تأمین رمزنگاری و کپسوله‌سازی داده از مسیرهای تصادفی شبکه استفاده کرده و بدین ترتیب ناشناسی، تعقیب‌ناپذیری و محربانگی را برای کاربر به ارمغان می‌آورد.

هسته‌ی بیت‌کوین تعداد زیادی گزینه‌ی پیکربندی دارد که اجازه می‌دهند یک گره بیت‌کوین مبادرات (ارسال/دریافت) خود را روی شبکه‌ی تور انجام دهد. علاوه بر آن، هسته‌ی بیت‌کوین یک سرویس مخفی تور نیز ارائه می‌کند که به گره‌های تور اجازه می‌دهد به طور مستقیم از طریق شبکه‌ی تور به گره شما متصل شوند.

از ویرایش ۰.۱۲ هسته‌ی بیت‌کوین، اگر یک گره بیت‌کوین به سرویس تور دسترسی داشته باشد، به طور خودکار این سرویس مخفی تور را فعال می‌کند. اگر در سیستم خود تور را نصب کنید و به فرآیندهای هسته‌ی بیت‌کوین مجوزهای کافی برای دسترسی به کوکی احراز هویت تور بدھید، این سرویس مخفی به طور خودکار فعال خواهد شد. برای وارد شدن به حالت دیباگ هسته‌ی بیت‌کوین برای سرویس تور، هسته‌ی بیت‌کوین را با پرچم `debug` اجرا کنید:

```
$ bitcoind --daemon --debug=tor
```

بادیدن پیام «`ADD_ONION successful`» در گزارش خروجی می‌توانید مطمئن باشید که هسته‌ی بیت‌کوین سرویس مخفی تور را برای این گره فعال کرده است. برای کسب اطلاعات بیشتر درباره اجرای هسته‌ی بیت‌کوین به عنوان یک سرویس مخفی تور می‌توانید به مستندات بیت‌کوین (`docs/tor.md`) یا منابع آموزشی اینترنتی مراجعه کنید.

احراز هویت و رمزنگاری همتا-به-همتا

دو سند پیشنهاد بهسازی بیت‌کوین BIP-150 و BIP-151 به ترتیب پشتیبانی از احراز هویت و رمزنگاری P2P شبکه‌ی P2P بیت‌کوین اضافه می‌کنند. این دو BIP سرویس‌های اختیاری را تعریف می‌کنند که می‌توان بین گره‌های بیت‌کوین سازگار به کار گرفت. استاندارد BIP-151 به دو گره سازگار با این استاندارد امکان می‌دهد تمامی مبادلات شبکه‌ی بین خود را رمزنگاری کنند. استاندارد BIP-150 به گره‌ها اجازه می‌دهد تا با استفاده از ECDSA و کلیدهای خصوصی هویت یکدیگر را احراز (شناسایی) کنند. برای استفاده از سرویس احراز هویت BIP-150 دو گره بایستی قبلً یک کanal ارتباطی با رمزنگاری BIP-151 بین خود برقرار کرده باشند.

تاریخ انتشار این کتاب استانداردهای BIP-150 و BIP-151 هنوز در هسته‌ی بیت‌کوین پیاده‌سازی نشده‌اند؛ با این حال، یکی دیگر از مشتری‌های بیت‌کوین موسوم `bcoin` این BIP ها را پیاده‌سازی کرده است. این دو استاندارد می‌توانند نقش مهمی در اتصال آمن گره‌های SPV به شبکه‌ی بیت‌کوین بازی کنند. استفاده از سرویس‌های احراز هویت می‌تواند مانع از حملات «نفر-در-وسط» شود، و سرویس رمزنگاری P2P نیز می‌تواند شبکه‌ی بیت‌کوین را در مقابل پایش و تحلیل ترافیک شبکه مقاوم کند. [برای اطلاعات بیشتر درباره این استانداردها به <https://github.com/bitcoin/bips/blob/master/bip-0150.mediawiki> و <https://github.com/bitcoin/bips/blob/master/bip-0151.mediawiki> مراجعه کنید.]

مخزن تراکنش

تقریباً تمامی گره‌های شبکه‌ی بیت‌کوین فهرستی موقتی از تراکنش‌های تأییدنشده دارند که به آن مخزن حافظه (memorypool) یا مخزن تراکنش (transaction pool) گفته می‌شود. گره‌ها از این مخزن برای نگهداری و پیگیری تراکنش‌هایی که برای شبکه شناخته شده هستند ولی هنوز در بلاک چین ثبت نشده‌اند، استفاده می‌کنند. برای مثال، گره‌های کیف‌پول مخزن تراکنش را برای پیگیری دریافت‌های ورودی به کیف‌پول کاربر که هنوز تأیید نشده‌اند، به کار می‌گیرند. تراکنش‌ها بعد از دریافت و اعتبارسنجی به این مخزن تراکنش اضافه شده و به گره‌های همسایه مخابره می‌شوند تا روی شبکه‌ی بیت‌کوین انتشار داده شوند.

برخی از پیاده‌سازی‌های خاص بیت‌کوین یک مخزن جداگانه برای تراکنش‌های یتیم نیز دارند. تراکنش یتیم (orphan transaction) به تراکنشی گفته می‌شود که خود آن قبل از تراکنش‌های ورودی اش به یک گره رسیده باشد؛ در این حالت تراکنش یتیم باید آنقدر در مخزن تراکنش‌های یتیم بماند تا تراکنش مادر (یا تراکنش والد؛ تراکنشی که در ورودی این تراکنش به آن ارجاع شده) از راه برسد.

وقتی یک تراکنش به مخزن تراکنش اضافه می‌شود، گره دریافت‌کننده ابتدا در مخزن تراکنش‌های یتیم جستجو می‌کند تا فرزند یتیم احتمالی آن (تراکنشی که در ورودی خود به این تراکنش ارجاع داده) را بیابد. اگر چنین تراکنشی یافته شود، آن تراکنش اعتبارسنجی خواهد شد؛ و اگر یک تراکنش معتبر باشد، از مخزن تراکنش‌های یتیم خارج شده و به مخزن تراکنش اصلی اضافه می‌شود تا زنجیره‌ای را که از تراکنش مادر آن شروع شده، تکمیل کند. با اضافه شدن تراکنش جدید به مخزن تراکنش (تراکنشی که دیگر یتیم نیست)، این فرآیند به صورت بازگشتی برای سایر تراکنش‌های خلف (فرزنده) آن تکرار می‌شود تا جایی که دیگر فرزندی یافت نشود. از طریق این فرآیند، ورود یک تراکنش مادر منجر به بازسازی پلکانی یک زنجیره از تراکنش‌های وابسته شده و در تمام طول این زنجیره، تراکنش‌های یتیم به مادر خود متصل خواهند شد.

مخزن تراکنش و مخزن تراکنش‌های یتیم (اگر در یک گره پیاده‌سازی شده باشد) هر دو در حافظه‌ی RAM سیستم نگهداری می‌شوند و به صورت دائمی روی دیسک ذخیره نخواهند شد؛ این دو مخزن با ورود پیام‌های جدید از شبکه‌ی بیت‌کوین به طور دائم پُر و خالی می‌شوند. وقتی یک گره بیت‌کوین راه‌اندازی می‌شود، این مخزن‌ها خالی هستند و به تدریج با تراکنش‌های جدیدی که از شبکه دریافت می‌شوند، پُر خواهند شد.

برخی از پیاده‌سازی‌های مشتری بیت‌کوین یک پایگاه داده یا مخزن UTXO نیز نگه می‌دارند، که چیزی نیست جز مجموعه‌ی تمامی خروجی‌های خرج‌نشده در بلاک‌چین. اگر چه نام «مخزن UTXO» شباهت زیادی به مخزن تراکنش دارد، ولی اینها دو چیز متفاوت هستند. برخلاف مخزن تراکنش و مخزن تراکنش‌های یتیم، مخزن UTXO در آغاز کار گره بیت‌کوین هرگز خالی نیست، بلکه حاوی میلیون‌ها خروجی تراکنش خرج‌نشده موجود در شبکه RAM (تمامی خروجی‌های تراکنش خرج‌نشده تا خود بلاک زاینده) است. مخزن UTXO می‌تواند در حافظه‌ی RAM سیستم نگهداری شود یا (به صورت یک جدول پایگاه داده‌ی مرتب‌سازی شده) روی دیسک آن ذخیره شده باشد.

در حالی که مخزن تراکنش و مخزن تراکنش‌های یتیم از گرهی به گره دیگر (بسته به این که یک گره کار خود را از چه زمانی شروع کرده یا از سر گرفته باشد) بسیار متفاوت هستند، مخزن UTXO حاصل اجماع در شرف تکوین تمامی شبکه‌ی بیت‌کوین بوده و تفاوت محتویات آن از یک گره تا گره دیگر بسیار انک است. علاوه بر آن، مخزن تراکنش و مخزن تراکنش‌های یتیم فقط حاوی تراکنش‌های تأیید‌نشده هستند، در حالی که مخزن UTXO فقط حاوی خروجی‌های تأیید‌شده است.